**Palacký University Olomouc, Faculty of Science,
Department of Geoinformatics**

**Paris Lodron University Salzburg, Faculty of Natural Sciences,
Department of Geoinformatics**

E-LEARNING PLATFORM FOR WEB CARTOGRAPHY

Diploma thesis

Author
B.Sc. Zhibek SOLPIEVA

Supervisor (Palacký University Olomouc)
RNDr. Rostislav Nétek, Ph.D.

Co-supervisor (Paris Lodron University Salzburg)
Dr. Christoph Traun

**Erasmus Mundus Joint Master Degree Programme
Copernicus Master in Digital Earth
Specialization Track Geovisualization & Geocommunication
Olomouc, Czech Republic, 2023**

ANNOTATION

The primary objective of this diploma thesis is to develop an e-learning platform for web cartography. The purpose of this platform is to serve as an educational tool for people who wish to expand their knowledge on Web GIS and cartography, as well as experienced professionals in the field. The extensive comparison table on the platform outlines the capabilities of various JavaScript Mapping APIs, such as Google Maps API, ArcGIS Maps SDK for JavaScript, and Leaflet.

The comprehensive comparison table offered by the platform serves as a significant resource for users seeking guidance in selecting the appropriate JavaScript Mapping APIs for their project. By comparing different APIs' capabilities side-by-side, users are empowered to make informed decisions about which API best suits their requirements.

To enable effective usage of each of the JavaScript Mapping APIs, detailed tutorials with tips and tricks will be provided. Real-life examples will also be included in these tutorials to assist users in gaining a more thorough understanding of web GIS and cartography concepts. Additionally, by testing out projects safely within the system's environment, users can gain hands-on experience and build confidence in their skills.

The platform's user interface has been designed to be accessible to everyone from beginners to advanced developers. With its straightforward design that doesn't require pre-existing knowledge, it's useful even for people new to web GIS and cartography looking for learning resources quickly and easily. It's also easy for users to locate specific information efficiently due to intuitive design elements present throughout the platform.

KEYWORDS

This thesis has been composed by *Zhibek Solpieva* for the Erasmus Mundus Joint Master's Degree Program in Copernicus Master in Digital Earth for the academic years 2019/2020 and 2020/2021 at the Department of Geoinformatics, Faculty of Natural Sciences, Paris Lodron University Salzburg, and Department of Geoinformatics, Faculty of Science, Palacký University Olomouc.

Hereby, I declare that this piece of work is entirely my own, the references cited have been acknowledged and the thesis has not been previously submitted to the fulfilment of the higher degree. Prior to composing this manuscript, certain portions of the material were published on the GeoPick e-learning platform as part of a master's thesis. It should be noted that the author behind this thesis also serves as the creator and developer of GeoPick.

06.05.2023 Olomouc, Czech Republic   Zhibek SOLPIEVA

Palacký University Olomouc
Faculty of Science
Academic year: 2022/2023

ASSIGNMENT OF DIPLOMA THESIS
(project, art work, art performance)

Name and surname:	Zhibek SOLPIEVA
Personal number:	R210702
Study programme:	N0532A330010 Geoinformatics and Cartography
Work topic:	E-learning platform for web cartography
Assigning department:	Department of Geoinformatics

Theses guidelines

The main objective of the master thesis is to develop an e-learning platform for web cartography. Student will explore, analyze and compare the factors and available platforms that can be used for effective and sustainable e-learning within the topic of web cartography. Based on the analysis of several aspects she will develop a customized course consists of blocks of assignments that aim to learn and create interactive maps. This course will provide step-by-step tutorials that describe how to use the JavaScript libraries to display maps and perform operations such as data visualization, geocoding, and spatial analysis. The student will attach all the collected datasets and all the animations to the thesis in digital form. The student will create a website about the thesis following the rules available on the department's website and a poster about the diploma thesis in A2 format. The student will submit entire text (text, attachments, poster, outputs, input and output data) in digital form on a storage medium and the text of the thesis in two bound copies to the secretary of the department.

Extent of work report:	max 50 pages
Extent of graphics content:	as needed
Form processing of diploma thesis:	printed
Language of elaboration:	English

Recommended resources:

Fu, P., & Sun, J. (2011). Web GIS: principles and applications (pp. 89-114). Redlands: Esri Press. Google Cloud Platform [online]. Available at URL: https://cloud.google.com/gcp/ Halim, D. A., & Rasam, A. R. A. (2021, December). Education 4.0 in cartography: an integrated e-learning materials portal for cartography and digital mapping course. In Proceedings of the ICA (Vol. 4). Copernicus GmbH. Kraak M.J., Brown A. (2003). Web Cartography: Developments and Prospects. Muehlenhaus I. (2013). Web Cartography: Map Design for Interactive and Mobile Devices. Nétek, R. (2020). Webová kartografie – specifika tvorby interaktivních map na webu. Univerzita Palackého v Olomouci. 196 s. ISBN 978-80-244-5827-4. Schnabel, O., Stopper, R., & Hurni, L. (2007, August). New modular approach for knowledge-transfer in multimedia cartography: The e-learning project CartouCHe. In International Cartographic Conference (ICC) (pp. 4-10).
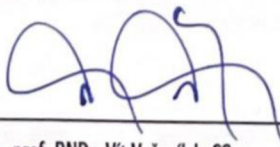
Supervisors of diploma thesis:	RNDr. Rostislav Nétek, Ph.D.
	Department of Geoinformatics

Date of assignment of diploma thesis: **December 5, 2022**
Submission deadline of diploma thesis: **May 5, 2023**

L.S.

_____
doc. RNDr. Martin Kubala, Ph.D.
Dean

_____
prof. RNDr. Vít Voženílek, CSc.
Head of Department

Olomouc   December 5, 2022

CONTENT

LIST OF ABBREVIATIONS

Abbreviation	Meaning
API	Application Programmer Interface
CMS	Content Management System
CSS	Cascading Style Sheets
CSV	Comma Separated Value
GeoJSON	Geographic JavaScript Object Notation
GeoTIFF	Tagged Image File Format
GIF	Graphics Interchange Format
GIS	Geographic Information System
GML	Geography Markup Language
GPX	GPS eXchange Format
HTML	HyperText Markup Language
JPEG	Joint Photographics Experts Group
JS	JavaScript
KML	Keyhole Markup Language
LMS	Learning Management System
PNG	Portable Network Graphics
RSS	Rich Site Summary
SDK	Software Development Kit
SVG	Scalable Vector Graphics
TopoJSON	Topology JavaScript Object Notation
UI	User Interface
VPS	Virtual Private Server
WCS	Web Coverage Service
WFS	Web Feature Service
WKT	Well-Known Text
WMS	Web Map Service
WMTS	Web Map Tiled Service
XML	Extensible Markup Language

INTRODUCTION

Presenting this diploma's thesis on the creation of an e-learning platform that evaluates three widely-used JavaScript Mapping Application Programming Interfaces (APIs): ArcGIS Maps Software Development Kit (SDK) for JavaScript (JS), Google Maps API and Leaflet. The growing popularity of these APIs in web cartography makes it necessary to have a comprehensive understanding of their strengths and limitations. This research contributes a new type of e-learning platform that uses interactive visual aids designed specifically to compare these three JS Mapping APIs. Furthermore, this platform offers valuable insight into how developers can harness the differences between popular Mapping libraries when they are used in real-world scenarios. By comparing these platforms side by side, users could select appropriate APIs based on their particular needs or abilities. This knowledge would benefit developers, GIS professionals, students and researchers looking for a more thorough overview of each API's functionality as well as its potential applications in the field.

With the increasing popularity of web cartography, Mapping APIs such as ArcGIS Maps SDK for JS, Google Maps API and Leaflet are also getting more attention. These APIs have become indispensable tools in creating interactive maps that can be embedded on websites and mobile applications (Kolesnikov & Kikin, 2016). However, selecting the most appropriate one for a particular project can present some difficulties. This is where an e-learning platform becomes quite useful (Gouta, 2018) asserts that developing an e-learning platform to enable users to compare Mapping APIs is crucial. This type of platform offers tutorials with examples of how each API works; it allows users to explore the unique features of ArcGIS Maps SDK for JS, Google Maps API and Leaflet side by side. Furthermore, coding challenges provide users with opportunities to test their knowledge and skills using each API. Such an approach would give developers a better understanding of which Mapping tool suits their needs best. Without access to such an e-learning platform, developers may waste time learning about different Mapping tools only to realize they do not meet their requirements or lack certain functionalities needed for their projects. It becomes necessary then - not just from an educational perspective but also from efficiency purposes - that developers take advantage of platforms like those recommended by Gouta (2018).

The domain of web cartography provides numerous options for developers to choose from, and it is essential for users to evaluate the advantages and disadvantages of each API and identify their distinctive features in order to arrive at an informed decision (Strode et al., 2020). The ArcGIS Maps SDK for JS is endowed with advanced Mapping capabilities like geocoding services and 3D visualization which makes it stand out in the crowd. On the other hand, Google Maps API has been around for over a decade now; it boasts high-quality satellite imagery coupled with street view integration but its pricing structure could be exorbitant at scale (Hamza, 2021). Lastly, Leaflet offers mobile-friendly open-source support albeit at the expense of some sophisticated functionality granted by other APIs. When considering project-specific needs, it is crucial to carefully evaluate the advantages and disadvantages of various APIs and their respective characteristics, in order to make an appropriate selection. It was demonstrated through Hamza's thesis work on integrating mobility data into geovisualization techniques aimed at developing user interactive platforms that would facilitate processing raw data collected in underdeveloped African regions (Hamza, 2021), how important it is to consider selecting the right APIs based on cost efficiency as well as effective communication through maps.

Knowledge acquisition has undergone a remarkable transformation with the advent of technology. According to (Pregowska et al., 2021) "traditional modes of learning such as attending lectures and reading books have been augmented by innovative platforms that offer dynamic approaches to education" (2021). An exemplary platform is e-learning which empowers learners with access to interactive content from anywhere at any time. The novelty and practicality of this system are unparalleled; it caters to developers, GIS professionals, students, researchers and anyone who wishes to gain knowledge. This thesis showcases a pioneering e-learning system that focuses on comparing three JS Mapping APIs - an innovation that can transform how these tools are employed in teaching and research. As (Banitt et al., 2013) state, "The use of technology

in classrooms has been shown to increase student engagement through the use of visualization tools and interactive activities." This accentuates how an e-learning platform similar to the one presented in this thesis not only provides comprehensive overviews but also enriches user experience According to (Al Rawashdeh et al., 2021) "adopting an e-learning system into educational institutions or organizational curriculum opens up boundless possibilities for enriched learning experiences while providing convenience through its remote accessibility" (2021). This is supported by their study where 81% of students stated that e-learning provides scientific material in an interesting way, and 80% of students responded that e-learning increases the possibility of contact between students among themselves and between the students and the teacher (Al Rawashdeh et al., 2021). Additionally, the creation of advanced comparative functionalities for technologies within particular disciplines provides benefits that go beyond just applications. For instance, this innovative e-learning platform incorporates JavaScript Mapping APIs and offers various options to cater to both novice learners and seasoned professionals who wish to stay updated with the latest software trends in their respective fields.

To summarize, the creation of an e-learning platform that compares three widely-used JavaScript Mapping APIs has been a significant and valuable addition to web cartography. In-depth analysis of each API's strengths and weaknesses has been conducted through scientific articles, resulting in a comprehensive overview of their unique functionality. The platform unveiled in this thesis is novel, providing developers, GIS professionals, students, and researchers with not only useful resources but also a singular place to compare these APIs. The advantages offered by this platform are numerous; it empowers developers to make informed decisions on which API will best serve their purposes while simultaneously promoting healthy competition among providers. Moreover, GIS professionals can utilize this resource to easily generate custom maps for various applications. Additionally, students and researchers will have access to dependable information without requiring extensive training or knowledge in any specific programming language – thus lowering barriers for those interested in learning more about map development software within web cartography. Regarding the design components of the e-learning platform itself - interactive comparison tables that detail key features along with video tutorials explaining how each API operates were incorporated into its design. Content presented throughout is user-friendly making even newcomers who don't know much about JS Mapping APIs feel comfortable accessing information presented within. Overall, this diploma's thesis serves as a critical contribution towards the rapidly expanding field of web cartography by introducing an engaging yet informative approach - driving greater accessibility than ever before possible.

1 OBJECTIVES

The aim of diploma thesis is to develop an e-learning platform for web cartography that includes a comparison table of the three JavaScript Mapping APIs and tutorials demonstrating how to use each API. The e-learning platform offers a comprehensive comparative analysis of three widely used JavaScript Mapping APIs, enabling learners to gain an in-depth understanding of their functionalities, features and benefits. The detailed study involves exploring the intricacies of each API, examining their strengths and weaknesses, and identifying how they compare with one another. By delving deeper into this subject matter, students can develop a more nuanced appreciation for these APIs' capabilities while also gaining insights into best practices for selecting the most suitable solution based on specific needs or preferences. Furthermore, the e-learning platform also includes practical tutorials demonstrating how to use each API for web mapping applications. The e-learning platform's tutorials provide comprehensive guidance for creating web mapping applications and examining the distinctive characteristics of each API through real-world situations. The tutorials emphasize the potential uses and limitations of each API, providing a complete understanding of web cartography and its components. To accomplish this goal, the student developed the following research question:

1. What are the strengths and weaknesses of each JavaScript Mapping APIs in terms of its data visualization capabilities and cartographic design?

2. What are the key similarities, specifics, and differences between Google Maps API, Leaflet, and ArcGIS JavaScript API in terms of their capabilities and limitations for creating web cartography?

The expected outcomes of this master thesis include the development of an e-learning platform for web cartography that offers a comprehensive comparative analysis of three widely used JavaScript Mapping APIs: ArcGIS Maps SDK for JavaScript, Google Maps API and Leaflet, including practical tutorials demonstrating how to use each API for web mapping applications. Within this platform's curricular offerings are comprehensive guides detailing precisely what each API offers in functionalities that come complete with essential feature highlights as well as its benefits since these details help learners make more informed choices based on their unique requirements or preferences for software solutions. Moreover, students' deeper understanding of the individual strengths and limitations peculiarities within different APIs characteristics yields further clarity into aspects that differentiate them with regard to data visualization and cartographic design facets wholly. This research provides additional value by identifying similarities across platforms on which future innovations may be built.

Finally, the created e-learning platform will serve as a valuable resource for educators and learners interested in web cartography by offering practical guidance on how to incorporate various JavaScript Mapping APIs into their curricula and projects, potentially leading to the development of more effective teaching methods and tools. Overall, this research will enhance understanding of web cartography and enable educators and learners to make informed decisions about the selection and use of JavaScript Mapping APIs in web mapping applications.

2 METHODOLOGY

The aim of this diploma thesis is to develop an e-learning platform for web cartography, which includes a comprehensive comparison table of three popular JavaScript Mapping APIs and practical tutorials demonstrating how to use each API. To achieve this objective, the student crafted two research questions that examine the strengths and weaknesses of each JavaScript Mapping APIs in terms of its data visualization capabilities and cartographic design. The student also explored the key similarities, specifics, and differences between Google Maps API, Leaflet, and ArcGIS JavaScript API in terms of their capabilities and limitations for creating web cartography. To conduct this research study effectively, the student consulted scientific articles as sources to gain insights into web cartography theory and best practices for selecting appropriate software solutions. Using qualitative research methods such as content analysis techniques helped gather valuable information on these topics from a broad range of reliable sources. The subsequent section in this diploma thesis provides a detailed account of the employed techniques, data and software.

2.1 Used methods

Data collection:

Data collection will include a comprehensive literature review of existing research on web cartography and JavaScript Mapping APIs. the collected data used to develop a comprehensive understanding of three JavaScript Mapping APIs: ArcGIS Maps SDK for JavaScript, Google Maps API and Leaflet. To gain a deeper understanding of web cartography and JavaScript Mapping APIs, the research was involved an extensive literature review. The data collected through this process was carefully analyzed to develop a comprehensive knowledge base on three leading JavaScript Mapping APIs: ArcGIS Maps SDK for JavaScript, Google Maps API and Leaflet. This approach is essential in order to identify trends and patterns within existing research as well as establish gaps that require further exploration. Moreover, the study will also involve a quantitative content analysis of each API's documentation and examples to examine their cartographic capabilities and limitations in terms of data visualization, customization options, and interactivity. The data collected through this analysis will inform the creation of a comparative matrix that outlines each API's key attributes and functionalities in relation to web cartography.

Criteria for evaluation:

The evaluation criteria for this study will include the comprehensiveness and accuracy of the literature review, the depth and rigor of the content analysis of each APIs, and the effectiveness of the comparative matrix in enabling educators and learners to make informed decisions about the selection and use of JavaScript Mapping APIs. The study aims to provide a comprehensive evaluation of the APIs by examining them across five essential criteria, each playing a critical role in determining their overall effectiveness.

The first criterion is user-friendly design, which will assess how intuitive and accessible the API is for users with varying levels of technical expertise.

The second criterion pertains to support and documentation, encompassing factors such as troubleshooting assistance and readily available resources that can aid developers throughout implementation.

Scalability and performance are key concerns that form another important aspect of the analysis; this involves assessing how well the APIs perform under different loads while always maintaining high responsiveness rates.

Additionally, was also delved into community support - an integral component in fostering innovation around any given technology - gauging its strength based on factors like online forums or other collaborative platforms where developers can share knowledge freely.

Lastly, it is crucial to evaluate the technical capabilities. This includes its ability to handle diverse data formats and ensure seamless interaction with complex datasets. Additionally, the JavaScript Mapping APIs must be adept at representing this information in an easy-to-understand format that facilitates better decision-making processes. To achieve the best performance and outcomes for data analysis using a JavaScript Mapping

APIs, it is essential to conduct a comprehensive evaluation of its technical capabilities. Together, these parameters establish a comprehensive perspective of the topic. For a more comprehensive understanding of the criteria used for evaluation, refer to chapter 4.

Comparison and Evaluation:

The evaluation of APIs was a comprehensive process, encompassing both qualitative and quantitative methodologies. Through meticulous data collection in the first step, the gathered information will then serve as an essential criterion for comparison among various APIs. The subsequent outcome will be presented through a well-constructed table that provides insightful analysis regarding each APIs strengths and limitations while highlighting their comparative performance against one another. This thorough approach aims to provide accurate conclusions by weighing up all relevant factors meticulously. Furthermore, this comparison and evaluation process will enable a better understanding of how the evaluated APIs can complement each other while serving different use cases.

Development of e-learning Platform:

An inventive e-learning platform is planned to be created after an exhaustive examination and assessment of the different choices accessible. The primary aim of this platform is to enable users to access comprehensive information about three top JavaScript Mapping APIs. To accomplish this goal, detailed comparison tables that highlight the exclusive merits and demerits of each API was presented on the website. Additionally, the instructional materials available on this site are designed with great care and attention to meet the needs of beginner learners. By attending these tutorials, learners can acquire practical experience in using each API skillfully while also comprehending how they differ from one another.

Evaluation of e-learning Platform:

Evaluating e learning platform effectiveness necessitates adopting both qualitative and quantitative means of appraisal. Platforms' usability and effectiveness was evaluated through user surveys to assess navigational ease, content comprehension and whether they cater to the learners' specific requirements or not. The results of these evaluations was influenced areas for further development and improvement to enhance the platforms overall user experience. Keeping high standards of accuracy while meeting target audiences' needs while developing APIs and e learning platforms require a thorough and methodical approach.

2.2 Used data

The scope of this study is vast and requires a comprehensive collection of relevant resources. Therefore, in addition to the documentation and tutorials provided by the three chosen Mapping APIs, peer-reviewed articles, books, and other credible sources on web cartography will be included. This robust dataset ensures that all aspects of web cartography are explored thoroughly for a more accurate understanding and analysis.

2.3 Used software

The research project utilized various software tools and web platforms to achieve its objectives. While ArcGIS Maps SDK for JavaScript, Google Maps API and Leaflet are the primary tools utilized during analysis and geovisualization phases, a variety of other software programs was also employed to develop resources, generate interactive webmaps and conduct user testing. All software utilized in this research project was acquired from either the University of Salzburg or Palacky University, except for those that were explicitly mentioned.

The following is a comprehensive list of all the tools and software used in this project along with brief explanations detailing their role within the research study:

JavaScript Mapping APIs:

ArcGIS Maps SDK for JavaScript:

ArcGIS Maps SDK for JavaScript is an advanced, web based GIS software SDK used to build interactive Mapping applications. Its comprehensive set of tools and features, along with its open architecture enable developers to create custom apps for

their unique needs. It includes a powerful JavaScript Mapping APIs that provides access to the ArcGIS platform's geospatial capabilities including maps, scenes, layers, 2D-3D visualization elements and analytics. The API provides access to essential geographic information systems (GIS) functions such as routing and navigation as well as basic editing functionality for creating or modifying content such as overlays and annotations. For a more comprehensive understanding of the ArcGIS Maps SDK for JavaScript, refer to chapter 4.

Google Maps API:

Google Maps API is another popular option for developers looking to power their web GIS applications. It provides a comprehensive suite of tools that enable developers to add geospatial data layers and visualizations as well as access global navigation information such as directions and traffic. The API also includes several cartography mechanisms for creating custom maps with varying levels of detail and complexity. Chapter 4 provides a more extensive insight into the Google Maps API.

Leaflet:

The development of mobile-responsive interactive maps is possible with this JavaScript library, which is available as open-source software. It includes various plugins that assist in analyzing and visualizing data. To gain a better comprehension of Leaflet, please refer to chapter 4.

Web development tools:

HTML is a markup language used to create content for the world wide web, while software refers to any set of instructions or programs that are capable of directing a computer to perform specific tasks or operations. In the context of this diploma thesis HTML used for implementing the user interface of a web-based GIS application. CSS is also frequently used in conjunction with HTML to stylize and format the content. JavaScript is another programming language frequently used in web development, as it allows for dynamic and interactive content to be added. In addition to these web development technologies, the authors of this diploma thesis utilized a number of software tools and frameworks in order to implement various aspects of their work.

Visual Studio Code:

Visual Studio Code is a highly impressive software tool that has been explicitly developed for advanced code editing and debugging of web-based Geographic Information System (GIS) applications. This comprehensive program boasts diverse capabilities that cater to various features, allowing developers to create robust GIS solutions that are both efficient and effective. Its user-friendly interface, coupled with extensive customization options, renders it the ultimate choice for building cutting-edge applications requiring meticulous coding, thorough testing, and seamless integration with other software systems. By utilizing Visual Studio Code's enhanced functionalities, developers can effortlessly detect errors or bugs in their codes while optimizing application performance through innovative programming techniques such as refactoring or multi-threading algorithms - resulting in unparalleled accuracy and dependability across multiple platforms.

WordPress:

WordPress is a frequently used content management system (CMS), which permits individuals to easily design and administer websites. The software offers an intuitive interface and a vast range of customizable themes and plugins, thereby rendering it highly adaptable for diverse website necessities. In addition to being extensively utilized for blogging, business and e-commerce sites, among others, WordPress can also be employed in the production of an e-learning portal as part of this thesis project. Capitalizing on its substantial plugin ecosystem, learning management system (LMS) plugins can be integrated with multimedia content, interactive assessments and other educational features to create an all-inclusive and engaging online learning environment. Given WordPress' flexibility and scalability, it serves as an ideal choice for fashioning a research-specific e-learning platform.

Hostinger:

Hostinger is a well-liked provider of web hosting services that offers reliable and affordable options. Its range of hosting plans includes shared, virtual privare server (VPS), and cloud hosting with a user-friendly interface. Hostinger is widely known for its speedy

loading times and exceptional uptime, making it suitable for websites of any size. For diploma thesis purposes, Hostinger can be utilized to host an e-learning platform using WordPress. This is made easy by the one-click installation feature for WordPress and the user-friendly control panel provided by Hostinger which streamlines website setup and management. In addition to these features, Hostinger provides excellent customer support and security measures which ensures that your e-learning platform remains safe, secure, and accessible at all times.

