

BRNO UNIVERSITY OF TECHNOLOGY

**Faculty of Electrical Engineering
and Communication**

BACHELOR'S THESIS

Brno, 2022

Adam Turek



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS

ÚSTAV TELEKOMUNIKACÍ

WEB APPLICATION FOR CYBERSECURITY JOB ADS ANALYSIS

WEBOVÁ APLIKACE PRO ANALÝZU PRACOVNÍCH INZERÁTŮ NA POZICE KYBERNETICKÉ BEZPEČNOSTI

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Adam Turek

SUPERVISOR

VEDOUCÍ PRÁCE

M.Sc. Sara Ricci, Ph.D.

BRNO 2022

Bachelor's Thesis

Bachelor's study program **Information Security**

Department of Telecommunications

Student: Adam Turek

ID: 211818

**Year of
study:** 3

Academic year: 2021/22

TITLE OF THESIS:

Web application for Cybersecurity Job Ads Analysis

INSTRUCTION:

The assignment is focused on the extension of the existing job ads database and machine learning analysis. The student's task will be to study the current database structure and analysis outcomes [1]. Then the student will study modern web application development and interactive interfaces. The output of this thesis is to create a world map that represents the database of job ads with filtering options for running the analysis. In particular, the job ads will be filtered by different parameters and then a machine learning analysis will be run on the selected data. Moreover, the map will show the number of job ads per country.

RECOMMENDED LITERATURE:

[1] REWIRE Deliverable R2.2.2 Skills Needs Analysis

https://rewireproject.eu/wp-content/uploads/2021/09/R2.2.2-SkillsAnalysis_Final.pdf

[2] Hajny, J., Ricci, S., Piesarskas, E., Levillian, O., Galletta, L., De Nicola, R.: Framework, Tools and Good Practices for Cybersecurity Curricula. IEEE Access 2021.

**Date of project
specification:** 7.2.2022

**Deadline for
submission:** 31.5.2022

Supervisor: M.Sc. Sara Ricci, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
Chair of study program board

WARNING:

The author of the Bachelor's Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

ABSTRACT

The bachelor's thesis aims to create an interactive global map showing a database of job advertisements in a web application and perform filtering according to various parameters, where machine learning analysis is then performed. The map also shows the number of job advertisements by country. The web application is created using the ReactJS JavaScript library associated with LeafletJS, which provides the main functionality. The machine learning and script change parts are implemented using Python's programming language. The thesis describes the theoretical part and implementation of individual map functions and deals with the description and successful modification of scripts to perform machine learning.

KEYWORDS

Web Application, JavaScript, ReactJS, LeafletJS, GeoJSON, Python, Data Analysis, Machine Learning, LSTM, Artificial Neural Network, Recurrent Neural Networks, Natural Language Processing

ABSTRAKT

Cílem bakalářské práce je vytvoření interaktivní celosvětové mapy zobrazující databázi pracovních inzerátů ve webové aplikaci a provedení filtrování podle různých parametrů, kde je následně provedena analýza strojového učení. Také mapa zobrazuje počet inzerátů na pracovní pozice podle příslušných států. Webová aplikace je vytvořena pomocí JavaScriptové knihovny ReactJS spojené s LeafletJS, které zajišťují hlavní funkcionalitu. Část se strojovým učení a změna skriptů je realizována pomocí programovacího jazyku Python. Práce popisuje teoretickou část a implementaci jednotlivých funkcí mapy a dále se zabývá popisem a úspěšnou úpravou skriptů pro účely provedení strojového učení.

KLÍČOVÁ SLOVA

Webová aplikace, JavaScript, ReactJS, LeafletJS, GeoJSON, Python, Analýza dat, Strojové učení, LSTM, Úmělá neuronová síť, Rekurentní neuronové sítě, Zpracování přirozeného jazyka

ROZŠÍŘENÝ ABSTRAKT

Cílem práce je vytvoření celosvětové interaktivní mapy zobrazující databázi pracovních inzerátů s možností jejich filtrování pro účely analýzy. Pracovní inzeráty budou filtrovány podle různých parametrů a poté je na vybraných datech spuštěna analýza strojového učení. Mapa světa zobrazuje i počet inzerátů na pracovní pozice podle příslušných států. Úkolem bylo nastudovat aktuální strukturu databáze a výsledky analýzy. Poté nastudovat vývoj moderních webových aplikací s interaktivním rozhraním.

Práce je rozdělena do tří hlavních částí, kterými jsou Teoretické zázemí, dále prostředky pro vývoj aplikace a úpravu jednoho ze skriptů, které popisuje kapitola Vývojové nástroje a celkové zpracování aplikace v kapitole Implementace, která zahrnuje celkový postup a řešení jednotlivých úkolů.

Teoretická část práce je rozdělena do tří kapitol, kterými jsou technologie webových aplikací, datová analýza a strojové učení. V teoretickém úvodu si práce klade za cíl vysvětlit co je to webová aplikace a jak se liší od webové stránky a desktopové aplikace. Každá webová aplikace se skládá z několika jednotlivých webových stránek. Jsou zde popsány hlavní výhody a nevýhody webové aplikace a druhy webových stránek, které jsou statické nebo dynamické. Dále se v této kapitole čtenář dozví o druzích síťové architektury, které lze pro vývoj webových aplikací využívat. Pro účely práce vyhovovala architektura three-layer, která je brána jako aplikační server a využívá databázi. Kapitola také zmiňuje odstavec o historii webových aplikací. Jsou představeny technologie pro vytváření webových aplikací, kde se používá několik jazyků. Je zaměřeno na JavaScript jazyk a jeho knihovnu ReactJS, která je využita při implementaci v práci a je popsána podrobně v kapitole dvě. Poslední části této kapitoly se zaměřují na popsání druhů webových serverů.

V druhé kapitole teoretické části následuje kratší povídání o datové analýze. Věnuje se definici a využití dat a jednotlivým důležitým druhům. Jedním z důležitých zmíněných pojmů jsou velká data, která se definují jako obrovský soubor dat. Tato množina pak roste v průběhu času exponenciální rychlostí. Dále jsou definována data jako jsou data v reálném čase nebo strojová data, která jsou vytvořena za pomoci strojů. Jak bylo zmíněno, analýza dat má několik obecně používaných druhů. Čtenář se dozví o pěti z nich a pro naše účely je nejdůležitější prediktivní a preskriptivní analýza, které se často využívají v odvětví strojového učení.

Poslední kapitola teoretické části se věnuje strojovému učení. Ze začátku kapitoly je specifikováno spojení mezi datovou analýzou a strojovým učením, následně se položíme do definice a účelu strojového učení. Prostor a čas, jako v každém jiném oboru počítačové vědy, jsou primární a kritická měřítka kvality těchto algoritmů. Po úvodu do problematiky se čtenář dozví o čtyřech typech strojového učení, které

se v praxi nejčastěji využívají a jsou upravovány člověkem. Algoritmy strojového učení jsou konstruovány, aby se učily z příkladů a prostřednictvím učení vyhledávají vzory, které odpovídají požadovaným výstupům. V poslední části této kapitoly se práce zabývá technikami a typy analýzy strojového učení, které jsou využity pro práci s webovou aplikací. Základním kamenem této části je zpracování přirozeného jazyka, od kterého se následně odráží ostatní techniky jako je například rekurentní neuronová síť a dále její vrstvy.

Po teoretické části následuje druhá hlavní část, kde se práce zaměřuje na popis použitých vývojových technologií. Tyto technologie jsou následně využity v kapitole s názvem Implementace. Ze začátku této kapitoly se čtenář dozví o vývojovém prostředí, které podporuje programovací jazyky jako je JavaScript a TypeScript a bylo pro tuto práci zvoleno. Dále se čtenář seznámí s open-source JavaScriptovou knihovnou ReactJS, na které je v práci aplikace postavena. V této kapitole jsou obecně popsány základní důležité funkce této knihovny využity pro účely práce. Dále se čtenář seznámí s další JavaScript knihovnou LeafletJS, která je v práci využita na tvorbu a stylování interaktivní celosvětové mapy. Výhoda této knihovny je její jednoduchost a komplexnost. V předposlední části této kapitoly je možné nahlédnout i do světa designu, konkrétně je představen designový systém, využívaný knihovnou React. Poslední část této kapitoly se nevěnuje technologiím k tvorbě webové aplikace, ale k její serverové části, která je programovaná v jazyce Python. Tento programovací jazyk je v práci obsažen pro změnu jednoho Python skriptu právě v serverové části. Python má mnoho knihoven, které uživatelům dovolují programovat algoritmy pro strojové učení a datovou analýzu.

V praktické části práce se čtenář dozví o celkovém vývoji interaktivní celosvětové mapy pro webovou aplikaci, dále také o využití strojového učení na pasáži serveru a úpravy jednoho ze skriptů pro přidělování pracovních dovedností do grafů dynamicky. Ze začátku kapitoly jsou popsány důvody použití vývojových technologií.

Následuje samostatná kapitola Vývoj mapy. V ní se práce nejprve zaměřuje na prvotní spuštění serveru a důležité instalace, které je potřebné provést před samostatnou tvorbou mapy. Pro spuštění serveru, který se otevře na jakémkoli webovém prohlížeči na adrese `http://localhost:3000`, existuje více způsobů. Pro účely práce, a také pro jednoduchost byl vybrán způsob otevření terminálu ve vývojovém prostředí Visual Studio Code, které ho obsahuje v základní verzi, v práci je použita verze 1.62. Dále je možné použít příkazový řádek a nebo PowerShell. Následná kapitola popisuje nastavení mapy pomocí LeafletJS knihovny a naimportování potřebných komponentů. Dále je důležitá kapitola jménem GeoJSON, která umožňuje interaktivitu mapy, kde je nejprve nutné naimportovat soubor `countries.geo.json`, který obsahuje údaje o zemích. Interaktivní funkce je název další kapitoly, ve které jsou podrobně popsány funkce, které zajišťují interaktivní část mapy jako je možnost

kliknutí na zemi a její obarvení nebo přejetí kurzoru myši na stát a jeho obarvení. Pro pracování s databází, která obsahuje pracovní nabídky jednotlivých evropských zemí je v práci obsažena další kapitola. Ta popisuje jak ji nainportovat do práce a následně s ní pracovat. Hlavní funkce je kapitola, která obsahuje podrobný popis funkce, která zajišťuje, aby mapa pracovala tak jak má a většina funkcí je do ní zapracována. Součástí této kapitoly jsou i obrázky, které zobrazují jednotlivé funkce mapy a v poslední kapitole se čtenář dozví o posledních úpravách mapy a její finální verzi. Poslední kapitola před serverovou částí popisuje filtrování v hlavní tabulce aplikace a její funkcionalitu.

Poslední kapitola v praktické části nese název REWIRE web server, která obsahuje prvky strojového učení. V ní se čtenář dozví o pozadí webové aplikace a využitých technikách strojového učení. Část úkolu práce byla pozměnit skript, aby zobrazoval graf a přiděloval k němu dynamicky názvy pracovních dovedností v oblasti kybernetické bezpečnosti. Z úvodu je popsán důležitý prvek pro realizaci úkolu a to je dataset. Obsahuje celkovou databázi pracovních inzerátů a dovedností k nim potřebným. Následná část popisuje potřebné instalace a jak provést zapnutí jednotlivých příkazů. V souboru README.md jsou obsaženy všechny potřebné příkazy, které jsou využity k tréninku a predikci. Posledí kapitola celé práce obsahuje podrobný popis jednoho ze skriptů, ve kterém je upravena funkcionalita pro splnění úkolu. Také jsou vyobrazeny obrázky grafu před úpravou a po úpravě.

V teoretické části bylo nutné si nastudovat technologie a možnosti webových aplikací. Dále si nastudovat možnosti datových analýz a také strojového učení, které je velmi užito. Následně dle znalostí byla naimplementovaná celosvětová mapa s možností filtrování a aplikace analýzy strojového učení, jak bylo požadováno dle zadání. Webová aplikace umožňuje přístup z jakéhokoliv webového prohlížeče a operačního systému. Webová aplikace bude součástí REWIRE projektu. Návod na spuštění webové aplikace popřípadě Python skriptu je součástí této práce.

TUREK, Adam. *Web application for Cybersecurity Job Ads Analysis* . Brno: Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 71 p. Bachelor's Thesis. Advised by M.Sc. Sara Ricci, Ph.D.

Author's Declaration

Author: Adam Turek
Author's ID: 211818
Paper type: Bachelor's Thesis
Academic year: 2021/22
Topic: Web application for Cybersecurity Job
Ads Analysis

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno

.....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to thank the supervisor of the bachelor's M.Sc. Sara Ricci Ph.D. for professional guidance, consultation, patience and inspiring suggestions for work. I would also like to thank Mr. Ing. Marek Sikora for all the advice and help throughout the work.

