

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Software pro správu slepých map



2024

Vedoucí práce:  
Mgr. Tomáš Kühn, Ph.D.

Michael Hanzl

Studijní program: Aplikovaná informatika,  
prezenční forma

## **Bibliografické údaje**

Autor: Michael Hanzl  
Název práce: Software pro správu slepých map  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2024  
Studijní program: Aplikovaná informatika, prezenční forma  
Vedoucí práce: Mgr. Tomáš Kühn, Ph.D.  
Počet stran: 35  
Přílohy: elektronická data v úložišti katedry informatiky  
Jazyk práce: český

## **Bibliographic info**

Author: Michael Hanzl  
Title: Blind Map Management Software  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2024  
Study program: Applied Computer Science, full-time form  
Supervisor: Mgr. Tomáš Kühn, Ph.D.  
Page count: 35  
Supplements: electronic data in the storage of department of computer science  
Thesis language: Czech

## Anotace

*V rámci práce je navržena a implementována webová aplikace pro tvorbu a správu slepých map pro výuku zeměpisu (případně i biologie, dějepisu či dalších předmětů). Aplikace umožňuje snadné nahrání podkladových map v klasicky podporovaných grafických formátech. Aplikace pak umožňuje kreslit přímo na mapy základní grafické tvary a vytvářet interaktivní testy, které lze jednoduše distribuovat studentům online.*

## Synopsis

*The thesis introduces a web application designed and implemented for creating and managing blank maps for geography education (potentially also for biology, history, or other subjects). The application allows the easy upload of base maps in commonly supported graphic formats. Users can directly draw basic graphical shapes on the maps and generate interactive tests that can be easily distributed to students online.*

**Klíčová slova:** webová aplikace; slepé mapy; testy;

**Keywords:** web application; blind maps; tests;

Děkuji panu Mgr. Tomáši Kührovi, Ph.D. za odborné vedení bakalářské práce.

*Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Analýza</b>	<b>8</b>
2.1	Způsob řešení . . . . .	8
2.2	Webová aplikace . . . . .	8
<b>3</b>	<b>Použité technologie</b>	<b>9</b>
3.1	HTML . . . . .	9
3.2	CSS . . . . .	9
3.3	JavaScript . . . . .	9
3.4	Node.js . . . . .	10
3.5	Node Package Manager . . . . .	10
3.6	Express . . . . .	10
3.7	EJS . . . . .	10
3.8	MongoDB . . . . .	11
3.9	Mongoose . . . . .	11
3.10	Fabric.js . . . . .	11
3.11	Bcrypt . . . . .	11
3.12	Dotenv . . . . .	11
3.13	jsonwebtoken . . . . .	12
3.14	Nodemailer . . . . .	12
3.15	Nodemon . . . . .	12
3.16	Postman . . . . .	12
3.17	Tippy.js . . . . .	12
3.18	Google Material Icons . . . . .	13
3.19	Visual Studio Code . . . . .	13
<b>4</b>	<b>Programátorská dokumentace</b>	<b>14</b>
4.1	Struktura projektu . . . . .	14
4.1.1	controllers . . . . .	14
4.1.2	middlewares . . . . .	17
4.1.3	models . . . . .	18
4.1.4	routes . . . . .	18
4.1.5	service . . . . .	19
4.1.6	views . . . . .	19
4.2	Struktura databáze . . . . .	20
<b>5</b>	<b>Uživatelská příručka</b>	<b>21</b>
5.1	Úvodní strana . . . . .	21
5.2	Přihlášení do aplikace . . . . .	21
5.3	Registrace . . . . .	21
5.4	Odhlášení z aplikace . . . . .	21
5.5	Zapomenuté heslo . . . . .	22

5.6	Správa testových slepých map . . . . .	22
5.6.1	Testy . . . . .	22
5.7	Tvorba slepé mapy . . . . .	22
5.7.1	Nahrání map . . . . .	23
5.7.2	Tvorba otázek . . . . .	23
5.7.3	Dokončení . . . . .	25
5.7.4	Uložení testu . . . . .	25
5.7.5	Absolvování testu . . . . .	25
5.7.6	Zobrazení výsledku . . . . .	26
5.7.7	Detail výsledku . . . . .	26
5.8	Tabulka výsledků . . . . .	26
5.9	Administrátorská sekce . . . . .	27
5.9.1	Přehled . . . . .	27
5.9.2	Uživatelé . . . . .	27
5.9.3	Testy . . . . .	28
	<b>Závěr</b>	<b>31</b>
	<b>Conclusions</b>	<b>32</b>
	<b>A Obsah elektronických dat</b>	<b>33</b>
	<b>Literatura</b>	<b>34</b>

# 1 Úvod

Existuje velké množství aplikací, které se věnují testování znalostí formou slepých map. Nenašel jsem však mezi nimi žádnou, která by umožnila uživatelům si takový test vytvořit a poté jej spravovat a jednoduše distribuovat. Výsledkem mé bakalářské práce bude takovou aplikaci vytvořit.

Mým cílem bude aplikace, která uživatelům umožní nejen již předem vytvořené testy absolvovat, ale také je mít možnost vytvářet, mazat, sdílet a absolvované testy pak automaticky vyhodnotit s možností tyto výsledky zobrazit, upravit a nebo smazat. Uživatelům, kteří si budou chtít takový test vytvořit, bude umožněno nahrání jedné nebo více vlastních podkladových map v klasických podporovaných grafických formátech. Nad těmito mapami pak budou moci snadno vyznačovat hledané oblasti ať už se jedná o bod, linii a nebo plochu. Test bude tvořen otázkami, ve kterých si uživatel zvolí mapu, nad kterou pak vykreslí hledaný pojem, nastaví bodování a vytvoří možnosti, ze kterých se bude vybírat. Pro každý test pak lze nastavit jeho název, známkování a časový limit ke splnění.

## 2 Analýza

### 2.1 Způsob řešení

Pro řešení bakalářské práce byla vytvořena webová aplikace, která nabídne dynamický a přístupný prostředek pro tvorbu, správu a provádění testů slepých map. Výhoda webové aplikace je, že ji uživatel nemusí instalovat. Aplikace běží na jakémkoli operačním systému a uživatel také nemusí řešit žádné aktualizace.

### 2.2 Webová aplikace

Webová aplikace je program, který je obvykle umístěn na vzdáleném serveru, ke kterému uživatelé mohou přistupovat z webového prohlížeče (Google Chrome, Firefox aj.). To znamená, že aby mohl uživatel aplikaci používat, tak nemusí na svém zařízení instalovat žádný software. Nejběžnější funkce webových stránek, jako je třeba nákupní košík, vyhledávání a filtrování produktů na vašem oblíbeném e-shopu.

Webové aplikace mají architekturu klient-server. V této architektuře přenáší webová aplikace data a komunikuje mezi dvěma hlavními komponentami: klientem a serverem. Klientská část je spuštěna na zařízení uživatele a zajišťuje interakci a prezentaci dat. Na druhé straně serverová část řídí logiku aplikace, zpracovává požadavky klienta a manipuluje s databází. Pro vývoj klientské části se používají jazyky HTML, CSS a JavaScript. Nejčastějšími jazyky pro vývoj serverové části jsou PHP, C#, Python, Java a run-time prostředí Node.js, které umožňuje JavaScriptu běžet na straně serveru. [1]



## 3 Použité technologie

V této části se stručně zaměřím na technologie, které byly při vývoji webové aplikace využity.