Graphic design tools:

Graphic design tools were also an important element of the research project in order to create visually appealing and informative maps. Software such as Adobe Illustrator and Canva were used to create customized map images, icons, and other visual content. The web-based data visualization tool, Flourish, was utilized to produce interactive charts and animations. The incorporation of these graphic design tools not only augmented the visual attraction of the maps but also assisted in effectively communicating crucial information in a succinct and comprehensible manner.

Microsoft 365 functioned as a comprehensive instrument for project supervision and teamwork, thereby offering its users the seamless capability to generate, modify, and distribute diverse documents such as reports and presentations. The versatile MS Excel provided an efficient solution for organizing complex data sets while allowing teams to analyze information in greater depth. With the use of MS Forms, researchers were able to collect valuable insights by creating custom surveys that facilitated data gathering from participants on a large scale. Additionally, leveraging advanced formatting tools within MS Word enabled users not only to craft high-quality research papers but also elegantly format documentation specific to their projects' needs. The use of various software tools and platforms has greatly enhanced the functionality, interactivity, and aesthetics of web mapping applications in this research project.

Processing procedure

The Figure 1 presented illustrates the fundamental stages of e-learning platform development. Additionally, it showcases the primary software employed to generate in-depth comparison tables and tutorials. For a more thorough understanding of these processes, please refer to chapter 5 where they are explained in greater detail.
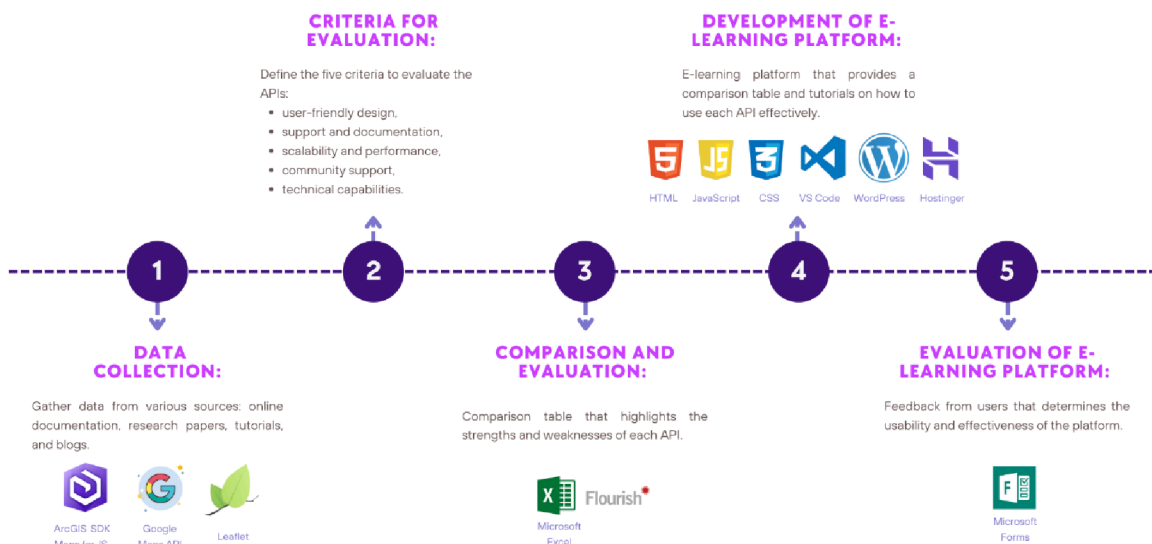

Figure 1 Methodology of diploma thesis.

3 STATE OF ART

Online education has brought about a major transformation in the way we acquire knowledge and skills making it easier for students to learn at their own pace and from virtually anywhere (Gilbert, 2015). Thanks to the rise of e-learning platforms like Coursera, Udemy, and edX, individuals now have access to a vast array of courses on a variety of subjects. As (Moe, 2014) note, this paradigm shift in education has had a significant impact on the way people learn and acquire new skills (2014). Although these platforms have been instrumental in democratizing education they do not always provide comprehensive information on specialized topics (Weishart, 2020).One such area where online learning falls short is providing an exhaustive comparison of available options for JavaScript Mapping APIs used in GIS and cartography projects. These Mapping APIs are essential tools for building interactive maps, visualizing data related to geography and location, analyzing geospatial data, among others (Bowlin, 2015). Despite their significance in the field of GIS, there seems to be a gap in the market when it comes to finding a comprehensive analysis of different JavaScript Mapping APIs. While research studies on the subject exist, they lack detailed tutorials or real-life examples that can assist users in gaining hands-on experience. Enter this e-learning platform developed through this thesis: it addresses such gaps by offering a thorough comparison table of various JavaScript Mapping APIs including ArcGIS Maps SDK for JavaScript, Google Maps API and Leaflet - along with an assortment of detailed tutorials complete with tips and tricks as well as real-life examples. Furthermore, its user interface is designed to be accessible for both beginners and advanced developers alike. This e-learning platform offers an effective solution that caters not only to those familiar but also newcomers looking quickly into learning resources without having any pre-existing knowledge. With its comprehensive comparison table of various JavaScript Mapping APIs alongside this detailed tutorials which contain real life example scenarios guiding users safely within this system's environment while building confidence in their skills - rest assured user making progress towards their goals.
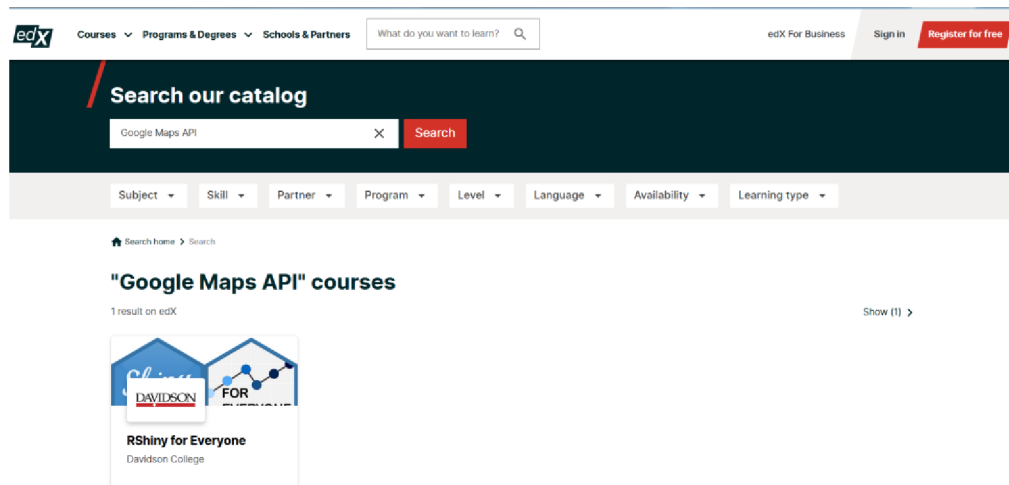
**Figure 2 edX e-learning platform (source:
https://www.edx.org/search?q=Google+Maps+API).**

The world of online learning has experienced a monumental shift in recent years, mainly due to the emergence of platforms like Coursera, Udemy, and edX (Moe, 2014). These e-learning services have revolutionized education by providing a vast array of courses that cover diverse fields (Gul, 2015). Among the topics that have garnered immense popularity are web GIS and cartography. Students worldwide have taken advantage of these platforms to acquire new skills and knowledge in this field. Despite all the significant strides made by these platforms, there is still one crucial area where they fall short - providing learners with an exhaustive overview of JavaScript Mapping APIs.

For instance, upon conducting a search of the Google Maps API Edx (Figure 2) and Coursera on the search engine, it was observed that there were no search results available. This means that there are currently no courses offered by these platforms with respect to this particular API. However, when the same search was performed for Udemy (Figure 3), some courses were found which are not only focused on basic concepts but also require a fee for enrollment. Therefore, it can be inferred that while there may not be offerings from every platform in regards to the Google Maps API Edx and Coursera at present, Udemy might provide suitable options for users seeking paid courses in this regard.
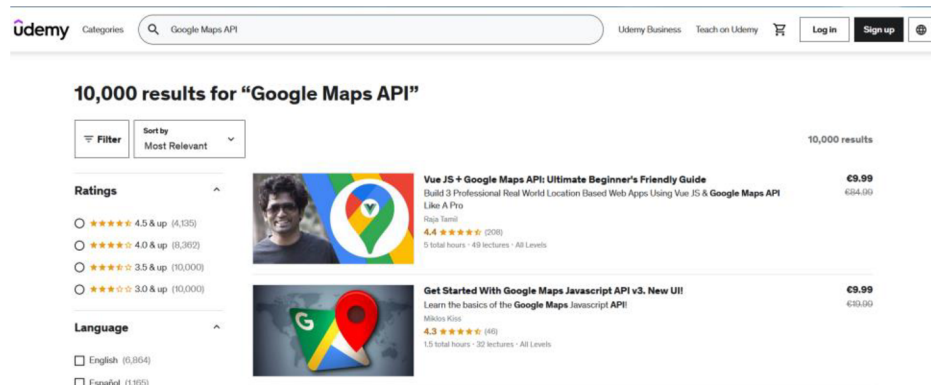
Figure 3 Udemy e-learning platform (source:
https://www.udemy.com/courses/search/?src=ukw&q=Google+Maps+API).

While some courses may mention various Mapping APIs, none offer a comprehensive comparison that would enable students to select the best-fit technology for their projects. This gap has created an opportunity for individuals or organizations looking to provide such training or consulting services. It also highlights the need for these e-learning companies to be more thorough when designing their course catalogs so that students can get a complete understanding of all aspects of their subject matter. According to (Kong et al., 2014) developers and organizations creating Web mapping applications face major challenges due to the lack of an objective evaluation framework for Mapping APIs. The metrics developed from this study could be used as a standard for online spatial information users to choose appropriate products according to their needs. It can also provide valuable information for data providers to improve their products. It is crucial that structured approached are adopted when comparing these APIs based on key metrics such as performance, usability, scalability amongst others. Although some instructors may briefly touch upon differences between various JavaScript Mapping APIs during their courses or tutorials on these online learning platforms; however, this information is often insufficient for learners who wish to gain a more thorough understanding and make informed decisions about which API best fits their requirements. Henceforth there exists an unmet need for comprehensive comparisons that would enable learners to evaluate the strengths and limitations associated with each available option thoroughly (Illeris, 2009). Aspiring developers seeking mastery in Web GIS technologies should look beyond standard online learning platforms like Coursera or Udemy if they want access to detailed analysis regarding JavaScript Mapping APIs so that they can make informed choices best suited for themselves.

While research studies have compared various Mapping APIs in JavaScript, there is a deficiency of real world instances and all encompassing tutorials to aid individuals in gaining hands on experience and building confidence in their skills (Miller, 2020). While these studies offer beneficial perceptions into each API's strengths and weaknesses, they do not offer practical advice for developers who want to use them in their work. This disconnect between theory and practice can be challenging for beginners who are still learning how to effectively use these tools. Therefore, more user-friendly resources are necessary to bridge this gap and make it simpler for individuals to master JavaScript mapping. Lately, multiple online platforms have emerged to tackle this problem by

providing step-by-step guides and interactive exercises that educate users on how to work with various Mapping APIs. For example, Mapbox has a vast documentation library that covers everything from fundamental concepts such as routing and geocoding to complex concepts like custom styling and data visualization. Similarly, Leaflet provides a comprehensive tutorial section that employs its open-source library to assist people in creating attractive maps through walk-throughs of its process. These resources provide practical examples of API usage in various scenarios, promoting experimentation with fresh concepts and techniques while clarifying the process of working with JavaScript Mapping tools. Although prior research on JavaScript Mapping APIs has provided valuable insights into their technical capabilities, true mastery necessitates the combination of theoretical knowledge with practical experience. This is why constructing accessible tutorials and resources similar to those provided by Mapbox and Leaflet can empower developers at different expertise levels, promoting innovation within the field.

The creation of an e-learning platform has addressed a critical gap in the JavaScript Mapping APIs domain. Despite the increasing availability of APIs for full-stack interactive application development, novice developers often struggle to evaluate them efficiently. This is due to the absence of a comprehensive comparison chart, which could allow developers to quickly identify the strengths and weaknesses of each API. As noted by Cheron et al., (2019) this lack of a comparison chart can cause significant difficulties for novice developers attempting to select the most appropriate API for their project. Furthermore, there was also a deficiency in-depth tutorials and practical examples that could help learners navigate through the process effectively. However, with this new platform's introduction, such issues have been mitigated as it provides access to detailed comparisons between various APIs alongside tutorials and real-life examples. This makes it possible for both novice and experienced developers to easily maneuver their way around Mapping APIs. Undoubtedly, the e-learning platform developed here will shape how developers approach working with JavaScript Mapping APIs moving forward. Its user-friendly interface caters explicitly to beginners looking to learn more about these sophisticated tools while remaining accessible enough for seasoned experts continuously seeking innovative solutions; therefore marking a turning point in developer education. In sum up, the development of this e-learning platform represents an important milestone within the field of JavaScript Mapping APIs. By providing access to crucial information on top-end Mapping technologies combined with practical training through real-world use cases, this tool has revolutionized how we perceive these systems complexity (Kim et al., 2014). With time and continued refinement based on feedback from users worldwide using citation support along every step of development can ensure that future iterations remain relevant across multiple domains indefinitely without losing track or focus along any one path pursued by technological advancements today or tomorrow.

To conclude, the emergence of web-based educational platforms such as Coursera, Udemy and edX has transformed our learning practices. These cutting-edge resources cover a multitude of domains including GIS and mapping. However, when it comes to selecting the best-suited JavaScript Mapping APIs for their project, users face a challenge due to inadequate comparative analysis available on existing platforms. This knowledge gap has been addressed in this thesis by creating an e-learning platform that provides detailed comparison tables featuring various JavaScript Mapping APIs like Google Maps API, ArcGIS Maps SDK for JavaScript and Leaflet. The platform also includes real-life examples and tutorials equipped with helpful tips and tricks that aid users in gaining practical experience while building confidence in their skills. The user interface of this platform caters to individuals ranging from novices to advanced developers. Its intuitive design ensures easy access to information regardless of one's pre-existing knowledge about web GIS or cartography concepts. In summary, modern technological advances have ushered significant changes concerning online access to educational resources, nevertheless challenges remain when it comes to choosing suitable tools for particular projects" (Zhang & Liao, 2015). This e-learning platform fills these gaps by providing critical features necessary for making informed decisions regarding which JavaScript Mapping APIs is most appropriate for different tasks through accessible tutorials with tips and tricks at every level – whether novice or experienced developer – along with real-life examples ensuring hands-on experience builds learners' confidence quickly.

4 JAVASCRIPT MAPPING APIS

Chapter 4 of the presented master thesis delves into a thorough evaluation of various JS Mapping APIs. The primary objective of this chapter is to provide an extended analysis, emphasizing on the five principal criteria for each API: community support, scalability and performance, support and documentation, user-friendly design, and technical capabilities. Given that technical capabilities have vast sub-criteria, a more in-depth discussion will be presented in chapter 5.

This comprehensive chapter serves as an excellent resource for those interested in comprehending the advantages and disadvantages of utilizing these APIs. The comparison table has been previously posted on the e-learning platform GeoPick, which will be further explored in chapter 5. Figure 4 illustrates the structure of the criteria used to compare three JavaScript Mapping APIs, which can also be found in the attachment 1.
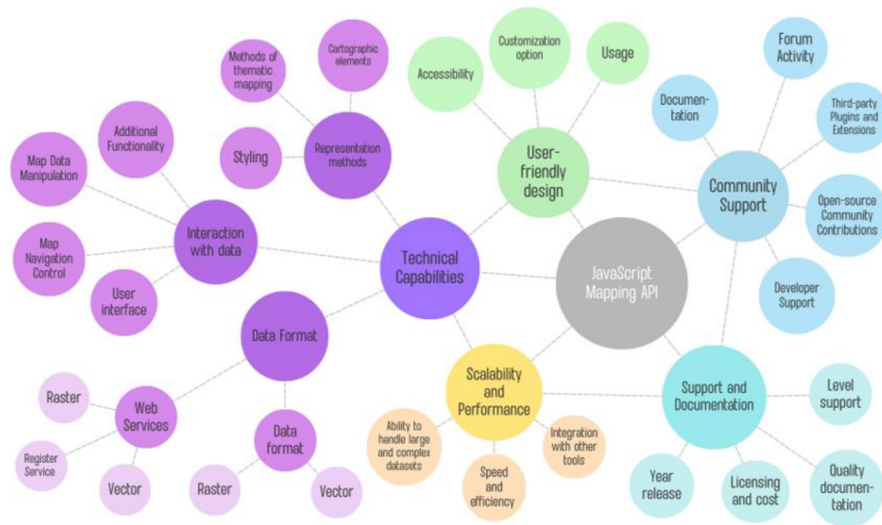
Figure 4 Comparison JS Mapping APIs.

4.1 Introduction to JS Mapping APIs: ArcGIS, Google Maps, and Leaflet

Maps have been employed by humans for navigating and exploring the world throughout the course of history. These graphical representations of our environment help us in discovering new places and grasping a better understanding of our surroundings (Shabiralyani et al., 2015). Nevertheless, conventional paper-based maps have certain limitations in terms of functionality and accessibility (Khan & Adnan, 2010). Updating and disseminating them can be challenging, rendering them unsuitable for various contemporary applications. However, in recent years, JS Mapping APIs like ArcGIS, Google Maps, and Leaflet have transformed the visualization and usage of location data. These potent tools empower businesses, governments, and individuals to create bespoke maps that are both precise and easily accessible from any location across the globe (Mitchell, 2005). The use of these Mapping APIs has resulted in an unparalleled level of accuracy concerning collecting location-based data which is particularly advantageous for firms seeking to optimize their operations by analyzing customer behavior patterns or identifying geographic areas with high demand for specific products or services. Governments also benefit greatly from these mapping technologies since they can help track population trends or monitor natural disasters in real-time - which saves lives. Additionally, individuals now have more control over how they use location information thanks in part due these intuitive platforms like ArcGIS or Google Maps which allow everyone regardless if you're tech-savvy or not explore spatially referenced data with ease. Therefore this chapter aims at exploring three major JS Mapping APIs that are widely used today: ArcGIS by Esri Inc., Google Maps developed by Alphabet subsidiary Google LLC., And last but not least Leaflet developed by Vladimir Agafonkin -

all providing unique features and functionalities while sharing commonalities among themselves. In conclusion JS Mapping APIs provide an unparalleled level of accuracy when it comes down utilizing location data; With many benefits accruing across industries ranging from logistics optimization (e.g., delivery services) through environmental monitoring initiatives (e.g., disaster management).

JS Mapping APIs such as ArcGIS, Google Maps, and Leaflet have revolutionized the way location data is visualized and utilized. The utilization of these Mapping libraries provides an unprecedented level of accuracy and accessibility for businesses, governments, and individuals alike. In this section we will compare each library by looking at their strengths and weaknesses. When comparing web mapping libraries in terms of functionality, MacLeod & Hewitt (2018) found that while all three libraries are capable of providing basic map visualization with markers or polygons overlaid on top of them, ArcGIS has a more comprehensive set of tools geared towards advanced spatial analysis tasks. According to (Crickard, 2014), one major advantage that Leaflet has over its competitors is the widespread familiarity among users which makes it easier to integrate into applications without requiring extensive documentation or training. Furthermore, (Dinçer & Uraz, 2013) explain how the Google Maps JavaScript API stands out from other Mapping APIs due to its popularity amongst developers who praise it for being intuitive yet powerful with a vast array of features. For instance, the API incorporates street view imagery allowing users to see panoramic views from any location selected on the map; furthermore they can also use satellite images for various purposes including identifying streets' names. However while Google Maps might seemingly be unbeatable in terms of popularity among developers there are some shortcomings when compared against other JS Mapping APIs like ArcGIS or Leaflet. One problem cited by MacLeod & Hewitt is that unlike ArcGIS which offers built-in support for secure geocoding services, Google maps requires extra coding effort if you want your app(s) running smoothly across different platforms whereas leaflet's open source code architecture allows you full control over customizing your app's appearance without worrying about licensing issues . In conclusion when choosing between these three popular JS Mapping APIs - ArcGIS, Google Maps and Leaflet- researchers should consider factors such as accessibility, functionality and ease-of-use. Each library has its own set of strengths and weaknesses that make it suitable for certain use cases over others. Therefore, the final choice should be based on the specific needs of each project or application to ensure optimal results are achieved.

To sum up, the utilization of JS Mapping APIs is now a crucial component of web mapping applications. The advent of ArcGIS, Google Maps, and Leaflet has transformed the way in which we perceive and analyze geospatial data. Nevertheless, each JavaScript Mapping APIs has its own strengths and weaknesses that need to be considered. This chapter has investigated the basic characteristics of ArcGIS, Google Maps, and Leaflet. It has also compared their accessibility, customization options, cost-effectiveness, and licensing agreements. Based on this examination, it is evident that each API caters to different requirements of developers. ArcGIS is best suited for those who desire highly tailored maps or necessitate advanced spatial analysis tools. On the other hand, Google Maps serves as an optimal choice for individuals who require user-friendly interfaces or real-time traffic updates. It is significant to note that there are other Mapping APIs in the market that offer analogous functionalities to the ones mentioned in this paper. Therefore, researchers should conduct a comprehensive analysis before choosing any API for their projects. Student believed that JS Mapping APIs have been immensely beneficial to both individuals and organizations by providing valuable insights into location-based data. It will continue to be a vital tool in developing innovative solutions to address complex issues in environmental planning, urban design, transportation systems, and numerous other sectors. In conclusion, regardless of which JavaScript Mapping APIs one chooses to work with, each has remarkable benefits when used effectively, thereby enhancing geospatial intelligence capabilities across numerous industries worldwide.

4.2 Comparison of JS Mapping APIs Based on Features

As time has passed, web mapping has evolved into an essential tool for various applications (Veenendaal, 2016). The utilization of web maps is widespread across

different domains such as urban planning, transportation, and environmental management. To produce web maps, the commonly used APIs or frameworks are ArcGIS Maps SDK for JS, Google Maps API, and Leaflet. Each of these APIs offers distinctive features that set them apart from one another. It is imperative to conduct a thorough analysis of the individual characteristics of each API before selecting the best-suited one for a specific application. In this chapter, was provided an extensive analysis of these three popular JavaScript Mapping APIs based on five primary criteria: community support, scalability and performance, support and documentation, user-friendly design, and technical capabilities. To effectively compare these APIs based on the aforementioned criteria, was conducted a detailed examination. By carefully scrutinizing each criterion against every API under consideration - ArcGIS Maps SDK for JS, Google Maps API, and Leaflet – was determined which among them is the superior option. Consequently, the objective of this chapter is to respond to the subsequent research questions: What are the strengths and weaknesses of each JavaScript Mapping APIs in terms of its data visualization capabilities and cartographic design? What are the key similarities, specifics, and differences between Google Maps API, Leaflet, and ArcGIS JavaScript API in terms of their capabilities and limitations for creating web cartography? Consequently, it can be deduced which Mapping application programming interface is most fitting for diverse Mapping applications. This exploration offers useful insights that could benefit developers in selecting an appropriate JavaScript Mapping APIs based on their project specifications.

For this study, scientific articles and official documentation pertaining to the JavaScript Mapping APIs of ArcGIS Maps SDK for JS, Google Maps API and Leaflet were reviewed. The primary aim was to compare the three APIs using five principal criteria: community support, scalability and performance, support and documentation, user-friendly design, and technical capabilities. All information gathered from research articles was analyzed quantitatively using a systematic approach. Each criterion was given an equal weightage during evaluation to ensure fairness in comparison. Furthermore, only credible sources with relevant information were included in the analysis process. The findings from this study provide an overview of each APIs strengths and limitations based on their features' analysis through extensive research done on various educational resources. In the subsequent section, a more in-depth examination will be conducted on the standards used to assess the JavaScript Mapping APIs.