Contents

Introduction	15
1 Theoretical background	16
1.1 Technology of web application	16
1.1.1 History	20
1.1.2 Technologies for creating web interfaces	21
1.1.3 Types of web servers	22
1.2 Data analysis	23
1.2.1 Types of Data Analysis	25
1.3 Machine learning	27
1.3.1 Types of Machine Learning	27
1.3.2 Types of Machine learning analysis	29
2 Development tools	32
2.1 Visual Studio Code	32
2.2 ReactJS	32
2.3 LeafletJS	35
2.4 Ant Design	36
2.5 Python	37
3 Implementation	38
3.1 Map Development	38
3.1.1 Start server and installations	38
3.1.2 Setting a map	40
3.1.3 GeoJSON	41
3.1.4 Interactive functions	43
3.1.5 Database import	44
3.1.6 Main function	45
3.1.7 Last modifications of map modal	49
3.1.8 Filtering in main table	50
3.2 REWIRE web server	52
3.2.1 Dataset	52
3.2.2 Start and necessary installations	53
3.2.3 Modification of Python Script	54
Conclusion	59
Bibliography	60

Symbols and abbreviations	65
List of appendices	67
A Directory tree for web application	68
B Directory tree for Server part	69
C Manual - running the web application and python script	70
C.1 Web application	70
C.2 Python script	71

List of Figures

1.1	Three-layer architecture	20
2.1	Example of React.Component in Visual Studio Code	33
2.2	Example of Hook in Visual Studio Code	34
2.3	Example of render() and componentDidMount() in Visual Studio Code	34
2.4	Example of Modal and Buttons	36
3.1	Successful npm start command	39
3.2	The left chart shows the Drawn line for Czechia and the right chart shows detail of the left chart	41
3.3	Example of country with Cybersecurity Job Ads with "mouseOver" function used	47
3.4	Example of country with Cybersecurity Job Ads with "selectedLayer" function used	48
3.5	Example of country with no Cybersecurity Job Ads with "selected- Layer" and "mouseOver" functions used	48
3.6	Overall look of the application	50
3.7	Filtering by Czechia country	51
3.8	Filtered table by Czechia country	51
3.9	Example of dataset	52
3.10	Part of the python script without modification	55
3.11	Part of the python script with modification	56
3.12	Radar_plot.png without labels	57
3.13	Radar_plot.png with labels	57
3.14	Detail of left plot with labels	58

List of Tables

3.1	Example of code for Map componets	40
3.2	Example code of countries.geo.json for Aruba state	42
3.3	Code for selecting a country	44
3.4	Code for adding number of job adds to the map	47

Introduction

In today's world, technology is an integral part of our lives. Almost every uses it daily, both when shopping in stores or playing various computer games, and to communicate with other people via mobile networks or the Internet [1]. In recent years, one of the most important fields has specifically risen, i.e., Cybersecurity. Cybersecurity deals with the protection of critical systems and sensitive in formations. More experts in this fileds are needed as shown by ISC2 [2] report that identify a shortfall of 2.7 million of expert in 2021. One of the reasons of this gap is that cyberspace cybersecurity is still a young science and its skills and knowledge are still not well defined. In fact, an analysis of existing European documents has shown that, cybersecurity work roles are poorly described in existing documents [3, 4] nowadays. ENISA [5] is dealing with the creation of a European Cybersecurity framework. In the meanwhile, it is necessary to understand which skills are needed.

That is why there is a web-based application that allows us to analyze job advertisements and cyber security to identify the skills we need. The bachelor thesis aims contribute to this web application by creating an interactive worldwide map showing the database of job advertisements in the web application and filtering according to various parameters. Machine learning analysis of selected job advertisements is then performed to identify needed skills. Therefore, display the number of ads on job positions according to the countries concerned. The map is going to be part of REWIRE project outcomes [6].

The bachelor thesis consists of 3 main chapters. The first chapter covers the theoretical part of the work and contains subchapters about website technology and its production. Other subchapters contain a general description of data analysis and what forms we can take. Lastly, in the theoretical part, part with machine learning, which contains a description of the possibilities of machine learning and a description of individual models designed for this work. Chapter 2 contains technologies for developing web applications and machine learning programs. The last chapter describes the overall implementation of development technologies and the web application's functionality. It also describes the functionality of the individual scripts that were used in machine learning and subsequent editing.

1 Theoretical background

The content of this chapter is divided into three parts. The first part explains how web applications work and describes the technology for their development. The second part is focused on data analysis. The third part focuses on types of machine learning and explains how it works and how we can apply it.

1.1 Technology of web application

A web application may look similar to a web page, but it is usually a more complex application that executes more complex tasks and contains a database. The main difference between web apps and websites is that websites usually contain static content and do not enable visitors to interact or communicate back to the site, whereas web applications allow this interaction. Most sites are built using HyperText Markup Language (HTML) and Cascading Style Sheets (CSS).

The web application is denoted to be an application running on a server with the user who communicates using a client on our computer, most often a web browser. Web apps are present on many websites e.g. Google.com¹, Facebook.com², LinkedIn.com³. Web apps are dynamic, and users can easily operate with them. These apps use server-side scripts and client-side scripts to present information. Moreover, it requires a server to manage requests from the users. In order to better understand web applications, the desktop application will be briefly introduced [7].

The desktop application is a classic program that has to be installed on our computer's hard drive. This is the most typical form of software. Rely on operation systems, so it can not be run in another medium. The critical point that distinguishes desktop applications from web ones is the independence of the Internet connection. In most cases (depending on the nature of the particular software), they can be used offline with unlimited file access. However, when using them, users are tied to one machine [8, 9].

The main advantages of web applications are as follow [10]:

- no need to installing any software,
- always get quick and easy updates,
- users have direct access,
- all is needed is a web browser,
- costs almost nothing.

¹www.google.com

²www.facebook.com

³www.linkedin.com

The main disadvantages of web applications are as follow:

- internet connection is required,
- takes time to be developed,
- might be slower(depends on speed of connection).

Static vs Dynamic websites

Static websites are a primary type of website. Their creation is the simplest and consists of the exact number of rendered web pages with hard-coded content. All users see the same content, regardless of who they are and which browser they use. Static web pages are programmed using HTML and CSS. HTML for structure settings and CSS for visualization and color settings. They are also set up independently without being connected to the database. Static web pages do not affect users. The webmaster makes changes by editing the HTML code on each page of the website [11].

Advantages of static websites:

- static websites are less complex,
- the basic code of the statistical website can be duplicated, the elements remain stable, but at the same time there are small changes that distinguish them,
- static websites usually contain less vulnerabilities and become more secure,
- creating static websites is easier and does not require aggregate software,
- thanks to simpler structures, they are also easily transferable from server to client in a short period of time, in the event of a failure or exploitation of a vulnerability by a Distributed Denial of Service (DDoS) attack, it is easier to recover it by relocating the code.

Disadvantages of static websites:

- updating static websites is more technically and time consuming,
- when updating, the site may malfunction due to its complexity,
- it is not possible to communicate with users / visitors of the site,

Dynamic web pages are created in proper intervals. Knowing the programming language (Hypertext Preprocessor, C-sharp, Python) is necessary. The content and structure of this site are very flexible to be adapted to the requirements of the user or his browser. According to user requirements, the code used to create them can create HTML pages in real-time. Dynamic websites are interactive and have changing components. They use the content of external databases or a content management system to process requests. Content is created and displayed after interaction with the user in response to his action. However, how much change will

occur depends on the skills of the developer and the complexity of the elements of the site [11].

Advantages of dynamic websites:

- dynamic websites offer more features and are interactive,
- their content is created according to user requirements,
- content can be edited by multiple users,
- it is possible to store and organize information in them,
- the connection to the content management system makes these pages flexible even afterwards,
- compared to a static site, editing is less expensive.

Disadvantages of dynamic websites:

- creating dynamic websites is expensive, but then you don't pay for updates and changes,
- pages may be slow to load and process instructions due to the complex technology that performs the functions.

Client-server architecture

This architecture is one of the most used in the Transmission Control Protocol/Internet Protocol (TCP/IP) [12] network. They were created primarily in response to the shortcomings of the original monolithic architectures, which they are trying to solve. One of the main problems was the load on the network due to the calculations of the assigned tasks of older structures on the user's computer. This computer obtained computational data from the file storage. The solution to this problem was to divide the responsibilities between two different parts of the architecture: the so-called client and server. The main reason for this solution was the decision to process the data where they are also stored. Such a place is a server located in a computer network and provides various services.

These include, for example:

- an email server that has services related to sending email messages,
- File Transfer Protocol (FTP),
- print servers for printing data on other devices.

The server is inactive, does not offer any services, but monitors events on the network and responds to requests from user computers (clients). It establishes communication and then processes requests. The second part of the architecture is the user computer or client. Compared to the server, it is active. It is in charge of establishing communication with the server, sending it data, and getting the result

back. It is common for the computer to be connected to multiple servers simultaneously. Its communication with the user most often occurs directly through the GUI (Graphical User Interface). We can distinguish two types of clients based on the degree of functionality - thick and thin client [13, 14]. The thick client involves a great deal of application logic and is in part server-independent. On the other hand, it has greater demands on the hardware and software of the user's computer. A thin client is the majority of the processes performed by a server. This is necessary high server performance and fast network connection on both sides. The Client-server model provides mainly three basic tasks:

1. presentation activity, where we can include communication with the user through the GUI, thanks to which the user can, for example, display the results of processes,
2. application logic, which is in itself the functionality of the services provided,
3. manage stored data, which includes storing, checking and deleting data.

These three basic tasks can be further divided into five categories according to the way they are divided between the client and the server:

1. distributed presentation, which consists in the division of presentation activities between the server and the client, the server generates character mode output in a text window, this output is sent to the client and the client displays it to the user,
2. remote presentation, which is no longer divided and remains only with the client, the server provides the rest of the work, that is, application functions and data management,
3. the distributed function is again divided, the client takes care of the presentation and the server takes care of data management, in terms of application functions, the server and client share them,
4. remote data management, presentation and application activities again remain only with the client and the server takes care of data management,
5. distributed database, the server remains with the data management together with the client, in addition, the client also takes care of application and presentation functions.

Three-layer architecture

The client-server architecture [15], which included a thin client (server and terminal), is referred to as a single layer. Her successor is a two-tier model that includes a thick client. This means that the user side also has part of the application logic. As the complexity of user applications increases, so makes the demands on computer

performance. With this, database machines also begin to become independent. Based on the services in the client-server structure, it is more logical to leave each layer its own layer, so a three-tier architecture is created by adding an application layer. The application server represents this layer. Concentrating application logic in one place has made it easier to share, manage, and expand its availability. As the location of the core network changes to the route between the application server and the data source, it is also necessary to reduce the network load. Thanks to this, the response could also be improved for clients with limited transmission speed.

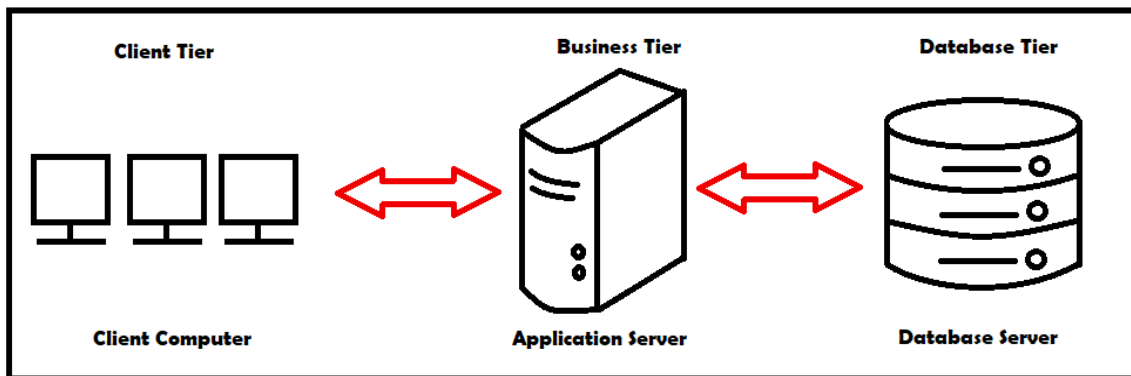


Fig. 1.1: Three-layer architecture

Figure 1.1, shows guiding principle of the three-layer architecture.

1.1.1 History

British scientist Tim Berners-Lee invented the World Wide Web in 1989. This invention aimed to connect the computer, data network, and hypertext technology into a global information system. The first web server was launched in 1990 at CERN [16, 17]. The browser was mainly used to display at this time, but as the number of web pages grew, it was also necessary to add user interaction. This idea created dynamic websites that provided users with feedback after using a mouse or keyboard. It was already possible, for example, to list and send a document at this time. Six years later, Flash technology was created on the Internet, allowing websites to use simple animations and images, increasing user interaction and making the overall content more visual. Around this time, a profitable way of structuring design, i.e., CSS, came to market. These separate the content that remained in the HTML from the presentation. Another major shift in web design development is web content editors (e.g., TinyMCE). The HTML formatting tools brought text customization (bold text, italics, alignment, etc.). At the end of 2000, the interactive content of

the website grew considerably, i.e., Web 2.0. Since then, despite the rise of mobile websites, there has been little missing from the desktop-type web applications we know today [18].

1.1.2 Technologies for creating web interfaces

Every developer will encounter an extensive set of rules and different techniques when developing web applications. It is required, and the developer wants the web application to work and look as it should. It is necessary to become familiar with web technologies. The development is mainly based on three primary and essential jazz, from which various libraries are derived. In this section, several languages used to develop web applications are introduced. We will focus on the JavaScript language and its ReactJS library used in thesis implementation, which is described in detail in Chapter 2.

HTML and CSS

The basis for a web application or website are HTML and CSS [19]. HTML is used to create web pages linked by hyperlinks. With its embedded elements, the HTML document creates a tree structure. Since 2014, the HTML5 version has been used. It supports media playback in the browser. HTML does not provide a visual presentation of content, so CSS has been created. CSS is a language that describes how elements appear on pages written in HTML, Extensible Hypertext Markup Language (XHTML), or Extensible Markup Language (XML). The World Wide Web Consortium (W3C) standardization organization designed the language.

JavaScript

JavaScript [20] syntax belongs to the C / C ++ / Java family of languages, but it is semantically different from these languages. It is written directly into the HTML code, which brings a big advantage. In the test, JavaScript is a client script. The program is sent with the page to the client (browser) and is only executed there. JavaScript is often confused with Java. Java is a standalone programming language. It has only a similar syntax to JavaScript.

PHP

Hypertext Preprocessor (PHP) [21] is a scripting programming language. It is designed primarily for dynamic programming websites and web applications such as HTML, XHTML, and Wireless Markup Language (WML). PHP can also be used

for creating console and desktop applications. It is integrated with several popular databases, including My Structured Query Language (MySQL), PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. PHP can perform system functions, i.e., from files on a system. It can create, open, read, write, and close them. Handle forms, i.e., gather data from files, and save data to a file. We can send data, return data to the user, and encrypt data through email.

1.1.3 Types of web servers

Just as there are many types of software, there are also many types of web servers. Web servers can be generally divided into categories, such as intranet and extranet web servers, or according to their purpose to portals, entertainment web servers, or personal sites. Each type of web server will have different design features that depend on the site's purpose. A proper website organization will help us accomplish its purpose. Structure is good however, the designed Web site is not always visible to the user. There are many criteria [22] for dividing web servers into groups. Criteria can be an audience, level of interactivity, the intensity of changes in content, size, and type of technology used.