### 3.1 HTML

HTML neboli HyperText Markup Language [2] je značkovací jazyk, který definuje strukturu a sémantiku obsahu webové stránky. Obsah webové stránky mohou tvořit texty, obrázky, tabulky, multimédia a další prvky. Hypertext znamená, že dokument obsahuje odkazy, které umožňují čtenáři přeskočit na jiná místa v dokumentu nebo na úplně jiný dokument. Jazyk HTML sestává z elementů, které umožňují definovat, jak by měl být obsah elementu prezentován uživateli. Každý element začíná otevírací značkou <název značky>, prezentovaným obsahem

a obvykle uzavírací značkou </název značky>. Elementy ještě často obsahují atributy (id, class atd.), které mohou poskytovat další informace. Od verze HTML5 lze také využívat vlastní atributů, které musí být označeny předponou data-

### 3.2 CSS

Kaskádové styly (Cascading Style Sheets) [3] je jazyk pro stylování a pozicování HTML prvků. Pomocí CSS můžeme měnit fonty, barvy, velikost, rozvržení obsahu na stránce, přidávat animace a spoustu dalších věcí. Kaskádové styly se zapisují jako pravidla. Každé pravidlo obsahuje selektor následovaný složenými závorkami, ve kterých se zapisují bloky deklarácí. Každá deklarace se sestává z identifikátoru vlastnosti následovaná dvojtečkou a pak hodnota té dané vlastnosti. Tato deklarace je pak ukončena středníkem.

```
1 h1 {  
2   color: red;  
3 }
```

Zdrojový kód 1: Pravidlo změni barvu html prvku h1 na červenou

### 3.3 JavaScript

JavaScript [4] je multiparadigmatický, jedno-vláknový, dynamický programovací jazyk podporující objektově orientovaný, imperativní i deklarativní způsob programování. Jedná se o jazyk založený na prototypch. JavaScript se převážně používá pro tvorbu webového front-endu. Lze pomocí něj interaktivně měnit obsah webové stránky na straně klienta. K tomu se používá tzv. DOM (Document Object Model). Ten je dostupný globálním kontextu v objektu "document".

Soubor s javascriptovým kódem se vkládá mezi HTML tagy `<script>`, který se umísťuje do libovolné části HTML souboru a nebo do externího souboru.

### 3.4 Node.js

Node.js [5] je open-sourcové, multiplatformní prostředí, které umožňuje spouštět JavaScriptový kód na straně serveru. Aplikace vytvořené pomocí Node.js jsou vysoce škálovatelné, a to hlavně díky jeho asynchronní architektuře. Node.js běží v jednom vlákně, které je neblokující. Jakmile například nějaký klient potřebuje načíst nějaká data z databáze, tak vlákno nečeká až se žádost vykoná, ale může začít zpracovávat nějaký další požadavek. Jakmile má databáze připraven výsledek, tak zprávu vloží do fronty událostí. Node.js tuto frontu na pozadí sleduje a jakmile se zde objeví nějaká událost, tak ji vyzvedne a zpracuje.

### 3.5 Node Package Manager

Node Package Manager [6] je open-source online knihovnu balíčků, která je nainstalována zároveň s Node.js. Umožňuje nám do projektu přidávat další javascriptové knihovny pomocí rozhraní příkazového řádku.

### 3.6 Express

Express [7] je minimalistický Node.js webový framework, který poskytuje bohatou sadu funkcí pro rychlý vývoj webových aplikací. Poskytuje mechanismy pro:

- Psaní obslužných funkcí pro požadavky s různými HTTP metodami na různých URL cestách.
- Snadnou integraci s renderovacími enginy jako je třeba EJS.
- Nastavování běžných nastavení webové aplikace, jako je port pro připojení a umístění šablon, které se používají k renderování odpovědí.
- Správu middleware.

### 3.7 EJS

Embedded JavaScript neboli EJS[8], je jednoduchý šablonovací jazyk, který umožňuje vložit JavaScriptový kód do jazyka šablony. Tento kód se pak používá ke generování HTML stránky. Šablona si ponechává syntax HTML a umožňuje provádět interpolaci dat. Typické použití je právě s výše zmíněným frameworkem Express, který umožňuje nastavit EJS jako renderovací systém a je pak používán back-endem ke generování stránek v reakci na nějaký HTTP požadavek. EJS dále umožňuje podmíněné renderování, iterace a nebo třeba vytvořit znovupoužitelné šablony, které pak lze vkládat do dalších šablon.

## 3.8 MongoDB

MongoDB [9] patří mezi tzv. NoSQL databáze. Jedná se o dokumentově orientovanou databázi, která k reprezentaci dat používá kolekce. V tom se liší od relačních databází, kde se k reprezentaci dat používají tabulky. Data jsou v MongoDB uložena jako binární JSON tzv. BSON. V datech jsou zakódovány informace o typu a délce, což umožňuje mnohem rychlejší průchod, než kdyby se používal k uložení dat JSON.

## 3.9 Mongoose

Mongoose [10] je knihovna pro objektové datové modelování (ODM) pro MongoDB a pro Node.js. Knihovna umožňuje snadné mapování datových modelů na dokumenty v MongoDB. Děje se tak pomocí uživatelem vytvořených schémat. Tyto schémata nám definují strukturu dokumentu. S pomocí schémat pak umožňuje vytvářet objekty, které pak lze ukládat do databáze.

## 3.10 Fabric.js

Fabric.js [11] je javascriptová knihovna pro práci s HTML5 canvasem. Jedná se o interaktivní objektový model nad HTML `<canvas>` elementem. Knihovna poskytuje vyšší úroveň abstrakce než klasické HTML Canvas API, kde musíme nejprve vytvořit 2d kontext, na který můžeme kreslit. Ve Fabric.js se nepracuje přímo s primitivními kreslicími operaci jako jsou tahy, obdélníky a obrazy, ale pracujeme přímo s objekty na plátně.

## 3.11 Bcrypt

Bcrypt [12] je javascriptová knihovna sloužící k hashování uživatelských hesel. Bcrypt používá ještě tzv. sůl (salt) což je náhodná hodnota, která je přidána k heslu před jeho zahashováním. Sůl je generovaná náhodně pro každé heslo. To znamená, že i když dva uživatelé mají stejné heslo, tak se jejich hash bude lišit.

## 3.12 Dotenv

Dotenv [13] je javascriptový balíček sloužící ke správě proměnných, které mohou obsahovat citlivé informace pro nastavení aplikace. Může jít například o heslo k databázi, tajný klíč sloužící k podpisu JSON web tokenu a nebo třeba přihlašovací údaje k emailu pokud implementujeme možnost resetování hesla. S nedávným vydáním Node.js verze 20.6.0 má nyní vestavěnou podporu pro soubor `.env` a není jej už nutné externě přidávat.

### 3.13 jsonwebtoken

Jsonwebtoken [14] je javascriptová knihovna implementující JSON Web Token, což je otevřený standard (RFC 7519), který definuje kompaktní a samostatný způsob pro bezpečný přenos informací mezi dvěma stranami. Tyto informace lze ověřit a důvěřovat jim, protože jsou digitálně podepsány. Struktura JSON web tokenu se skládá ze tří částí oddělených tečkou, tyto části jsou:

- Header – Nese informaci o algoritmu kódování a typu tokenu.
- Payload – Obsahuje užitečné informace, které mohou být dekodovány kýmkoli, kdo má přístup k tokenu.
- Signature – Podpis celého tokenu, který slouží k ověření pravosti.

### 3.14 Nodemailer

Nodemailer [15] je balíček pro Node.js, který umožňuje posílat e-maily z webových aplikací. Odeslání emailu probíhá ve 3 krocích:

1. Vytvoření objektu transportér Nodemailer pomocí SMTP nebo jiného transportního mechanismu.
2. Nastavení možností zpráv (kdo komu co posílá).
3. Doručení objektu zprávy pomocí metody `sendMail()` dříve vytvořeného transportéru.

### 3.15 Nodemon

Nodemon [16] je nástroj, který nám zajistí automatické restartování Node.js aplikace kdykoli v projektu nastane změna. Bez něj bychom museli při každé změně znovu restartovat server.

### 3.16 Postman

Postman [17] je klient pro odesílání a emulaci HTTP požadavků. Nejčastěji se používá k testování webového API.

### 3.17 Tippy.js

Tippy.js [18] je knihovna pro vytváření přizpůsobitelných a interaktivních popisků. Tato knihovna umožňuje snadno přidávat informativní bubliny s obsahem k různým prvkům na stránce.