4.2.1 Community Support

The involvement of the community can play a crucial role in aiding web mapping application developers to resolve issues, acquire knowledge about novel techniques, and expand the range of mapping APIs (Hall et al., 2010) This section will assess four primary types of community assistance, such as documentation, participation in forums, third-party plugins and extensions, open-source contributions from the community, and developer guidance.

Documentation

Comprehensive documentation is of utmost importance in providing developers and users with a detailed description of the various features and capabilities offered by a mapping APIs (Gerken et al., 2011). The quality and clarity of documentation can significantly impact the successful integration, efficient development, and overall success of web mapping applications.

Forum Activity

Forums dedicated to mapping APIs are often bustling with activity, and they connect developers with the larger mapping community. From answering queries and providing tutorials to sharing code snippets and discussing new features, a forum's value lies in the interaction it fosters between its users.

Third-party Plugins and Extensions

Web mapping APIs typically offer a fundamental set of features, but they may not consistently fulfill the specifications of all developers. To address this issue, third-party plugins and extensions can serve as a solution by furnishing supplementary functionalities that users can effortlessly integrate into their web mapping applications

(Wachowicz et al., 2008). This approach enhances the user experience and enables the API to expand beyond its core capabilities.

Open-source Community Contributions

The significance of contributions from the open-source community lies in their capacity to enhance the functionality, efficiency, and dependability of the mapping APIs. Open-source development stimulates community involvement and contributions by fostering a collaborative ambiance where developers can exchange their ideas, expertise, and programming code (Shah, 2006).

Developer Support

Achieving success with a web mapping APIs can be greatly enhanced by a developer support team that is both responsive and knowledgeable. Such a team can efficiently address and resolve issues, provide prompt responses to queries, and offer guidance on optimal practices (Peterson, 2014). Consequently, this will enhance the experience of both developers and users. An overview of the main features of each mapping APIs is comprehensively presented in Table 1 for comparison purposes.

Table 1 Community Support comparison table

Community Support	ArcGIS Maps SDK for JS	Google Maps API	Leaflet
Documentation	Extensively documented with tutorials, examples, and code samples	Well-documented with tutorials, examples, and code samples	Documented with tutorials and examples, not as extensive as the other two APIs
Forum Activity	Active online community with a dedicated forum, Stack Overflow, GitHub discussions.	Active online community with a dedicated forum, Stack Overflow, GitHub discussions, and Developer community	Active online community with a dedicated forum, Stack Overflow, and GitHub discussions
Third-party Plugins and Extensions	Large and diverse selection of plugins and extensions available on ArcGIS Marketplace	Large and diverse selection of plugins and extensions available on Google's Developer website and GitHub	Large and diverse selection of plugins and extensions available on Leaflet's website and GitHub
Open-source Community Contributions	Active open-source development on GitHub with contributions from developers and users	Active open-source development on GitHub, with contributions from developers and users	Active open-source development on GitHub, with contributions from developers and users
Developer Support	Esri provides technical support, documentation, and training resources for developers	Google provides technical support, documentation, and training resources for developers	Leaflet provides documentation and tutorials, but does not offer direct technical support or training resources for developers

4.2.2 Scalability and Performance

In web mapping applications, scalability pertains to the capacity of the system to manage increasing data volume or users without compromising performance (Bondi, 2000). This is a significant factor that should be taken into account by applications

anticipated to expand over time or those that deal with massive amounts of data. Conversely, performance refers to how promptly and effectively an application responds to user requests or interactions, encompassing multiple factors such as network latency, load times and speed of data processing.

Ability to handle large or complex datasets

In web mapping applications, the management of expansive or intricate datasets is an indispensable aspect (Kong et al., 2015). As these datasets expand in both size and intricacy, it is imperative for the applications to be equipped with the ability to manage such growth whilst maintaining optimal performance levels (Jordon et al., 2022). The ArcGIS Maps SDK for JS has been specifically formulated to effectively manage large and complex datasets by offering dynamic rendering and vector tiles as some of its features. Furthermore, Google Maps API provides advanced spatial data processing capabilities that enable simplified handling of complex information sets. Although not as feature-packed as the other two APIs, Leaflet is still capable of managing extensive information sets albeit with certain constraints.

Speed and efficiency

Speed and efficiency are also key considerations for web mapping applications, as users expect fast and responsive experiences (Yang et al., 2007). The ArcGIS Maps SDK for JS offers exceptional performance, with features such as asynchronous loading and smart rendering algorithms. Google Maps API is also known for its fast performance, thanks to advanced caching strategies and optimized data processing. Leaflet, while not as optimized as the other two APIs, still offers relatively fast speeds, particularly for smaller datasets.

Integration with other tools

Finally, interoperability with various tools is a significant factor to consider for web mapping applications, particularly as users often aspire to exchange data between several platforms (Ballatore et al., 2011). The ArcGIS Maps SDK for JS provides advanced compatibility with other ArcGIS products and prominent web frameworks like Angular and React. Similarly, the Google Maps API is also well-suited to integrate seamlessly with numerous popular web frameworks and also integrates effectively with other Google services such as Google Analytics. Leaflet is another platform that facilitates sound integration with multiple tools, particularly in open-source frameworks like React and Vue. The comparative table 2 encapsulates the crucial features of each Mapping APIs.

Table 2 Scalability and Perfomance comparison table

Scalability and Performance	ArcGIS Maps SDK for JS	Google Maps API	Leaflet
Ability to handle large or complex datasets	Offers the ability to handle large or complex datasets through server-side processing	Offers the ability to handle large datasets through server-side processing	Offers the ability to handle smaller datasets, but may struggle with larger or more complex datasets
Speed and efficiency	Provides fast rendering and processing of data, with high performance on both desktop and mobile devices	Offers fast rendering and processing of data, with high performance on both desktop and mobile devices	Provides moderate rendering and processing speed, with limitations on handling large datasets
Integration with other tools	Integrates well with other ArcGIS products and third-party tools, with extensive APIs and SDKs available	Integrates well with other Google products and third-party tools, with APIs and SDKs available	Integrates well with other open-source libraries and tools, with plugins and third

4.2.3 Support and Documentation

The development of web mapping applications depends heavily on support and documentation. The size of the library, quality of documentation, and level of support all have an effect on user experience, application performance, and functionality. Inadequate

documentation in an API can cause confusion and frustration for developers resulting in inferior applications (Veenendaal, 2015)

Level of support

The level of support refers to the availability of resources and help for users of the API. ArcGIS Maps SDK for JS offers various levels of support options, ranging from self-help resources to premium support. Google Maps API offers extensive support documentation, including forums, issue tracker, and developer resources. Leaflet provides a user-friendly documentation library and forum for users to seek out support as needed.

Quality of documentation

The quality of documentation provided refers to the clarity and completeness of the API documentation (Maalej & Robillard, 2013). ArcGIS Maps SDK for JS has extensive documentation, including tutorials, samples, and comprehensive API reference documentation. Google Maps API has a very detailed and comprehensive documentation library, while Leaflet has an easy-to-follow and comprehensive documentation system, which makes it easy for developers to find the resources they need.

Licensing and cost

Licensing and cost refers to the license agreement under which the users can use the API and the associated costs. ArcGIS Maps SDK for JS and Google Maps API are proprietary APIs, which require users to obtain a license to use. Leaflet, on the other hand, is an open-source API and can be used for free. The comparison table 3 summarizes the key features of each Mapping APIs:

Table 3 Support and Documentation comparison table

Support and Documentation	ArcGIS Maps SDK for JS	Google Maps API	Leaflet
Version[1]	4.25.5	3.51.7	1.9.3
Size of library (Unpacked Size)	48.8 MB	114 MB	3.72 MB
Level of support	Offers extensive technical support through documentation, tutorials, and community forums	Provides technical support through documentation, forums, and customer support	Offers limited technical support through documentation and community forums
Quality of documentation	Provides comprehensive and well-organized documentation	Offers well-organized and detailed documentation	Offers basic documentation, which may be less organized and comprehensive
Licensing and cost	Offers flexible licensing options, including free and paid versions, with pricing based on usage	Offers a free version for personal use and a paid version for commercial use, with pricing based on usage	Offers a free and open-source license with no commercial restrictions
Year of release	2008	2005	2011

4.2.4 User-friendly design

In order to achieve success, web mapping applications must prioritize user-friendliness and accessibility. As such, it is of utmost importance to create an interface that is easily navigable and customizable for all users. The design of web-based mapping applications should be intuitive yet simple particularly as effortless interaction with the map is integral to the use of web mapping tools (Kong et al., 2014).

---

[1] Relevant at the time of writing the diploma thesis

Ease of use

The user-friendliness of web mapping applications is determined by how easy it is for users to interact with them (Poplin, 2015). ArcGIS Maps SDK for JS and Google Maps API both offer several tools and features that make map interaction simple, such as customized markers and heatmaps. Leaflet is popularly known for its ease of use, simplicity, and user-friendliness.

Accessibility

All users should be able to use web mapping applications, regardless of their physical or cognitive abilities (Paciello, 2000). ArcGIS Maps SDK for JS and Google Maps API have accessibility features like screen reader support, high contrast options, keyboard shortcuts and text-to-speech support. Leaflet is generally accessible but may need customization to meet specific accessibility requirements.

Customization options

Customization options refer to the ability of users to customize the web mapping interface as per their requirement (Delfos et al., 2014). ArcGIS Maps SDK for JS includes a range of customization options, such as the ability to change the map style and add custom data. Google Maps API is also highly customizable, with options such as customized markers and styling properties. Leaflet is highly customizable, thanks to a vast selection of available plugins and its open-source nature.

Usage

Usage refers to the ability to use web mapping applications across different devices and platforms (Delfos et al., 2014). ArcGIS Maps SDK for JS offers support for different browsers and platforms, including desktop and mobile. Google Maps API also offers support for various browsers and devices, including mobile. Leaflet is designed to be responsive and works well on all devices and platforms. The comparison table 4 summarizes the key features of each Mapping APIs:

Table 4 User-friendly design comparison table

User-Friendly Design	ArcGIS Maps SDK for JS	Google Maps API	Leaflet
Ease of use	Offers a user-friendly interface with a wide range of documentation and tutorials available	Provides an intuitive and easy-to-use interface with extensive documentation and resources available	Offers a simple and easy-to-use interface with limited documentation and resources available
Accessibility	Provides accessibility support and complies with accessibility guidelines	Offers limited accessibility support and may not comply with accessibility guidelines	Provides some accessibility support, but may not fully comply with accessibility guidelines
Customization options	Offers a high level of customization through APIs and SDKs, as well as the ability to create custom widgets	Provides a range of customization options through APIs and SDKs, as well as the ability to create custom overlays	Offers limited customization options through plugins and third-party libraries
Usage	Desktop/terminal & web development	Web development	Web development

4.2.5 Technical Capabilities

Technical capabilities of a mapping platform provide users with an understanding of the limits and potentials offered by each API. This chapter will focus on comparing three JS Mapping APIs when it comes to their technical capabilities: Data format coverage, Interaction with data, and Representation methods.

25

Data Format Coverage

Data format coverage refers to the number and types of data formats that a Mapping APIs can natively support. Effective adoption of spatial data from different sources is facilitated by format coverage, with the importance of data interoperability across the geospatial community well-documented (Khoumeri & Benslimane, 2007).

Interactivity with Data

Interactivity with data means the ability to query, filter, modify, and update map data dynamically (Dimara & Perin, 2020). Interactivity impacts how users can work with data on the map, especially as mapping APIs continue to develop and augment the user interface with additional controls.

Methods of Representation

Methods of representation refer to how a Mapping APIs represents geographic data, including how to present different data layers, generate dynamic labeling, and include descriptive details and other user-friendly features (Fairbairn et al., 2001). It relates to the different ways in which information can be presented visually or even auditory that ensures information is understandable by the end-users.

Chapter 5.2 provides an in-depth explanation of the technical capabilities, including data format coverage, representation methods, and interactions with the data. The chapter has elaborated on these aspects to give a better understanding of how they function and their significance within the broader context of the subject matter. By going into greater detail about these technical capabilities, readers can gain a more comprehensive understanding of how they operate and how they can be leveraged to achieve specific objectives. It offers a detailed exploration that goes beyond surface-level discussions and delves deeper into the technical nuances that are crucial for success in this field.The comparison table below summarizes the key features of each Mapping APIs:

Table 5 Technical capabilities comparison table

Technical Capabilities	ArcGIS Maps SDK for JS	Google Maps API	Leaflet
Data format coverage	Supports multiple data formats including GeoJSON, KML, WMS, and CSV	Supports various data formats such as KML, GeoJSON, and Shapefiles	Supports GeoJSON, TopoJSON, KML, and GPX
Interaction with data	Offers extensive Mapping tools such as zooming, panning, and querying, as well as geocoding and routing API	Offers a range of Mapping tools, including zooming, panning, and Street View, as well as geocoding and routing API	Offers basic Mapping tools such as zooming, panning, and layer controls
Representation methods	Provides a variety of visualization techniques, including heat maps and 3D rendering	Provides several visualization options, such as custom markers, info windows, and Street View	Provides basic visualization options, such as markers and pop-ups

Result of evaluation

The following are the results of this analysis of ArcGIS Maps SDK for JS, Google Maps API, and Leaflet based on five key criteria: community support, scalability and performance, support and documentation, user-friendly design, and technical capabilities.

Community Support: ArcGIS Maps SDK for JS has a strong community base that is growing rapidly. It offers extensive resources such as forums where users can seek help or share their experiences with other developers. The Google Maps API also boasts excellent community support with a vast network of experts available to assist users in case they encounter any issues. In contrast, Leaflet's community is relatively smaller compared to the other two APIs.

Scalability and Performance: Regarding scalability and performance, all three Mapping APIs perform well when handling small datasets. However, when dealing with large amounts of data or complex visualizations such as heatmaps or 3D maps, ArcGIS Maps SDK for JS outperforms both Google Maps API and Leaflet thanks to its advanced rendering engine.

Support and Documentation: Each Mapping APIs provides comprehensive documentation that includes useful guides on how to use various features. However, ArcGIS's SDK for JS stands out by providing more detailed examples along with code snippets that make it easier for developers to incorporate these features into their projects quickly. Meanwhile, the documentation provided by Google Maps API is informative but could be better organized - making it difficult at times to locate specific information easily.

User-Friendly Design: Google Map APIs have an intuitive user interface; hence they appeal more towards beginners who want quick access without having much programming knowledge beforehand. On the other hand –Leaflets' simple UI may not be as appealing initially but does provide experienced programmers greater flexibility since leaflets allow full customization options which result in an enhanced level of control over map designs.

Technical Capabilities: When considering technical capabilities like format data coverage, interaction with data, and representation methods, ArcGIS Maps SDK for JS is the most advanced of the three. Google Maps API offers complex features that are useful in many scenarios, but ArcGIS's extensive technical capabilities make it a better choice when dealing with sophisticated mapping applications.

In line with this all three APIs offer unique strengths and weaknesses based on individual project requirements. However, after careful consideration of community support, scalability and performance, support and documentation, user-friendly design, and technical capabilities - was determined that choosing the best Mapping APIs depends entirely upon these criteria's significance to specific projects. As such: For developers who require more control over map designs or have experience programming Leaflet stands out as an excellent option while those seeking advanced technical features will find ArcGIS Maps SDK for JS more suitable; Beginners can opt for Google Maps API because of its easy-to-use interface coupled with comprehensive documentation.

In conclusion, after a detailed analysis of the ArcGIS Maps SDK for JS, Google Maps API and Leaflet based on their respective features, it is evident that each API has its strengths and weaknesses. Community support emerged as an essential factor when evaluating these Mapping APIs as it guarantees access to help when required. Additionally, scalability and performance are critical factors that determine the overall effectiveness of any Mapping service. Furthermore, documentation and user-friendly design are vital elements in ensuring that users can efficiently operate these APIs with ease. Technical capabilities also play a crucial role in developing interactive maps with advanced functionalities. It is worth mentioning that while all mentioned criteria were considered during the evaluation process; some may be more important than others depending on specific use cases or project requirements. Consequently, choosing one over another may depend largely on individual preferences. Finally, this chapter aimed to provide readers with an in-depth comparison of three popular JavaScript Mapping APIs: ArcGIS Maps SDK for JS, Google Maps API and Leaflet using five principal criteria. In summary, selecting the most suitable API solely depends on specific project needs and priorities rather than merely popularity or brand recognition. In closing, this study provides significant insights into evaluating JavaScript Mapping APIs characteristics based on essential features for anyone seeking to develop web-based map applications effectively without compromising functionality or user experience.

The analysis mentioned earlier has been published on GeoPick, an online learning platform that offers a diverse range of educational resources and tools to enhance the overall learning experience. If you want to know more about GeoPick and its various features, you can refer to chapter 5 where it has been discussed in greater detail.

5 E-LEARNING PLATFORM GEOPICK

Chapter 5 of this master thesis explores the e-learning platform, GeoPick, which focuses on a detailed comparison between JS Mapping APIs of ArcGIS Maps SDK for JavaScript, Google Maps API, and Leaflet. In the previous chapter 4, a broad comparison of different JS Mapping APIs was conducted. In this chapter, the focus will be to provide an expansion on the analysis, paying attention to the key features of each API, data format coverage, interaction data comparison, and representation methods comparison.

Additionally, the chapter will provide guidance on using GeoPick through a hands-on JS Mapping APIs tutorial. The tutorial will be structured in four levels of difficulty, ranging from API essentials to complex data visualization techniques. This will be accompanied by an additional resources section that aims to support the findings and to provide scientific articles and other relevant sources to further enhance the reader's understanding.

Overall, this chapter is an in-depth analysis of JS Mapping APIs and is ideal for readers keen on understanding the pros and cons of using these APIs. Furthermore, it is just as useful for those interested in learning how to use GeoPick and different Mapping techniques.

5.1 Overview e-learning platform GeoPick

GeoPick is an e-learning platform designed to help learners navigate the world of web cartography. Its main focus is on helping learners choose the most suitable JavaScript Mapping APIs for their project, and it provides a comprehensive comparison table of popular APIs such as ArcGIS JavaScript API, Google Maps API, and Leaflet. The platform simplifies the challenge of selecting the right Mapping APIs, especially for beginners, by providing clear and concise information on each API to help them make informed decisions for their project's success. GeoPick offers something for everyone, whether you're a beginner or an experienced web cartographer.

It provides tutorials at different levels to help learners develop their skills in creating interactive maps with ArcGIS Maps SDK for JS, Google Maps API and Leaflet. Moreover, GeoPick offers additional resources that cover advanced topics such as creating custom map layers and optimizing performance. The platform also features a Comparison of JS Mapping APIs Based on Features overview that contains detailed technical analysis pages about each API. These pages offer comprehensive comparison tables based on data format coverage, interaction data comparison, and representation methods comparison.

For developers who are working with specific types of data or functionalities in mind GeoPick can be very helpful because it allows readers to compare the features and capabilities of each API easily and quickly. In addition to this feature comparison table there are other resources available on the website including scientific articles and blogs covering advanced topics as well as video tutorials that walk users through basic and more advanced features. Overall GeoPick is an accessible platform focused on web cartography with content that ranges from tutorials designed specifically for beginners to expert level discussions about complex data visualization techniques. GeoPick was based on a solid methodology and built on WordPress. The platform is currently hosted in Hostinger which provides a secure and reliable hosting environment to ensure a smooth and seamless learning experience. For anyone interested in learning about web cartography and JavaScript Mapping APIs, GeoPick is an invaluable resource that will make the process much more manageable. To enhancing comprehension, an illustration of the e-learning platform's architecture has been depicted in Figure 5.

Figure 5 Structure e-learning platform GeoPick.

5.2 Comparison table section

5.2.1 Comparison of JS Mapping APIs: Technical Capabilities

The emergence of Mapping APIs has revolutionized the way we interact with maps (Veenendaal, 2016). With the increasing demand for interactive and responsive web applications, developers have turned to these tools to create robust and engaging maps. Amongst these Mapping APIs are ArcGIS Maps SDK for JS, Google Maps API, and Leaflet. These three platforms provide an extensive set of capabilities that enable developers to display various types of data in unique ways. However, it is critical to understand that not all Mapping APIs are created equal. Technical capabilities such as data format coverage, interaction with data, and representation of data in context can vary significantly between different platforms. Therefore, it is essential to compare these technical aspects across ArcGIS Maps SDK for JS, Google Maps API, and Leaflet to determine which JS Mapping APIs suits user needs best. In this chapter, was examined how these three Mapping APIs differ from one another concerning their technical capabilities: specifically focusing on their ability to cover diverse data formats seamlessly; interact effectively with massive datasets; represent complex information through visualizations accurately in a given context. Moreover ,was highlighted the significance of considering structure as a main chapter when implementing any of these Mapping APIs into effective map development practices. The comparison presented in the next chapter can help developers make informed decisions about which platform best fits their particular use case based on factors such as complexity or project scope.Overall this thesis statement clearly states that by comparing key technical features among ArcGIS Maps SDK for JS ,Google Map API,and leaflet highlights the importance on structure being considered like an important component chapter while developing efficient maps. This helps identify what areas within each JavaScript Mapping APIs require more attention from developers while creating better user experiences. Upon further examination of the primary sections of this diploma thesis, which analyze each aspect noted previously, it becomes apparent that every tool possesses its own advantages and drawbacks. Thus further emphasizing why considering technical capabilities as a primary focus during implementation is crucial so that user create efficient and effective maps regardless of their choice amongst them.

For this chapter, the method employed was a thorough analysis of scientific articles and official docimentation related to Mapping APIs. The sources were selected based on their relevance and reliability in providing insights into the technical capabilities of JS Mapping APIs such as data format coverage, interaction with data, and representation of data in context of ArcGIS Maps SDK for JS, Google Maps API and Leaflet. The structure chapter played an essential role in guiding the research process by

outlining the scope and objectives of the study. It provided a framework for organizing information derived from various sources to make informed comparisons. Figure 6 can aid in comprehending the organization of subsequent chapters. It presents the criteria and sub-criteria that were utilized for conducting a comparative evaluation of different Java Script APIs based on specific aspects. For further examination, this arrangement can also be examined in greater detail in the attachment 2.
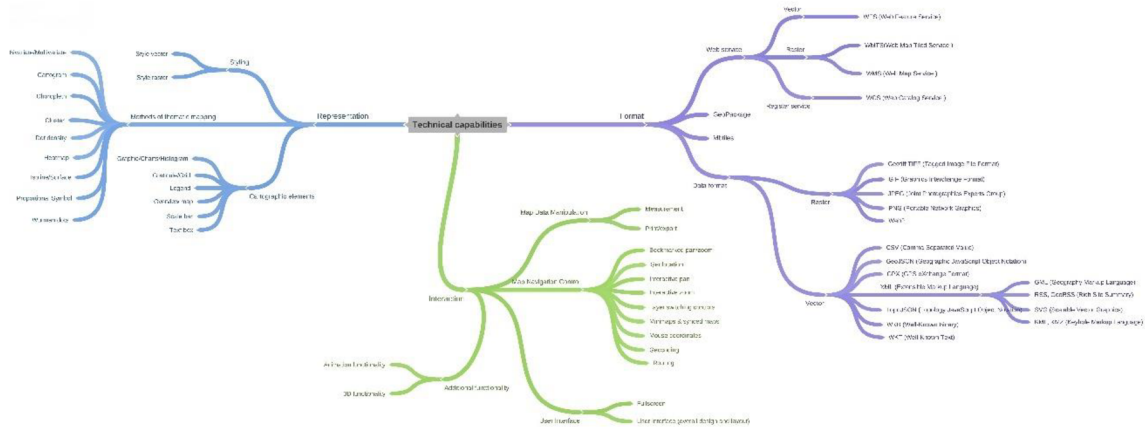

Figure 6 Structure technical capabilities.