Web servers by interactivity:

Many websites are currently interactive. It consists mainly of static content that users can view. The interactive web server [22] allows users to process content directly or communicate with other users. To some extent, all web servers have some interactivity. However, real interactive sites for users allow them to process content and add content. A Web server that allows the user to ask technical support questions that other users may view may be requested as interactive, whereas a Web server that lets the user only browse pre-existing question answers will be marked as static.

Web servers by size:

Another option for dividing web servers is according to their size [22]. The size of multiple web servers does not mean much, mainly if they are created from the content saved in a database, but regardless of this fact, the number of pages is still used to describe web servers. Although there is no exact breakdown of what a small and extensive Web site is. Several people usually form small web servers and usually have limited technological elements. These servers usually contain less than ten pages. On the contrary, medium and large web servers can be managed by a small group of people and can be followed by more sophisticated technologies. These servers

usually contain around a few thousand pages. Multiple web servers can maintain many individuals due to their complex technical and transmission requirements and contain the most significant number of pages, e.g., tens of thousands.

Commercial web servers:

Commercial web servers [22] are designed mainly to boost the business of a particular organization. Users of the commercial web server are potential and current customers of the company. The primary purpose of all commercial servers is to serve users in a way that brings the company direct or indirect profits. The purpose of providing the information is to get customers to buy a product or service from the company. Whether it is a direct or indirect way of convincing the user, the purpose is always the same: to carry out a business transaction.

Personal and artistic web servers:

Personal pages can be created to provide information to friends and relatives or pass on experiences such as knowledge of HTML and CSS, and many others. Some personal sites are a sanctuary for their creators when they try to become famous on the web. Other personal pages are only a review of knowledge and data. An ideal web server is experiencing a similar concept. Such a site could be to encourage, instruct or entertain visitors. Designing artistic web servers can purposefully bypass standard web conventions [22].

1.2 Data analysis

Data analysis [23] is a process that usually involves multiple activities: organization, cleaning, and data collection. Moreover, it may require data analysis software. All of these activities are important to prepare data for business purposes. Data comes to processing in various types and forms, such as Big data [24]. Big data is a huge set of data. This set then grows at an exponential rate over time [24]. Four basic concepts characterize big data:

- diversity (data sources),
- speed (data growth),
- volume (data volume),
- variability (data quality and business value).

Structured/unstructured data

Structured data [23] is a data model, such as a row and column database, that is predefined. In contrast, unstructured data is in a format that does not fit in rows and columns and can also contain videos, photos, text, or audio. Structured data compared to unstructured data show that their analysis and management are more straightforward.

Metadata

Metadata [23] is a form of data that provides information about various other data. For example, metadata includes information about the author, creation date, or file type. They also help users better organize unstructured data into different categories, corresponding to working with them.

Real-time data

Real-time data [23] is data that is presented immediately after it is obtained. This type of data is efficient, especially if it is for any decision with up-to-date information. For example, a stockbroker may need to use a stock market ticker to track active stocks in real-time.

Machine data

Machine data [23], as reflected in their name, is data created only with the help of machines, without human interaction. Using the Internet of Things (IoT), sensors, and various technologies, these data are generated by factory systems, information technology, intelligent cars, infrastructure, and many other devices. The basic techniques for data analysis are quantitative and qualitative analysis. These two techniques can be used alone or combined with other techniques concentrated using business knowledge from different types of data.

Quantitative data analysis: Quantitative analysis generally has numerical analysis and therefore works with numerical variables such as measurements, statistics, calculations, etc. As a technique for their analysis, mathematical analysis tools, algorithms, and software for data manipulation and knowledge detection is also essential.

Qualitative data analysis: On the other hand, qualitative data analysis works with rather non-numerical information. This information means working with unique identifiers such as statistics, measurements, properties and labels, categorical variables, etc. The data analyst can use direct access, observations, interviews, revise documents, or lead focus groups in qualitative analysis.

1.2.1 Types of Data Analysis

In this section, we will describe the general types of data analysis. We divide data analysis into five types: descriptive analysis, exploratory analysis, predictive analysis, prescriptive analysis, and mechanics analysis.

Descriptive Analysis

Descriptive Analysis [25] forms the basis for all quantitative data analysis together with simple graphics. Summaries of samples and measures are drawn from it, and it is also used to describe the basic essential features of data in research. For many people, descriptive analytics may not differ from inferential statistics, but the two terms are usually different. The descriptive analysis describes what the data displays and what happens in them. On the other hand, through inference statistics, we need to come to some conclusion that goes beyond the data itself. We can use in various surveys, we can have many factors, such as measuring a large number of people in any section, or we can have many measures. At the same time, this analysis helps us to simplify large amounts of data intelligently and logically. Each type of this analysis limits the amount of data to a more straightforward overview.

Types of descriptive analysis:

- measurement frequency,
- central tendency procedure,
- degrees of dispersion,
- position levels.

Exploratory Analysis

Exploratory Data Analysis (EDA) [26] is used to research data files and recapitulate their basic features. A method called data visualization is usually used for this research. With this visualization, we can determine how to handle data sources to optimally find the answers we require. It also makes it easier for scientists to obtain formulas, discover deviations, verify hypotheses or monitor presumptions. The EDA is widely used to find out what specific data reveals. Thanks to it, it is possibly better to understand variable data files and their ratio. It also helps determine if the statistical techniques we want to use for data analysis are appropriate. This type of analysis is a widely used method in data collection today. The types of explanatory data analysis are:

- one-dimensional graphics,
- one-dimensional non-graphical,
- multidimensional graphics,
- multidimensional non-graphical.

Predictive Analysis

This analysis [27] uses the results to develop a model that can be used to predict future values for different or completely new data. These results appear to be the probability of an interchangeable variable whose basic principle is to estimate significance from a set of initial variables. The fundamental goal of this analysis is to use statistics to determine future performance based on history or presence data.

We distinguish these types of predictive analysis:

- regression models that predict a number, such as how long a customer will generate revenue for a given company over the next year or the number of months or years before a particular part of a machine will stop working,
- classification models that predict class competence, for example, when we classify whether the probability of someone leaving or whether they are responding to a given challenge or whether they have a positive or negative credit risk, often the model is based on 0 or 1, where 1 means the event we intend.

Prescriptive Analysis

This analysis [28] aims to use technologies that help companies make better decisions by analyzing data that is not analyzed. This analysis informs about situations that are available from past or current performance sources and thus determines the strategy or future process. With prescriptive analysis, decisions can be made at any time from an immediate or long-term perspective. It uses artificial intelligence. This intelligence, for example, includes machine learning, which means running a computer program without the intervention of a person who understands and follows data. This learning provides processing of a large amount of generated data. When we get new data, programs from the computer automatically adapt to use that data in a faster and more complex process.

Mechanistic Analysis

The primary purpose of the mechanistic analysis [29] is to understand the accuracy of deviations in the data, which is close to the deviations in other data. This analysis is often used for circumstances that tend to be accurate and have little room for error. This test, which various engineers use, scientists, or doctors, can also be used to test the safety and effectiveness of a given product.

1.3 Machine learning

Machine learning [30] is a data analysis technique that automates the complete data analysis workflow and analytical model and provides more profound, rapid, and complex statistics. It is a unit of artificial intelligence founded on the view that systems can learn from data, recognize ways and make conclusions with minimal human intervention.

The general definition of machine learning [31] is that it is a computational method that uses experience to improve performance or make accurate predictions. In this version, experience represents information from the past that can be freely disposed of by the pupil and usually takes the form of electronic data collected and made available for analysis. There is a possibility that this data could be in the form of digitized training sets or other types of information obtained through interaction with the environment.

The most significant success rate predictions made by students are determined by the quality and size of the information. Using the final sample of randomly selected documents, each marked motif, to predict this topic of ungrateful documents accurately is an example of an educational problem. Of course, the larger the sample, the easier the task. Since labels may not always be all correct, it follows that the difficulty of the tasks must depend on the quality of the labels that are assigned to the documents in the sample. Machine learning is based on designing efficient and accurate prediction algorithms. Space and time, as in any other field of computer science, are the primary and critical measures of the quality of these algorithms.

1.3.1 Types of Machine Learning

This section will explain the types of machine learning that are generally used. We divide types of machine learning into four parts: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

Supervised Learning

This machine learning industry is the most widely used and commonly used today. Supported Learning Machine Learning Algorithms are designed to learn by example. The overall idea of Supervised Learning [32] is formed in such a way that training this type of algorithm is like having a teacher control the whole process. When training a supervised learning algorithm, the data will consist of inputs matched with the accurate outputs. Through training, the algorithm will scan for patterns in the data that correspond with the wanted outputs. After the training, the supervised learning algorithm will accept the new hidden inputs and determine, like a sticker,

that the new inputs will be classified based on data from the initial training. The goal of the escorted learning model is to predict the correct specification for the newly introduced input data. There are three main tasks in supervised learning: classification, regression, and forecasting.

Unsupervised Learning

Unsupervised learning [33] is the opposite of supervised learning. If we want to discover the basic structure of data, unsupervised learning is a great choice. However, unlike supervised learning, we cannot apply uncontrolled methods to regression style or classification style problems. We need to know the mapping of inputs and outputs to perform regression or classification. Because machine unsupervised learning algorithms learn patterns with unmarked ones.

Our unsupervised learning algorithm essentially finds hidden patterns or groupings in the data without the person (or anyone else) having to tag the data or otherwise interfere. This learning method is usually used when our data is not flagged. For example, suppose we wanted to determine the target market for the new product we wanted to market because we would use unsupervised learning. In that case, we have no historical data on the target market's demographics. There are three main tasks in conducting unsupervised learning: clustering, association rules, dimensional reduction.

Semi-supervised Learning

Semi-supervised learning [34] is a model that examines how computers and natural systems, such as people, learn in the attendance of both tagged and untagged data. Traditionally, learning has been investigated either in an independent model, where all data is unmarked or in a supervised model, where all data is marked. The purpose of semi-controlled learning is to explain how a sequence of tagged and untagged data can change learning functions and design algorithms that use such combinations.

Semi-supervised learning is exciting in machine learning and mining because it is simply available for unmarked data and can be used to increase supervised assignments when marked data is unique or expensive. Semi-supervised learning also displays potential as a quantitative tool for understanding the social category of learning, where most inputs are unmarked.

Reinforcement Learning

This technical type of machine learning [35] works on the principle of trial and error through feedback from your actions and experiences. In contrast to unsupervised learning, strengthening various factors is the goal. It uses mapping between input

and output, as opposed to supervised learning, where the feedback provided to the agent is the right set of actions to accomplish the task. This learning aims to find a suitable action regime that would maximize the agent's overall cooperative remuneration.

1.3.2 Types of Machine learning analysis

In this section, we describe the types of machine learning mechanisms that are relevant to a web application. We describe four types of machine learning mechanisms: Natural Language Processing (NLP), Artificial Neural Network (ANN), Recurrent Neural Network (RNN), and Long Short Term Memory (LSTM).

Natural language processing

NLP [36] are techniques that combine linguistics, artificial intelligence, mathematics, and more. These combined techniques are devoted to analyzing and preparing natural language data. Their purpose is to achieve similarity to the common language used for various applications and tasks. NLP applications commonly include machine translation, automatic proofreading, and the like.

Artificial neural network

The basis of an ANN [37] is a collection of units or nodes called artificial neurons. This neuron is the basic computing unit of all neural networks. The function of a neuron is to create an output signal based on an internal potential. The internal potential is given by the evaluated incoming signals and the threshold. The output signal is passed to other neurons connected by evaluated connections or synapses. The neuron has one output and several inputs. Each input adds a certain contribution to the inner potential. The neuron itself cannot solve more complex tasks. For this reason, neurons are among the layers that make up the highest layer building block. Neural networks can consist of three types of layers:

- Input layer: the input layer is the first layer of the neural network, and this layer is responsible for receiving data, functions, and measuring quantities, and for making the mathematical operations of neural networks more accurate, and the inputs are typically normalized within the limits of the activation functions,
- Hidden layers: hidden layers are located between the input and output layers, and they are made up of neurons that are responsible for gaining information and are associated with the analysis process or system,
- Output layer: the output layer has the task of creating the final outputs of the network, and these are the result of processing all previous layers.

Recurrent neural network

A RNN [38] is an artificial neural network. This network uses time series or sequence data. Its algorithms are used for temporal or ordinal issues, which include: speech recognition, image description, NLP, language translation, and the like. They are present in popular applications: Siri and Google Translate. Like convolutional and forward networks, RNN use training data for learning. This data is typical of its memory. It obtains information from previous inputs and thus influences the current inputs and outputs. Unlike traditional neural networks, the outputs of recurrent networks depend on the previous elements in a given sequence.

Four types of RNN are used [39]:

- One-to-One: one-to-one is the simplest type of RNN, allowing one input and output, and input and output sizes are fixed and have the function of a traditional neural network,
- One-to-Many: one-to-many is a more complex type of RNN, one-to-many has multiple outputs per input, and it requires a fixed input size and offers a sequence of data outputs,
- Many-to-One: many-to-one is used when one output from multiple input units or sequences of units is needed, and a sequence of inputs is required to display the fixed output, and common type of this type of network is sentiment analysis,
- Many-to-Many: many-to-Many is used to generate a sequence of output data from a sequence of output units, this type of network is divided into:
 - Same unit size: the number of input and output units is the same, and a common application can be found in Name-Entity Recognition,
 - Different unit sizes: inputs and outputs have different numbers of units, and a common application can be found in machine translation.

Long short-term memory

LSTM [40] is a particular type of feedback neural network. LSTM can learn long-term dependencies, meaning that the current output is dependent on the previous remote input. Classical layers did this only on a theoretical level. Thanks to this advantage, LSTM is used for a wide range of problems: speech recognition, time series prediction, text classification, and the like. Like other layers, LSTM has a structure of concatenated and repetitive modules. However, LSTM has an important property, which is the cell's internal state. It runs through the entire chain and is only affected by linear interactions. The information can thus pass through the string unchanged. LTSMs can modify information in their internal state, thanks to " gateways " elements. Gates are composed of a nonlinear sigmoid function and a

multiplication operation. This LSTM variant is equipped with a forgetting entrance and exit gate.