### 3.18 Google Material Icons

Google Material Icons [19] jsou sada ikon vyvinutých společností Google, které jsou součástí designového systému Google Material Design. Tyto ikony jsou jednoduché, srozumitelné a optimalizované pro použití na webových stránkách nebo v mobilních aplikacích.

### 3.19 Visual Studio Code

Visual Studio Code [20] je lehký, ale výkonný editor zdrojového kódu. Obsahuje vestavěnou podporu pro JavaScript, TypeScript a Node.js a má bohatý ekosystém rozšíření pro další jazyky a běhové prostředí (jako je C++, C#, Java, Python, PHP, Go, .NET).

## 4 Programátorská dokumentace

### 4.1 Struktura projektu

Aplikace byla postavena na frameworku Express, který nechává kostru aplikace kompletně na vývojáři. Složka s aplikací obsahuje následující soubory.

- controllers – Složka obsahující soubory, kde každá reprezentuje kontrolér zodpovědný za specifický aspekt aplikace a obsahuje logiku zpracování HTTP požadavků.
- middleware – Složka obsahuje middlewary, které se vykonávají mezi požadavkem a obsluhou požadavku v kontrolerech.
- models – Složka obsahující datové modely aplikace.
- node\_modules – Tato složka obsahuje závislosti projektu, které jsou instalovány pomocí nástroje npm.
- public – Složka obsahující klientské skripty pro vykreslování map a tvorbu otázek v testech, dále pak obrázky a kaskádové styly.
- routes – Složka, která definuje, jak aplikace zpracovává HTTP požadavky na různé cesty. Každá cesta může být spojena s konkrétní funkcí, která provádí odpovídající operace při příslušné HTTP metodě.
- service – Složka obsahující soubor se službou pro odesílání emailů při obnově hesla.
- views – Složka obsahující šablony pro zobrazení různých stránek.
- .env – Konfigurační soubor obsahující citlivá data aplikace.
- gitignore – Soubor definující soubory a složky, které mají být ignorovány verzovacím systémem Git.
- app.js – Hlavní vstupní soubor, kde je inicializován Express server a kde jsou načteny a použity další části aplikace.
- package.json – Soubor s metadaty a závislostmi projektu.
- README.md – Popis aplikace.

#### 4.1.1 controllers

Složka controllers obsahuje

- authController.js - Obsahuje funkce sloužící k obsluze přihlašování, registrace, odhlašování z aplikace.

- getLogin
  - \* Slouží k renderování stránky pro přihlášení.
- postLogin
  - \* Zpracovává POST požadavek pro přihlášení uživatele.
- getRegister
  - \* Slouží k renderování stránky pro registraci.
- postRegister
  - \* Zpracovává POST požadavek pro registraci uživatele.
- getLogout
  - \* Slouží k odhlášení uživatele.
- passwordController.js – Obsahuje funkce sloužící k obsluze resetování hesla:
  - forgotPassword
    - \* Zpracovává GET požadavek na stránku pro zapomenuté heslo.
    - \* Připravuje prostředí pro zadání e-mailové adresy pro resetování hesla.
  - postForgotPassword
    - \* Zpracovává POST požadavek, který obsahuje e-mailovou adresu uživatele pro zapomenuté heslo.
    - \* Provádí kroky pro inicializaci procesu resetování hesla, např. generování a odeslání resetovacího tokenu na uživatelskou e-mailovou adresu.
  - getResetPassword
    - \* Zpracovává GET požadavek na stránku pro resetování hesla po odeslání resetovacího tokenu.
    - \* Zajišťuje, že uživatel má platný resetovací token pro provedení resetu hesla.
  - postResetPassword
    - \* Zpracovává POST požadavek s novým heslem po resetování.
    - \* Aktualizuje heslo uživatele v databázi na základě platného resetovacího tokenu.
- resultController.js – Obsahuje funkce sloužící k obsluze výsledků.
  - getResult
    - \* Zpracovává GET požadavek na stránku pro zobrazení výsledku.
  - postResult

- \* Zpracovává POST požadavek, který vytváří nový výsledek testu do databáze.
- deleteResult
  - \* Zpracovává DELETE požadavek pro smazání výsledku z databáze.
- updateResult
  - \* Zpracovává PUT požadavek pro aktualizaci výsledku v databázi.
- testController.js – Obsahuje funkce sloužící k obsluze testů.
  - getTests
    - \* Zpracovává GET požadavek na stránku pro zobrazení testů vytvořených právě přihlášeného uživatele.
  - getAllTests
    - \* Zpracovává GET požadavek a vrací seznam všech testů ve formátu JSON.
  - getTestsCount
    - \* Zpracovává GET požadavek, který vrací počet všech vytvořených testů.
  - getNewTest
    - \* Zpracovává GET požadavek, který zobrazí stránku pro vytvoření nového testu.
  - getTestById
    - \* Zpracovává GET požadavek, který má za cíl získání konkrétního testu podle jeho id a zobrazení stránky pro vytvoření nového testu.
  - getTestByIdTest
    - \* Zpracovává GET požadavek, který má za cíl získání konkrétního testu podle jeho id ve formátu JSON.
  - getExam
    - \* Zpracovává GET požadavek, který získá test podle jeho id a zobrazí stránku *exam*.
  - getResults
    - \* Zpracovává GET požadavek, který získá všechny výsledky testu podle jeho id a zobrazí stránku *results*.
  - postTest
    - \* Zpracovává POST požadavek, který do databáze uloží nový test a přeměruje uživatele na stránku *tests*.



- publishTest
  - \* Zpracovává PUT požadavek, který u konkrétního testu provede jeho aktivaci a přesměruje na stránku *tests*.
- updateTestById
  - \* Zpracovává PUT požadavek, který aktualizuje test podle jeho id a zobrazí stránku *tests*.
- deleteTest
  - \* Zpracovává DELETE požadavek, který smaže test podle jeho id a zobrazí stránku *tests*.
- userController.js – Obsahuje funkce k obsluze uživatelů
  - getAllUsers
    - \* Zpracovává GET požadavek a vrací seznam všech uživatelů ve formátu JSON
  - getUsersCount
    - \* Zpracovává GET požadavek, který vrací počet všech vytvořených uživatelů.
  - deleteUser
    - \* Zpracovává DELETE požadavek, který smaže uživatele podle jeho id.
  - getUserByEmail
    - \* Zpracovává GET požadavek, který získá uživatele podle jeho emailové adresy.

#### 4.1.2 middlewares

- CheckUser
  - Middleware pro ověření uživatelského tokenu. Přidá informace o přihlášeném uživateli do objektu locals.
  - Pokud není token platný, přidá hodnotu null do objektu locals.
- requireAuth
  - Middleware pro ověření, zda uživatel je přihlášený. Pokud ne, přesměruje na stránku přihlášení.
- checkAuthor
  - Middleware pro ověření, zda uživatel je autorem daného testu. Pokud není, přesměruje na seznam testů.

- isAuthor
  - Middleware pro ověření, zda uživatel je autorem výsledku. Přidá hodnotu true do locals objektu, pokud je autorem, jinak null.
- isAdmin
  - Middleware pro ověření, zda uživatel má administrátorská práva. Pokud ne, přesměruje na domovskou stránku.
- checkAdmin
  - Middleware pro ověření, zda uživatel má administrátorská práva. Přidá hodnotu true do locals objektu, pokud má práva, jinak null.

### 4.1.3 models

- Result.js
  - Modul Result definuje a manipuluje s modelem Result v MongoDB. Tento model reprezentuje výsledek testu a obsahuje informace o testu, osobě, která test absolvovala.
- Test.js
  - Modul Test slouží k definici a manipulaci s modelem Test v MongoDB.
  - Model představuje test vytvořený učitelem, obsahující název, otázky, mapy, hodnoty pro hodnocení a další atributy.
- User.js
  - Modul User definuje a manipuluje s modelem User v MongoDB.
  - Model reprezentuje registrovaného uživatele.
- Token.js
  - Modul Token slouží k definici a manipulaci s modelem Token v MongoDB.
  - Model představuje token, který se používá při tvorbě nového hesla.