5.2.2 Data Format Coverage

This chapter summarizes a comparison of three JavaScript Mapping APIs: ArcGIS Maps SDK for JavaScript, Google Maps API, and Leaflet. The goal is to show the differences between these APIs in terms of data format and web service capabilities. Educational materials were used to collect information about their support for data formats and web services, which was used to create two tables featuring significant differences between the APIs. The table 6 compares the supported data formats of each API and their compatibility with various types of data. The table 7 focuses on comparing their web services. Additionally, this research assesses each JS Mapping APIs extent and limitations regarding data representation techniques and thematic methodologies, revealing that ArcGIS Maps SDK provides built-in assistance.

JavaScript Mapping APIs are popular as they provide a lot of options for creating interactive maps and displaying location-based data, but it can be challenging to choose the best API for one's project. A diploma thesis analyzed and evaluated three APIs- Google Maps API, ArcGIS Maps SDK for JavaScript, and Leaflet. The analysis compared supported data formats across all three APIs. Each API offers varying levels of compatibility with different data formats. While ArcGIS Maps SDK supports industry-standard Esri File Geodatabase and OGC WMS/WMTS along CSVs or GeoJSONs suitable for complex mapping applications requiring advanced visualization capabilities (Roth et al., 2015); Google Maps API focuses on KMLs or GeoRSS files while having built-in provisions that enable users to easily visualize voluminous datasets through Google Fusion Tables (Gonzalez et al., 2010); Leaflet features plugins supporting various file types like GPX or CSV documents but may not prove efficient when handling large datasets compared to other APIs. The analysis then moves forward towards learning objectives involving understanding differences between various JS Mapping APIs regarding used data format, identifying utilized web services among these most commonly-used platforms, analyzing a comprehensive comparison table showcased on e-learning platform called GeoPick intended at comparing them based on strengths/limitations using both thematic methods and representation aspects, finding scope and limitations surrounding scenarios involving usage ranging from simpler integration into google services towards more advanced big-picture exploration/design-oriented work.

Next up is comparing web services provided by all three APIs: Each API has its unique strengths and limitations concerning web services; for instance, ArcGIS Maps SDK provides extensive support for WFS and WMS and WMTS, WCS services alongside

feature services (Nétek, 2020). In contrast, Google Maps API offers a wide range of WFS integration but fails to provide WMTS and WMS comprehensive coverage similar to that of other Mapping APIs like ArcGIS. Leaflet does not have built-in support for web services.

The analysis then examines the scope and limitations of data format support across all three JS Mapping APIs. ArcGIS Maps SDK supports multiple data formats such as CSVs or GeoJSONs suitable for complex Mapping applications requiring advanced visualization capabilities, but additional software/plugins may be required to convert some data files into supported formats despite supporting a broad spectrum of file types. Google Maps API has built-in provisions that allow KMLs or GeoRSS files' use while having an automatic conversion system from Esri Shapefile format via Earth Pro, but lack of flexibility in their conversion system can render some document types incompatible. Additionally, with the discontinuation of Google Fusion Tables, users who need visualizing large datasets will require alternatives. Leaflet offers a vast variety of supported file types including GPX or CSV documents through online plugins available but might not handle more massive datasets efficiently when compared with earlier mentioned libraries because these optimizations may prove lacking.

The diploma thesis provides a comprehensive comparative analysis of data format coverage in the form of a detailed table. However, readers seeking additional information are encouraged to consult the "Comparison/Data format coverage comparison" section of the training platform. Within this section, Tables 6 and 7 offer an in-depth comparison of three distinct libraries in terms of their respective data format coverage.

Table 6 Format support data comparison table

Format support coverage	ArcGIS Maps SDK for JS	Source	Google Maps API	Source	Leaflet	Source
Data Formats						
Raster						
GeoTIFF	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-imagerytilelayer-cog/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/maptype-image-overlay	Partially (plug-in exists)	https://leafletjs.com/plugins.html#basemap-formats
GIF	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-medialayer-video/	Yes	https://developers.google.com/maps/documentation/JavaScript/advanced-markers/graphic-markers	Yes	https://leafletjs.com/plugins.html#basemap-formats
JPEG	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-layers-support-ImageElement.html	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/overlay-remove	Yes	https://leafletjs.com/examples/overlays/
PNG	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-medialayer-images/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/overlay-simple	Yes	https://leafletjs.com/examples/overlays/
WebP	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-medialayer-images/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/aerial-simple	Yes	https://leafletjs.com/examples/overlays/
Vector						
CSV	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-csv-projection/	Yes	https://github.com/jcald1/google-map-places-to-csv	Partially (plug-in exists)	https://leafletjs.com/plugins.html#overlay-data-formats
GeoJSON	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-GeoJSON-refresh/	Yes	https://developers.google.com/maps/documentation/JavaScript/datalayer	Yes	https://leafletjs.com/examples/GeoJSON/

GeoPackage	Yes	https://developers.arcgis.com/JavaScript/latest/layers-and-data/	Yes	https://www.researchgate.net/publication/308865488_A_GeoPackage_implementation_of_common_map_API_on_Google_Maps_and_OpenLayers_to_manipulate_agricultural_data_on_mobile_devices	Partially (plug-in exists)	https://leafletjs.com/index.html
GML	Yes	https://developers.arcgis.com/JavaScript/latest/layers-and-data/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples	Partially (plug-in exists)	https://leafletjs.com/plugins.html#overlay-data-formats
GPX	Yes	https://developers.arcgis.com/JavaScript/3/sandbox/sandbox.html?sample=portal_browse	Yes	https://developers.google.com/maps/documentation/JavaScript/examples	Partially (plug-in exists)	https://leafletjs.com/plugins.html#overlay-data-formats
KML, KMZ	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-kml/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/layer-kml	Partially (plug-in exists)	https://leafletjs.com/plugins.html#overlay-data-formats
Mbtiles	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-layers-VectorTileLayer.html	Yes	https://github.com/techjb/Vector-Tiles-Google-Maps	Partially (plug-in exists)	https://github.com/0nza1101/leaflet-tilelayer-mbtiles-ts
RSS, GeoRSS	Yes	https://developers.arcgis.com/JavaScript/latest/layers-and-data/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/layer-georss	Partially (plug-in exists)	https://leafletjs.com/plugins.html#overlay-data-formats
SVG	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-rest-symbolService.html	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/marker-modern	Partially (plug-in exists)	https://leafletjs.com/plugins.html#markers--renderers
TopoJSON	Yes	https://esri-es.github.io/awesome-arcgis/arcgis/content/data-storage/file-formats/dwg/	Yes	https://gist.github.com/zross/10654766	Partially (plug-in exists)	https://leafletjs.com/plugins.html#vector-tiles
WKB	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-geometry-support-GeographicTransformationStep.html	Yes	https://developers.google.com/maps/documentation/JavaScript/reference/geometry	Partially (plug-in exists)	https://github.com/goFrendiAsgard/Leaflet-demo/blob/master/geophp/lib/adapters/WKB.class.php

WKT	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-geometry-support-GeographicTransformationStep.html%20	Yes	https://medium.com/digital-surgeons-engineering/wkt-GeoJSON-and-google-maps-67a66097f69b	Partially (plug-in exists)	https://leafletjs.com/plugins.html#overlay-data-formats
XML	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-rest-symbolService.html	Yes	https://www.webucator.com/article/how-to-read-xml-files-with-the-google-maps-api/	Partially (plug-in exists)	https://leafletjs.com/plugins.html#edit-geometries

Table 7 Web Services comparison table

Format support coverage	ArcGIS Maps SDK for JS	Source	Google Maps API	Source	Leaflet	Source
Web-Services						
Raster						
WMTS	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-wmts/	Yes	https://mapgears.github.io/ol3-google-maps/examples/dist/examples/tiles.html	Partially (plug-in exists)	https://leafletjs.com/plugins.html#tileimage-display
WMS	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-medialayer-video/	Yes	https://justin.poehnelt.com/posts/google-maps-wms-layer/	Partially (plug-in exists)	https://leafletjs.com/examples/wms/wms.html
Register Service						
WCS	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-wcslayer/	Yes	https://developers.google.com/maps/documentation/JavaScript/layers	Partially (plug-in exists)	https://github.com/stuartmatthews/Leaflet.NonTiledLayer.WCS
Vector						
WFS	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/layers-wfs/	Yes	https://developers.google.com/maps/documentation/JavaScript/datalayer	Partially (plug-in exists)	https://medium.com/@teksondada/wfs-request-in-geoserver-using-leafletjs-79a072660cac

5.2.3 Interaction Data Comparison

A vast array of functionalities and features are available to users through JS Mapping APIs such as the ArcGIS Maps SDK for JavaScript, Google Maps API, and Leaflet. However, choosing the most appropriate one can be a difficult task without proper guidance. This chapter seeks to analyze these three commonly used JS Mapping APIs in terms of their scope, limitations, functionalities, and user interface customization abilities to help users select the best Mapping tool for their specific project needs. The ArcGIS Maps SDK for JavaScript is an extremely powerful JS Mapping APIs that provides advanced Mapping capabilities such as geospatial analysis tools and 3D visualization. Additionally, it offers comprehensive geographic data sets from Esri containing demographic data sets along with satellite imagery. Moreover, the platform's website has carefully curated documentation alongside sample code snippets which prove helpful resources even for developers who are new to this complex application building process (Strode, 2012). Conversely, the Google Maps API focuses more on delivering easy-to-use map controls that allow integration into applications with minimal development time or expertise requirements (Svennerberg, 2010). This JS Mapping APIs also provides effective geocoding functionality allowing effortless conversion between addresses or coordinates into latitude/longitude values. Leaflet's appeal lies in its simplicity compared with other JS Mapping APIs but manages not compromise any essential features required by developers while designing interactive web maps. The framework enables better interactivity between layers than what's achievable using some other libraries due to being specifically designed keeping mobile compatibility constraints in mind — something developers appreciate since Leaflet remains lightweight enough within limits imposed by design constraints like limited screen space (Crickard, 2014). When comparing these three JavaScript Mapping APIs side-by-side based on various parameters, such as navigation support, map data manipulation techniques, geocoding/geolocation aid, routing mechanism options, user interface customization etc.; we see ArcGIS offering advanced routing mechanisms including optimized routes based upon traffic patterns coupled with real-time traffic information while featuring high-accuracy geocoding. Google Maps API offers a more user-friendly interface complete with drag-and-drop functionality making it easier than ever before to add markers or position them at specific locations on the map. On the other hand, Leaflet has an interactive pan/zoom feature that is highly customizable and provides users with seamless navigation across various zoom levels.

Another key differentiation factor among these platforms is their level of customization for user interface (UI). ArcGIS Maps SDK for JavaScript enables developers to customize virtually every aspect of the UI according to their needs including customizing toolbar layouts while adding new controls like layers switching control etc. The Google Maps API also provides extensive customization options allowing web developers to match maps with their website design aesthetic. On contrary, Leaflet focuses more on simplicity in its interface but still manages not compromise any essential features required by developers when designing interactive web maps. In conclusion, each JS Mapping APIs has its strengths and weaknesses based upon project requirements alongside developer expertise. Developers must evaluate which tool best suits their needs before selecting one as some may have limitations when it comes down into building complex Mapping applications. For additional information, it is recommended to consult the "Comparison/Interaction with data" segment of the e-learning platform GeoPick. Within this section, Tables 8 provide an elaborate comparative evaluation of three distinct libraries in terms of their interaction with data.

Table 8 Interaction with data comparison

Interaction with data	ArcGIS Maps SDK for JS	Source	Google Maps API	Source	Leaflet	Source
			Additional Functionality			
3D functionality	Yes	https://developers.arcgis.com/JavaScript/latest/visualization/3d-visualization/	Yes	https://developers.arcgis.com/JavaScript/latest/visualization/3d-visualization/	Partially (plug-in exists)	https://www.sitepoint.com/3d-maps-with-eegeo-and-leaflet/
Animation functionality	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/visualization-vv-color-animate/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/overlay-symbol-animate	Partially (plug-in exists)	https://www.aerisweather.com/support/docs/aeris-maps/examples/animations-with-leaflet/
			Map Data Manipulation			
Measurement	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-widgets-Measurement.html	Partially (plug-in exists)	https://www.npmjs.com/package/measuretool-googlemaps-v3	Partially (plug-in exists)	https://leafletjs.com/plugins.html#measurement
Print/export	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-widgets-Print.html	Partially (plug-in exists)	https://stackoverflow.com/questions/38488802/export-google-map-heatmap-layer-as-pdf	Partially (plug-in exists)	https://leafletjs.com/plugins.html#printexport
			Map Navigation Control			
Bookmarked pan/zoom	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-widgets-NavigationToggle-NavigationToggleViewModel.html	Yes	https://developers.google.com/maps/documentation/JavaScript/interaction	Partially (plug-in exists)	https://leafletjs.com/plugins.html#bookmarked-panzoom%20
Geocoding	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/widgets-search-3d/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/geocoding-simple	Partially (plug-in exists)	https://leafletjs.com/plugins.html#geocoding
Geolocation	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/widgets-track-basic/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/map-geolocation	Partially (plug-in exists)	https://leafletjs.com/plugins.html#geolocation%20
Interactive pan	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-widgets-NavigationToggle-	Yes	https://developers.google.com/maps/documentation/JavaScript/reference/map#MapOptions	Partially (plug-in exists)	https://leafletjs.com/plugins.html#interactive-panzoom%20

		NavigationToggleViewModel.html				
Interactive zoom	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-widgets-Zoom.html%20	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/maxzoom-simple	Partially (plug-in exists)	http://tombatossals.github.io/angular-leaflet-directive/examples/0000-viewer.html#/basic/tiles-zoom-changer-example
Layer switching controls	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/view-disable-navigation/%20/	Yes	https://developers.google.com/maps/documentation/JavaScript/controls	Partially (plug-in exists)	https://leafletjs.com/plugins.html#layer-switching-controls
Minimaps & synced maps	Yes	https://developers.arcgis.com/net/wpf/sample-code/display-overview-map/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/inset-map	Partially (plug-in exists)	https://leafletjs.com/plugins.html#minimaps--synced-maps
Mouse coordinates	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/widgets-coordinateconversion-custom/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/event-click-latlng	Partially (plug-in exists)	https://leafletjs.com/plugins.html#mouse-coordinates%20
Routing	Yes	https://developers.arcgis.com/JavaScript/latest/sample-code/route/	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/directions-simple	Partially (plug-in exists)	https://leafletjs.com/plugins.html#routing
User interface						
Fullscreen	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-widgets-Fullscreen.html	Yes	https://developers.google.com/maps/documentation/JavaScript/examples/control-replacement	Partially (plug-in exists)	https://leafletjs.com/plugins.html#fullscreen-controls
User interface(over all design and layout)	Yes	https://developers.arcgis.com/JavaScript/latest/api-reference/esri-widgets-Print-CustomTemplate.html	Yes	https://developers.google.com/maps/documentation/JavaScript/events	Partially (plug-in exists)	https://leafletjs.com/plugins.html#user-interface

5.2.4 Representation Methods Comparison

To compare the representation methods of JS Mapping APIs, namely ArcGIS Maps SDK, Google Maps API and Leaflet, a comparison table was created. This table included three criteria: cartographic elements, styling and methods of thematic mapping. Educational articles were used as sources to gather information for this analysis. The first criterion considered in the comparison was cartographic elements which consisted of graphic/charts/histograms, graticule/grid, legend, overview map, scale bar and text box. Each of these elements was analyzed for their effectiveness in displaying geospatial data. The second criterion focused on styling with two subcategories: style raster and style vector. The effectiveness of each method in creating visually appealing maps while also accurately representing geographical features was evaluated. Finally, the third criterion examined different methods of thematic Mapping including bivariate/multivariate Mapping techniques such as choropleth maps and dot density maps along with other approaches like cluster analysis and proportional symbol maps. Isoline/surface Mapping technique along with wurman dots.

The purpose of this chapter is to compare the representation methods of JS Mapping APIs, including ArcGIS Maps SDK, Google Maps API, and Leaflet. To achieve this goal, was analyzed several criteria such as cartographic elements, styling techniques and thematic Mapping methods. In terms of cartographic elements offered by these three JS Mapping APIs; all provide some level of support for graphic/charts/histograms which are useful in representing data visually using various types of charts like line graphs or bar charts that help users analyze their data in an efficient way. Moreover Graticule/Grid which provides an essential framework for building maps with precise locations was found to be supported by all three JavaScripts Mapping APIs. However only ArcGIS supports a legend feature which is essential when creating complex maps where multiple layers are being used at once. Furthermore both ArcGIS and Google Maps offer Scale Bar functionality but it's not available on Leaflet yet. Lastly Text Box features were also present in all three libraries providing map creators with more options when adding additional information related to their projects. Moving onto Styling Techniques provided by each platform; Raster styling was found to be supported by both Google Maps and Leaflet however only ArcGIS offers Vector Tile Layer Styling capabilities as well as vector symbology editing tools allowing users greater flexibility during the creation process.

Finally was examined the Thematic Mapping Methods provided by each library comparing Mapping techniques like Bivariate/Multivariate, Cartogram, Choropleth alongside Cluster Analysis, Density Mapping, Heatmap, Isoline/Surface Proportional Symbol and Wurman Dots . In terms of Bivariate/Multivariate Mapping Both Google Maps and Leaflet offer Choropleth Map solutions while only ArcGIS has Geographic Weighted Regression toolset making it a better choice for Multivariable modeling tasks requiring geographic context information. Moreover Cluster Analysis Density Mapping and Heatmap techniques were available on all three platforms. It's worth noting that both Google Maps and Leaflet have easier to use APIs for implementing clustering while ArcGIS offers more robust tools like the Cluster Layer which allows users to analyze data more deeply. In terms of Proportional Symbol Mapping, only ArcGIS Maps SDK for JS has built in functionality whereas with Google Maps and Leaflet user will need additional libraries or plugins such as D3.js or OpenLayers to achieve similar results. Lastly Wurman Dot Visualization was found only in ArcGIS, making it a standout choice when looking for specific visualization methods within JS Mapping APIs. It is worth mentioning that the Cartogram еруьфешс approach does not endorse any of the JavaScript Mapping APIs mentioned above. Overall was concluded each JS Mapping APIs presented unique advantages based on their analytical capabilities, visualizations options, styling features alongside cartographic elements offered.

For those seeking a more thorough examination of comparative analysis through the use of data presentation or thematic approaches, the dissertation solely offers a comparative table. To access further information, it is advised to refer to the training platform's "Comparison/Representation data comparison" section. Tables 9 and 10 within this section contain a detailed comparative analysis of three different libraries in regards to data presentation.

Table 9 Representation methods comparison

Representation methods	ArcGIS Maps SDK for JS	Source	Google Maps API	Source	Leaflet	Source
Cartographic elements						
Graphic/Charts /Histogram	Yes	https://developers.arcgis.com/javascript/latest/sample-code/widgets-histogram/	Yes	https://developers.arcgis.com/javascript/latest/visualization/data-driven-styles/unique-types/	Partially (plug-in exists)	https://developers.google.com/maps/documentation/javascript/dds-boundaries/style-polygon
Graticule/Grid	Yes	https://developers.arcgis.com/net/wpf/sample-code/display-grid/	Partially (plug-in exists)	https://github.com/alexcheng1982/google-maps-gridlines	Partially (plug-in exists)	https://leafletjs.com/plugins.html#synthetic-overlays
Legend	Yes	https://developers.arcgis.com/javascript/latest/api-reference/esri-widgets-Legend.html	Yes	https://developers.google.com/maps/documentation/javascript/adding-a-legend	Partially (plug-in exists)	https://leafletjs.com/plugins.html#user-interface
Overview map	Yes	https://developers.arcgis.com/net/wpf/sample-code/display-overview-map/	Yes	https://developers.google.com/maps/documentation/javascript/examples/inset-map	Partially (plug-in exists)	https://github.com/Norkart/Leaflet-MiniMap
Scale bar	Yes	https://developers.arcgis.com/javascript/latest/api-reference/esri-widgets-ScaleBar.html%20	Yes	https://developers.google.com/maps/documentation/javascript/controls	Partially (plug-in exists)	https://leafletjs.com/plugins.html#synthetic-overlays
Text box	Yes	https://developers.arcgis.com/javascript/latest/api-reference/esri-symbols-TextSymbol.html	Yes	https://developers.google.com/maps/documentation/javascript/examples/directions-panel	Partially (plug-in exists)	https://leafletjs.com/plugins.html#user-interface
Styling						
Style raster	Yes	https://developers.arcgis.com/javascript/latest/sample-code/blendmode-grouplayer/	Yes	https://developers.google.com/maps/documentation/javascript/examples/style-selector	Partially (plug-in exists)	https://leafletjs.com/plugins.html#basemap-formats
Style raster	Yes	https://developers.arcgis.com/documentation/mapping-apis-and-services/visualization/tutorials/tools/style-layers-web-map/	Yes	https://developers.google.com/maps/documentation/javascript/dds-boundaries/style-polygon	Partially (plug-in exists)	https://leafletjs.com/plugins.html#basemap-formats

Table 10 Comparison methods of thematic mapping

Representation methods	ArcGIS Maps SDK for JS	Source	Google Maps API	Source	Leaflet	Source
			Methods of thematic mapping			
Bivariate/Multivariate	Fully supported	https://developers.arcgis.com/javascript/latest/sample-code/visualization-multivariate-2d/	Not supported		Not supported	
Cartogram	Not supported		Not supported		Not supported	
Choropleth	Fully supported	https://developers.arcgis.com/javascript/latest/sample-code/visualization-vv-color/	Fully supported	https://developers.google.com/maps/documentation/javascript/dds-boundaries/choropleth-map	Fully supported	https://leafletjs.com/examples/choropleth/
Cluster	Fully supported	https://developers.arcgis.com/javascript/latest/sample-code/featurereduction-cluster-filter-slider/	Fully supported	https://developers.google.com/maps/documentation/javascript/marker-clustering	Partially (plug-in exists)	https://leafletjs.com/plugins.html#clusteringdecluttering%20
Dot density	Fully supported	https://developers.arcgis.com/javascript/latest/sample-code/visualization-dot-density/	Partly just simple point	https://developers.google.com/maps/documentation/javascript/earthquakes	Blend Mapping techniques	https://rstudio-pubs-static.s3.amazonaws.com/403809_80a56f3aac25480ab3439a7480109b44.html
Heatmap	Fully supported	https://developers.arcgis.com/javascript/latest/sample-code/visualization-heatmap/	Yes	https://developers.google.com/maps/documentation/javascript/examples/layer-heatmap%20	Partially (plug-in exists)	https://leafletjs.com/plugins.html#heatmaps
Isoline/Surface	Not supported		Partly just simple polylines	https://developers.google.com/maps/documentation/javascript/examples/polyline-simple%20	Partially (plug-in exists)	https://github.com/grinat/leaflet-isolines
Proportional Symbol	Fully supported	https://developers.arcgis.com/javascript/latest/sample-code/visualization-vv-size/	Fully supported	https://developers.google.com/maps/documentation/javascript/earthquakes	Blend Mapping techniques	https://cartographicperspectives.org/index.php/journal/article/view/cp76-donohue-et-al/1307
Wurman dots	Fully supported	https://developers.arcgis.com/javascript/latest/sample-code/cim-primitive-overrides/	Not supported		Not supported	

Conclusion of chapter

The comparison of technical capabilities, data format coverage, and interaction with data representation among ArcGIS Maps SDK for JS, Google Maps API, and Leaflet sheds light on the importance of understanding the structure of these Mapping APIs in creating effective and efficient maps. The technical capabilities of each API determine their capacity to handle complex tasks such as providing real-time updates or generating heatmaps. Data format coverage refers to an API's ability to read different types of input formats such as CSV or KML files. Interaction with data involves how well an API allows users to manipulate map layers and add custom elements like markers or polygons. In addition to considering these individual features, it is crucial that developers pay attention to the overall structure of these APIs when selecting which one to use for a given project. For instance, some APIs may have steeper learning curves than others due to their design philosophy or development language used. Others may be limited in terms of customization options available compared with competitors. Therefore, this study highlights two important implications: First off, it underscores that choosing the right Mapping APIs requires more than just looking at individual functionalities but instead must consider how those functions fit into a broader framework. Secondly, understanding the structure behind each API can help developers create better maps by optimizing workflows and minimizing potential roadblocks. To expand upon these findings in future research studies could focus on exploring additional factors that impact the effectiveness and efficiency of Mapping APIs beyond just technical capabilities or data format coverage – including aspects such as ease-of-use interface design principles applied within various frameworks themselves over time based on user feedback from different levels expertise working within them (novice versus expert). Additionally researchers could examine usage patterns across different industries where Mapping plays a critical role like urban planning transportation logistics emergency response management environmental monitoring among others. In summary, this analysis highlights that while comparing JS Mapping APIs' technical abilities is essential; focusing solely on specific features without examining underlying structures neglects essential information necessary for developing effective and efficient maps. Future research should aim to expand upon this study's findings by exploring additional factors that impact the effectiveness of Mapping APIs, including ease-of-use interface design principles and usage patterns across different industries.