2 Development tools

This chapter will explain all the essential development tools selected for this work and then used in Chapter 3. Many development tools have been considered for subsequent implementation. Those that the reader learns are selected for several criteria. The main one is user accessibility and simplicity.

From the beginning of the first section, we will get acquainted with the Integrated Development Environment (IDE) and the editor, thanks to which the work would not be able to. Moreover, in the following sections of this chapter, we will get acquainted with two JavaScript libraries. The next one will be a design language to diversify our application, and last but not least, it will be the programming language used in the machine learning part of the thesis.

2.1 Visual Studio Code

Visual Studio Code [41] is a free source code editor. It has built-in programming support for JavaScript and TypeScript languages and assists in other languages within a rich distribution ecosystem. Visual Studio Code uses a folder or workspace system to cooperate with the project and document system to manipulate the source code files project. This feature enables to be linguistically agnostic and select language-specific expansion features.

Although this IDE is from Microsoft, it has grown successful among Linux [42] users as well. Its features - such as IntelliSense, code refactoring, syntax highlighting support for various programming languages, and snippet capabilities - have acquired the favor of web admins and programmers. Although Visual Studio Code is also available in many other Linux packages, users also can compile directly from source code.

2.2 ReactJS

First of all, it is necessary to define what ReactJS [43] is. ReactJS is an open-source JavaScript library used to develop user interfaces, especially for single-page applications. React was first built by Jordan Walke, a software designer working for Facebook. React was first deployed on Facebook's newsfeed in 2011 and Instagram.com in 2012. It is used for manipulating the view layer for web and mobile apps. React also enables us to create reusable user interface components. In React, rather than applying regular JavaScript concerning templating, it applies JavaScript Extensible Markup Language (JSX). JSX is an uncomplicated JavaScript that enables HTML quoting and uses these HTML tag language syntax to execute

subcomponents. HTML syntax is processed inside JavaScript calls of React Framework. However, it can additionally write in classic old JavaScript.

Furthermore enables developers to design large web applications that can change data without reloading the page. The main idea of React is to be fast, scalable, and straightforward. It runs only on user interfaces in the application. This matches the view in the model view controllers template. It can be used by a combination of different JavaScript libraries or frameworks.

Now let us look a little deeper into the world of the React library. We will explain the main parameters and concepts. Let us start with one of the main things, and that is React Components [44]. Components enable us to separate the user interface into independent, reusable elements and think about each element individually. Every component has some "lifecycle methods" that you can edit and run code at particular points in the process. The only method which must be defined in Component is Lifecycle Method **render()**. This method is used for executing other methods and parameters in Class.

```
class SampleThesis extends React.Component{
  render(){
    | return <h1> Hello word </h1>;
  }
}
```

Fig. 2.1: Example of React.Component in Visual Studio Code

In Figure 2.1, we can see the lifecycle method **render()**. After a lighter performance of React. Component, it is time to say something about, i.e., Hooks [45]. This feature is an innovative addition to React 16.8. Hooks are functions that "hook" to the React state and lifecycle properties of functional components. The hooks inside the classes do not work. The hooks are backward compatible, so they do not involve any changes when broken. If we write a functional component and then want to add some state, we will do it first by converting it to a class. Now, however, we can do so by using a hook inside the current functional component. Hooks are related to JavaScript features, but there are two rules for using them:

- hooks should always be called out of conditions and loops or other nested functions, the rule includes that hooky called in the same order each time a component renders,
- hooks can be called from other hooks and cannot be called from the traditional Javascript function.

```

import React, {useState} from 'react'
const [subtraction, setSubtraction] = useState(0)
return(
  <div>
    <button onClick={() => setSubtraction(subtraction - 1)}></button>
  </div>
)

```

Fig. 2.2: Example of Hook in Visual Studio Code

Figure 2.2 displays an example of React function called **Hook**. The last thing we will look at is the React lifecycle methods [46]. Each component of the React class works through several "lifecycle methods." Everything in React consists of either components or parts of components. One lifecycle method has already been mentioned, namely the **render()** method. It is the usual lifecycle method because it is turned on within the class component in the response. Now that the component has been attached and ready, that is when the next React lifecycle method **componentDidMount()** becomes in action. Described as soon as the component is attached and ready. Unlike the **render()** method, **componentDidMount()** enables the application of **setState()**. Pronouncing the **setState()** here will update the state and make another rendering but it will happen before the browser updates the user interface. Another lifecycle method is **componentDidUpdate()**, is caused as soon as the update starts. The most commonly used is updated in status change responses. The last important lifecycle method is used before any component is destroyed and disconnected and is called **componentWillUnmount()**.

```

class SampleThesis extends React.Component {
  constructor(props) {
    super(props);
    this.state = {color: "black"};
  }
  componentDidMount() {
    setTimeout(() => {
      this.setState({color: "blue"})
    }, 100)
  }
  render() {
    return (
      <h1>The rewire web color is {this.state.color}</h1>
    );
  }
}

```

Fig. 2.3: Example of **render()** and **componentDidMount()** in Visual Studio Code

On Figure 2.3, is shown example of code in Visual Studio Code that shows model example of using two lifecycle methods called **render()**, and **componentDidMount()**. The output of this functions is "The rewire web color is blue".

2.3 LeafletJS

In this section, we go through one of the essential JavaScript library for this project [47]. Interest in map applications has increased rapidly in recent years. Just look around, and almost every web application uses these applications. They are designed for navigation systems or e-shops, and there is much more. After 2013, an avalanche of these applications broke out, but the focus was on the Leaflet Javascript library. It is a library for creating interactive web maps. The advantage of this library is its simplicity and complexity, as it is possible to have a simple map using a few lines of code. However, it is also possible to have an extensive map with hundreds of lines, fulfilling all interactivity. In order to create a Leaflet map, we need to have the following four things:

- create a `<div>` element to hold map,
- need a link to a CSS file,
- create a map object,
- add base layer.

There are two possible variants to loading Leaflet into the code: via a link to the guest source or downloading a copy to our local system and linking that copy. This description is for clean leaflet documentation. The project uses the React Leaflet [48], which provides links between the library and ReactJs 2.2. It is important to say that this does not replace the Leaflet, but the leverages are to the abstract layers of the Leaflet as React components. The Leaflet itself does the rendering of the Document Object Model (DOM). It is not as it might seem at first glance, namely that it would draw React in the DOM. React renders `<div>` only when rendering Map Container components. Respond Leaflet uses the context of the Application Programming Interface (API) to make the necessary children's elements available. The MapContainer component creates its context for the Respond and each instance of the Leaflet. Other components and hooks that are only like React Leaflet are only descendants of MapContainer. It is important to watch out for restrictions. One of them is that the Leaflet immediately calls the DOM after loading, so the responding Leaflet is not agreeable to server-side rendering.

Moreover, another limitation is that the exposed components are ideas for leaflet layers, not DOM elements. Some produce features that can be updated immediately by requesting the Leaflet-exposed setters. In contrast, others should be replaced

by placing unprecedented value on their key features because React’s algorithm manages them precisely.

2.4 Ant Design

This section is lightly introduced to the open-source enterprise-level design system, and user interface library React [49]. It comes with a high-quality React component set with an excellent option for customizing the theme. Ant Design was created in 2015. Since then has over half a million downloads a week from the Node Package Manager (NPM) registry and is supported by dozens of programming languages. It is based on four design values [50]: Natural, Certain, Meaningful, Growing.

The goal is to reduce redundancy and high user costs. This goal also includes trying and unifying user interface specifications to improve the user experience. Some functions are connected to the project. Therefore, it would be good to say something about the main ones. The map that is being executed stands on, ie. Modal [50]. It is used so that the user can interact with applications without jumping to a new page and the current process on the current page. Another big component of any project that uses Ant Design is the confirmation button, cascader, or menu. Ant Design makes all of these features available. The advantage of this designer is that, besides the design component in question, sample codes are also made available, which can be easily modified for the needs of everyone.

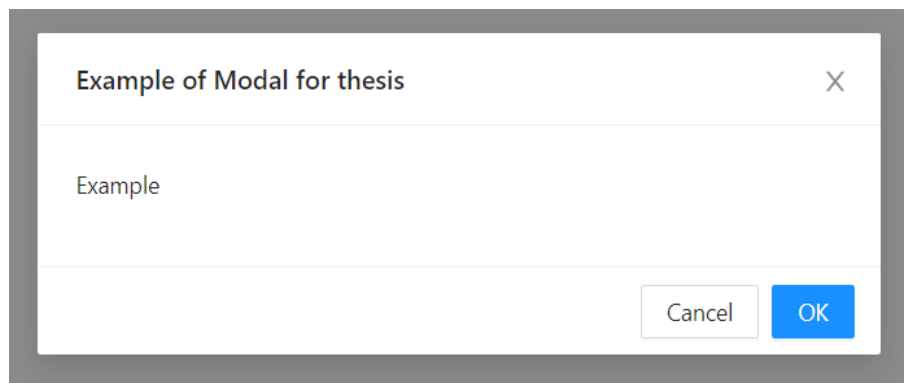


Fig. 2.4: Example of Modal and Buttons

The Figure 2.4, shows a Modal that is created for effective demonstration and also used in our project with interactive map inside it.

2.5 Python

This section presents the programming language used in the server part of the thesis using the machine learning algorithm. Python [51] is an open-source, user-friendly language and is often recommended as the first language for beginning programmers because its illustration, automatic datatype recognition, or automatic memory allocation that Python is about will take maintenance of the programmer without extraordinary effort. These properties are typical of interpreted languages, which are translated every time the program is run, which inevitably directs to a significant slowdown. Another advantage is the commonness with which new versions of Python are released (a new version is released approximately every year). Python has a rich standard library, including a simple calculator.

Python has evolved a foundation in data science, enabling data analysts and other specialists to use the language to perform complex statistical calculations and develop data visualizations. Moreover, construct machine learning algorithms, manage and analyze data, and perform other data-related tasks.

Python can create various data visualizations, such as line and bar graphs, pie charts, histograms, and 3D graphs. Python also has many libraries that allow coders to write data analysis and machine learning programs faster and more efficiently.

3 Implementation

This chapter consists of two main subchapters. The first chapter describes the overall implementation of our dynamic web application. The second chapter will look at a preview of back-end development and machine learning for the web application. The task of the work is to extend an existing web interface and create a worldwide interactive map offering a database of job advertisements with their filtering for data analysis.

At first, this chapter briefly describes why development tools are used in the thesis and why they are essential. Eventually, we move on to the more significant part of this chapter, starting the server. Therefore, it is necessary to explain how the used libraries in the application are uploaded to the development environment. The next part of this chapter is an internal view of the standalone application and the functions used with a more detailed description.

Many development technologies were used for this work as a development environment where the rest of the technologies are applied. We chose visual code studio version 1.62 because it does not need additional installation of extensions for the required application as in other environments. Furthermore, ReactJS is used as a programming language and HTML5 and CSS. ReactJS is more straightforward to grasp for novices than pure Javascript. Thanks to a component-based attitude, a well-formulated lifecycle, and a simple JavaScript application. Based on its simple and user-friendly interface, we chose this library's application because its composition allows adding different resources freely. Next, a design visualization tool called Ant Desing was used, which uses the antd library 4.17.0-alpha.10. We chose this so that our web application had the same design elements as it distributed. As the last Python programming language used for machine learning, we use version 3.10.4.

3.1 Map Development

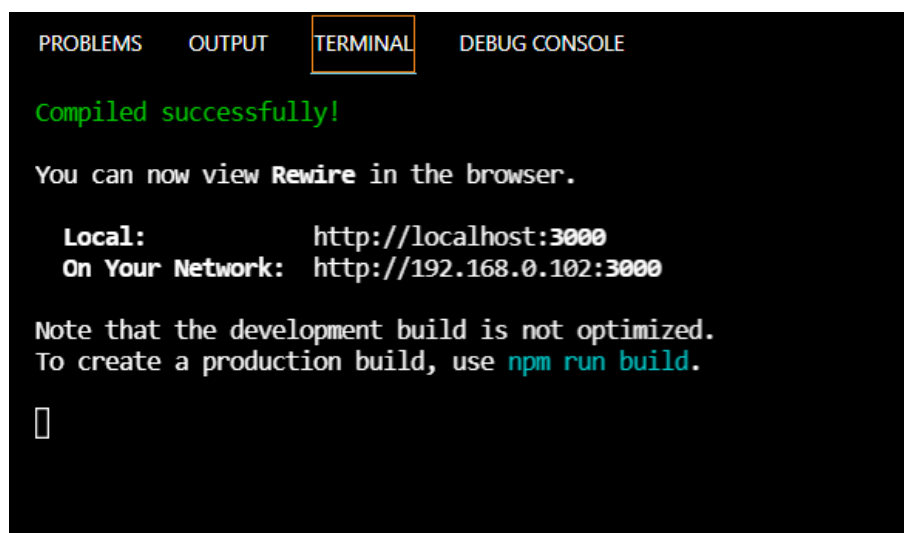
This section covers the overall development of the map. This section aims to explain to the reader the functionality of the individual elements of the map.

3.1.1 Start server and installations

This section will look at the necessary installations to start the application. Let us start with the complete basics, namely by installing the JavaScript platform Node.js. Node.js can be easily downloaded from their website [52], then is just needed to follow the instructions. React is the most advantageous base for hosting and running a web server for React. There are two reasons for this [53]:

- using NPM, Node works with the NPM registry and easily installs any package via the NPM Command Line Interface (CLI),
- node combines React into one file for easy assembly using the webpack and many other Node modules.

As we mentioned, we are creating an extension for a web application that a colleague is developing. Therefore, it is necessary to align all packages and libraries first. There are two ways to do this, by opening the command prompt and using the `cd..` command to get to the folder where we saved the project. The second option requires opening the **file -> open Windows PowerShell** in the project folder. This option is simpler and faster than the command prompt option. Once in the CLI, type **npm install**. This action will cause all additional and necessary packages to be installed and, without error, run ReactJs. After completing the necessary packages, we can start the development server. There are three ways to run the **npm start** command. In any case, it is always necessary to be in the maid with the project. Visual Code Studio has the advantage of having its terminal, where it is possible to run all commands, such as in Windows PowerShell. The command starts the development server with base port 3000. After loading the command, the type-check is first activated to verify the data types in the source code by the compiler. Once everything is set up correctly, and the server does not respond with the error, a message about a successful launch will appear, and the Uniform Resource Locator (URL) address **http:localhost:3000** will be loaded in the browser. To shut down the development server, it is necessary to press the combination of keys Ctrl + C and then confirm the batch job.