### 4.1.4 routes

- authRoutes.js
  - Obsahuje definice cest a příslušných kontrolerů pro registraci, přihlášení a odhlášení uživatelů.
- passwordRoutes.js

- Zahrnuje definice cest a příslušných kontrolerů pro správu zapomenutého a resetování hesla uživatelů.
- resultRoutes.js
  - Obsahuje definice cest a příslušných kontrolerů pro manipulaci s výsledky testů, včetně vytváření, získávání, mazání a aktualizace.
- testRoutes.js
  - Definuje cesty a kontrolery pro správu testů, včetně jejich získávání, vytváření, publikování, aktualizace a mazání.
- userRoutes.js
  - Zahrnuje definice cest a příslušných kontrolerů pro administrátorskou správu uživatelů, včetně získávání, mazání a hledání podle emailu.

#### 4.1.5 service

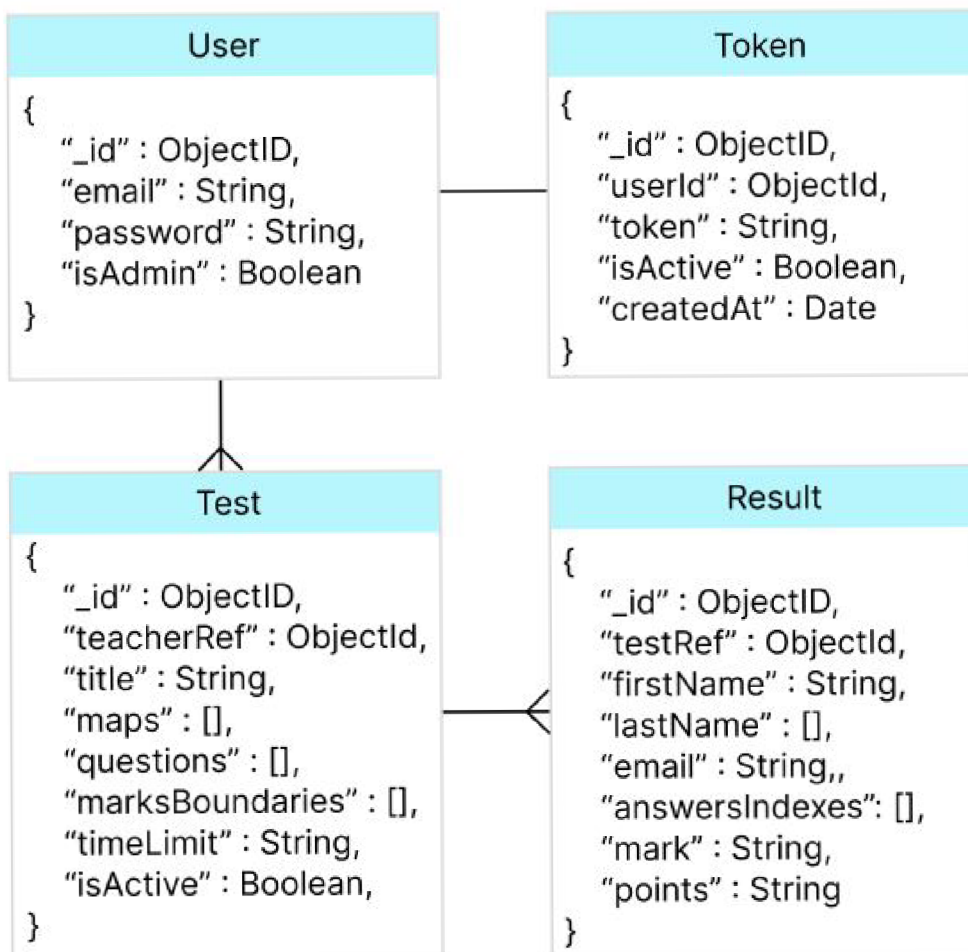
- sendEmail.js
  - Modul definuje funkcionalitu odesílání emailů pro resetování hesla. Tato funkce využívá knihovnu nodemailer a obsahuje logiku pro vytváření a odesílání emailových zpráv s odkazem na resetování hesla.

#### 4.1.6 views

Složka obsahuje šablony v jazyce EJS, které definují strukturu a obsah uživatelského rozhraní aplikace. Každá šablona v této složce reprezentuje určitý pohled nebo stránku aplikace a obsahuje HTML kód s integrovaným JavaScriptem pro dynamické zobrazování dat. Tyto šablony jsou používány při renderování odpovědí serveru a umožňují vytvářet uživatelsky přívětivé a interaktivní stránky, které odpovídají funkcionalitě aplikace. Složka views ještě obsahuje složku *partials*, která obsahuje šablony pro opakovaně používané části HTML kódu. Tyto části lze snadno integrovat do hlavních šablon.

## 4.2 Struktura databáze

V této aplikaci je použita NoSQL databáze MongoDB, která obsahuje níže uvedené kolekce.



Obrázek 1: Náhled databáze

## 5 Uživatelská příručka

Rozmístění ovládacích prvků v aplikaci a některé funkce jsou běžné jako ve většině webových aplikací a měly by být uživatelům dobře známé. Při návrhu aplikace jsem se zaměřil na odlehčený a minimalistický design, abych uživatelům poskytl intuitivní a nenarušené prostředí. To mělo za cíl zvýraznit klíčové funkce, usnadnit navigaci a zlepšit celkový uživatelský zážitek.

### 5.1 Úvodní strana

Na úvodní stránce se nachází stručný popis toho, k čemu aplikace slouží, a výrazné tlačítko *Začít tvořit* pro další pokračování. V pravém horním rohu se nachází tlačítko pro přihlášení a v levém horním tlačítko s názvem aplikace, které slouží vždy pro návrat na hlavní stránku. Po kliknutí na tlačítko *Začít tvořit* nás aplikace přesměruje na přihlašovací okno.

### 5.2 Přihlášení do aplikace

K přihlášení do aplikace slouží tlačítko *Přihlásit se*. Formulář pro přihlášení obsahuje 2 vstupy. Jeden pro zadání emailu a druhý pro zadání hesla. Pokud zadáte email, který ale není zaregistrován, tak vás aplikace upozorní, že email není registrován. Pokud zadáte správný email, ale nesprávné heslo, tak vás aplikace rovněž upozorní, že jste zadal nesprávné heslo. Pokud zadáte správné údaje v obou vstupech, tak po kliknutí na tlačítko *Přihlásit se* vás aplikace přihlásí.

### 5.3 Registrace

Pokud nemáte účet založený, tak pod tlačítkem pro přihlášení je odkaz na stránku pro jeho založení. Po kliknutí vás aplikace přesměruje na stránku, kde si účet můžete vytvořit. Stránka *Registrace* obsahuje 3 vstupy. První pro zadání platného emailu, pro který si chcete účet založit a zbylé 2 pro zadání vašeho přihlašovacího hesla. Pokud nevyplníte platný formát emailu, tak vás aplikace upozorní a stejně tak pokud nevyplníte heslo nebo se hesla nebudou shodovat, tak opět dostanete od aplikace upozornění s chybovou hláškou. Poté, co si myslíte, že máte vyplněné správné údaje, tak stačí kliknout na tlačítko *Registrovat se* a aplikace vám následně vytvoří účet a přesměruje na stránku pro tvorbu testů. Pokud již účet máte, tak je tu opět možnost si přepnout stránku registrace na stránku pro přihlášení a pomocí svých údajů se přihlásit.

### 5.4 Odhlášení z aplikace

Odhlášení z aplikace můžete provést po kliknutí na tlačítko *Odhlásit se*, které bude v pravém horním rohu na stejné pozici, kde je tlačítko pro přihlášení.