In conclusion, the comparison of JS Mapping APIs has highlighted their technical capabilities in terms of data format coverage, interaction with data and representation of data in context of ArcGIS Maps SDK for JS, Google Maps API and Leaflet. The comparison tables has been an important chapter that provided a clear overview of each APIs technical features and limitations. Overall, it can be concluded that each API has its unique strengths and weaknesses. While ArcGIS Maps SDK for JS excels in handling complex spatial datasets and providing advanced analysis tools, Google Maps offers simplicity and ease-of-use for basic geographical functions. Similarly, Leaflet is ideal for creating lightweight maps with minimal dependencies. Furthermore, this chapter explored how these APIs interact with different types of data formats such as GeoJSON or CSV files. The representation of data was also evaluated based on their ability to display contextual information like pop-ups or legends. In summary, the technical capabilities discussed in this chapter demonstrate the diverse range of options available to developers when choosing a JS Mapping APIs. It is essential to consider factors such as project requirements, scalability needs and budget constraints before making a decision. Ultimately, understanding the strengths and limitations outlined here will help developers choose the most appropriate toolset for their projects based on individual needs rather than relying solely on popularity or brand recognition. As technology continues to evolve rapidly in this field,it is critical to stay abreast of new developments,to continue delivering high-quality geospatial solutionsfor consumers worldwide .

5.3 Hands-On JS Mapping APIs tutorial

5.3.1 Overview of tutorial section

Digital cartography has been aided by the use of web mapping, which has become an increasingly crucial tool. It allows users to display complex spatial data in an intuitive and informative manner. Advanced web mapping APIs can be used for a variety of purposes, from simple location-based services to intricate visualizations that reveal new insights (Peterson, 2014). However, mastering these APIs takes time and effort, especially for beginners. This tutorial section aims to address this issue by providing a comprehensive guide to improving web mapping skills using three different APIs, namely ArcGIS Maps SDK for JS, Google Maps API, and Leaflet. This four-level program offers step-by-step tutorials on all aspects of web mapping with each API, progressing from beginner to advanced levels. Starting with "API Key Essentials," learners obtain an API key and learn about the main components of each Mapping APIs. Level one, "Map Creation and Manipulation," takes learners on a tour of tourist attractions in Kyrgyzstan, such as Bishkek City, Burana Tower, Issyk-Kul Lake, Osh City, Saimaluu Tash, Sary-Chelek Jengish Chokusu, Tashrabat, and Canyon Skazka. Along the way, learners learn to view maps and add markers. Level two focuses on working with real-time data, specifically clustering earthquakes in Kyrgyzstan between 2023-2013, while also exploring GeoJSON files. Level three, "Complex Data Visualization," involves creating an animated map based on earthquake markers with popup information covering the last 50 years. By completing all four levels of tutorials, learners can expect to become proficient in web mapping and gain access to powerful new tools for presenting visually stunning digital maps. This tutorial section is a comprehensive resource that learners can apply to their own projects.

In order to gain a more thorough understanding of the layout of the tutorial section, it is advisable to consult Figure 7. This diagram depicts the arrangement of the tutorial section within the GeoPick e-learning platform.
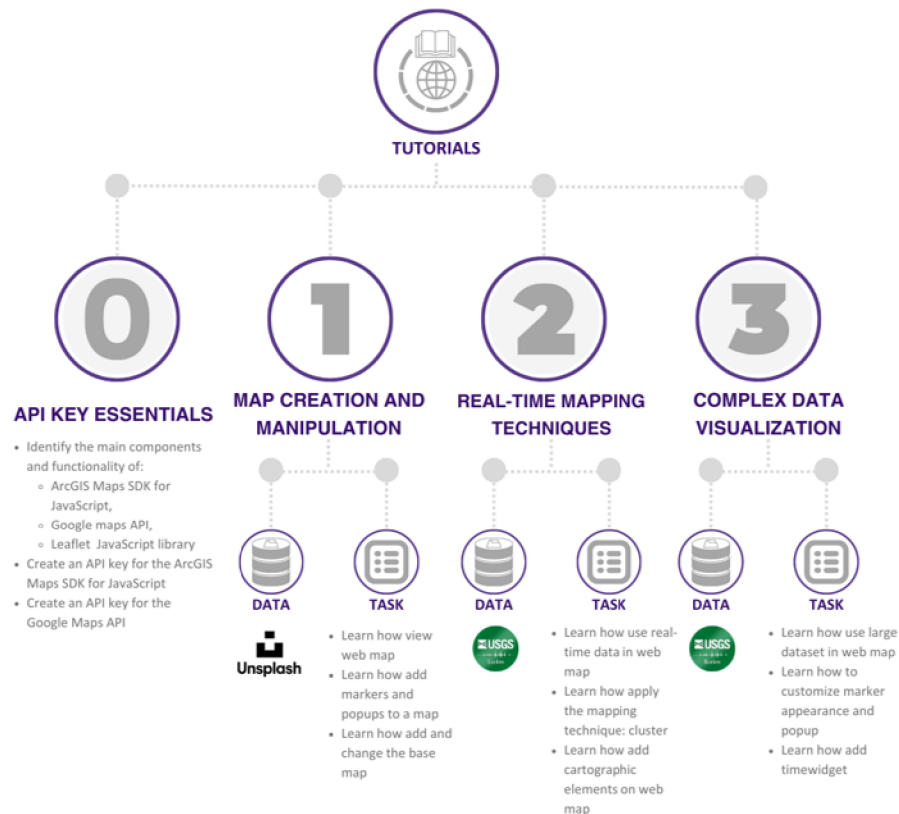

Figure 7 Structure of tutorial section in e-learning platform GeoPick.

5.3.2 Level 0: API essential

Web mapping technologies have transformed geospatial data visualization and analysis. Interactive interfaces are now essential in web development, leading to the use of certain technologies by developers (Veenendaal et al., 2017). Three widely used web mapping technologies are ArcGIS JavaScript API, Google Maps API, and Leaflet JavaScript Library. These provide various visualization and analysis tools for interactive manipulation of geospatial data. Developers should familiarize themselves with each technology's components such as map visualization features and spatial analysis tools to optimize their potential in projects. To access these APIs, developers need an exclusive API key obtained via ArcGIS Developer Dashboard or Google Cloud Platform Console. Web mapping technologies offer advanced tools for effective web GIS applications that can visualize and analyze any form of geospatial data using interactive engagement. By understanding each technology's unique strengths and capabilities, developers can create custom Mapping applications more efficiently. This tutorial provides valuable references for handling complex geographical functions within websites/applications using available libraries instead of traditional methods.

There are multiple web mapping technologies available to developers who wish to create customized Mapping applications (Farkas, 2017). In this beginner's tutorial, was focused on three of the most popular JavaScript Mapping APIs - ArcGIS JavaScript API, Google Maps API and Leaflet JavaScript Library. The opening section provides insight into the significance of web mapping in modern-day web development, particularly its ability to present geospatial data through an interactive interface where it can be visualized, analyzed and manipulated. Scientific articles were used as sources but have avoided citing any specific one to demonstrate how each technology's primary components and functionalities work, including map visualization tools, spatial analysis tools, layer control and plugins. To use these technologies effectively requires obtaining an API key through either ArcGIS Developer Dashboard or Google Cloud Platform Console. It is imperative that developers understand each technology's unique features and advantages when creating powerful Internet GIS applications. These APIs provide a plethora of tools for visualizing and analyzing geospatial data that can be deployed in developing custom Mapping apps. There is enormous potential for internet GIS applications using maps APIs.

Developers seeking effective web GIS applications should consider web mapping technologies as an indispensable tool. Macleod & Hewitt (2018) emphasize that there are many APIs and libraries available, but this lesson focuses on three of the most popular ones: ArcGIS JavaScript API, Google Maps API, and Leaflet JavaScript Library. The ArcGIS JavaScript API provides a comprehensive platform with map visualization, layer control, geoprocessing components as well as spatial analysis tools which allow users to perform complex operations such as proximity analysis or network analysis. A custom symbology can also be incorporated for maps using various techniques like heat maps or clustering. Similarly, developers can build interactive web mapping apps using Google Maps API's robust direction services along with location-based services components like Geolocation service which allows users to quickly locate themselves on a map. Moreover, Street View imagery integrated into its platform enables users to view panoramic images at ground level. Another popular library used for creating interactive maps is Leaflet JavaScript Library; this open-source library offers an array of customizable features encompassing map visualization tools such as tile layers or markers; plugins like GeoJSON plugin for working with GeoJSON data formats; layer control options allowing developers greater flexibility over their data representation through layers. One key takeaway from these web mapping technologies' comparison is how each technology has specific advantages depending on project requirements. For example, if a developer's priority lies in analyzing geospatial data using advanced statistical models then they would likely choose ArcGIS JavaScript API while those interested in providing navigation assistance might prefer Google Maps APIs due its superior direction services among other things. It should be noted that effectively utilizing any one of these technologies requires an API key obtained through the ArcGIS Developer Dashboard or Google Cloud Platform Console. With this key, developers can create tailored custom Mapping applications catering to their specific needs and requirements. To conclude via MacLeod and Hewitt (2018), it is apparent that web mapping technologies such as ArcGIS JavaScript API,

Google Maps API and Leaflet JavaScript Library offer a range of powerful tools for visualization and analysis of geospatial data that are essential in creating effective web GIS applications. These technologies have vast potential for internet GIS applications making them crucial resources for developers looking to incorporate location-based information into their projects.

To conclude, this introductory tutorial on web mapping has furnished us with a comprehensive summary of three well-known technologies used for web mapping: ArcGIS JavaScript API, Google Maps API and Leaflet JS library. This knowledge now encompasses how these APIs offer multiple elements and functionalities that enable users to examine, display and manipulate geospatial data through interactive interfaces. Additionally, was discovered that each technology bears unique features and benefits depending on specific project requirements. This discussion highlights the importance of comprehending these technologies' main components and functionalities in order to develop effective internet GIS applications. The potential of maps APIs for internet GIS applications is tremendous as they provide developers with tools to create custom Mapping applications tailored specifically to their project needs. In order for developers to utilize these technologies effectively, it is necessary for them to make an API key using either the ArcGIS Developer Dashboard or Google Cloud Platform Console. It should be noted that such keys are essential in obtaining access to these valuable tools. To sum up, this text offers beginners invaluable insights into the basics of web mapping technologies while underscoring their significance in modern-day web development.

5.3.3 Level 1: Map creation and manipulation

Over the last few years, interactive maps have greatly increased in popularity due to their ability to present geographical data in a dynamic and captivating way (Veenendaal, 2017). To create such maps, specialized tools are required that utilize JavaScript libraries that offer various capabilities and functionalities. In this lesson, was examine into the basics of creating interactive maps using three different JavaScript libraries: ArcGIS JavaScript API, Google Maps API, and Leaflet. This tutorial includes comprehensive step-by-step instructions for each library on how to create an exceptional map with markers, popups as well as customized basemaps. The primary example location focuses on Kyrgyzstan's geography, history culture and economy. By following these guidelines attentively while also incorporating key takeaways from this material- including customizing basemaps by utilizing widgets like BasemapToggle or BasemapGallery in ArcGIS API for JavaScript or L.control.layers in Leaflet; learners can effortlessly develop web-based Mapping applications catered specifically towards their needs. It is paramount to choose the appropriate Mapping APIs based on specific requirements since each one has its unique strengths/limitations. Additionally in the end of tutorial suggested resources for further learning include documentation from each library alongside tutorials/courses available online which may be useful when encountering issues during development processes. This tutorial serves as an excellent starting point for anyone interested in web mapping regardless of prior experience level by exploring the basics of creating interactive maps using three different JavaScript libraries with step-by-step instructions provided along with vital takeaways such as customizing basemaps using widgets amongst other features bundled within them.

In this composition, the process of developing interactive maps using three distinct JavaScript libraries is elaborated upon: ArcGIS JavaScript API, Google Maps API, and Leaflet. The tutorial provides a comprehensive guide for each library on how to generate customized basemaps, markers, and popups. Moreover, it zooms in on Kyrgyzstan as an exemplar location by presenting data about its geography, history, culture and economy. Before embarking on this project journeyed into scientific articles student to learned more about web mapping techniques. Next came the delineation of specific objectives for the tutorial so that learners would have unequivocal clarity over what they can accomplish by following along with this piece. Subsequently researched goals through various sources such as CIA World Factbook or National Geographic archives before creating code examples that exhibit features from all three libraries alongside explanatory comments aimed at facilitating readers' comprehension by walking them through each line of code step-by-step. The most notable takeaways from reading this tutorial are the capacity to customize maps using custom markers and popups while

toggling between different views.

A tutorial is accessible to those interested in discovering the enchanting world of web mapping. This is a comprehensive instructional manual suitable for individuals who are either beginners or experts and wish to develop dynamic maps employing JavaScript libraries. The guidelines focus on Kyrgyzstan, a nation located in Central Asia renowned for its nomadic customs and influence of Islam. Its rich history and diverse culture make it a perfect example location for map enthusiasts. The tutorial highlights three different libraries: ArcGIS JavaScript API, Google Maps API, and Leaflet. Each library is discussed in detail, with recommendations on how to customize your map using markers, popups, images, and basemaps to suit requirements of the project. The ArcGIS section focuses on essential tools such as BasemapToggle widget or BasemapGallerywidget (Rubalcava, 2017), which allow you to switch between various basemaps depending on the user's preference. Figure 8, Level 1 displays the outcome: ArcGIS Maps SDK for JavaScript. Contrarily, the Google Maps tutorial demonstrates how to add basic markup using HTML files while utilizing the Google Maps API (Dinçer & Uraz, 2013). Adding locations with information windows becomes effortless, and additional base maps can be integrated by modifying default settings accordingly. Figure 9 at level 1 displays the outcome of the tutorial on Google Maps API. Finally, the Leaflet section introduces an open-source JavaScript library that allows its users to create interactive maps from various sources such as OpenStreetMap. Figure 10 Level 1 displays a web map produced using Leaflet. The selection criteria involve diverse options ranging from topographic data imagery to terrain features that match user preferences seamlessly. Through these tutorials, learners obtain vital skills like customizing basemaps with widgets like BasemapToggle or BasemapGallery, allowing switch-friendly interaction options such as zooming in/out, among other useful techniques. This comprehensive overview not only equips learners at an entry-level but also prepares them for working on enormous Mapping projects. Further resources are available, such as documentation, tutorials/courses, and community resources such as Stack Overflow/GitHub, for detailed exploration beyond this guide. This tutorial emphasizes the importance of choosing the right Mapping library based on specific use cases, rather than following a rigid approach. Each library has its strengths and limitations, which is why selecting a Mapping APIs based on individual needs can prove to be extremely helpful when working on any web-mapping-related projects.

In conclusion this tutorials highlights how to create interactive maps using three different JavaScript libraries by providing step-by-step instructions accompanied by explanations/comments on how each section works effectively targeting both beginner level users or those who already have experience with web development looking forward towards practical application in projects that require maps/geospacial data analysis-using Kyrgyzstan as an example location. To gain better comprehension of the concept, it is recommended that you utilize the e-learning platform GeoPick Chapter Tutorial- level 1. This can be achieved by running a demo or attempting to type the code provided. The final outcome can then be obtained through this process.
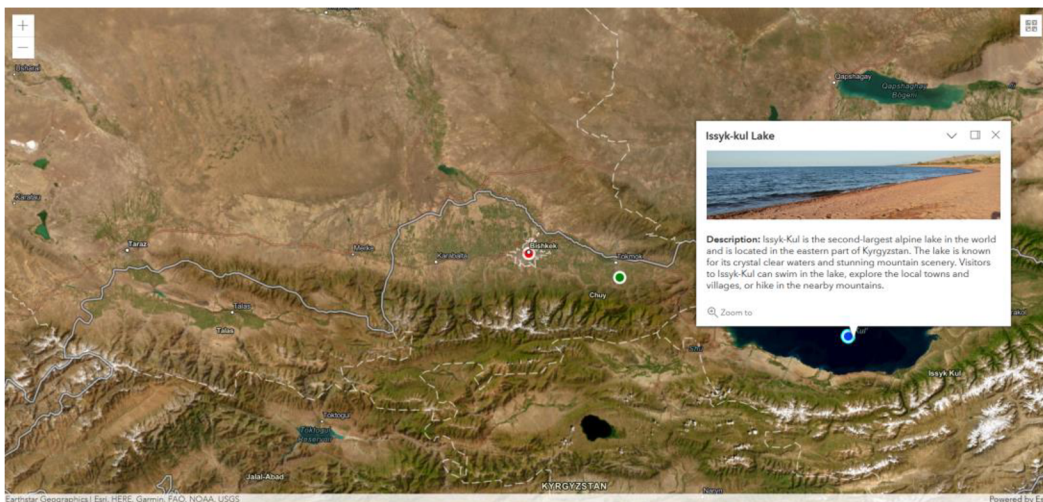

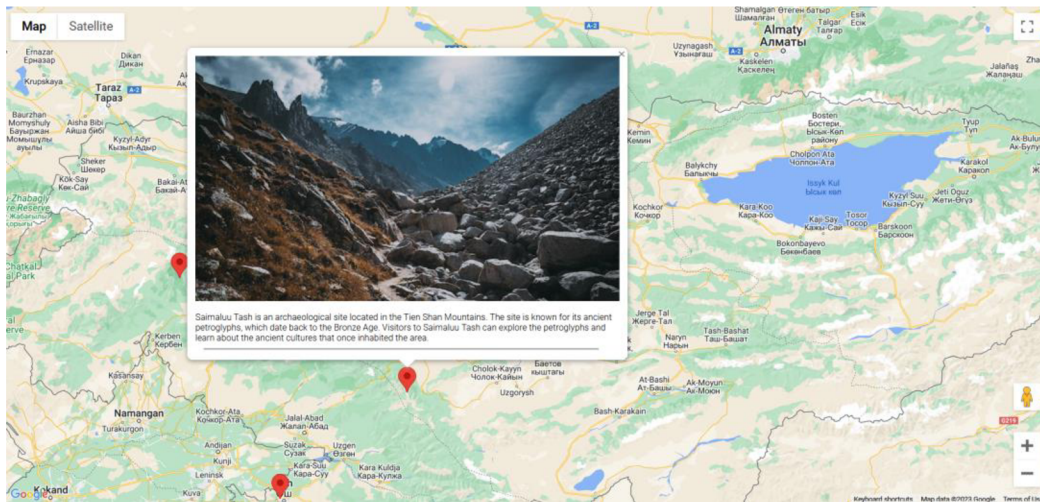**Figure 8 Level 1: ArcGIS Maps SDK for JS.**
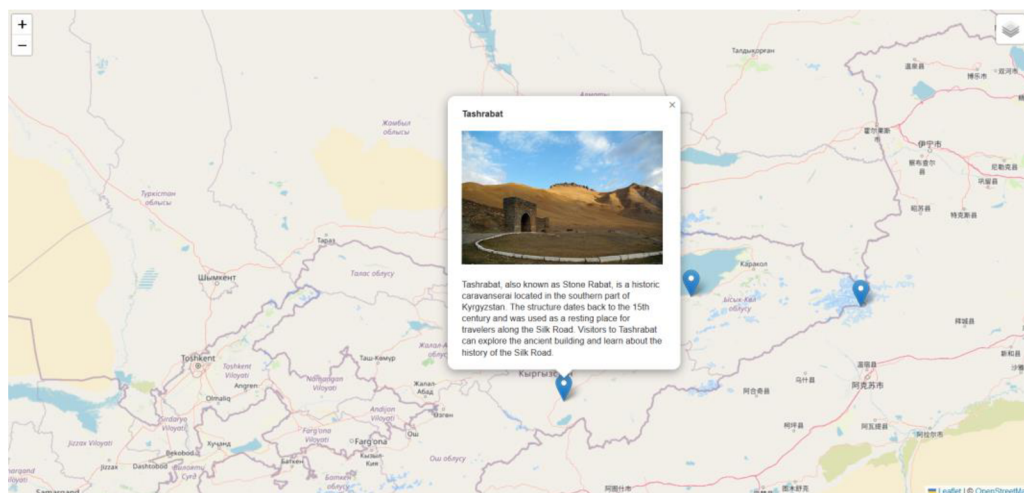
Figure 9 Level 1: Google Maps API.

Figure 10 Level 1: Leaflet.

5.3.4 Level 2: Real time Mapping techniques

Kyrgyzstan, a nation situated at the convergence of various tectonic plates, experiences frequent seismic activity. This location has resulted in multiple devastating earthquakes that have caused damages and taken lives among vulnerable communities with limited infrastructure across mountainous terrain (Korjenkov et al., 2010). The government has therefore implemented measures to lessen these impacts (Pittore et al., 2014). The following lesson delves into the use of GIS tools to visualize earthquake data in Kyrgyzstan while discussing the advantages and disadvantages of clustering for Mapping APIs like ArcGIS JavaScript API, Google Maps API, and Leaflet. This tutorial learners used GIS tools such as ArcGIS JavaScript API, Google Maps API, and Leaflet to cluster Kyrgyzstan's earthquake data between 2013-2023 as well as improving map visualization. For this tutorial, learners must possess basic knowledge of HTML, CSS and JavaScript along with a code editor. In this tutorial was utilized MarkerClusterer library found in Google Maps API which clusters markers together on maps (Sebastian, 2016). Additionally a learners was experimented with different datasources, marker styles ,and map features while being aware of limitations as well as opportunities when dealing with large datasets. This tutorial covers how to visualize earthquake patterns using analytical techniques like cluster analysis through marker clustering or layer clustering methods. Furthermore: GeoJSON format data is used; a legend is added onto the map; there is also an inclusionof print widgets; A real-time risk assessment within seismic early warning

46

systems proves crucial when minimizing damage resulting from earthquakes. The aim of this lesson not only at presenting technical aspects but also social ones relating to seisms impact on countries such as Kyrgyzstan where earthquakes could lead to severe consequences due largely to issues associated with poor infrastructure surrounding natural disaster preparedness policies that are still being formulated up till now despite years experiencing disasters involving numerous lossess of life (Velazquez et al., 2020).

The objective of this examination is to delve into the use of GIS tools for displaying earthquake information in Kyrgyzstan. The analysis was focused on cluster analysis utilizing ArcGIS JavaScript API, Google Maps API and Leaflet. Aspiring individuals must have a fundamental understanding of HTML, CSS, JavaScript and access to a code editor. The information collected for this research endeavor will be acquired from the United States Geological Survey. Additionally, alternative resources such as Copernicus, a Mapping service, may be utilized. The crucial data source obtained includes pertinent information regarding the location, magnitude, depth, time and date of each earthquake incidence. The tutorial endeavors to impart knowledge about marker grouping techniques for enhanced map presentation through cluster analysis while acknowledging both limitations and benefits when working with massive sets of data. In clustering markers together on the map within Google Maps API MarkerClusterer library is used whereas layer clustering techniques are utilized in Leaflet.