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Compiled successfully!

You can now view Rewire in the browser.

Local:      http://localhost:3000
On Your Network: http://192.168.0.102:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

□
```

Fig. 3.1: Successful npm start command

In the Figure 3.1 we can see successfully compiled command and also local ad-

dresses. The last thing what is needed to do is to install another library for map development which is LeafletJS library, using the **npm install react-leaflet** command in the terminal. This allows binding React and Leaflet.

3.1.2 Setting a map

The Leafletjs library, installed in the previous chapter, is used to create the interactive map environment. As a next step, it is necessary to upload this library to the project so that we can continue working with it. Upload is done at the beginning of the program using the command **import {"components from the library, for me it is MapContainer, GeoJSON and TileLayer"} from "react-leaflet"**.

Every function in the library has its functionality. The first point that emerged from the analysis is that we set and created a map. The {MapContainer} component is used for this. This component causes the position of the entire map, functions such as zoom and centering, and much more, {MapContainer} is a cornerstone. Without it,, we could not build a map, but ,it has no design or style so far. Design or style fixes another component, namely "TileLayer," which gives the map a subtext of how the continents, states, and water are depicted. Style maps can be found on open-source websites. We chose an international format because the format is the clearest and simple.

```
return(  
< div >  
  <MapContainer  
    style = {height : "80vh"}  
    zoom = {2.4}  
    scrollWheelZoom = {true}  
    minZoom = {2.4}  
    center = {[30,0]} >  
    <TileLayer  
      url = ' https : //s.tile.openstreetmap.org/z/x/y.png'  
    />  
  </MapContainer>  
< /div >  
);
```

Tab. 3.1: Example of code for Map componets

In Table 3.1 is shown example of the code with some components of leaflet library.

3.1.3 GeoJSON

At this point, the map is visible with specific properties. For the correct functionality, namely the possibility of interactivity, we must add the most crucial component, "GeoJSON," for our purposes, a component that adds the required options. It is a component that adds, i.e., Layers. For the map to be globally interactive, i.e., for it to have the required functionality and the user to be able to work with it, it is necessary to add a file to our project that contains a worldwide database of state layers. This file is called GeoJSON. The JSON file is an open standard file format and data exchange format. For example and understanding, we created this file for the Czech Republic. This creation can quickly be done through an open-source website [54].

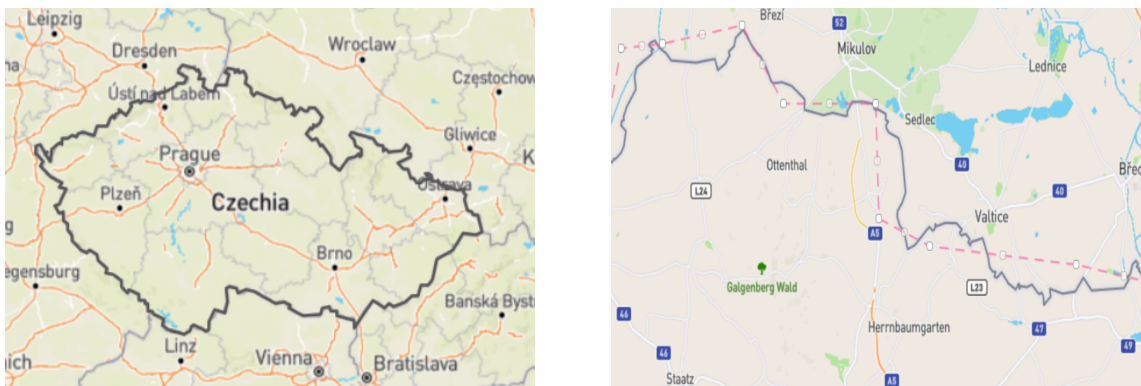


Fig. 3.2: The left chart shows the Drawn line for Czechia and the right chart shows detail of the left chart

As shown on the right of Figure 3.2, the state outline and the GeoJSON component are points or coordinates. The whole and every country made in this style always contain geographical coordinates, longitude, and latitude, respectively. If someone wanted to make a state outline according to the borders, the set would have countless coordinates. In this example, we have about two thousand coordinates. All this is then displayed in the generated file. We made this task easier and did not create the outlines of the state for everyone due to time constraints. There are many freely available generators for the whole world on the internet, so we used one and generated our file, which is called "**countries.geo.json**". It contains essential data used in our work.

The main reason is that the already mentioned plotting coordinates highlight the states on the map, using the attribute "geometry," type "polygon." In addition to this reason, there is another reason: when we click on the state, the state's names are displayed. This ensures that the attribute features other properties and after mallow Admin in this file.

```

"type" : "FeatureCollection",
"features" : [
{
"type" : "Feature",
"properties" : {
"ADMIN" : "Aruba",
"ISO - A3" : "ABW"
},
"geometry" : {
"type" : "Polygon",
"coordinates" : [
[
[
-69.996937628999916,
12.577582098000036
],
[
-69.936390753999945,
12.531724351000051
],

```

Tab. 3.2: Example code of countries.geo.json for Aruba state

Table 3.2 is displayed an example of the GEO.JSON file that we use. This file needs to be imported into the development environment. Once this point is made, it is necessary to define this component as previous ones in the return function to work with it. It can be used to set its style, height, width, etc. However, unlike the previous ones, it contains the **"data"** attribute, which is an object that displays the highlighted states after assigning the correct element. The **"onEachFeature"** attribute is used for the functionality of individual states, after assigning the **"onEachCountry"** function, which is used for various purposes in our program. This feature is the cornerstone and functionality of the entire map as a complex and lastly **"style"**, which allows us to change the styles of the map dynamically. Specifically, it is deselected after sending data for filtering in our case. The user always comes to a clean map after sending it. This causes another function called in this function called **"style"**, where a negated condition is used to determine if the selected states contain features from the GeoJSON file.

3.1.4 Interactive functions

At the moment, the map is displayed, but so far, without any interactivity. We want three ways of interactivity on the map. If the user moves the mouse cursor to one of the states, the state on which the cursor is colored will be colored to that color. To do this, we need to create a completely new function, which is named "mouseover."

At the beginning of the function, the parameter "e" is defined, an abbreviation for the event object. An event is an event that causes a function to execute. Subsequently, the layer variable is defined in the "mouseover" function. An event with the "target" property is stored in this variable. We are working with this variable in the next steps. For the functionality to display a popup window with the content that we define in the next steps in the "onEachCountry" function, it is necessary to add the "openPopup" leaflet feature to the "layer" variable. The popup does not appear at this time because it is still undefined.

To make this function work, it is necessary to make another variable, namely "ishighlightedlayer," in which we store the content for the condition, which we will then use. We store in this variable "layer.options.weight" equals 2. This number can be any if it is not 1. We will use this variable in the condition if, using the variable "ishighlightedlayer" as its condition. Since the value of weight equals 1, this condition will not apply, and it will always use its other part to fulfill it. If the weight value is equal to 2, then less transparency is added to this style, and this will cause the state to be additionally highlighted the next time we move the mouse over the state. In this part of the condition, the state coloring style is set. It is specifically the color cyan ("78DFFF").

This function is then used in the "onEachCountry" function, which we get in. So we have prepared the first way of map interactivity. Because leaving the cursor out of the map would leave the map styles and not reset them, creating a second way of interactivity is necessary to solve this problem. Therefore, we will create a "mouseout" function that looks essentially the same as the "mouseover" function. The only difference is that "openPopup()" is not defined in the "mouseout" function and also in the conditional part, where white ("FFFFFF") is applied instead of cyan. Now we have two functions ready.

The last function to set is the "selectedLayer" function, which is more complicated. This function is used to indicate the state after a user click. This feature is the main reason for the whole map. Here we will use the functionality to select the state and its information, which will then be used to filter the parameters of the state assistance on the main page of the web application. The function is named "selectedLayer," and the main structure is no different from the previous two. This function is different mainly in the part where the condition is used. The condition

is to set the map styles again after clicking, but React Hooks are also. To set these Hooks, it is necessary first to define them in a program outside this function. This hook is named "selectedCountries." In the first part of the condition, the setting part of this hook creates an arrow function with a parameter. This parameter then uses the JavaScript array function, the "filter." It then compares it with a new parameter if this parameter does not equal the state's name in the "countries.geo.json" file. That is, it creates and finds filtered arrays of state names. The second part of this condition, i.e., the map styles are set again in the other part. The React hook setting part, where the arrow function is created again, for which a parameter is also created. The collision and multiplication library is used. This library is called "lodash" and is imported at the beginning of the program. The array function is applied again. This time, it is the "concat" function of the state names. The "concat" function causes one or more arrays to be combined into one. All arrays of states that have been clicked into one will be connected, and there will be no duplication, thanks to the "lodash" library.

```

function selectLayer(e) {
  let layer = e.target
  let ishightedlayer = (layer.options.weight === 2)
  if (ishightedlayer) {
    setSelectedCountries((oldArray) =>
      oldArray.filter(item => item !== layer.feature.properties.ADMIN))
    layer.setStyle({fillColor : '78dff', weight : 1, fillOpacity : 0.2})
  } else {
    setSelectedCountries((oldArray) =>
      _.uniqWith(oldArray.concat(layer.feature.properties.ADMIN)))
    layer.setStyle({fillColor : '3898ff', weight : 2, fillOpacity : 0.5})
  }
}

```

Tab. 3.3: Code for selecting a country

In Table 3.3, is displayed the code for selecting countries after clicking on the country in the map, and also basic functionality for subsequent filtering.

3.1.5 Database import

For the job to be done, it is necessary to start working with the job ads data and add it to a certain country. The database currently contains job ads only for Europe. For other continents, a job position is not yet offered. To add specific offers to

certain countries, we need to familiarize ourselves with the database. The database is transferred to the project in the new **init_data.js** file. In this file, it is necessary to make a common database constant variable array. It is named the same as the file, i.e., **init_data**. In this array, there are data with job ads. These contain several variables that are very important for the project. There is, for example, a source, a link to the source that the company offers this position, a job description, and much more. Two aspects are the most important for our work, namely **"title"** and **"country"**. After this little and quick introduction to the `init_data` field, it is time to start working with it. Since this field is an external file, we must insert it into the project to work with the data. It is important that our created map has access to this data and can also work with it. This problem can be solved by importing a **init_data.js** file in our project.

3.1.6 Main function

At the beginning of the project, a global constant is created in the **MapModal.js** file, named **"MapModal"**. One of the variables that contain our databases is `"data."` For the final functionality, it is necessary to unify our data.

We move to the function **"onEachCountry"**, where all the functions takes place. In this function, it is necessary to create a new constant named **"countryName"**, in which we take the country parameter, which we defined at the beginning of the function, and then add properties.ADMIN to it, which is the path to the names of individual states from the `"countries.geo.json"` file.

After creating this constant, it is vital to create a second constant called **"countryAds."** This constant use the JavaScript array method filter (`()`) attached to our global data variable (job offers). In this method, a new parameter `"ad"` is created using the array function. Two variables are compared: the states with the parameter `"ad"` and `"countryName"` create an array of several objects. So in the `"onEachCountry"` function, we have filtered and ready states so far. Now is the time to start using them.

To display the names of states, the number of current job offers, and other labels, after clicking on that state, it is necessary to create another variable, namely **"popupContent."** HTML commands are applied here in combination with JavaScript. First, a `<div>` is created to group HTML tags with the class, then a title is added, to which the constant `"countryName"` is added, which means that the titles of the names of certain countries are displayed in this form for the time being. For better clarity, we will use the `` tag, which is used to define text with different formatting so that we can add supporting text for clarification.

The last important part of this variable is the condition that compares the length

of the constant "countryAds" with zero. Thus, if the length of this variable is more significant than zero, that is, if there is a job offer in the country. The number of job offers in the country will be added to the "popupContent" variable in bold by calling the constant "countryAds" and the "length" parameter attached to it. "

The second part of the condition is that if there is no job offer in the country, a simple label "none" is added to the "popupContent" variable. The last part closes the total <div> tag attached to this variable. The function currently has a pop-up functionality set. However, if we want to run the application, these windows will not pop up. It is necessary to set the pop-up window in the "onEachCountry" function. We will do it quickly and use the "bindPopup" function. This feature is selected from the leaflet library, and it will cause the window to pop up. We insert the "popupContent" variable programmed by us into the parameter of this function. After launching the application, it is possible to click on the country and display a pop-up window with the respective country's name and the number of job offers.

As a last resort, in order for everything to work, it is necessary to connect our interactive Functions 3.2.1 with the map. This connection is set again in the "onEachCountry" function. We will use the leaflet library function again. Add "on" to our "layer" parameter. Simply put, this will activate the map layers. We then insert three properties into this function, which are "**mouseover**," "**mouseout**" and "**click**." We then use these colons to combine these three properties with our functions, which are "mouseover," "mouseout" and "selectLayer." After the application, the states are now colored after the mouse cursor is moved and reset again when the mouse is moved. When clicked, we will see the relevant countries, and at the same time, we will see a pop-up window with the name of the country and the number of ads in it.