## 5.5 Zapomenuté heslo

V případě zapomenutí hesla, aplikace nabízí možnost si vytvořit nové. Pro jeho resetování se musíte dostat na stránku pro přihlášení, kde se nachází možnost *Zapomenuté heslo*. Po kliknutí proběhne přesměrování na stránku *Zapomenuté heslo*, která obsahuje vstup pro zadání emailové adresy, na kterou bude žádost o nové heslo odeslána. Po kliknutí na tlačítko *Resetovat heslo* aplikace ověří, jestli je pod zadaným emailem založen nějaký účet. Pokud není, tak vypíše zprávu, že uživatel s tímto emailem neexistuje. Pokud je zadán správný email, tak se zobrazí zpráva o úspěšném odeslání zprávy na zadanou emailovou adresu pro vytvoření nového hesla. Tento odkaz bude platný po dobu 15 minut.

Obdržení email obsahuje odkaz, který po kliknutí přesměruje uživatele na stránku *Nové heslo*. Tato stránka obsahuje vstup pro zadání nového hesla. Po kliknutí na tlačítko *Změnit heslo* aplikace vypíše, že heslo bylo úspěšně aktualizováno.

## 5.6 Správa testových slepých map

Pokud proběhne přihlášení do uživatelského účtu v pořádku, tak se zobrazí hlavní stránka s tlačítkem *Nový test* a s vytvořenými testy. Pokud uživatel žádné testy vytvořené nemá, tak zde bude jen zpráva, že uživatel nemá žádné vytvořené testy.

### 5.6.1 Testy

Vytvořené testy se zobrazují na hlavní stránce ve formě karet. Test se může nacházet ve 2 stavech:

- *Neveřejný* – V tomto stavu je test po vytvoření uživatelem. Test v tuto chvíli nelze sdílet s uživateli a lze ho ještě upravovat.
- *Aktivní* – Test se stane aktivní až po kliknutí na zelené tlačítko *ZVEŘEJNIT*, které se vyskytuje na každé kartě testu.

Karta obsahuje název testu, datum vytvoření, datum poslední změny. Pokud je již test zveřejněn, tak obsahuje link pro sdílení testu a tlačítko pro jeho zkopírování do schránky. Ve spodní části karty se nachází 3 tlačítka pro úpravu, smazání a nebo zobrazení výsledků testu. Pokud je test zveřejněn, tak zde tlačítko pro úpravu chybí.

## 5.7 Tvorba slepé mapy

Po kliknutí na tlačítko *Nový test* se zobrazí rozhraní sloužící k vytváření nového testu. Na této stránce se nachází tlačítko *Zpět*, které je umístěno v levém horním rohu. V horním středu stránky se nachází navigace, která slouží k pohybu mezi různými fázemi vytváření testu.



Obrázek 2: Náhled stránky s testy (vlevo neaktivní, vpravo aktivní)

### 5.7.1 Nahrání map

K nahrání podkladových map slouží první stránka, která obsahuje tlačítko textitNahrát mapy. Po kliknutí na tohle tlačítko se zobrazí uživatelův souborový systém. Je možné vybrat jednu nebo více podkladových map. Podporované formáty jsou jpg, jpeg, svg, png. Maximální velikost nahrávané mapy je omezena na 15 MB a jejich počet na 10. Po vybrání map se vytvoří jejich zmenšený náhled a jejich název tak, jak je má uživatel pojmenované. Smazání mapy pak proběhne kliknutím na ikonu křížku v pravém horním rohu každého náhledu. Mapa nelze vymazat pokud je vykreslena v nějaké otázce, pro vymazání musíme nejprve otázku buď odstranit nebo v ní změnit mapu.

### 5.7.2 Tvorba otázek

Na stránce *Tvorba otázek* se v horní části nachází lišta, na které je zprvu jen tlačítko se symbolem plus, které slouží pro vytvoření nové otázky. Po kliknutí na tohle tlačítko se v liště vykreslí ikona s číslem otázky a v pravé straně stránky se zobrazí panel který obsahuje další nastavení otázky.

Panel pro nastavení otázky obsahuje v horní části číslo otázky následovanou ikonou koše, která slouží pro smazání otázky. Kliknutím na tuto ikonu pak dojde k vymazání otázky a zobrazí se předchozí otázka. Pokud smazaná otázka byla rovněž poslední otázkou, tak dojde ke skrytí panelu pro nastavení otázek. Panel dále obsahuje tlačítko pro výběr mapy, které obsahuje názvy nahraných map. Po výběru mapy dojde k jejímu vykreslení. Ve vykreslené mapě se v levém horním rohu nachází navigační tlačítko se čtyřmi šipkami, které slouží k posouvání náhledu mapy pokud je náhled přiblížen respektive oddálen. V pravém rohu vy-

kreslené mapy se nachází 4 tlačítka. Tlačítka s ikonou lupy slouží k přibližování a oddalování náhledu mapy (náhled mapy lze přibližovat a oddalovat také rolovacím kolečkem myši, pokud se kurzor nachází někde nad náhledem mapy), tlačítka s ikonou koše slouží k vymazání obsahu mapy a vedle pak ještě tlačítka které vrátí mapu do své původní pozice. Pro kreslení nad mapou jsou na výběr tvary bod, linie a polygon. Po kliknutí na jeden z tvarů se aktivuje režim kreslení.

- Vykreslení bodu – Pohybem myši nad místo, kde chci bod umístit a kliknutím levého tlačítka myši bod umístit.
- Vykreslení linie – Pohybem myši nad místo, kde chci začít tvořit linii a levým kliknutím myši umístím první bod. Pohybem pak myši najedu nad místo, kde bude linie pokračovat a dalším kliknutím levého tlačítka umístím další bod. Dvojklikem levého tlačítka myši je pak tvorba linie ukončena.
- Vykreslení polygonu – Pohybem myši nad místo, kde chci začít tvořit polygon a levým kliknutím umístím první bod. Pohybem myši pak najedu na místo, kde chci umístit další bod a znovu levým kliknutím umístím další bod. Pokud chci tvorbu polygonu ukončit, tak stačí umístit poslední bod do blízkosti prvního bodu a kliknutím levého tlačítka myši se polygon automaticky uzavře. Druhá varianta uzavření polygonu je je pak opět jako u linie levým dvojklikem myši kdekoli nad mapou.

Po kliknutí na vytvořený tvar se zobrazí obrys, se kterým lze manipulovat a nad pravým horním rohem obrýsu je ikona pro odstranění aktuálně označeného tvaru. V sekci *PŘIDAT TVAR* se pak dále nachází 1 tlačítka

- *Magická výplň* – Při kliknutí se aktivuje (pod myši se zobrazí ikona kbelíku) a při dalším kliknutí do nějaké oblasti jej vyplní barvou. Slouží pro rychlé vyplňování polygonů, které mají jasnou hranici a jednobarevné pozadí (například pro mapy států).

Sekce *ODPOVĚDI* slouží k vytvoření odpovědí, ze kterých pak bude uživatel v testu vybírat. Po najetí myši na jednotlivé ikony se zobrazí nápověda k čemu slouží.

- *Přidat odpověď* – Vytvoří prázdnou odpověď.
- *Zamíchat odpovědi* – Prohodí pořadí u již vytvořených odpovědí.
- *Přidat náhodnou odpověď* – Přidá jednu náhodně vygenerovanou odpověď z předchozích vytvořených otázek.

Po vytvoření odpovědí je potřeba zvolit, která z nich bude správná. Toto je povinný krok, bez kterého nelze pokračovat ve vytváření dalších otázek. Jako poslední je zde sekce *BODY*, ve které se nastavuje počet bodů za otázku.



### 5.7.3 Dokončení

Poslední stránka tvorby testu je *Dokončení* a obsahuje 3 sekce:

- *NÁZEV TESTU* – Povinné pole, bez vyplnění se test neuloží.
- *NASTAVENÍ ZNÁMKOVÁNÍ* – Body se nesmí překrývat. A vždy se nastavuje první ve tvaru *od - do*. Sekce obsahuje tlačítko s náповědou pro správné vyplnění vstupů. Pod tabulkou je i vypočtená hodnota celkového počtu bodů ze všech otázek.
- *ČASOVÝ LIMIT* – Výběr z předem nastavených hodnot. Po uplynutí času se test automaticky ukončí a vyhodnotí.