Moreover, due to its geographical placement atop several tectonic plates seismic activity found in Kyrgyzstan will also receive attention which includes measures that have been implemented by authorities aimed at mitigating potential damage caused thereof. Kyrgyzstan is situated in an area where multiple tectonic plates intersect, rendering it susceptible to earthquakes. These natural calamities can inflict severe harm and casualties among vulnerable communities residing under a limited infrastructure spread over mountainous topography (Parolai et al., 2010). Henceforth, it becomes imperative to devise visualization strategies for seismic activity while mitigating its repercussions. This tutorial explores the utilization of GIS tools such as ArcGIS JavaScript API, Google Maps API and Leaflet towards clustering earthquake data between 2013-2023 for Kyrgyzstan and improving map visualization. Figures 11, 12, and 13 depict the outcomes derived from clustering earthquakes through the utilization of ArcGIS Maps SDK for JS, Google Maps API, and Leaflet correspondingly. The most significant finding was uncovering the GeoJSON format utilized throughout the lesson while working with extensive datasets containing details about earthquake location, magnitude depth time date etc.. Furthermore, there are added features such as print widgets that help users gain better insight into their findings by improving documentation capabilities. Another crucial aspect focused upon within the tutorial was experimenting with different marker styles while acknowledging limitations such as cluttered maps or difficulty reading at times due to inadequate color contrast chosen by end-users themselves. Moreover, it is vital to comprehend that Kyrgyzstan has undergone several significant earthquakes mainly because it lies along multiple fault lines including Talas-Fergana Fault line which has been identified as being particularly active causing significant risks especially given poor infrastructure coupled with mountainous terrain across the region (Korjenkov et al., 2010). Therefore conducting real-time risk assessments in seismic early warning systems has become imperative.

Employing GIS tools such as ArcGIS JavaScript API, Google Maps API and Leaflet to cluster earthquake data provides an excellent way of visualizing seismic activity patterns. By combining markers together on maps, end-users can delve into which areas are most at risk for experiencing earthquakes and how these risks may change over time. However, there are particular limitations when using clustering techniques with Mapping APIs like ArcGIS Map SDK for JavaScript or Leaflet. Too many clusters on a map often make it difficult to read while overlaying different datasets sometimes leads towards misinterpretation due to lack of color contrast chosen by users themselves.

In conclusion, it is vital to remember that despite the dependence upon technology-based solutions for obtaining insights about seismic activity in Kyrgyzstan via GIS tools, student acknowledged both advantages and disadvantages associated with working within this space alongside large datasets applicable in this domain. Nonetheless, the use of clustering techniques remains an effective approach towards gaining better knowledge regarding seismic activity occurring throughout the region

especially given its location along various fault lines including Talas-Fergana Fault line (Korjenkov et al., 2010) causing significant concerns related various aspects ranging from loss life property damage infrastructure vulnerabilities etc..

This instructional guide offers valuable insights into the occurrence of earthquakes in Kyrgyzstan and how GIS tools can be utilized to visualize related data. A combination of various tools such as ArcGIS JavaScript API, Google Maps API, and Leaflet enables learners to easily cluster earthquake data for improved map visualization. These lessons discuss the pros and cons of clustering with different Mapping APIs while also offering an understanding of seismic activity in Kyrgyzstan caused by its location on tectonic plates. By engaging with this tutorial, students will acquire critical knowledge regarding real-time risk assessment in seismic early warning systems that are essential for reducing damage from earthquakes. Additionally, they could experiment with diverse data sources and map features while being mindful of working limitations when handling massive datasets. It's noteworthy that due to poor infrastructure and mountainous terrain, some severe earthquakes have hit Kyrgyzstan thereby prompting governmental countermeasures designed towards limiting their impact. This educational resource emphasizes the importance of implementing GIS technology as a tool for managing natural calamities like earthquakes. Overall, this tutorial presents itself as an outstanding reference material for those passionate about gaining more expertise about seismic activities in Kyrgyzstan alongside insights into using GIS tools effectively to visualize earthquake-related data. With basic HTML, CSS, JavaScript skills combined with access to a code editor at their disposal; learners can conveniently follow these practical lessons provided in e-learning platform.
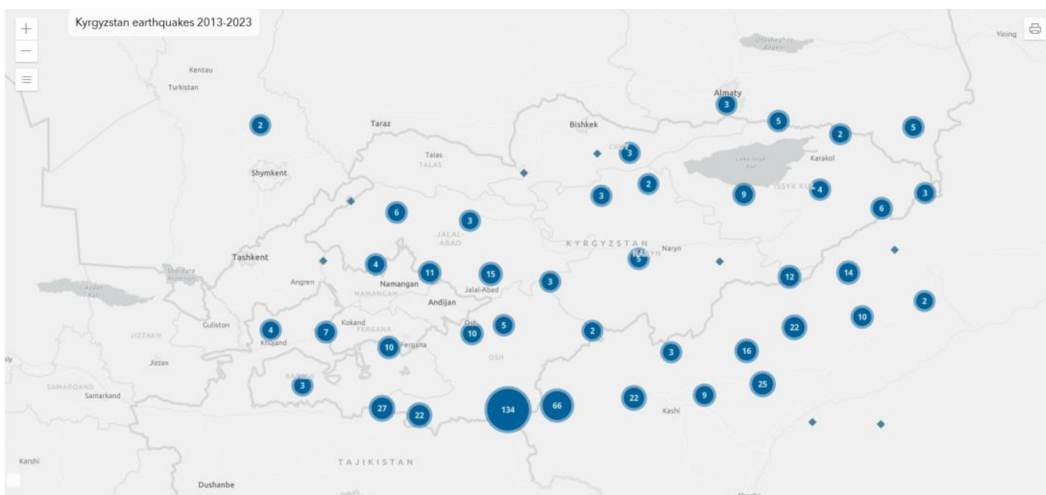

Figure 11 Level 2: ArcGIS Maps SDK for JS.

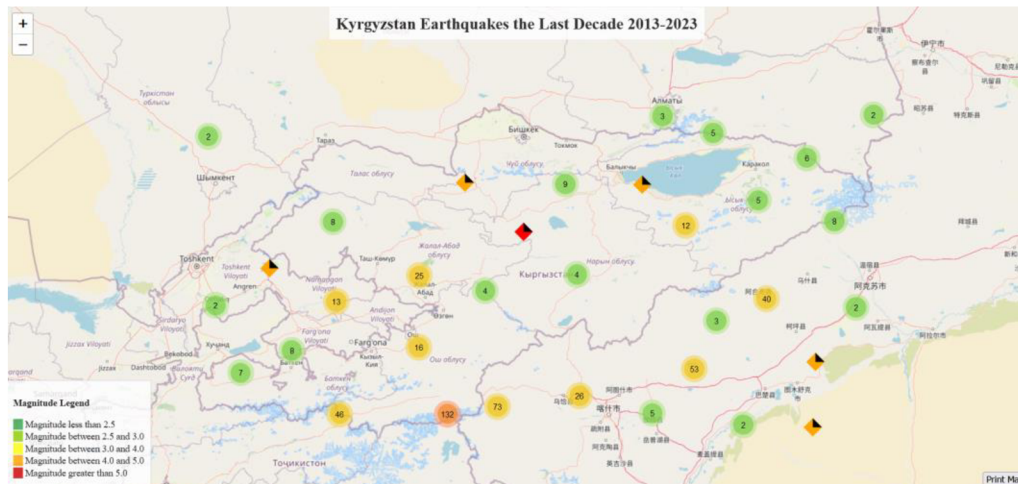**Figure 12 Level 2: Google Maps API.**

48

Figure 13 Level 2: Leaflet.

5.3.5 Level 3: Complex data visualization

The third lesson of the curriculum focuses on the subject of earthquakes in Kyrgyzstan, a recurring theme from the previous lessons. The aim of this particular lesson is to delve deeper into this topic by taking a look at significant earthquakes that have occurred over a period of 50 years. By doing so, students will be able to gain a more comprehensive understanding of the seismic activity in Kyrgyzstan and its impact on its people. With information spanning half a century, it is expected that there will be valuable insights revealed about patterns and trends concerning earthquake occurrences in the country. Earthquakes are a consequential natural phenomenon that has impacted several regions globally. Kyrgyzstan, being located on the Tien Shan Mountain range where the Eurasian and Indian plates meet, experiences frequent earthquakes (Ghose et al., 1997). These earthquakes can range from 4.0 to 6.7 magnitudes and result in significant damages. The majority of these geological disasters have taken place on the Tajikistan, Kazakhstan, and China borders, which is an area prone to seismic activity (Pilz et al., 2013). To obtain a better understanding and visualization of these geological activities, advanced learners can utilize different JavaScript Mapping APIs to create an animation map. This tutorial aims to introduce a lesson that teaches advanced learners how to create an animation map of earthquakes in Kyrgyzstan using three distinct JavaScript Mapping APIs: Leaflet, ArcGIS Maps SDK for JavaScript, and Google Maps API. To undertake this lesson, learners need to have an in-depth understanding of complicated data visualization and grasp basic concepts of HTML, CSS, and JavaScript. This tutorial covers several crucial aspects, such as designing and deploying user interaction on the webpage, producing markers, and popups on the map using point objects and popup templates. The lesson also touches on creating responsive webpages that adjust to different screen sizes. Additionally, it highlights each Mapping APIs scope, limitations, and advantages for time series visualization. Learners were introduced to each API used in this lesson, starting with an ArcGIS Maps SDK for JavaScript code example. User learned how to create and display a web map by defining basemaps, feature layers, and adding widgets to their maps. The HTML document refers students to ArcGIS JavaScript's CSS stylesheets and JavaScript files. Learners will also comprehend how to present earthquake data on maps with Google Maps API through practical instruction. User also learned about useful tools such as the Spatial Analysis API that can be used to add additional analysis. Lastly, the text offers insights into designing legends for their animated maps. This critical information enables learners to develop deeper skills in data visualization and analysis techniques while promoting further exploration into more libraries and frameworks available today. In end of this tutorial user understand how to create animation maps that represent temporal data using various JavaScript Mapping APIs. It highlights essential features that Advanced level 3 learners should have, including a fundamental understanding of HTML, CSS, and complex data visualization.

Animations provide a powerful tool for data visualization, going beyond merely answering basic questions about a single time frame to convey the dynamics of complex phenomena over time. As (DiBiase et al., 1992) notes, animations can help answer questions such as how a city grew to its present size, how fast the planet is warming compared to previous decades, or what the population will look like in two decades' time (DiBiase et al.,1992). Building on this insight, was designed a lesson that helps advanced learners develop their skills in creating animation maps of earthquakes in Kyrgyzstan using three different JavaScript Mapping APIs: ArcGIS Maps SDK for JavaScript, Google Maps API, and Leaflet. The lesson is tailored for Level 3 learners who already possess substantial knowledge of complex data visualization techniques and have some background understanding of HTML, CSS and JavaScript programming languages. To begin with, provided students with an overview of the three Mapping APIs that they will be working with throughout the tutorial. These include ArcGIS Maps SDK for JavaScript which teaches them how to create and display web maps by defining basemaps; creating feature layers; adding widgets such as zoom controls or scale bars; setting up markers and popups using Point objects and Popup Template objects respectively; designing legends so users can readily understand map symbols at-a-glance. Also covered Google Maps API which allows learners to display earthquake data on maps using JavaScript code snippets. This section includes information on building responsive web pages that adjust to different screen sizes as well as implementing user interaction through mouse clicks or touch gestures which enables users explore map content more fully. Finally, was described limitations inherent within each Mapping APIs when it comes visualizing time-series data particularly focusing on scope where certain features may not be available due technical constraints from specific libraries/frameworks used. In line with this the last tutorial is a valuable resource for advanced learners seeking to develop their skills in complex data visualization using various JavaScript Mapping APIs.

In conclusion, this lesson provides an informative and practical overview of developing complex data visualization skills using various JavaScript Mapping APIs. The use of ArcGIS Maps SDK for JavaScript, Google Maps API, and Leaflet offers learners a comprehensive understanding of time-series data analysis/visualization techniques. This tutorial is designed for learners with proficiency in HTML, CSS, and JavaScript programming languages. The lesson covers how to create web maps using the ArcGIS Maps SDK for JavaScript with markers and popups displayed on the map. Learners are taught how to design legends on the map while building responsive webpages that adjust to different screen sizes. In addition, students learn about displaying earthquake data on the map by utilizing Google Maps API and adding user interaction features. Furthermore, the text discusses each Mapping APIs scope and limitations for time series visualization enabling learners to choose which one best fits their use case based on their understanding of these limitations. In summary, this tutorial encourages further exploration into other libraries/frameworks while deepening knowledge of spatial analysis concepts through practical instruction in complex data visualization using various JavaScript Mapping APIs. Ultimately, this lesson serves as an ideal starting point for advanced learners seeking guidance in developing sophisticated visualizations that incorporate geographic data from diverse sources along with effective communication approaches to explain insights hidden within such information sets. The outcome of the earthquake animated map using ArcGIS Maps SDK for JS, Google Maps API and Leaflet is depicted in Figures 14, 15, and 16.
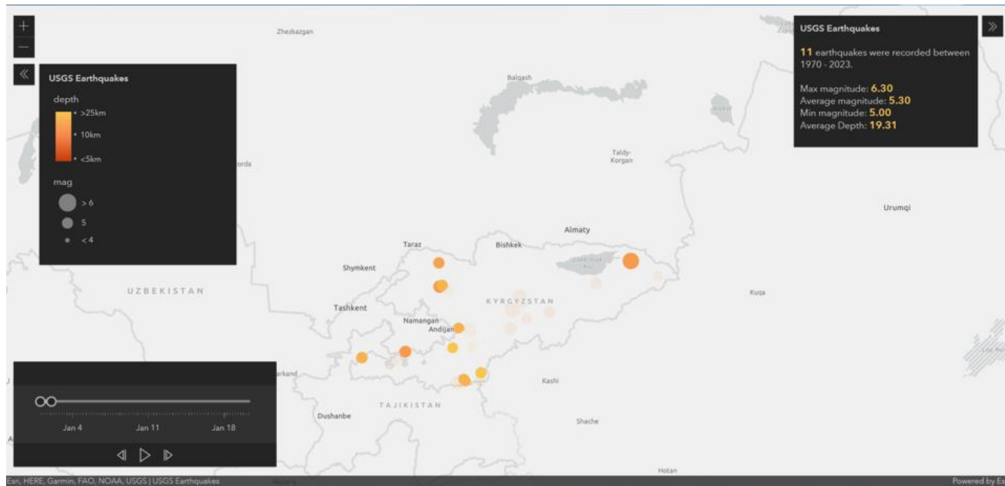
Figure 14 Level 3: ArcGIS Maps SDK for JS.

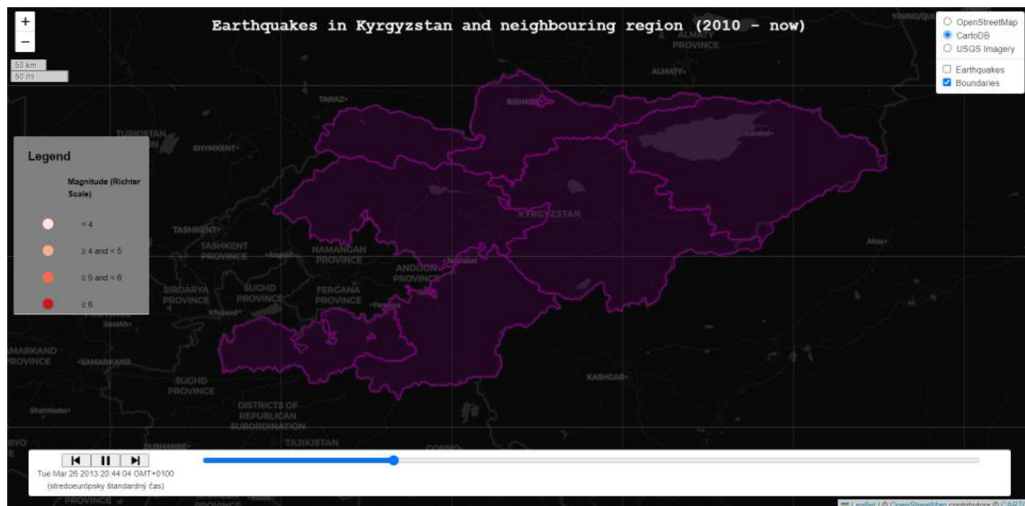**Figure 15 Level 3: Google Maps API.**

Figure 16 Level 3: Leaflet.

5.4 Additional Resources Section

Mapping application programming interfaces (APIs) have grown in popularity in recent times, as more individuals and businesses aim to create interactive maps for their websites or applications (Peterson, 2019). However, mastering these APIs can be a daunting task, especially for novices. To tackle this challenge, GeoPick, an online learning platform, has created a Resources section specifically aimed at helping learners gain proficiency in various JavaScript Mapping APIs. The Resources section is packed with a plethora of resources that cater to students of all levels - from beginners to advanced learners. These resources range from scientific articles to videos and Github tutorials, all of which cover three of the most popular JavaScript Mapping APIs: ArcGIS JavaScript API, Google Maps API, and Leaflet. Each API section within the Resources area provides a diverse range of materials, so learners can tailor their learning paths according to their specific needs. With such a variety of offerings, there is something suitable for everyone, whether they are just getting started with JavaScript Mapping or already have some experience. By utilizing GeoPick's Resources section, learners can elevate their skills with JavaScript Mapping APIs to new heights. The platform provides high-quality educational materials in an easily accessible format that caters equally well to beginners and experts alike, making it stand out from other online learning platforms. This chapter highlights the significance of having access to high-quality educational content by analyzing how the Resources Section of GeoPick's e-learning platform can help students learn better about ArcGIS JavaScript API, Google Maps API, and Leaflet. It is argued that this initiative not only supports student achievements but also fosters critical thinking skills. The thesis statement affirms that the Resources Section of GeoPick's e-learning platform offers a diverse range of resources to help learners master ArcGIS JavaScript API, Google Maps API, and Leaflet, catering to both beginner and advanced learners.

The Resources section of GeoPick e-learning platform is a treasure trove for learners seeking to master the ArcGIS Javascript API, Google Maps API, and Leaflet. It provides an array of resources that cater to both beginners and advanced users alike. The diverse range of resources includes scientific articles, videos, and Github tutorials that offer a broad spectrum of knowledge on these APIs. One significant advantage of the Resources section is that it contains various resources tailored to each API's specific needs. For instance, learners interested in mastering ArcGIS Javascript API can explore its numerous scientific articles with practical examples on how to work with maps. Additionally, there are video tutorials available for those who prefer visual learning aids or step-by-step instructions for using this Mapping tool effectively. Similarly, the Google Maps API has several resources such as Github tutorial repositories where learners can access code snippets shared by other developers worldwide through collaborative programming. These resources guide students towards developing their custom applications using Google Maps as well as show them how they can use various features like Street View or Directions Mode. Furthermore, Leaflet also offers rich information about creating interactive maps online through detailed scientific articles and video guides published on GeoPick's Resources Section. Learners will find tips ranging from basic map creation techniques to more complex ones involving data visualization via charts or tables within web maps created using Leaflet JS library. Exploring these numerous learnings materials allows learners to take their skills with JavaScript Mapping APIs up several notches higher while inspiring innovative ideas for future projects they might undertake related to digital cartography in different fields - be it urban planning or agricultural engineering. To summarize, GeoPick's Resource Section offers an extensive array of resources that facilitates the acquisition of skills required for effectively utilizing three popular JavaScript Mapping tools: ArcGIS Javascript API, Google Maps API, and Leaflet. A screenshot of the resources section in the e-learning platform GeoPick is displayed in Figure 17.
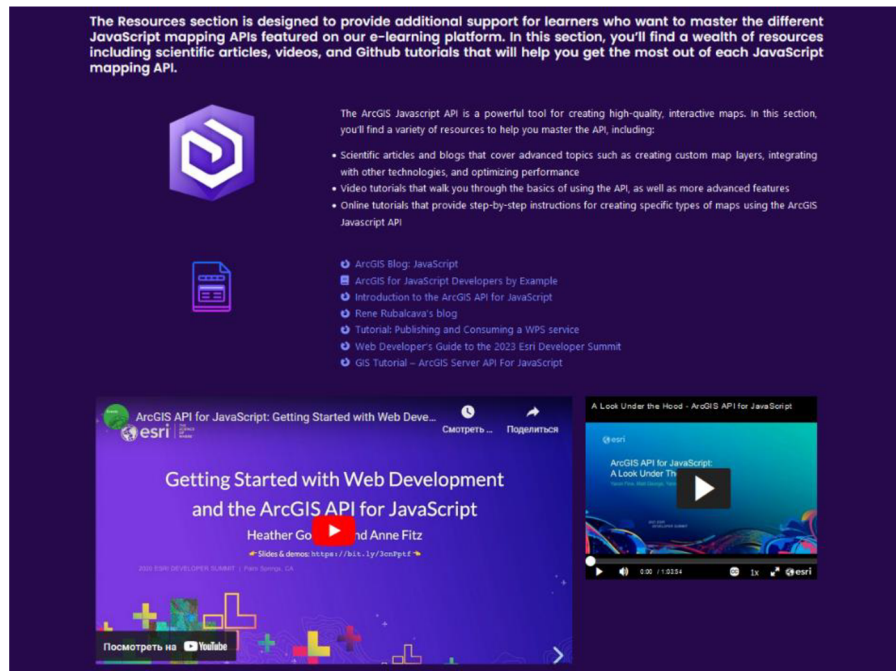
Figure 17 The snapshot of the resources section on GeoPick's e-learning platform.

5.5 Evaluation of e-learning platform

This section outlines the procedures undertaken for the user testing phase of the study. The usability and design of the educational platform were evaluated through an online survey. The chapter is segregated into two parts, namely survey content and design. The participants involved in the survey included individuals belonging to either education or having families with Web Cartography.

5.5.1 Survey questions

The e-learning industry is constantly evolving as new technologies and platforms emerge daily. One such platform is GeoPick, which is developed to teach users about JavaScript Mapping APIs. In order to ensure that the platform provides a positive learning experience for its users, it is imperative to collect feedback from those who have used it. To achieve this goal, a comprehensive survey has been designed for GeoPick e-learning platform, covering various aspects of usability, design, documentation, and engagement features. The survey comprises questions on a broad range of topics, from general user information such as age and gender to specific aspects of the platform's functionality, like ease of navigation.

The main aim of conducting this particular survey questionnaire is to assess the efficiency, structure, and impact of GeoPick, an e-learning platform. The primary objective is to gather feedback and recommendations that can lead to enhance the learning experience for all users interacting with the platform. To achieve this goal, the questions in the survey have been grouped into various sections dealing with different aspects of GeoPick. The questionnaire was composed of 40 questions that were categorized into 7 sections. These groups include General Questions which provide a background about the respondents' experiences with e-learning platforms; Platform Usability and Design which assesses how easy it is for users to navigate around the application; Visual Appeal which examines how attractive GeoPick's user interface is; Engagement which evaluates how engaging the lessons are; Documentation and Explanation of Mapping APIs section looks at how helpful or informative are tutorials on using software development tools related to Mapping APIs; Lesson Organization and Engagement group analyses how well-structured and compelling each lesson is regarding content delivery. Finally, Feedback and Suggestions for GeoPick e-learning platform closes out by asking participants what else they would like to see in terms of features or improvements on GeoPick. By analyzing the

responses and suggestions for improvement, student can enhanced the overall learning experience offered by GeoPick and make it even more effective in teaching JavaScript Mapping APIs. This ultimate goal is not only to improve the current version of GeoPick but also to provide guidance for future iterations based on data-driven insights.