```

const countryName = country.properties.ADMIN
const countryAds = data.filter((ad) => ad.country === countryName)
let popupContent = < divclass = "popover - content" >
< h2 > {countryName} < /h2 >
< spanclass = "ant - typographyant - typography - secondary" >
Number of job adds : < /span > ‘
if (countryAds.length > 0) {
popupContent+ = ‘ < strong > {countryAds.length} < /strong > ‘
} else {
popupContent+ = ‘none‘
}
popupContent+ = ‘ < /div > ‘

```

Tab. 3.4: Code for adding number of job adds to the map



Fig. 3.3: Example of country with Cybersecurity Job Ads with "mouseOver" function used

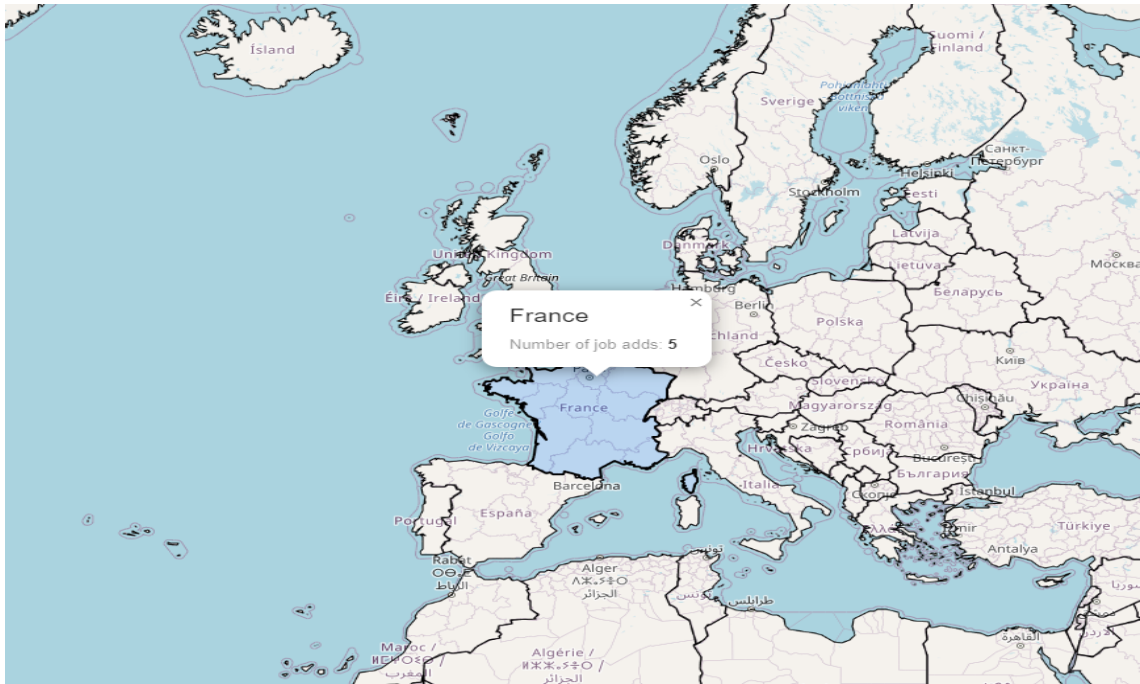


Fig. 3.4: Example of country with Cybersecurity Job Ads with "selectedLayer" function used



Fig. 3.5: Example of country with no Cybersecurity Job Ads with "selectedLayer" and "mouseOver" functions used

Table 3.4 shows the code for adding the number of jobs that will be added to the pop-up window. Figure 3.3 shows an example of a country with the number of job opportunities using the "mouseover" function. Figure 3.4 shows an example of a country with the number of job opportunities using the "selectedLayer" function, and Figure 3.5 shows an example of a country that does not contain any job offers using the "selectedLayer" and "mouseover" functions. These features belong in their essence together and show the overall functionality to display numbers of job adds in European countries.

3.1.7 Last modifications of map modal

In this last part of the map development, we will look at the latest modifications in the Ant Design library element. In this element, namely "Modal," it is necessary to set a certain functionality. Our modal contains several parameters. Such as "className," which is named "mapModal." This property sets or returns the class attribute of the element. Then there are elements such as centering, title, and width. However, the main thing for us is the "footer." In this property, we set the text with the text "Selected **{selectedCountries.length}** countries."

"SelectedCountries" is the second part of React Hook for display, which will cause a digit with the number of clicked states to appear instead of this function when we click on this function. In the footer, we set buttons, reset, which resets all selected countries, further return, which will take us back to the original page and submit button, which confirms our selection and overwrites the filtering table in the main application. After reopening the map, the styles are subsequently deleted, and the user can overwrite the table. The "mapFilteredData" parameter is created in the "filterViaMap" function in the main application, as well as the new React Hook "filteredInfo." countries. Furthermore, "ModalVisible" is set to "false." When we click the button in Map Modal, the value is saved in "filterViaMap." Moreover, using React Hook, the countries are then transposed into the table filters on the main page.



Fig. 3.6: Overall look of the application

Figure 3.6 shows the overall appearance of a functional application.

3.1.8 Filtering in main table

As mentioned, the web application has, in addition to the map, also the central part, where the table is located, in which all job offers that are contained in the database are displayed. There is a bar with the possibility of filtering in the table. Our task is a filter bar for several criteria. Namely, the criteria are source, company, country, partners, and type. Everything must work dynamically when adding more jobs to the database. We need to create a function that selects keywords from the database and creates new arrays from them.

We named this function "getColumnOptions," where a condition is created that compares if it is equal to the date, then selects the data and converts it to "YYYY" format. Otherwise, it creates an array of other keywords, assigns them text and value, and then sorts them. Next, the constant "tableComuns" is created, to which several parameters are assigned. We add the "filters" parameters to them, to which we always add a specific criterion name to the "getColumnOptions" function.

A filter icon appears in the toolbar, in which the individual data of a specific criterion is located. In order for the data to be applied after selection, it is necessary to add the parameter "onFilter," where the parameters "value" and "record" are

loaded, and the names of the appropriate criteria are added to them, which are transferred to the lower case.

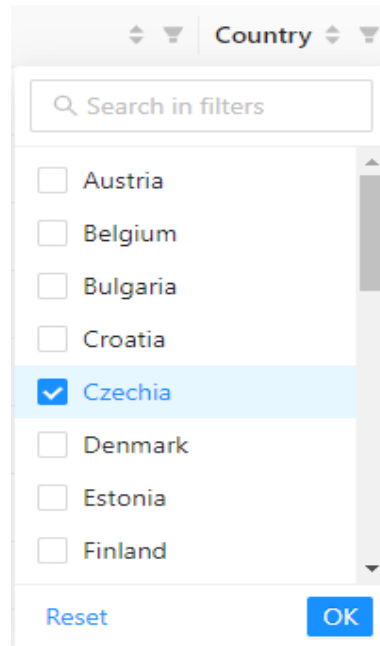


Fig. 3.7: Filtering by Czechia country

Title	Source	Company	Country
Senior Information Security Analyst	LinkedIn	Conduent	Czechia
Security Analyst	LinkedIn	Oracle	Czechia
Infrastructure Security Architect - Prague	LinkedIn	AVON	Czechia
Tier 1 Security Engineer	LinkedIn	Verizon	Czechia
CYBER SECURITY & IT COMPLIANCE MANAGER (M/F)	LinkedIn	Rieter	Czechia
Group Product Manager	LinkedIn	Avast	Czechia
Threat Hunting & Response Senior Analyst	LinkedIn	Novartis Czech Republic	Czechia
Associate Security Engineer	LinkedIn	Blackboard	Czechia
Security Engineer	LinkedIn	CGI Space	Czechia
Sr. Security Engineer	LinkedIn	Mavenir	Czechia

Fig. 3.8: Filtered table by Czechia country

Figure 3.7 shows the filtering options in the table by country. Map filtering activates this filter in the same way. Figure 3.8 shows the filtering output of Figure 3.7.

Figure 3.7 displays example of cybersecurity skills, other IT skills and soft skills for the first two jobs in the dataset.

3.2.2 Start and necessary installations

We will now start working with the Python programming language and explain how to install the necessary elements to run the tests and applications themselves. At first, it is essential to have Python installed. It is easy to install. Search the Python website [55], and in the download section, choose the platform on which you want to install it. The latest version is 3.10.4. It is then recommended to check several criteria in the installation, such as "Add python 3.10 PATH". This criterion will cause us to be able to use Python at the command prompt. There is also the possibility of ticking other criteria, which are voluntary. "Documentation" and "preferred installer program (pip)" are necessary to us. All options can be re-installed if necessary. We have Python installed at this time. Many programs such as PyCharm can be used to open Python programs. Because we have Visual Studio Code installed, we will use this environment. We have access to the codes provided to us and can view and edit them.

For example, if we want to run the most important program, we must first find the path to this program, which we will copy. Then we insert this path into the command line and put the command "cd" in front of it. Another option is to open the program folder and then open it in the folder by clicking the right mouse button on the "Open in Terminal" button. A PowerShell terminal with a direct set path will open. The last option is to open the terminal in visual studio code. We will use this option for simplicity. To run a program, in our case, for example, "**main.py**," defines the main paths to folders and other programs that use machine learning. A separate run command (`python main.py --run-type both --output-folder result --token-raw-truth data/large_dataset/ex_list.xlsx --token-raw-txt-folder data/large_dataset/ads_txt_ftr`) is written in this file. We insert it into the command line and confirm. Several problems appear. The main problem is that we need to install all the necessary libraries to run this command. There are many of these modules. One of the most important is "argparse." Used to define parameters that are written to the command line. In our case, it is all that follows the words "**main.py**." Additional modules such as "numpy," "pandas," "transformers," and others are required. The installation is performed using the "**pip install module_name**" command at the command prompt. After the installation is complete, we can run the program. The "README.md" file contains the types of commands to run. It can be run either to run the overall implementation or only an individual to control, such as a tokenizer with statistics or a standalone LSTM

model. We can also run commands for prediction only and also for learning only. The beginning of the command always remains the same. The following parameters of the type, the resulting component, and more change.

We have created the command (`python main.py --run-type statistics --output-folder output_model --token-raw-truth data/dataset.xlsx --token-raw-txt-folder data/ads_txt_fldr`) to make things easier so that we do not have to run time-consuming programs. This command is stored in the "main.py" file. We also had to modify the code at least and add another argument of the "statistics" types it runs. It runs similarly to other commands.

3.2.3 Modification of Python Script

The most important part of this section is a modification of one of the scripts that generate an image (**radar_plot.png**) with a graph containing the analyzed job offers using the NLP model. The picture generates two sets, namely training and testing. First, it is necessary to determine which script modifies and saves this image. All scripts that have functionality are stored in the "src" folder. All we have to do is find the "statistics" folder and the **save_statistics.py** file in this folder. This file stores the currently mentioned image in the "output_model/statistics" folder. At this point, let us look at what the script looks like and how it works. Several libraries are imported from the beginning of the script mentioned above. Under import, the functionality of the image begins. The variable set named "statistics_folder" merges folder paths using the "join" library.

Furthermore, this variable is a parameter of the function that checks if a folder is created that is attached to "join." If the folder does not exist, it will create it. However, the script has a condition that finds and retrieves the dataset path. For us, the dataset, in this case, is the file **truth_csv**," which is essentially an extract of our already described dataset.

Truth_csv contains only skills and the values 1 and 0. In addition, the dataset is loaded into memory in the condition. Next, the data is divided into train and test parts, and a new data frame is created. It contains 90 train parts and 10 test parts. Then a numeric index is added. Then the names of the skills are added, and 37 + 1 places in angles are created. An array of angles from 0 to 2 of those labels is then created, and one is added to complete the circle. Number 1 is then deleted so that it is not displayed. Another function assigns a color to the graph using the "cm" or "colormap" module. Next, the size of the overall image is set, and the background is set to "white." A for loop is created, where the numeric index is first erased, the color of the graph is added, then the connection is added, and a centerline is created. The graph is drawn first to the right and then to the left. For

there to be a space between them, the indentation must also be set by adding rows and columns. It must also cause the graph to rotate correctly. These functions are performed by the variable "ax."

```
for i, row in df.iterrows():
    color = my_palette(i)
    data = df.iloc[i].drop('index').tolist()
    data += data[:1]

    ax = plt.subplot(3, 2, i+1, polar=True)
    ax.set_theta_offset(pi / 2)
    ax.set_theta_direction(-1)

    ax.tick_params(axis='x', which='major', pad=15)

    ax.set_rlabel_position(0)
    ax.plot(angles, data, color=color, linewidth=2, linestyle='solid')
    ax.fill(angles, data, color=color, alpha=0.4)
```

Fig. 3.10: Part of the python script without modification

Figure 3.10 shows part of a python script that generates a graph without dynamically adding job skills labels.

At this point, we will edit the script and add labels to the chart. To add a label to the graph, we must create a condition and assign individual labels to the created angles using the "matplotlib" module and the "thetagrids" function into which angles and labels are loaded. Now we have the names stored in the image, but the names are unfortunately badly rotated and overlap each other. Therefore, it is necessary to turn them so that we can read them. First, we find the x and y coordinates. These coordinates return the position of the names and assign a label to the first graph. We store new positions with the addition of text in the variable "lab." The names now hit the chart, and this fact is confusing and illegible. Therefore, it is necessary to add to the variable "lab" the indent from the center, which we chose 0.2. The smaller the number, the smaller the indent. Since the circle is 360 °, the text rotates 180 ° for better user readability. Next, the graph is plotted using the variable "ax," and the fill is also determined.