### 5.7.4 Uložení testu

Kliknutí na tlačítko *ULOŽIT TEST* se test uloží a proběhne přesměrování na hlavní stránku se všemi testy.

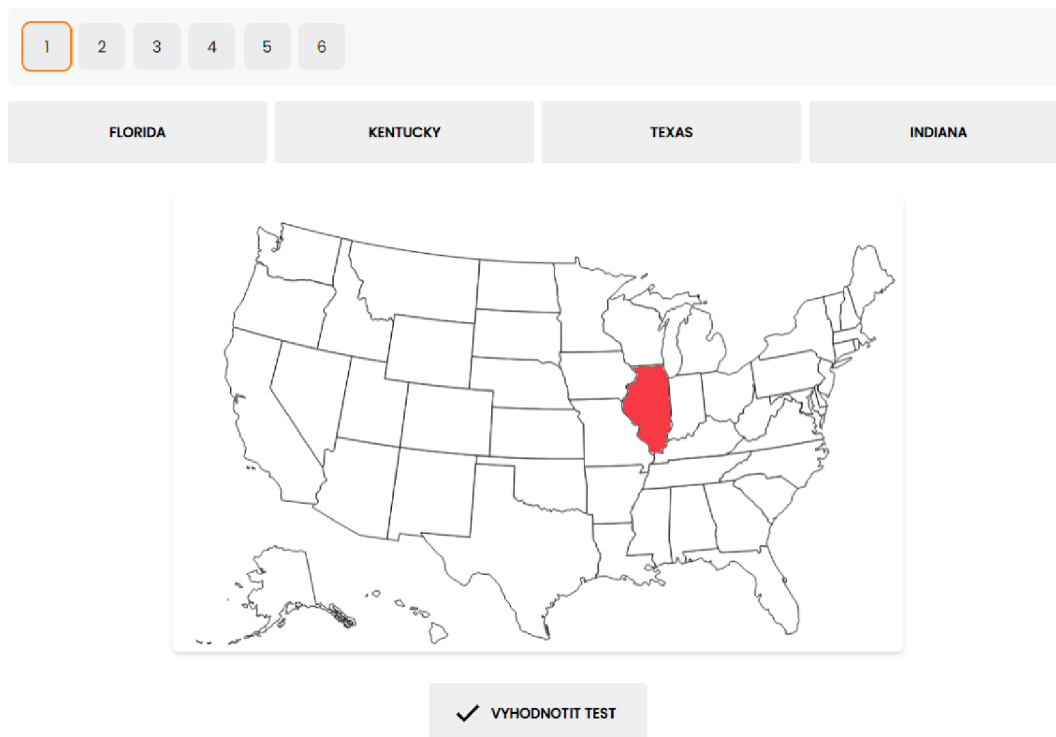
### 5.7.5 Absolvování testu

Po otevření odkazu se uživateli zobrazí formulář s názvem testu a časovým limitem, který bude potřeba ke splnění. Dále pak 3 vstupy:

- Jméno
- Příjmení
- Emailová adresa

Všechny vstupy jsou povinné a bez jejich vyplnění nelze spustit test. Po kliknutí na tlačítko *SPUSTIT TEST* se zobrazí náhled testu které obsahuje:

- Název testu – Obsahuje název aktuálního testu.
- Zbývající čas – Nachází se v pravém horním rohu. Po jeho uplynutí se test ukončí a vyhodnotí.
- Lišta otázek – Oranžový rámeček okolo ikony čísla otázky znamená aktuální otázku.
- Možnosti – Každá otázka obsahuje 2 a více odpovědí. Po kliknutí na jednu z možností se změní pozadí na modrou a otázka se tím považuje za zodpovězenou. Odpověď na otázku lze měnit a mezi otázkami lze náhodně přeskakovat.
- Mapa – Slepá mapa, která obsahuje červeně vyznačený hledaný pojem. Může se jednat o bod, linii, polygon, nebo i skupina všech zmíněných.
- Tlačítko *VYHODNOTIT TEST* – Dojde k okamžitému ukončení testu a jeho vyhodnocení.



Obrázek 3: Průběh testu

### 5.7.6 Zobrazení výsledku

Po kliknutí na tlačítko *VYHODNOTIT TEST* se zobrazí náhled výsledku s dosaženou známkou a počtem bodů a tlačítkem *ZOBRAZIT VÝSLEDEK* pro zobrazení detailu výsledku.

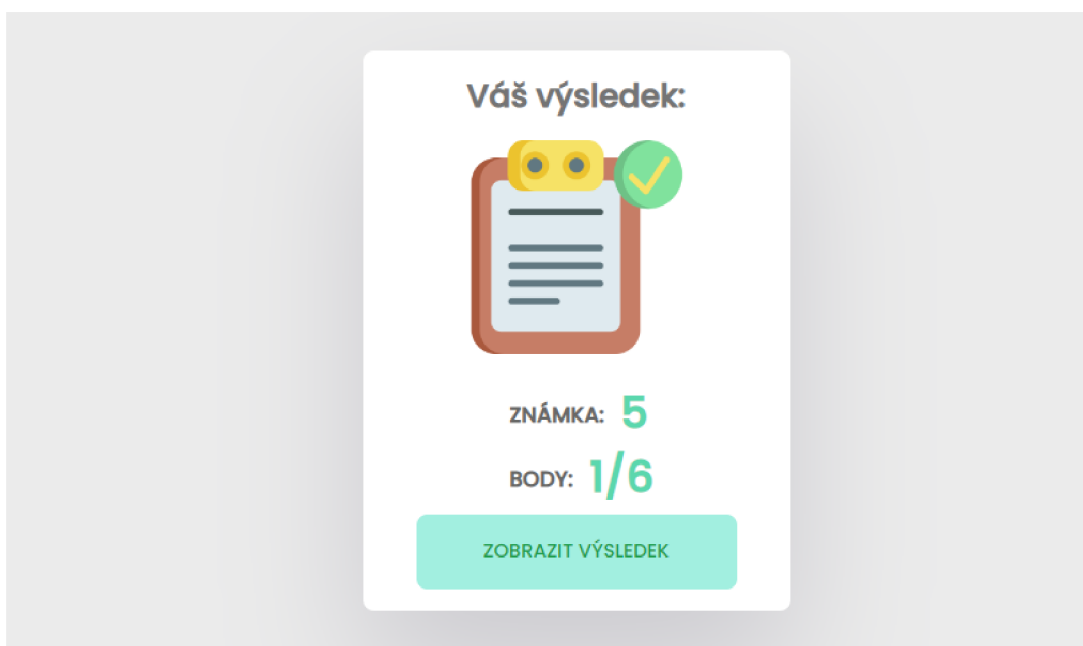
### 5.7.7 Detail výsledku

Stránka s detailem výsledku obsahuje 2 navigační tlačítka *VÝSLEDEK* a *ODPOVĚDI*.

- Karta *VÝSLEDEK* – Zobrazí celkový počet získaných bodů a známku (viz obr. číslo 5).
- Karta *ODPOVĚDI* – Zobrazí detail odpovědí na jednotlivé otázky (viz obr. číslo 6).

## 5.8 Tabulka výsledků

K tabulce výsledků má přístup pouze autor testu a je k ní přístup z hlavní stránky přes ikonu *VÝSLEDKY*, která se nachází na kartě testu. Po kliknutí se zobrazí



Obrázek 4: Náhled výsledku

tabulka s výsledky, která obsahuje jméno a příjmení studenta, jeho známku, dosažené body a možnost výsledek smazat nebo zobrazit podrobnosti přes tlačítko *DETAIL*. Toto tlačítko nás přesměruje na detail výsledku (viz předchozí podkapitola). Pokud je přihlášen autor testu, tak má na stránce s detailem výsledku ještě možnost výsledek upravit přes tlačítko *UPRAVIT VÝSLEDEK* (viz obr. 7).

## 5.9 Administrátorská sekce

Pokud je přihlášen uživatel s administrátorským oprávněním, tak jeho hlavní stránka má ještě možnost přejít do administrátorské sekce pomocí tlačítka *ADMINISTRACE* (viz obr. 8). V administrátorské sekci se nachází postranní vertikální panel s nabídkou odkazů.