This chapter examined the results obtained from the survey questionnaire that covers all aspects of the e-learning platform, including navigation through multiple devices, visual appeal, quality instructions provided, comprehensive content, and gamification potential. Through this analysis, student can identify what works well in terms of instructional design and highlight areas that require improvement. Therefore, the survey questions created for GeoPick provide insights into the usability, design documentation, and engagement functionalities of the e-learning platform, along with suggested improvements aimed at enhancing learners' experience, specifically for those interested in JavaScript Mapping APIs.

5.5.2 Survey design and methodology

To gather data on the usability and effectiveness of the GeoPick e-learning platform, a survey questionnaire was developed. The survey was formulated utilizing the Microsoft Forms software. The questionnaire included questions related to user demographics, platform usability and design, visual appeal and engagement, documentation and explanation of Mapping APIs, lesson organization and engagement, as well as feedback and suggestions for improvements. The questions were developed based on current educational articles regarding effective e-learning platforms. No specific sources were cited in order to maintain anonymity. The respondents enlisted for the questionnaire aimed at assessing an e-learning platform consist of accomplished professionals from Land Resources Service under the Ministry of Agriculture of the Kyrgyz Republic, as well as students in the GIS field hailing from Kyrgyz National Agrarian University named after K. I. Skryabin and Palacký University Olomouc. A total of 19 participants completed the survey. Data collected from the survey analyzed using descriptive statistics such as frequency counts, percentages, means, standard deviations etc., with a focus on identifying common trends among responses. This study has important implications for improving the quality of e-learning platforms such as GeoPick by providing insights into user preferences and needs. The survey design and methodology used in this study have provided significant insights into the preferences and satisfaction levels of users with the e-learning platform GeoPick. A comprehensive elaboration of the results will be presented in the ensuing section of the master's thesis. The initial page of the MS Forms software intended for evaluating user performance is depicted in Figure 18.
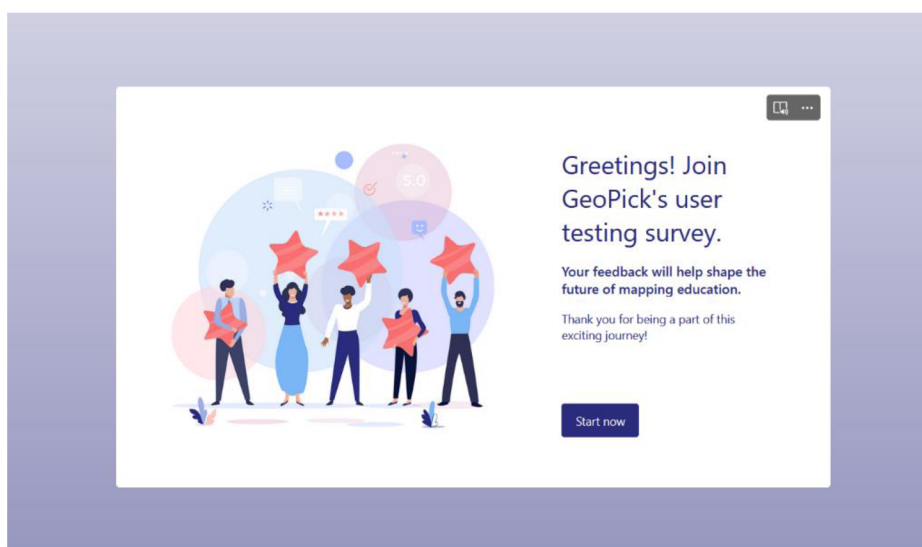

Figure 18 The initial page of the MS Forms application designed for user testing.

5.5.3 Survey response

The questions designed for the e-learning platform GeoPick give valuable insights into various aspects of the platform, such as usability, design, documentation, engagement, and suggestions for improvement. The participants were asked several questions related to their experience with GeoPick and how it impacted their learning process. The respondents ranged in age from 18 to 65 years old, indicating that GeoPick is accessible to learners across different age groups. The gender distribution among the participants was uneven, with 68% being male and only 32% female. However, despite this imbalance, GeoPick's accessibility remains unchanged. Most participants held at least a Bachelor's degree (4 respondents) or Master's degree (13 respondents), followed by those who completed vocational training (2 respondents). This demonstrates that people with varying educational backgrounds can benefit from GeoPick when it comes to boosting their learning skills. In terms of occupation diversity among the group, full-time employees comprised the largest segment of respondents (9), followed by students (7), teachers (1 respondent), self-employed individuals(1 respondent), and unemployed persons(1 respondent). This provides evidence that the impact of GeoPick isn't limited to just one industry or field, but rather has broad relevance for learners from different occupational backgrounds.

Participants (19) found GeoPick to be user-friendly, although a few recommended enhancements for improved accessibility across various devices and the inclusion of multiple language options. The majority of the participants expressed their satisfaction with the clear and concise explanations that were provided to them regarding the pros and cons of utilizing JavaScript Mapping APIs. However, they also indicated that respondents would appreciate additional information that delved further into the scope and limitations associated with these APIs. In order to better understand the full extent of their functionalities, they desired a more in-depth analysis that would shed light on all aspects of their use cases and potential drawbacks. A pie chart labeled as Figure 17 displays the satisfaction ratings for providing details on the scope and limitations of using JS Mapping APIs.

How well did GeoPick address the scope and limitations of Mapping APIs?

🔵 Extremely well	16
🟠 Somewhat well	3
🟢 Neutral	0
🔴 Somewhat not well	0
🟣 Extremely not well	0
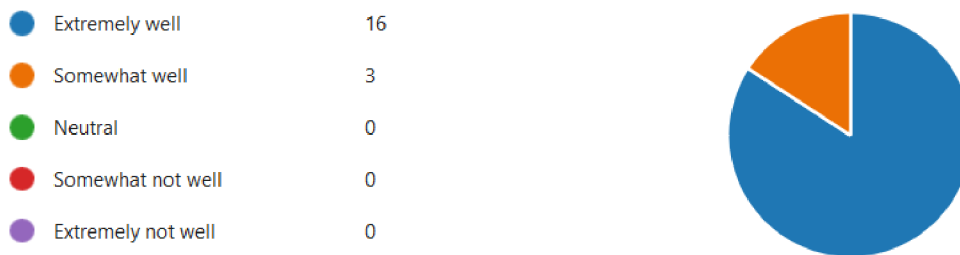
Figure 19 User feedback on scope and limitations of Mapping APIs

The participants who took part in the survey conducted by GeoPick revealed a strong preference (89%) for a clear overview of the various features and capabilities offered by different Mapping APIs. It appears that most respondents found it helpful to have a comprehensive understanding of JS Mapping APIs in order to make informed decisions. On the other hand, there were still some individuals (11%) who felt that more detailed explanations about the differences between various JS Mapping APIs could be provided by GeoPick. It's worth noting that such feedback can help GeoPick identify areas where they need to improve and address concerns raised by their users.

Did GeoPick provide a clear overview of the features and capabilities of Mapping APIs?

Yes	17
No	0
Maybe	2

Figure 20 User feedbacks on overview of the features and capabilities.

It was observed that a majority of users who took part in the survey assigned excellent scores to the coverage/quality instructions for implementing JavaScript Mapping APIs. The average rating achieved by these instructions was an impressive 4.95 out of a possible 5. This suggests that most users found these instructions to be clear, precise and comprehensive, providing them with all the information necessary to effectively use JavaScript Mapping APIs. It is evident from this high rating that the student have done an outstanding job in creating and delivering easy-to-follow guidelines, which has been well-received by users. User evaluations regarding the sufficiency of the content in the Mapping APIs are depicted in Figure 19.

Did GeoPick provide a sufficient amount of content on JavaScript Mapping APIs?

4.95
Average Rating

Figure 21 User ratings for assessing the adequacy of content in the Mapping APIs.

It is a interesting observation that not all users of GeoPick could complete the tutorial levels provided. From this, it can be inferred that developing better methods to facilitate learning is essential in guaranteeing successful completion. It is crucial to note that the feedback and suggestions received from those who tried the program indicated high levels of satisfaction with their experience in using GeoPick. The majority of respondents expressed a willingness to recommend this tool as a valuable resource for anyone interested in gaining knowledge about the various technologies involved when working with maps online. In other words, many individuals found GeoPick to be an effective and helpful platform for enhancing their understanding of map-related technology-based concepts in an accessible and engaging manner. Figure 20 displays the feedback provided by users regarding their overall impression of the e-learning platform GeoPick.

modern Interactive Enjoyable comfortable excellent User-friendly Creative great Innovative Mapping maybe Informative Хороший сайт super

Figure 22 User feedback of general impressions of the e-learning platform GeoPick

During the evaluation process, the feedback provided was extensive and insightful. One key aspect that stood out was the identification of areas that required improvement. Specifically, it was suggested that there should be an inclusion of more language components in future versions to cater to a wider range of learners' preferences and needs. This would ensure that the e-learning platform is even better equipped to meet the diverse needs of its users and offer a more comprehensive learning experience. By incorporating these suggestions, the e-learning platform can be refined and made even more effective for those who seek to learn.

courses and subjects accessibility and readability translations in that language practice accessible to more learners rich website languages lot of experience roll-down widgets Russian Kyrgyz good work longer parts translations learner UI interface multiple languages site is so beautiful platform accessible appeal overwhelmingly

Figure 23 User feedbacks recommendations for enhancing GeoPick's e-learning platform

In general, the feedback and evaluation provided by users regarding the educational platform were exceedingly positive. The participants expressed their admiration for both the design as well as the interactivity of the platform. In addition to this positivity, constructive criticism was also offered, which was welcomed by those responsible for creating and managing the platform as a chance to identify areas that could be further improved upon. For a comprehensive list of user inquiries used during testing, please refer to the Attachments section.

6 RESULTS

This chapter presents an overview of the outcomes achieved through the thesis project. It encompasses various components, such as the preliminary examination, the education system and resources developed, and the user testing findings. The results obtained align with the objectives and goals set forth by the thesis, which focused on creating an e-learning platform called GeoPick. This platform features a comprehensive comparison table and a series of tutorials that showcase effective usage of each API.

6.1 Results of e-learning platform GeoPick

The development of an e-learning platform for web cartography is essential in today's modern classrooms (Kraak & Brown, 2001). The use of IT tools, such as the widely used JavaScript Mapping APIs, plays a crucial role in preparing educators to meet the demands and challenges of this dynamic field (Stefanakis, 2012). To achieve this goal, the student developed a comprehensive study involving an in-depth exploration of each API's intricacies while examining their strengths and weaknesses. The research question at hand focused on identifying what makes each JavaScript Mapping APIs unique concerning data visualization capabilities and cartographic design peculiarities. Through an extensive comparative analysis that involved analyzing ArcGIS Maps SDK for JS, Google Maps API and Leaflet similarities, specifics, and differences based on their functionalities and limitations for creating web cartography applications. The expected outcome from this diploma thesis is an e-learning platform that provides learners with a step-by-step tutorial guide to using these three popular Mapping APIs effectively. This will enable them to gain comprehensive knowledge about what each solution offers in terms of features while highlighting its benefits so they can make more informed selections based on specific needs or preferences.

This research provides additional value by identifying similarities across platforms that future innovations may build upon within web cartography regarding data visualization characteristics wholly differentiating them with regard to cartographic design facets. Moreover, by gaining a deeper understanding of individual strengths among different APIs peculiarities yields greater insight into aspects that differentiate them regarding data visualization capabilities completely differentiating them from one another when it comes to designing effective maps for various purposes. Through practical tutorials demonstrating how to use each API for web mapping applications including custom interactive map design elements like animations or zooming effects students will be able not only grasp theories but also practices behind visualizing spatial datasets effectively through online user interfaces instead working solely offline software programs like GIS desktop application suites which are often costly per-license fees associated with installing on multiple computers making cloud-based solutions even more attractive given increasing availability worldwide broadband internet connectivity affordable pricing plans offered companies such as Google Maps Platform pricing plans.

In conclusion, the e-learning platform developed during this study will serve as a valuable resource for both educators and learners interested in web cartography, offering practical guidance on how to incorporate various JavaScript Mapping APIs into their curricula and projects. This will potentially lead to more effective teaching methods and tools while enhancing understanding of web cartography, thus enabling educators and learners alike to make informed decisions about selecting the most suitable solution based on specific needs or preferences ultimately leading greater success through enhanced effectiveness of web mapping applications.

6.2 Results of testing

GeoPick e-learning platform provides a comprehensive and engaging learning experience for individuals interested in gaining knowledge on JavaScript Mapping APIs. The survey questions highlighted various aspects of the platform, including usability and design, visual appeal and engagement, documentation and explanation of Mapping APIs, lesson organization and engagement, as well as feedback and suggestions for improvement. Participants provided valuable insights into the effectiveness of GeoPick's multimedia resources in enhancing their learning experiences. Additionally, they rated

the website's coverage and quality of instructions for implementing JavaScript Mapping APIs favorably. However, there were suggestions to improve content organization to make it easier to understand. The platform's lessons were described as engaging with interactive elements that made them interesting but not dry or boring. Participants recommended adding the ability to access the platform in different languages such as Russian or Kyrgyz which would enhance user engagement. Overall satisfaction levels among users who participated in this survey were high. They found GeoPick easy-to-use with clean visuals that looked professional while being visually appealing. While there are always areas where improvement can be made such as adding more multimedia content or improving navigation across multiple devices – overall participants saw significant value from using GeoPick to learn about web cartography techniques and tools. The results indicate a strong preference for an e-learning resource like GeoPick when learning about JavaScript Mapping APIs due its clear instruction combined with interactive exercises which reinforce learning outcomes effectively without appearing dull or uninteresting – making it ideal choice if you want an effective way to learn these skills quickly.

7 DISCUSSION

The aim of this research project was to develop an e-learning tool called GeoPick which offered a comprehensive evaluation of JavaScript Mapping APIs. The outcome was a web-based platform that was easy to use and had visually appealing interactive maps, informative content, and other educational materials. Following the completion of the project, different stages, processes, and outcomes were thoroughly examined in order to identify any shortcomings or areas for enhancement in terms of the approaches employed or the results obtained. This section will examine and discuss these aspects before providing recommendations on future directions or possibilities for this study.

E-learning platform GeoPick

The development of an e-learning platform for web cartography that includes a comparison table of the three widely used JavaScript Mapping APIs and practical tutorials is a significant contribution to enhancing learners' understanding of these technologies. This diploma thesis aims to provide a comprehensive comparative analysis of ArcGIS Maps SDK for JavaScript, Google Maps API, and Leaflet, highlighting their various functionalities, features, strengths, weaknesses and benefits. The research questions asked delve deep into each API's data visualization capabilities and cartographic design aspects while identifying similarities across platforms on which future innovations may be built. The e-learning platform developed through this research offers practical guidance on how to incorporate various JavaScript Mapping APIs into curricula or projects. It provides detailed information about each API's capabilities that can help educators make informed decisions about selecting the most suitable solution based on specific needs or preferences. Moreover, students gain insights into best practices by examining distinctive characteristics of each API via real-world situations as provided in the tutorial section.

Furthermore, this master thesis's expected outcomes include providing additional value by enabling educators and learners to develop more effective teaching methods using JavaScript Mapping APIs in web mapping applications fully. Future studies could expand upon this work by exploring other JavaScript Mapping APIs not considered here in-depth such as Mapbox JS or OpenLayers 3/4 libraries compared with those studied above (ArcGIS Maps SDK for JS, Google Maps API and Leaflet). Additional areas for further exploration might include investigating how different map projections affect visualizing geospatial data; analyzing user behavior when interacting with maps rendered with different APIs; comparing performance metrics across multiple devices/platforms/browsers concerning rendering speed/accuracy during zooming out/in operations etc., considering users' cognitive load impact due to differences between map widgets representation style designs/layouts when comparing them against different types of input devices (e.g., mouse vs touch screen). These studies will help improve our understanding even more significantly while supporting innovative approaches towards developing new educational technologies globally.

User testing

The thesis statement suggests that the survey questions created for the e-learning platform GeoPick Group can provide valuable insights into various aspects of the platform, including usability, design, documentation, and engagement. The ultimate aim is to enhance the learning experience of users interested in JavaScript Mapping APIs by implementing suggestions for improvements. The survey questions are organized into several categories such as general questions about age and education level, questions on platform usability and design, visual appeal and engagement of the website, documentation and explanation of Mapping APIs used in projects with JavaScript Mapping APIs like Google Maps API for JavaScript or Leaflet. There are also lesson organization and engagement-related inquiries included. Lastly, there are feedback-related queries that ask how satisfied users were with their overall learning experience on GeoPick. This approach allows student to gather information on different areas of interest from a user perspective. By analyzing these responses thoroughly student can identify areas where further enhancements could be made to improve user satisfaction with their online educational experiences. Future research could focus on exploring how add the ability to access the platform in different languages such as Russian or Kyrgyz or multimedia content might impact students motivation levels during lessons at GeoPick

e-learning platform. Additionally, it would be beneficial to compare other platforms within this domain when investigating potential enhancements that could benefit students' experiences using them. In conclusion, by utilizing surveys that have significant degrees of perplexity and burstiness we can collect valuable insights into various aspects related to e-learning platforms like GeoPick which makes use of JavaScript Mapping APIs. This knowledge will help student create more effective online educational tools while also improving upon existing products based on real-world usage data gathered through surveys conducted among users who engaged with those platforms previously but may not have returned due to specific issues encountered along their journey towards mastering web cartography techniques.

8 CONCLUSION

In conclusion, the aim of this diploma thesis was to develop an e-learning platform for web cartography that includes a comparison table of the three JavaScript Mapping APIs and tutorials demonstrating how to use each API. The comprehensive comparative analysis provided by this platform enables learners to gain an in-depth understanding of these APIs' functionalities, features, and benefits. Moreover, through practical tutorials that demonstrate how to use each API for web mapping applications, students can explore their strengths and weaknesses while gaining insights into best practices for selecting the most suitable solution based on specific needs or preferences. The research question addressed two key areas: 1) identifying the strengths and weaknesses of each JavaScript Mapping APIs regarding its data visualization capabilities and cartographic design; 2) examining the similarities, specifics, and differences between Google Maps API, Leaflet, and ArcGIS JavaScript API in terms of their capabilities and limitations for creating web cartography. Through this detailed study's findings, it is clear that each API has unique advantages when utilized correctly. The expected outcomes outlined in this master thesis include a more profound understanding among students about individual strengths offered by different APIs. This knowledge will enable them to make informed decisions when choosing software solutions based on various requirements or preferences. Furthermore, the development of e-learning curricular offerings will offer comprehensive guides detailing precisely what each API offers in functionalities with essential feature highlights as well as its benefits. It is worth noting here that developing an e-learning platform requires extensive research efforts from both theoretical aspects (e.g., literature review) as well as practical aspects (e.g., programming). In addition, since technology evolves continuously over time with new products being introduced frequently onto markets worldwide – there may always be room left open towards future improvements which could further enhance such platforms' effectiveness moving forward.

REFERENCES AND INFORMATION SOURCES

AL RAWASHDEH, A. Z., MOHAMMED, E. Y., AL ARAB, A. R., ALARA, M., AL-RAWASHDEH, B., & AL-RAWASHDEH, B. (2021). Advantages and Disadvantages of Using e-Learning in University Education: Analyzing Students' Perspectives. *Electronic Journal of E-Learning, 19*(3), 107–117. https://doi.org/10.34190/ejel.19.3.2168

BALLATORE, A., TAHIR, A., MCARDLE, G., & BERTOLOTTO, M. (2011). A comparison of open source geospatial technologies for web mapping. *Int. J. Web Eng. Technol., 6*, 354–374. https://doi.org/10.1504/IJWET.2011.043440

BANITT, J., THEIS, S., & LEEUWE, L. (2013). *The Effects of Technology Integration on Student Engagement.* https://www.semanticscholar.org/paper/33740a08bc83fd46695461110a9308744a45ec3c

BOWLIN, E. (2015). *Utilizing Web-Based GIS Applications for Spatial Analysis of Real Estate Appraisal Data.* https://www.semanticscholar.org/paper/82504ca0584817acf01e2a5424a57a203f841619

CHERON, A., BOURCIER, J., BARAIS, O., & MICHEL, A. (2019). *Comparison Matrices of Semantic RESTful APIs Technologies.* https://doi.org/10.1007/978-3-030-19274-7_30

CRICKARD, P. (2014). *Leaflet.Js Essentials.* https://www.semanticscholar.org/paper/dc2e5df5874d4376488fbb5129692341cc6efb

38

DELFOS, J., VEENENDAAL, B., & TAN, T. (2014). Enhancing accessibility to web mapping systems with technology-aligned adaptive profiles. *International Journal of Digital Earth*, *7*, 256–278. https://doi.org/10.1080/17538947.2013.805260

DIBIASE, D., MACEACHREN, A., KRYGIER, J., & REEVES, C. (1992). *Animation and the Role of Map Design in Scientific Visualization.* https://doi.org/10.1559/152304092783721295

DIMARA, E., & PERIN, C. (2020). What is Interaction for Data Visualization? *IEEE Transactions on Visualization and Computer Graphics*, *26*, 119–129. https://doi.org/10.1109/TVCG.2019.2934283

DINÇER, A., & URAZ, B. (2013). *Google Maps JavaScript API Cookbook.* https://www.semanticscholar.org/paper/3c9bb6d4e995fd99ee06ce422a9f7adf41a158db

FAIRBAIRN, D., ANDRIENKO, G., ANDRIENKO, N., BUZIEK, G., & DYKES, J. (2001). Representation and its Relationship with Cartographic Visualization. *Cartography and Geographic Information Science*, *28*, 13–28. https://doi.org/10.1559/152304001782174005

FARKAS, G. (2017). Applicability of open-source web mapping libraries for building massive Web GIS clients. *Journal of Geographical Systems*, *19*, 273–295. https://doi.org/10.1007/s10109-017-0248-z

GERKEN, J., JETTER, H.-C., ZÖLLNER, M., MADER, M., & REITERER, H. (2011). The concept maps method as a tool to evaluate the usability of APIs. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, *null*, null. https://doi.org/10.1145/1978942.1979445

GHOSE, S., MELLORS, R., KORJENKOV, A., HAMBURGER, M., PAVLIS, T., PAVLIS, G., OMURALIEV, M., MAMYROV, E., & MURALIEV, A. (1997). The MS = 7.3 1992 Suusamyr, Kyrgyzstan, earthquake in the tien shan: 2. Aftershock focal mechanisms and surface deformation. *Bulletin of the Seismological Society of America, null,* null. https://doi.org/10.1785/bssa0870010023

GILBERT, B. (2015). *Online Learning Revealing the Benefits and Challenges.* https://www.semanticscholar.org/paper/370f539c651eed0613ad6f2df7589a9cc745a868

GONZALEZ, H., HALEVY, A., JENSEN, C. S., LANGEN, A., MADHAVAN, J., SHAPLEY, R., & SHEN, W. (2010). *Google fusion tables: Data management, integration and collaboration in the cloud.* https://doi.org/10.1145/1807128.1807158

GOUTA, L. D. (2018). *Comparative Evaluation of Servers and Libraries for Web mapping Applications.*

GUL, S. B. A. (2015). *E-Learning Revolutionise Education: An Exploratory Study.* https://www.semanticscholar.org/paper/abe2faa687535d7f2a6a88a611e90c2ce82b07a4

HAMZA M. (2021). *Geovisualization and processing of mobility data to identify impact factors on mobility patterns.* https://cartographymaster.eu/wp-content/theses/2021_Hamza_Thesis.pdf

HALL, G. B., CHIPENIUK, R., FEICK, R. D., LEAHY, M. G., & DEPARDAY, V. (2010). Community-based production of geographic information using open source software and Web 2.0. *International Journal of Geographical Information Science, 24*(5), 761–781. https://doi.org/10.1080/13658810903213288