```

for i, row in df.iterrows():
    color = my_palette(i)
    data = df.iloc[i].drop('index').tolist()
    data += data[:1]

    ax = plt.subplot(3, 2, i+1, polar=True)
    ax.set_theta_offset(pi / 2)
    ax.set_theta_direction(-1)

    ax.tick_params(axis='x', which='major', pad=15)

    if i==0:
        lines, labels1 = plt.thetagrids(np.degrees(angles), labels=labels)

        labels1 = []
        for label, angle in zip(ax.get_xticklabels(), np.degrees(angles[::-1])):
            x,y = label.get_position()
            lab = ax.text(x,y-0.2, label.get_text(), transform=label.get_transform(),
                ha=label.get_ha(), va=label.get_va())

            lab.set_rotation(angle+100)

            if angle < 180:
                lab.set_rotation(angle+280)

            labels1.append(lab)
        ax.set_xticklabels([])

    ax.set_rlabel_position(0)
    ax.plot(angles2, data, color=color, linewidth=2, linestyle='solid')
    ax.fill(angles2, data, color=color, alpha=0.4)

```

Fig. 3.11: Part of the python script with modification

Figure 3.11 shows part of a python script that generates a graph with dynamically adding job skills labels.

The last function is to save the image using the "savefig" function and assign the path using "join" to "statistics" under the name "**radar_plot.png**." The last command of this script saves skills to a Comma-Separated Values (CSV) file, which is saved in the same folder as our image.

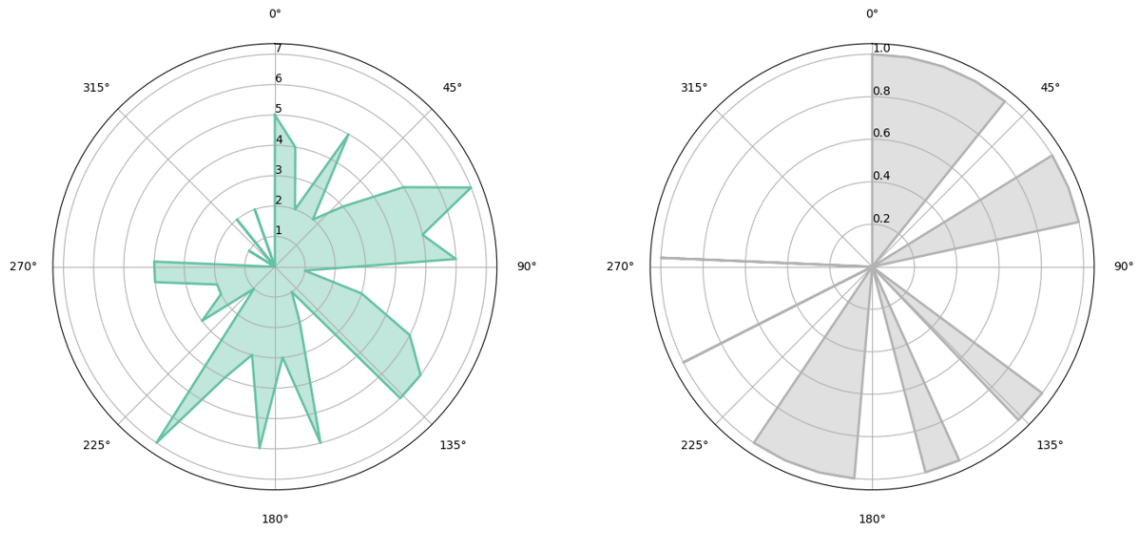


Fig. 3.12: Radar_plot.png without labels

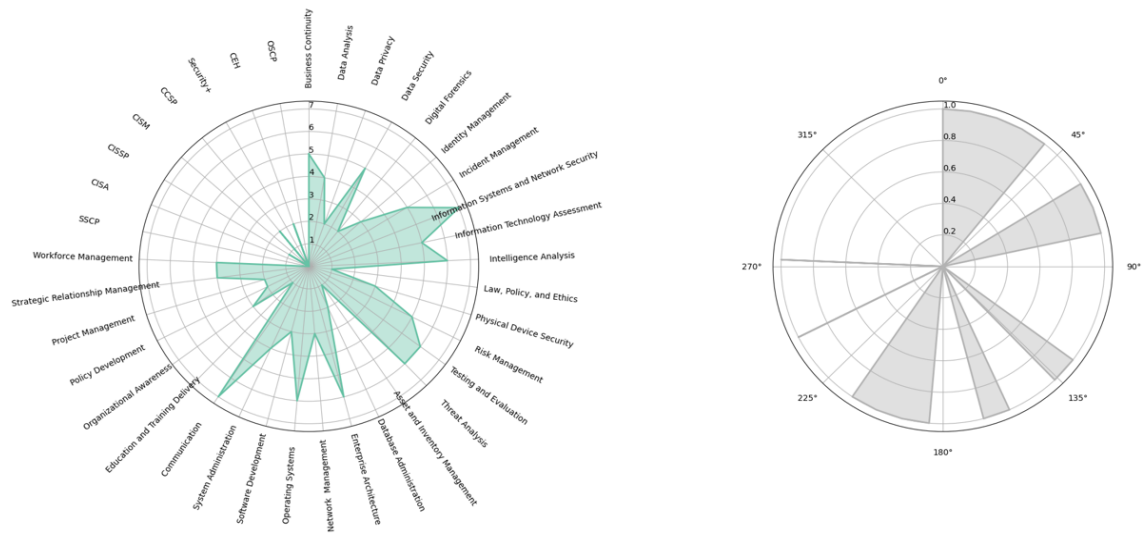


Fig. 3.13: Radar_plot.png with labels

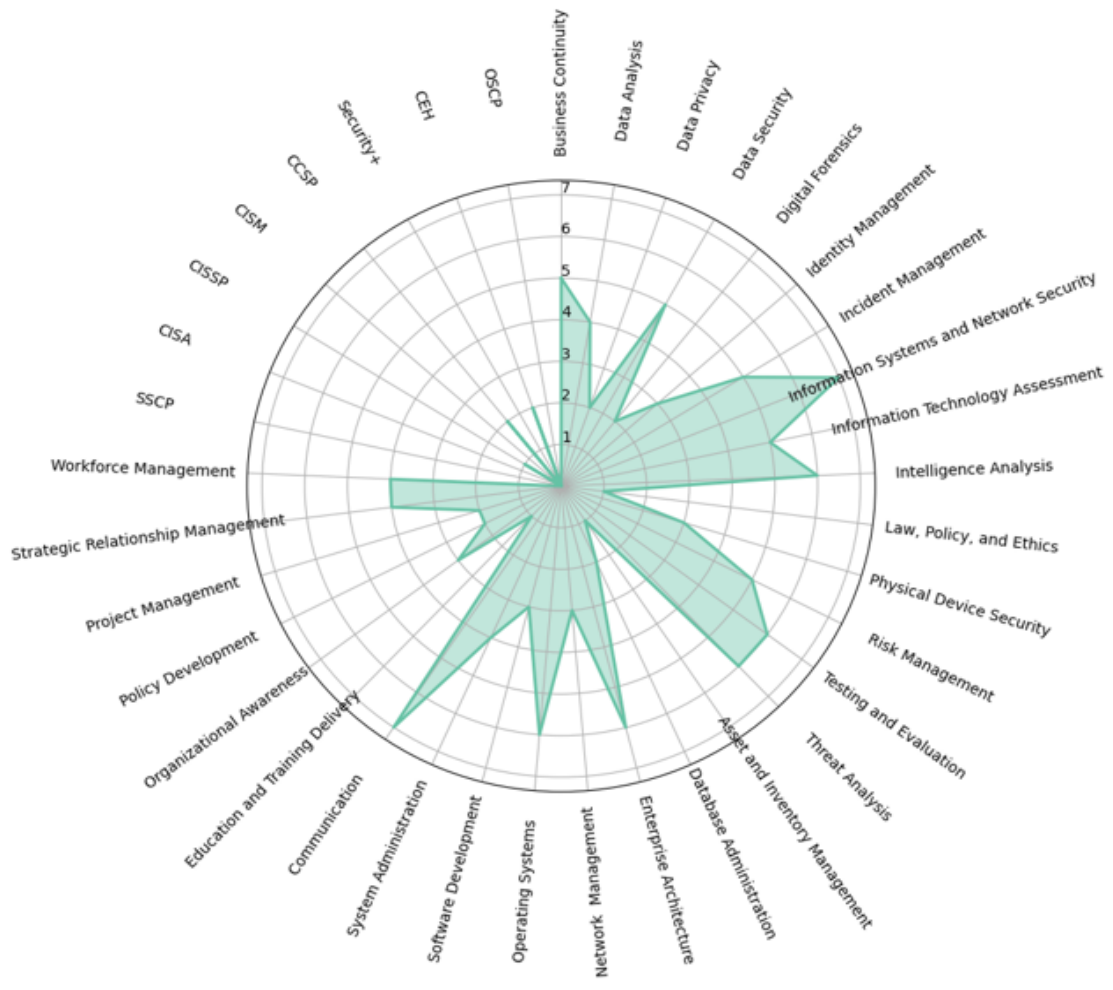


Fig. 3.14: Detail of left plot with labels

Figure 3.8 shows the unmodified generated image radar_plot.png, the performed test result. Figure 3.9 shows the modified radar_plot.png with added job skill labels from the dataset. Figure 3.10 shows a detail of the left part of Figure 3.9

Conclusion

This thesis aimed to create a world map web application that visualizes generic data in geographical coordinates and filters for job ads cybersecurity in Europe. The map is integrated into a web application that can filter job ads according to several parameters and then use an analysis of machine learning of the filtered data to display job skills and graphs of them.

To achieve the goals of the thesis, it was necessary to study the ways of data visualization. It was also necessary to get acquainted with the technologies and possibilities for creating interactive maps. It works with various libraries, but the most important was the ReactJS JavaScript library associated with LeafletJS. Therefore, it was necessary to study these libraries, their architecture, philosophy, and how it works with user data. It was also important to study the principles of machine learning used for machine learning analysis, such as LSTM, artificial neural network, recurrent neural networks, and natural language processing. It was also necessary to get acquainted with the already created server and the subsequent editing of one of the scripts containing the image generation. Advanced filtering using several factors had to be changed to single-factor filtering using a country filter during the semester. Due to the non-functional coloring of the state after selecting the country in the side menu.

The result of the implementation part extends the existing REWIRE website with a web application that displays a world map with interactive functions of LeafletJS library. Map contains the possibility of showing a country the sum of their job positions within the cyber security of a particular country. Option to select, color, and remove the country and filter the countries, passed to the main application table. Also, edit and add filtering of the database several parameters in the main application table. Subsequently, the use of machine learning and modification of the python script generates the figure located on the server part of the application. The figure of graph now shows the dynamically assigned labels of job skills from the dataset. In the future, this application could be extended worldwide. It also could serve as a database for people looking for a job in cyber security and help further analyze the most commonly used skills in the industry.

Bibliography

- [1] Domo. *Data Never Sleeps 9.0*, 2021. [online]. URL: <https://www.domo.com/learn/infographic/data-never-sleeps-9>.
- [2] (ISC)². (isc)² cybersecurity workforce study, 2021. URL: <https://www.isc2.org/-/media/ISC2/Research/2021/ISC2-CybersecurityWorkforce-Study-2021.ashx>.
- [3] ENISA. *Stocktaking of information security training needs in critical sectors*, 2017. [online]. URL: <https://www.enisa.europa.eu/news/enisa-news/stocktaking-of-information-security-training-needs-in-critical-sectors>.
- [4] Margot James. *Cyber Security Skills Strategy*, 12 2018. [online]. URL: <https://www.gov.uk/government/publications/cyber-security-skills-strategy>.
- [5] ENISA. *Ad-Hoc Working Group on the European Cybersecurity Skills Framework*, 2020. [online]. URL: https://www.enisa.europa.eu/topics/cybersecurity-education/european-cybersecurity-skills-framework/adhoc_wg_calls.
- [6] *Rewire*, 2021. [online]. URL: <https://www.rewireproject.eu/>.
- [7] Mary MacPherson. *Website vs Web App*, 07 2019. [online]. URL: <https://medium.com/@essentialdesign/website-vs-web-app-whats-the-difference-e499b18b60b4>.
- [8] *What is a desktop application*, 2021. [online]. URL: <https://www.rascasone.com/cs/blog/desktop-web-aplikace>.
- [9] *Desktop and web applications*, 2021. [online]. URL: <https://sluzby-internet.webnode.cz/web/desktopove-a-webove-aplikace-rozdil/>.
- [10] Paul Stanley. *Advantages and Disadvantages of Web Applications*, 2021. [online]. URL: <https://www.pssuk.com/AdvantagesWebApplications.aspx>.
- [11] Tars Geerts. *Static and dynamic websites*, 08 2021. [online]. URL: <https://www.mlytics.com/blog/static-and-dynamic-websites-whats-the-difference/>.
- [12] *TCP/IP Services and Client/Server Architecture*. [online]. URL: <http://etutorials.org/Linux+systems/red+hat+linux+9+professional+>

secrets/Part+I+Setting+Up+Red+Hat+Linux/Chapter+6+Network+Setup/
TCP+IP+Services+and+Client+Server+Architecture/.

- [13] *Fat Server*, 2021. [online]. URL: <https://www.techopedia.com/definition/7142/fat-server>.
- [14] *Thin Server*, 2021. [online]. URL: <https://www.techopedia.com/definition/15315/thin-server>.
- [15] *3-Tier Architecture*. [online]. URL: <https://digitalthinkerhelp.com/what-is-client-server-architecture-diagram-types-examples-components/>.
- [16] *History of internet*, 2008. [online]. URL: <https://www.tvorba-webu.cz/php/historie.php>.
- [17] *A short history of Web*, 2021. [online]. URL: <https://home.cern/science/computing/birth-web/short-history-web>.
- [18] Gustavo Rossi, Oscar Pastor, Daniel Schwabe, and Luis Olsina. *Web engineering: modelling and implementing web applications*. Springer Science & Business Media, 2007.
- [19] *HTML and CSS*, 2016. [online]. URL: <https://www.w3.org/standards/webdesign/htmlcss>.
- [20] Dušan Janovský. *Introduction to JavaScript*. [online]. URL: <https://www.jakpsatweb.cz/javascript/javascript-uvod.html>.
- [21] *Introduction to PHP*. [online]. URL: https://www.tutorialspoint.com/php/php_introduction.htm.
- [22] Thomas A Powell. *Web design: kompletní průvodce*. Computer Press, 2004.