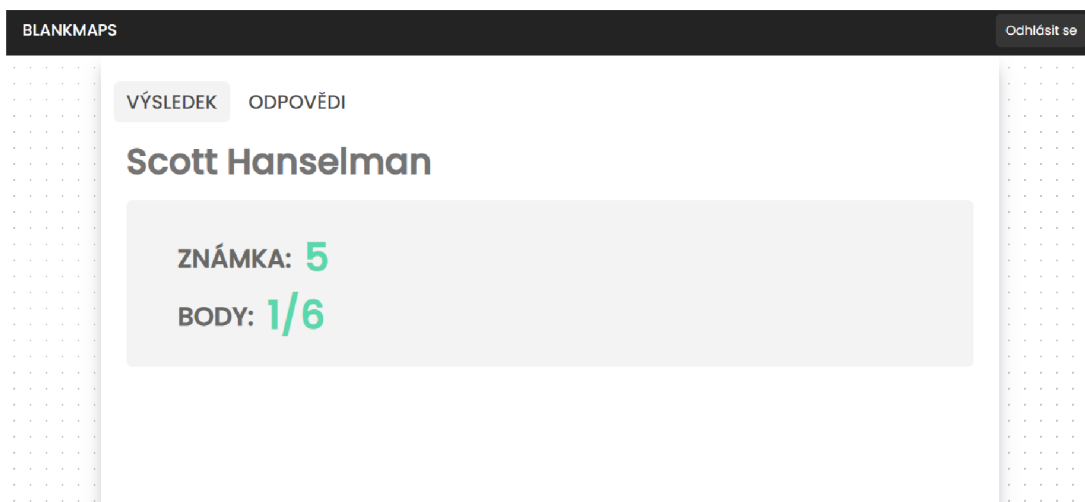
### 5.9.1 Přehled

Odkaz přehled zobrazuje statistiky aplikace.

- Celkový počet registrovaných uživatelů
- Celkový počet vytvořených testů

### 5.9.2 Uživatelé

Odkaz *Uživatelé* zobrazuje prvních 10 uživatelů ve formě tabulky, která obsahuje:



Obrázek 5: Detail výsledku - výsledek

- Emailovou adresu uživatele
- Uživatelskou roli
- Tlačítko *SMAZAT*

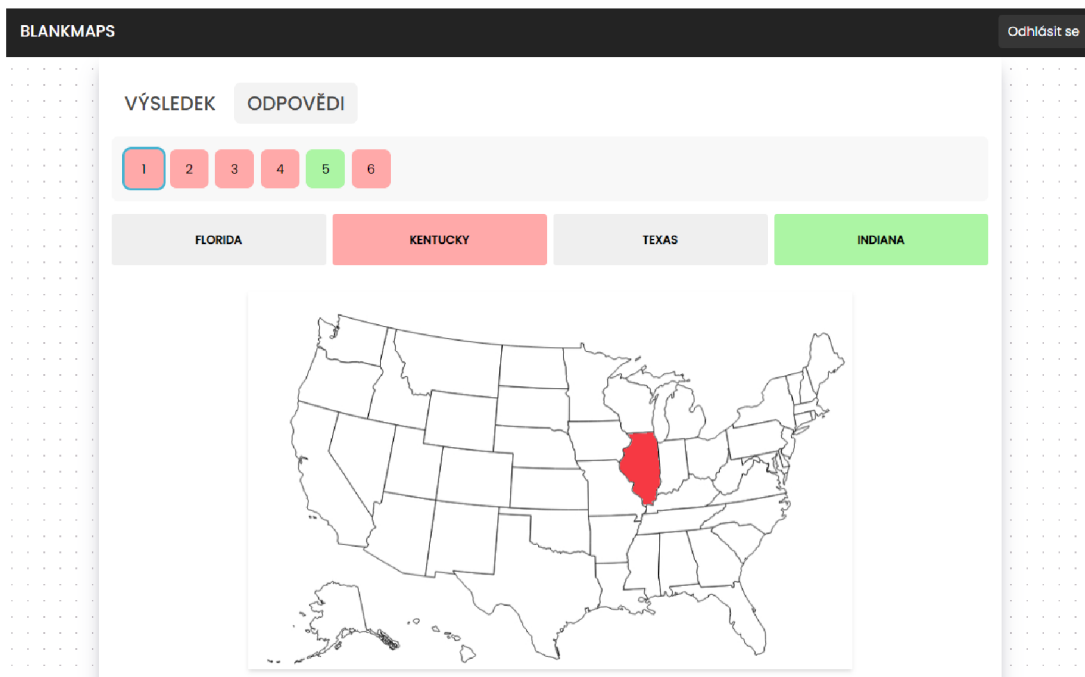
Vyhledání v tabulce funguje pomocí vyhledávacího pole nad tabulkou. Do vstupu je potřeba napsat hledaný email a potvrdit kliknutím na tlačítko lupy. Pokud byl uživatel nalezen, tak se zobrazí hledaný uživatel. Pokud by nebyl nalezen, tak se vypíše textová hláška, že uživatel nebyl nalezen. Pro opětovné zobrazení celé tabulky stačí do pole *Email uživatele* nezadávat žádný text a potvrdit tlačítkem lupy. Pro načtení dalších 10 uživatelů stačí kliknout na tlačítko *NACÍST DALŠÍ*, které zobrazí dalších 10 uživatelů. A to pouze v případě, že lze ještě nějaké zobrazit.

### 5.9.3 Testy

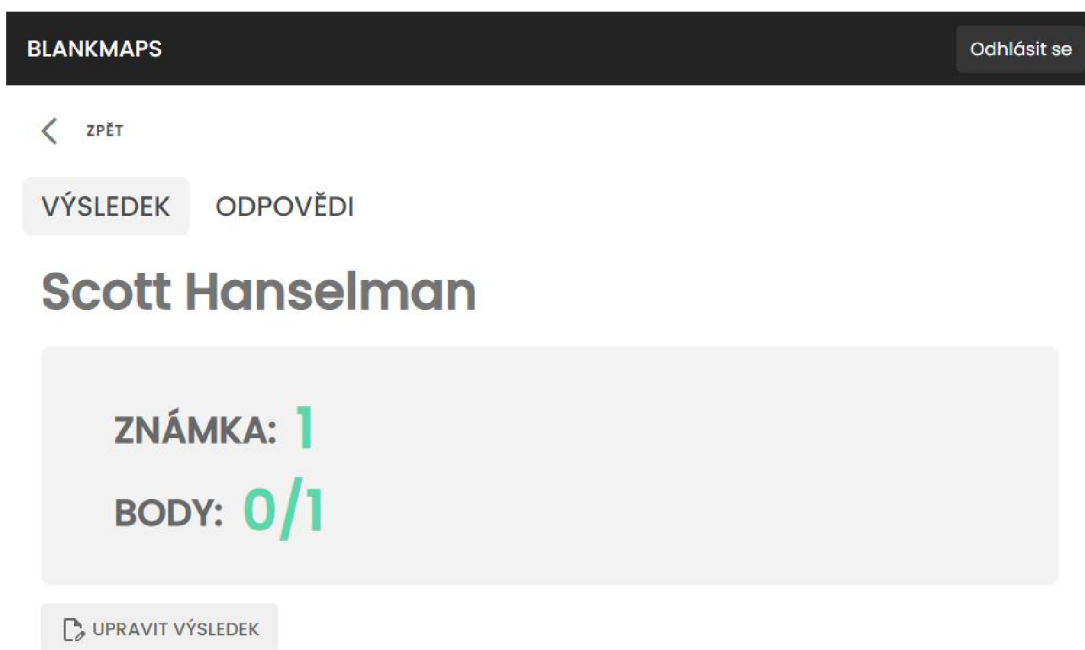
Odkaz *Testy* zobrazuje prvních 10 testů ve formě tabulky, která obsahuje:

- Autora testu
- Název testu
- Tlačítko *SMAZAT*

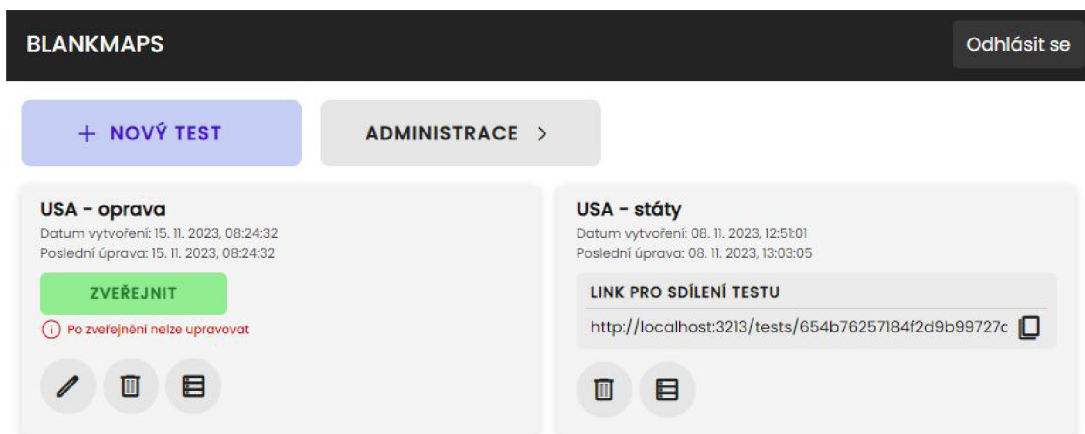
Pro načtení dalších 10 testů stačí kliknout na tlačítko *NACÍST DALŠÍ*, které zobrazí následujících 10 testů. A to pouze v případě, že lze ještě nějaké zobrazit.



Obrázek 6: Detail výsledku - odpovědi



Obrázek 7: Detail výsledku - z pohledu autora testu



Obrázek 8: Hlavní stránka - z pohledu administrátora

## Závěr

Cílem mé bakalářské práce bylo navrhnout a implementovat nástroj pro správu slepých map pro výuku zeměpisu a případně i dalších předmětů. Aplikace měla umožňovat snadno vkládat pozadí mapy a definovat oblasti odpovídající hledaným objektům. Toho bylo docíleno s pomocí elementu canvas dostupné v HTML5 a javascriptové knihovny Fabric.js, která mi hodně zjednodušila práci s vykreslováním objektů nad kreslícím plátnem.

Aby vytvořené mapy mohl uživatel spravovat, tak aplikace umožňuje vytvořit si uživatelský účet. Registrovaný uživatel má pak možnost mapy tvořit, upravovat, sdílet, pak také prohlížet a upravovat výsledky. V aplikaci existuje ještě uživatel s administrátorským oprávněním, který si může zobrazit statistiky aplikace, spravovat uživatele a mazat jejich testy.

Jako možné rozšíření aplikace by se nabízelo umožnit uživatelům zveřejňovat své testy ve formě šablon pro ostatní uživatele. Tím by se vytvořilo komunitní prostředí, kde uživatelé mohou sdílet své slepé mapy a přispívat k rozmanitosti obsahu aplikace. Dalším rozšířením by mohlo být přidání již prázdných slepých map podle kategorií a předmětů.

## Conclusions

The goal of my bachelor's thesis was to design and implement a tool for managing blind maps for teaching geography and possibly other subjects. The application should allow easy insertion of the map as the background and definition of areas corresponding to the searched objects. This was achieved with the help of the canvas element available in HTML5 and the Fabric.js javascript library, which greatly simplified my work with drawing objects over the drawing canvas.

In order for the user to manage the created maps, the application allows you to create a user account. The registered user then has the option to create, edit, share maps, then also view and edit the results. There is also a user with administrator rights in the application who can view application statistics, manage users and delete their tests.

As a possible extension of the application, users could be allowed to publish their tests in the form of templates for other users. This would create a community environment where users can share their blank maps and contribute to the diversity of the application's content. Another extension could involve adding pre-existing blank maps categorized by subjects.



## A Obsah elektronických dat

### **text/**

Adresář s textem práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech (textových) příloh, a všechny soubory potřebné pro bezproblémové vytvoření PDF dokumentu textu (případně v ZIP archivu), tj. zdrojový text textu a příloh, vložené obrázky, apod.

### **src/**

Adresáře a soubory s veškerými ostatními autorskými daty práce (případně v ZIP archivu) – typicky spustitelné a další soubory software vytvořeného v rámci práce potřebné pro bezproblémový provoz software, případně jeho instalační program, a kompletní zdrojové texty software a další data nutná pro plně reprodukovatelné korektní vytvoření spustitelných souborů.

### **data/**

Adresář obsahující instalační soubor Node.js a dvě slepé mapy které lze použít při vytváření testu.

### **README.txt**

Instrukce pro spuštění aplikace.

## Literatura

- [1] Synodus. *What is Web Application? How it works? A Detailed Guide* [online]. [cit. 2023-11-21]. Dostupný z: <https://synodus.com/blog/web-development/web-application/>.
- [2] HTML. *HTML* [online]. [cit. 2023-11-21]. Dostupný z: <https://html.com>.
- [3] Mozilla. *CSS: Cascading Style Sheets / MDN* [online]. [cit. 2023-11-21]. Dostupný z: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
- [4] Javatpoint. *Learn JavaScript Tutorial - javatpoint* [online]. [cit. 2023-11-21]. Dostupný z: <https://www.javatpoint.com/javascript-tutorial>.
- [5] FreeCodeCamp. *What Exactly is Node.js? Explained for Beginners* [online]. [cit. 2023-11-21]. Dostupný z: <https://www.freecodecamp.org/news/what-is-node-js/>.
- [6] NPM. *npm* [online]. [cit. 2023-11-21]. Dostupný z: <https://docs.npmjs.com/about-npm>.
- [7] Mozilla. *Express* [online]. [cit. 2023-11-21]. Dostupný z: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction).
- [8] EJS. *EJS* [online]. [cit. 2023-11-21]. Dostupný z: <https://ejs.co>.
- [9] TechTarget. *What is MongoDB? Features and how it works – TechTarget Definition* [online]. [cit. 2023-11-21]. Dostupný z: <https://www.techtarget.com/searchdatamanagement/definition/MongoDB>.
- [10] Mongoose. *Mongoose ODM v8.0.0* [online]. [cit. 2023-11-21]. Dostupný z: <https://mongoosejs.com>.
- [11] Fabric.js. *Fabric.js Javascript Canvas Library* [online]. [cit. 2023-11-21]. Dostupný z: <http://fabricjs.com>.
- [12] NPM. *Bcrypt* [online]. [cit. 2023-11-21]. Dostupný z: <https://www.npmjs.com/package/bcrypt>.
- [13] NPM. *Dotenv* [online]. [cit. 2023-11-21]. Dostupný z: <https://www.npmjs.com/package/dotenv>.
- [14] JWT. *JWT* [online]. [cit. 2023-11-21]. Dostupný z: <https://jwt.io>.
- [15] Nodemailer. *Nodemailer* [online]. [cit. 2023-11-21]. Dostupný z: <https://nodemailer.com>.
- [16] Nodemon. *Nodemon* [online]. [cit. 2023-11-21]. Dostupný z: <https://nodemon.io>.
- [17] Postman. *Postman* [online]. [cit. 2023-11-21]. Dostupný z: <https://www.postman.com>.

- [18] Tippy.js. *Tippy.js - Tooltip, Popover, Dropdown, and Menu Library* [online]. [cit. 2023-11-21]. Dostupný z: <https://atomiks.github.io/tippyjs/>.
- [19] Google. *Google icons* [online]. [cit. 2023-11-21]. Dostupný z: <https://fonts.google.com/icons>.
- [20] Microsoft. *Visual Studio Code - Code Editing. Redefined* [online]. [cit. 2023-11-21]. Dostupný z: <https://code.visualstudio.com>.