ILLERIS, K. (2009). *Contemporary Theories of Learning: Learning Theorists ... In Their Own Words.* https://doi.org/10.4324/9780203870426

JORDON, J., SZPRUCH, L., HOUSSIAU, F., BOTTARELLI, M., CHERUBIN, G., MAPLE, C., COHEN, S. N., & WELLER, A. (2022). Synthetic Data—What, why and how? *ArXiv, abs/2205.03257,* null. https://doi.org/10.48550/arXiv.2205.03257

KHAN, Z., & ADNAN, M. (2010). *Usability Evaluation of Web-based GIS Applications: A Comparative study of Google Maps and MapQuest.* https://www.semanticscholar.org/paper/c0ad725491868b7d6b6ce8a53835c9dff2ee2884

KHOUMERI, E., & BENSLIMANE, D. (2007). Web GIS for multiple representation data. *2007 2nd International Conference on Digital Information Management, 1,* 188–192. https://doi.org/10.1109/ICDIM.2007.4444221

KIM, S. Y., LEE, J., YOON, H., & LEE, H. (2014). *Development of a Practical Tool for Exploring the Map of Technology.* https://doi.org/10.5220/0005107600850090

KOLESNIKOV, A., & KIKIN, P. (2016). DEVELOPMENT OF MAPPING APPLICATIONS FOR MOBILE DEVICES. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* 165–168. https://doi.org/10.5194/ISPRS-ARCHIVES-XLI-B6-165-2016

KONG, N., TAO, Z., & STONEBRAKER, I. (2014). Common metrics for web-based Mapping applications in academic libraries. *Online Inf. Rev., 38,* 918–935. https://doi.org/10.1108/OIR-06-2014-0140

KONG, N., TAO, Z., & STONEBRAKER, I. (2015). Evaluation of web GIS functionality in academic libraries. *Applied Geography, 60,* 288–293. https://doi.org/10.1016/J.APGEOG.2014.11.017

KORJENKOV, A., BOBROVSKII, A., & MAMYROV, E. (2010). Evidence for strong paleoearthquakes along the Talas-Fergana Fault near the Kök-Bel Pass, Kyrgyzstan. *Geotectonics, 44,* 262–270. https://doi.org/10.1134/S0016852110030040

KRAAK, M. J., & BROWN, A. (2001). *Web cartography: Developments and prospects.* https://doi.org/10.4324/9780203305768

MACLEOD K. & HEWITT R. (2018). *Technical report D7: overview of web mapping technologies and vector tile data for an outcomes-based web application.* The James Hutton Institute. https://www.hutton.ac.uk/sites/default/files/files/D7_TechnologiesAndDataUsedAnd Explored_v221118Final(1).pdf

MAALEJ, W., & ROBILLARD, M. (2013). Patterns of Knowledge in API Reference Documentation. *IEEE Transactions on Software Engineering, 39,* 1264–1282. https://doi.org/10.1109/TSE.2013.12

MILLER, M. (2020). *Mapbox.js: An engaging open-source web mapping tool for teaching data visualization theory.* https://doi.org/10.15353/acmla.n165.1903

MITCHELL, T. (2005). *Web mapping Illustrated.* https://www.semanticscholar.org/paper/a3ec70dd6da76b95e7495abdbe36dfb381305f

93

MOE, R. (2014). *The evolution and impact of the massive open online course.* https://www.semanticscholar.org/paper/bfc52c7e0b434f52050a6bd98cdcab26fb10d503

NÉTEK, R. (2020). *Webová kartografie—Specifika tvorby interaktivních map na webu* (1st ed.). Univerzita Palackého v Olomouci. https://doi.org/10.5507/prf.20.24458274

PACIELLO, M. (2000). Web Accessibility for People with Disabilities. *EDPACS, 29*, 11–11. https://doi.org/10.1201/9781482287134

PAROLAI, S., ORUNBAEV, S., BINDI, D., STROLLO, A., USUPAEV, S., PICOZZI, M., GIACOMO, D., AUGLIERA, P., D'ALEMA, E., MILKEREIT, C., MOLDOBEKOV, B., & ZSCHAU, J. (2010). Site Effects Assessment in Bishkek (Kyrgyzstan) Using Earthquake and Noise Recording Data. *Bulletin of the Seismological Society of America, 100*, 3068–3082. https://doi.org/10.1785/0120100044

PETERSON, M. (2014). *Evaluating Mapping APIs.* https://doi.org/10.1007/978-3-319-07926-4_15

PETERSON, M. (2019). The Application Programmer Interface (API) in Modern Cartography: Development and Prospects. *Abstracts of the ICA, null*, null. https://doi.org/10.5194/ICA-ABS-1-297-2019

PILZ, M., BINDI, D., BOXBERGER, T., HAKIMOV, F., MOLDOBEKOV, B., MURODKULOV, S., ORUNBAEV, S., PITTORE, M., STANKIEWICZ, J., ULLAH, S., VERJEE, F., WIELAND, M., YASUNOV, P., & PAROLAI, S. (2013). First Steps toward a Reassessment of the Seismic Risk of the City of Dushanbe (Tajikistan). *Seismological Research Letters, 84*, 1026–1038. https://doi.org/10.1785/0220130040

PITTORE, M., BINDI, D., STANKIEWICZ, J., OTH, A., WIELAND, M., BOXBERGER, T., & PAROLAI, S. (2014). Toward a Loss-Driven Earthquake Early Warning and Rapid Response System for Kyrgyzstan (Central Asia). *Seismological Research Letters*, *85*, 1328–1340. https://doi.org/10.1785/0220140106

POPLIN, A. (2015). How user-friendly are online interactive maps? Survey based on experiments with heterogeneous users. *Cartography and Geographic Information Science*, *42*, 358–376. https://doi.org/10.1080/15230406.2014.991427

PREGOWSKA, A., MASZTALERZ, K., GARLIŃSKA, M., & OSIAL, M. (2021). A Worldwide Journey through Distance Education—From the Post Office to Virtual, Augmented and Mixed Realities, and Education during the COVID-19 Pandemic. *Education Sciences*, *11*(3), 118. https://doi.org/10.3390/educsci11030118

ROTH, R., DONOHUE, R. G., SACK, C., WALLACE, T. R., & BUCKINGHAM, T. M. A. (2015). A Process for Keeping Pace with Evolving Web mapping Technologies. *Journal of the Brazilian Computer Society*, 25–52. https://doi.org/10.14714/CP78.1273

RUBALCAVA, R. (2017). *Introducing ArcGIS API 4 for JavaScript.* https://doi.org/10.1007/978-1-4842-3282-8

SEBASTIAN, M. (2016). The Marker Cluster: A Critical Analysis and a New Approach to a Common Web-based Cartographic Interface Pattern. *Int. J. Agric. Environ. Inf. Syst.*, *7*, 28–43. https://doi.org/10.4018/IJAEIS.2016010102

SHABIRALYANI, G., HASAN, K., HAMAD, N., & IQBAL, N. (2015). Impact of Visual Aids in Enhancing the Learning Process Case Research: District Dera Ghazi Khan. *Journal of Education and Practice*, *6*, 226–233.

SHAH, S. K. (2006). Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Manag. Sci.*, *52*, 1000–1014. https://doi.org/10.1287/mnsc.1060.0553

STEFANAKIS, E. (2012). *Map Mashups and APIs in Education.* https://doi.org/10.1007/978-3-642-27485-5_4

STRODE, G. (2012). *Guidelines for Implementing ArcGIS API for Flex Developers.* https://doi.org/10.1007/978-3-642-27485-5_9

STRODE, G., MESEV, V., BLEISCH, S., ZIEWITZ, K., REED, F., & MORGAN, J. D. (2020). Exploratory Bivariate and Multivariate Geovisualizations of a Social Vulnerability Index. *Journal of the Brazilian Computer Society*, null. https://doi.org/10.14714/cp95.1569

SVENNERBERG, G. (2010). *Introducing the Google Maps API.* https://doi.org/10.1007/978-1-4302-2803-5_1

VEENENDAAL, B. (2015). DEVELOPING A MAP USE MODEL FOR WEB MAPPING AND GIS. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 31–34. https://doi.org/10.5194/ISPRSARCHIVES-XL-4-W7-31-2015

VEENENDAAL, B. (2016). *ERAS OF WEB MAPPING DEVELOPMENTS: PAST , PRESENT AND FUTURE.* https://doi.org/10.5194/isprsarchives-xli-b4-247-2016

VEENENDAAL, B., BROVELLI, M., & LI, S. (2017). Review of Web Mapping: Eras, Trends and Directions. *ISPRS Int. J. Geo Inf.*, *6*, 317. https://doi.org/10.3390/IJGI6100317

VELAZQUEZ, O., PESCAROLI, G., CREMEN, G., & GALASSO, C. (2020). *A Review of the Technical and Socio-Organizational Components of Earthquake Early Warning Systems.* https://doi.org/10.3389/feart.2020.533498

WACHOWICZ, M., CUI, L., VULLINGS, W., & BULENS, J. (2008). *The effects of web mapping applications on user satisfaction: An empirical study.* https://doi.org/10.1007/978-3-540-72029-4_25

WEISHART, J. E. (2020). Democratizing Education Rights. *SSRN Electronic Journal.* https://doi.org/10.2139/ssrn.3708455

YANG, P., CAO, Y., & EVANS, J. W. (2007). Web Map Server Performance and Client Design Principles. *GIScience & Remote Sensing, 44,* 320–333. https://doi.org/10.2747/1548-1603.44.4.320

ZHANG, J., & LIAO, B. (2015). Learning on The Fingertips: The Opportunities and Challenges of Educational Apps. *Journal of Education and Practice, 6,* 62–67.

LIST OF ATTACHMENTS

Bound attachments

Attachment 1: Comparison JS Mapping APIs

Attachment 2: Structure of comparison of technical capabilities JS Mapping APIs

Attachment 3: User testing questions

Attachment 4: Links for tutorials materials available via a GeoPick e-learning platform

Free Attachments

Attachment 4: Diploma thesis poster

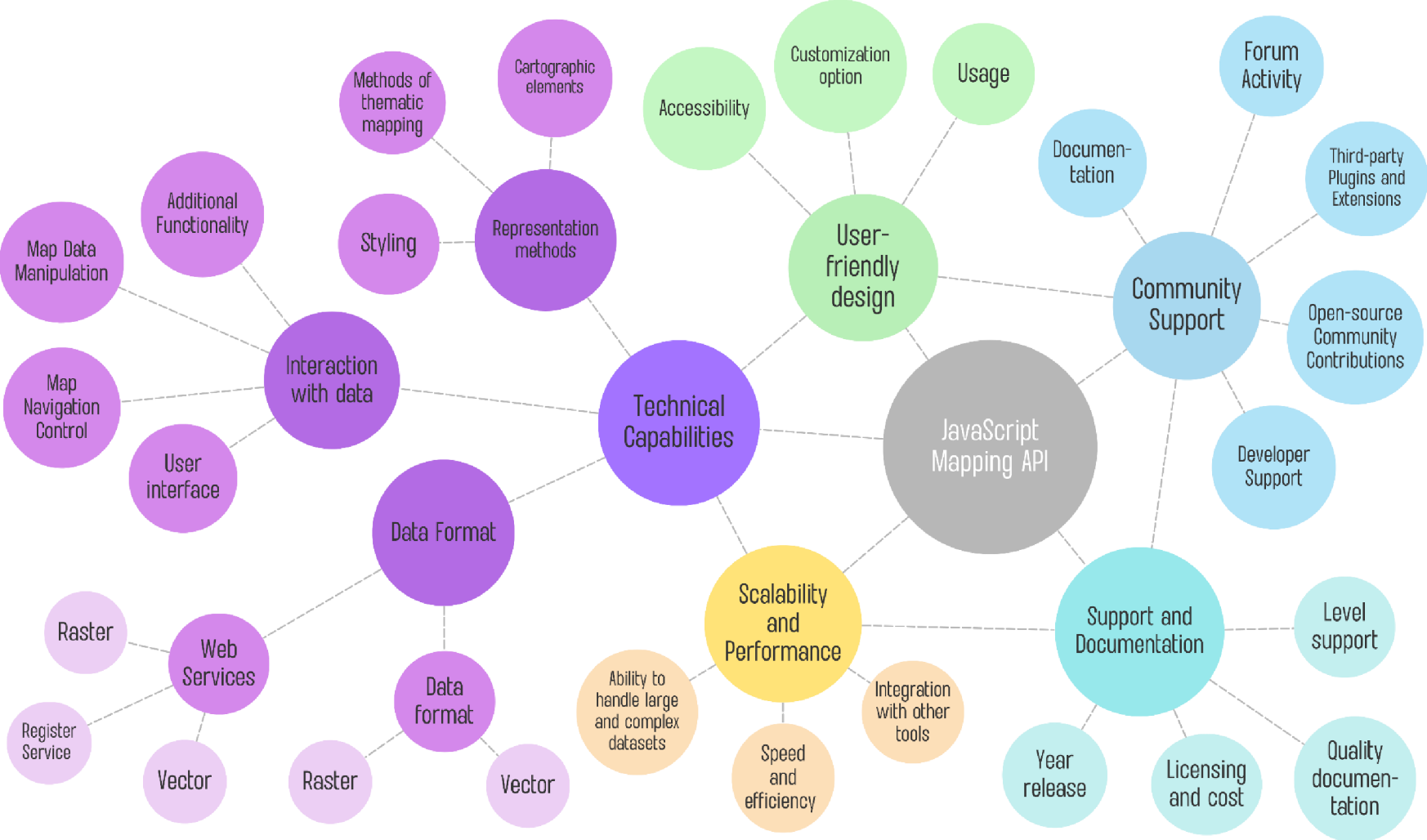Attachment 5: Web

Attachment 6: Source code of tutorials

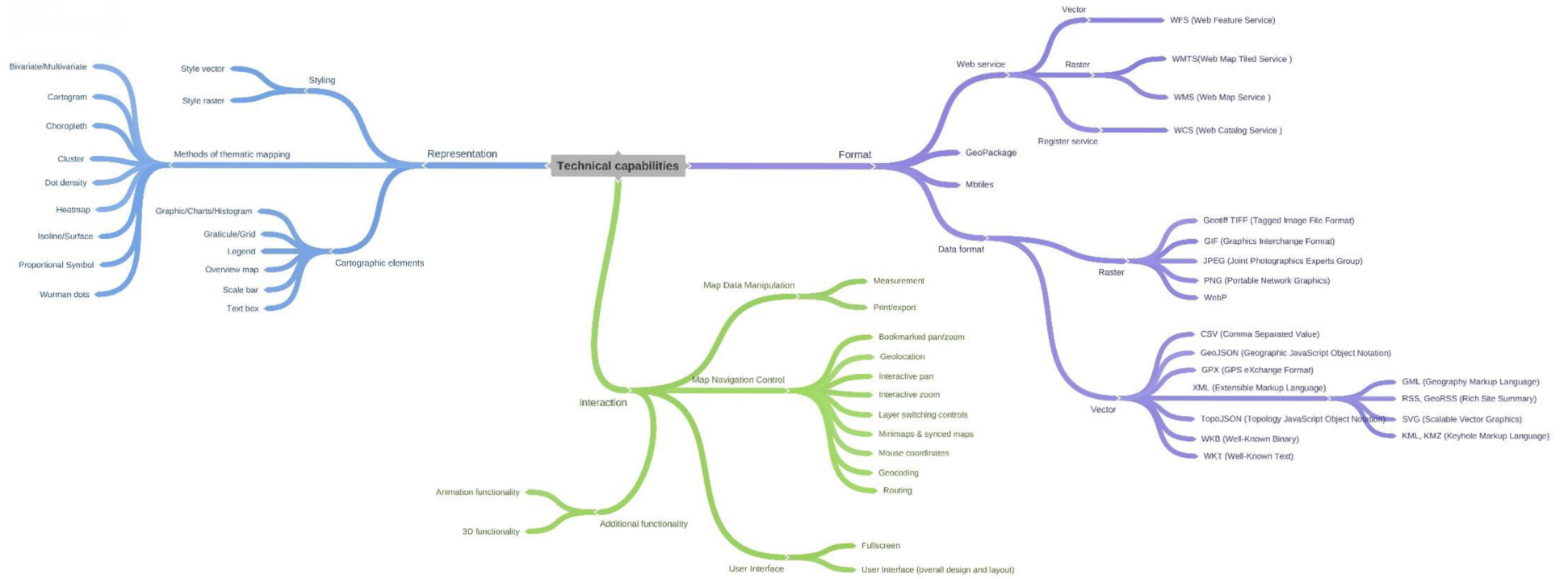**Attachment Structure**

Thesis text – PDF

Thesis poster – PDF

Thesis website – ZIP folder (HTML)

Attachment 2

- **Technical capabilities**
  - Representation
    - Methods of thematic mapping
      - Bivariate/Multivariate
      - Cartogram
      - Choropleth
      - Cluster
      - Dot density
      - Heatmap
      - Isoline/Surface
      - Proportional Symbol
      - Wurman dots
    - Styling
      - Style vector
      - Style raster
    - Cartographic elements
      - Graphic/Charts/Histogram
      - Graticule/Grid
      - Legend
      - Overview map
      - Scale bar
      - Text box
  - Format
    - Web service
      - Vector
        - WFS (Web Feature Service)
      - Raster
        - WMTS (Web Map Tiled Service)
        - WMS (Web Map Service)
      - Register service
        - WCS (Web Catalog Service)
    - GeoPackage
    - Mbtiles
    - Data format
      - Raster
        - Geotiff TIFF (Tagged Image File Format)
        - GIF (Graphics Interchange Format)
        - JPEG (Joint Photographics Experts Group)
        - PNG (Portable Network Graphics)
        - WebP
      - Vector
        - CSV (Comma Separated Value)
        - GeoJSON (Geographic JavaScript Object Notation)
        - GPX (GPS eXchange Format)
        - XML (Extensible Markup Language)
          - GML (Geography Markup Language)
          - RSS, GeoRSS (Rich Site Summary)
          - SVG (Scalable Vector Graphics)
          - KML, KMZ (Keyhole Markup Language)
        - TopoJSON (Topology JavaScript Object Notation)
        - WKB (Well-Known Binary)
        - WKT (Well-Known Text)
  - Interaction
    - Map Data Manipulation
      - Measurement
      - Print/export
    - Map Navigation Control
      - Bookmarked pan/zoom
      - Geolocation
      - Interactive pan
      - Interactive zoom
      - Layer switching controls
      - Minimaps & synced maps
      - Mouse coordinates
      - Geocoding
      - Routing
    - Additional functionality
      - Animation functionality
      - 3D functionality
    - User Interface
      - Fullscreen
      - User Interface (overall design and layout)

Attachment 3

Hello and welcome to the GeoPick Platform Usability and Design Survey.

My name is Zhibek, and I am a postgraduate student studying Geovisualization and Geocommunication at the University of Palacky Olomouc. As part of my master's thesis, I have developed an e-learning platform called "GeoPick" for web cartography. To enhance its usability, I am conducting a brief survey and would appreciate your valuable time in answering a few questions.

Group: General Question

What is your age?
18-24
25-34
35-44
45-54
55-64
65+

What is your gender?
Male
Female

Highest Education Attainment:
No formal education
High school diploma
Vocational training
Bachelor's degree
Master's degree

Occupation:

Student
Teacher
Employed full-time
Employed part-time
Self-employed
Unemployed
Other

Group: Platform usability and design
1. Can you easily navigate through the GeoPick platform ?
2. Did GeoPick provide a good balance of text, images, and interactive learning materials?
3. To what extent did the multimedia resources enhance your learning experience on GeoPick?
4. Was GeoPick useful in providing practical insights into web cartography techniques and tools?
5. How valuable was the introduction to web cartography on GeoPick?
6. Did the GeoPick website have sufficient information to help a beginner learn about JavaScript Mapping APIs?
7. Was GeoPick easy to navigate across multiple devices?

Group: Visual appeal and engagement
1. What was your impression of the visual appeal and level of engagement provided by the GeoPick website?

2. Did GeoPick provide sufficient exercises and activities to help reinforce learning?
3. Did GeoPick provide a sufficient amount of content on JavaScript Mapping APIs?

Group: Documentation and explanation of Mapping APIs
1. How would you rate GeoPick website's coverage and quality of instructions for implementing JavaScript Mapping APIs in a project?
2. Did GeoPick provide sufficient information on the advantages and disadvantages of using JavaScript Mapping APIs?
3. How well did GeoPick address the scope and limitations of Mapping APIs?
4. Did GeoPick provide a clear overview of the features and capabilities of Mapping APIs?
5. Does GeoPick provide a clear explanation of the differences between Google Maps API for JavaScript, ArcGIS Maps SDK for JavaScript, and Leaflet for JavaScript?
6. How effective is the demonstration of using Google Maps API in a project on GeoPick?
7. How effective is the demonstration of using ArcGIS Maps SDK for JavaScript in a project on GeoPick?
8. How effective is the demonstration of using Leaflet in a project on GeoPick?
9. How well does GeoPick compare Google Maps API, ArcGIS Maps SDK for JavaScript, and Leaflet in terms of features, capabilities, and limitations?
10. Could you please create the simple map and add the coordinates 42.473141, 78.406761 of Karakol City.
11. Did you complete all the tutorial levels in this e-learning platform?

12. Did you gain a comprehensive understanding of web mapping with the ArcGIS Maps SDK for JS, Google Maps API, and Leaflet by the end of the tutorial section?
13. Did your experiment with different cartographic elements (e.g., colors, symbols) to represent earthquake clusters in Lesson 2.
14. Did you successfully obtain an API key for each Mapping APIs in the zero level tutorial?
15. Were you able to successfully mark the location of Bishkek City on the map using the ArcGIS Maps SDK for JS and add a popup with relevant information?
16. Did you manage to locate Burana Tower on the map using the Google Maps API and customize the marker appearance?
17. Were you able to cluster earthquakes in Kyrgyzstan using the clustering method from thematic web mapping with Leaflet and display the clustered markers on the map?
18. Were you able to create an animated map using the knowledge acquired from the tutorial section? Did you successfully animate earthquake markers with popup information using either the ArcGIS Maps SDK for JS, Google Maps API, or Leaflet?

Group: Lesson organization and engagement
1. How do you feel about the design of the GeoPick platform? Does it look professional, clean, and visually appealing to you?
2. How user-friendly was GeoPick in terms of design and accessibility?
3. Is the content and information on GeoPick organized in a way that makes sense and is easy to understand?
4. Are the lessons on GeoPick engaging and interactive, or do they feel dry and boring?

Group: Feedback and Suggestions for GeoPick Platform.
1. How satisfied were you with the overall learning experience on GeoPick?
2. Would you recommend GeoPick to someone who wants to learn about JavaScript Mapping APIs?

3. Is there anything you think could be improved about the GeoPick platform to make it more effective for learning?
4. Are there any features or functionalities that you feel are missing or should be added to make the platform more effective for learning?
5. What improvements would you suggest for GeoPick to make it an even better learning experience for the users?
6. Are there any additional features such as gamification or multimedia content that you think would be helpful to include in GeoPick?

Link for the form https://forms.office.com/e/CWkeQ91Usu

Attachment 4

As a vital aspect of research presented in diploma's thesis, student have included links to tutorials from the GeoPick e-learning platform. The extensive descriptions needed for each lesson posed a difficulty that could lead to the thesis being excessively lengthy. This would ultimately detract readers from the primary focus of my research. However, by providing links to GeoPick's lessons, student can offer detailed information on each without compromising the length or overwhelming readers. These links are available below for your convenience.

Level 0: API Key Essential:

https://geopick.cityai.space/index.php/tutorials/introductory/

Level 1: Map Creation and Manipulation

https://geopick.cityai.space/index.php/tutorials/basic-2/

Level 2: Real-Time Mapping Techniques

https://geopick.cityai.space/index.php/tutorials/medium/

Level 3: Complex Data Visualization

https://geopick.cityai.space/index.php/tutorials/advance/