- [23] *Top 4 Data Analysis Techniques That Create Business Value*. [online]. URL: <https://online.maryville.edu/blog/data-analysis-techniques/>.
- [24] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.
- [25] Prof William M.K. *Descriptive Statistics*. [online]. URL: <https://conjointly.com/kb/descriptive-statistics/>.
- [26] *IBM Cloud Education. Exploratory Data Analysis*, 08 2020. [online]. URL: <https://www.ibm.com/cloud/learn/exploratory-data-analysis>.

- [27] *Predictive Analytics*. [online]. URL: https://www.sas.com/en_us/insights/analytics/predictive-analytics.html.
- [28] Troy SEGAL. *Prescriptive Analytics*, 07 2021. [online]. URL: <https://www.investopedia.com/terms/p/prescriptive-analytics.asp>.
- [29] Neelam Tyagi. *Mechanistic Analysis*, 01 2021. [online]. URL: <https://www.analyticssteps.com/blogs/7-types-statistical-analysis-definition-explanation>.
- [30] *Data Analytics and Machine Learning*, 2020. [online]. URL: <https://www.answerrocket.com/data-analytics-machine-learning/>.
- [31] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- [32] Aidan Wilson. *A Brief Introduction to Supervised Learning*, 09 2019. [online]. URL: <https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590>.
- [33] Kurtis Pykes. *Unsupervised Machine Learning Explained*, 08 2021. [online]. URL: <https://towardsdatascience.com/unsupervised-machine-learning-explained-1ccc5f20ca29>.
- [34] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.
- [35] Shweta Bhatt. *Reinforcement Learning 101*, 03 2018. [online]. URL: <https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292>.
- [36] IBM Cloud Education. *What is natural language processing?*, 07 2020. [online]. URL: <https://www.ibm.com/cloud/learn/natural-language-processing?fbclid=IwAR2pDgxsizyi5HCqAH0TpcsmJp1EA34Uv1XASVmkjU-v7bebWiNzhyMAbE>.
- [37] *Artificial Neural Network*. [online]. URL: <https://www.sciencedirect.com/topics/computer-science/artificial-neural-network>.
- [38] IBM Cloud Education. *What are recurrent neural networks?*, 09 2020. [online]. URL: https://www.ibm.com/cloud/learn/recurrent-neural-networks?fbclid=IwAR0UHBqrR-bSZ9i_kzWHowGcVM_bSegNzt9Rue7Xr015M0qXEZmEfJ7vPxA.

- [39] Arslan Bajwa. *What are the types of RNN?* [online]. URL: <https://www.educative.io/edpresso/what-are-the-types-of-rnn?fbclid=IwAR03XdBY6C16SeAohTGBadHyUKwNCCD0uewMn-a3z8K-Q9f5BbvU91pXGps>.
- [40] *Understanding LSTM Networks*, 08 2015. [online]. URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/?fbclid=IwAR0yxy3NpTyWjI0h9hKg13a-zXD1NjR1LC1cACaHur-jopcZvuJFjEzF8Bw>.
- [41] Jonas Kjær Rask, Frederik Palludan Madsen, Nick Battle, Hugo Daniel Macedo, and Peter Gorm Larsen. Visual studio code vdm support. *John Fitzgerald, Tomohiro Oda, and Hugo Daniel Macedo (Editors)*, page 35, 2021.
- [42] Ivan Kmeto. *VSCodium - VSCode bez telemetrie*, 05 2019. [online]. URL: <https://www.linuxexpres.cz/software/vscodium-vscode-bez-telemetrie>.
- [43] Nitin Pandit. *What And Why React.js*, 02 2021. [online]. URL: <https://www.c-sharpcorner.com/article/what-and-why-reactjs/>.
- [44] *React.Component*, 2021. [online]. URL: <https://reactjs.org/docs/react-component.html#static-getderivedstatefromprops>.
- [45] *React Hooks*. [online]. URL: <https://www.javatpoint.com/react-hooks>.
- [46] Mosh Hamedani. *React Lifecycle Methods - A Deep Dive*, 11 2019. [online]. URL: <https://programmingwithmosh.com/javascript/react-lifecycle-methods/>.
- [47] Paul Crickard III. *Leaflet. js essentials*. Packt Publishing Ltd, 2014.
- [48] Paul Le Cam. *Introduction*, 2021. [online]. URL: <https://react-leaflet.js.org/docs/start-introduction/>.
- [49] Alex Taylor. *Crafting Beautiful UIs in React Using Ant Design*, 09 2018. [online]. URL: <https://www.digitalocean.com/community/tutorials/react-beautiful-uis-ant-design>.
- [50] *Introduction*, 2021. [online]. URL: <https://ant.design/docs/spec/introduce>.
- [51] *What is Python? Executive Summary*, 2022. [online]. URL: <https://www.python.org/doc/essays/blurb/>.
- [52] *NodeJS*, 2021. [online]. URL: <https://nodejs.org/en/>.

- [53] Prayaag Kasundra. *Use nodejs with react*, 3 2020. [online]. URL: <https://www.simform.com/blog/use-nodejs-with-react/>.
- [54] *Web page for creating GeoJson file*, 2021. [online]. URL: <https://geojson.io>.
- [55] *Python*, 2022. [online]. URL: <https://www.python.org/>.

Symbols and abbreviations

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
DDoS	Distributed Denial of Service
TCP/IP	Transmission Control Protocol/Internet Protocol
FTP	File Transfer Protocol
GUI	Graphical User Interface
PHP	Hypertext Preprocessor
XML	Extensible Markup Language
XHTML	Extensible HyperText Markup Language
WML	Wireless Markup Language
SQL	Structured Query Language
IoT	Internet of Things
EDA	Exploratory Data Analysis
NLP	Natural Language Processing
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
IDE	Integrated Development Environment
JSX	JavaScript Extensible Markup Language
DOM	Document Object Model
API	Application Programming Interface
NPM	Node Package Manager
CLI	Commnad Line Interface
URL	Uniform Resource Locator

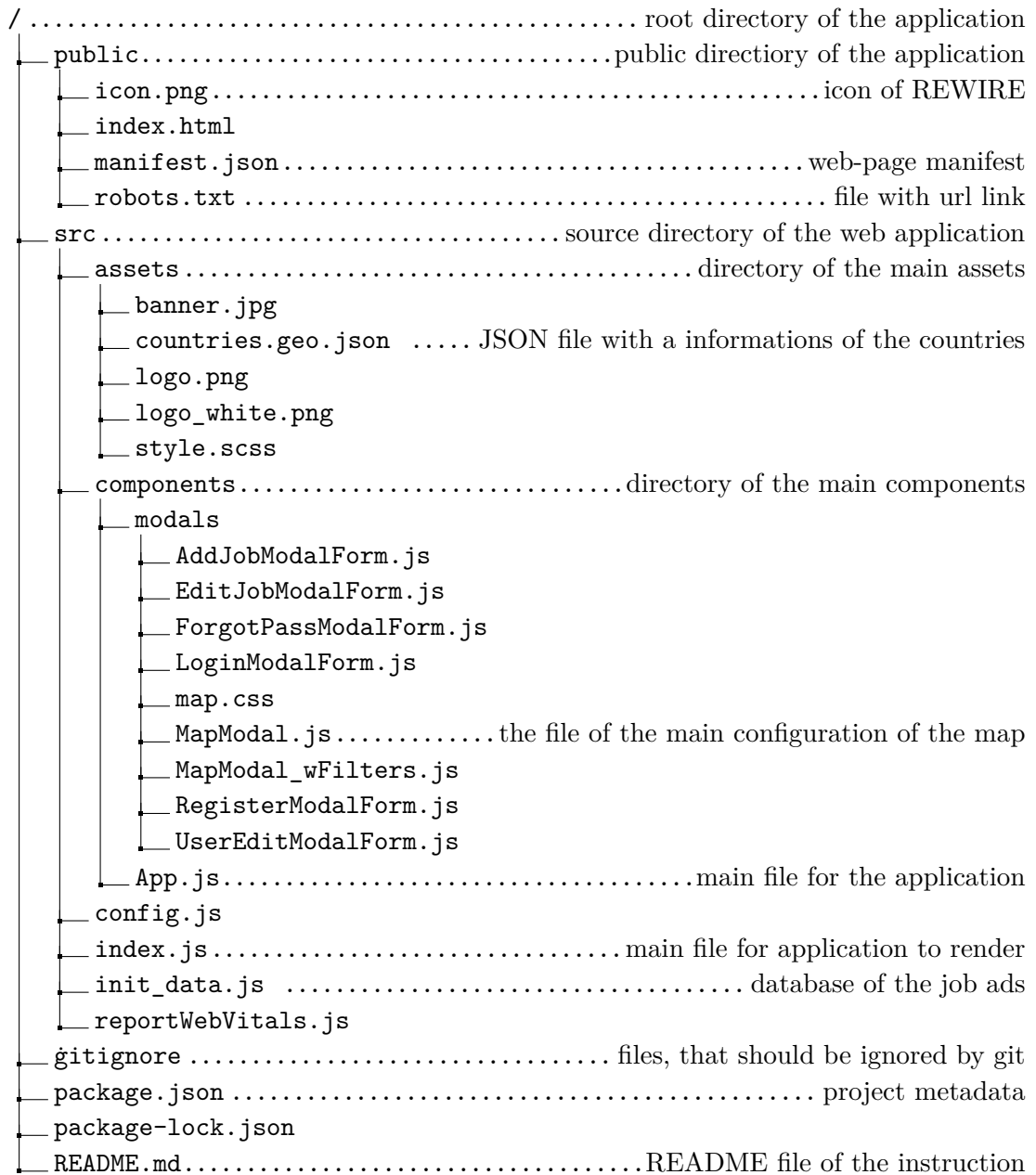
pip preferred installer program
CSV Comma-Separated Values

List of appendices

A	Directory tree for web application	68
B	Directory tree for Server part	69
C	Manual - running the web application and python script	70
C.1	Web application	70
C.2	Python script	71

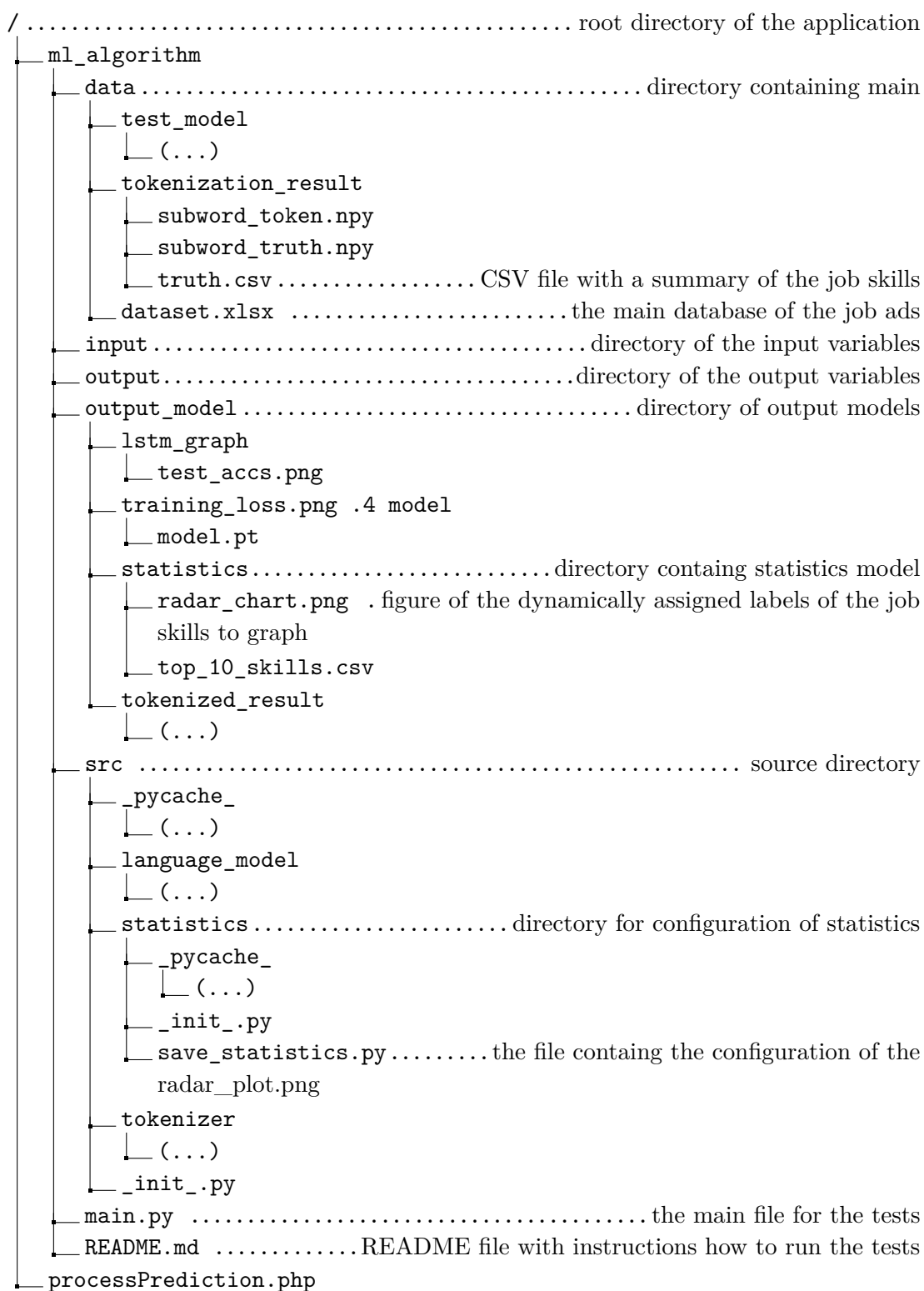
A Directory tree for web application

In this chapter we will go through the directory tree for the whole web application.



B Directory tree for Server part

In this chapter we will go through the directory tree for the whole server part.



C Manual - running the web application and python script

In this chapter we will explain how to run a web application and also a modified python script

C.1 Web application

To start the application it is necessary:

1. download Node.js,
2. start terminal:
 - command prompt:
 - use the "cd" command to get to the application folder.
 - powershell:
 - right-click and select "open in terminal" in the application folder.
 - visual studio code(recommended):
 - download the development environment,
 - upload the application folder,
 - run the terminal directly in visual studio code.
3. execute the "npm install" command,
4. execute the "npm start" command:
 - starting the server takes a few seconds,
 - do not close the terminal!,
 - when the server is turned on, the browser opens on <http://localhost:3000>.
5. the application can also be opened on a web page <https://rewire.informacni-bezpecnost.cz/>.

C.2 Python script

To start the Python script with radar_plot.png it is necessary:

1. download Python,
2. start terminal,
 - command prompt:
 - use the "cd" command to get to the application folder.
 - powershell:
 - right-click and select "open in terminal" in the application folder.
 - visual studio code(recommended):
 - download the development environment,
 - upload the application folder.
3. execute "python main.py --run-type statistics --output-folder output_model --token-raw-truth data/dataset.xlsx --token-raw-txt-folder data/ads_txt_ftr" to run a file with a radar_plot.png (nothing will happen),
4. download necessary packages using command "pip package",
5. make some change in save_statistics.py file and open the figure of graph,
6. execute the Python command once more.