HOTEL ANIPIA

Webová aplikace slouží pro uživatele (zákazníky), které chtějí vytvořit rezervaci pokoje pro svého domácího mazlíčka. Zatím tato aplikace umožňuje vyplnit kontaktní formulář na stránce, který se následně:

- uloží do databáze (H2)
- odešle jako e-mail prostřednictvím Mailtrap

Taky stránka Login umožňuje registraci nového uživatele a následovně jeho přihlášení do svého profilu, kde pak se může odhlásit.

Celkově projekt je postaven pomocí:

Frontend

- HTML
- CSS
- JavaScript
- Skript pro AJAX odesílání dat

Backend

- Java 17
- Spring Boot
- Spring Web, Spring Data JPA, Spring Mail
- H2 databáze
- Mailtrap (pro simulaci e-mailového serveru)
- Maven (pro správu závislostí)

Databáze H2

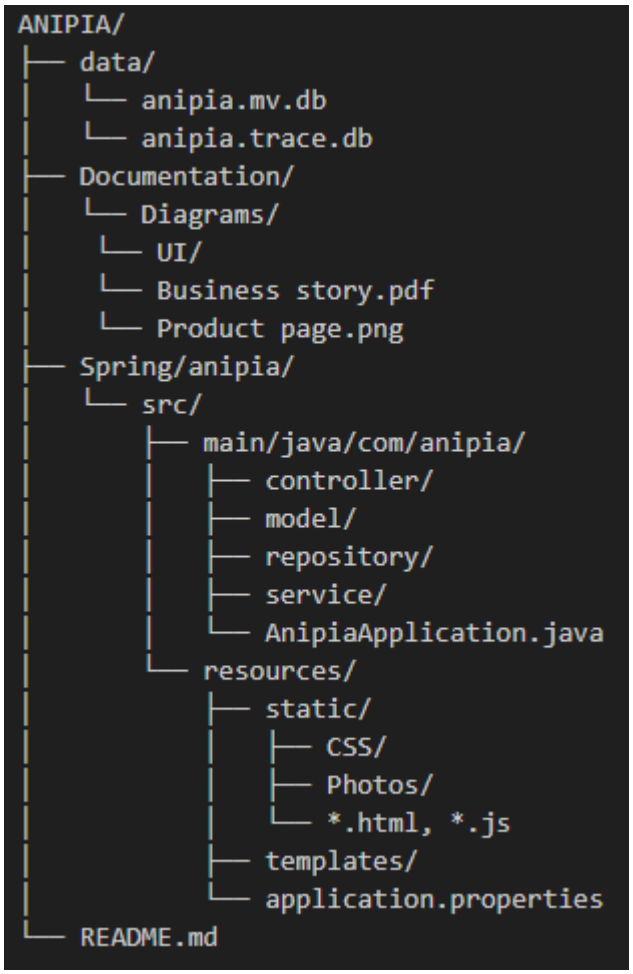
Byla zvolena především kvůli její jednoduchosti, rychlosti a snadné integraci s frameworkem Spring Boot. H2 je ideální pro testování, protože umožňuje běžet jako in-memory databáze a její velkou výhodou je také webová konzole, která usnadňuje správu a testování SQL dotazů. Všechny zprávy se automaticky ukládájí do data/anipia.mv.db. Aby se dostat do konzole H2, musí se v prohlížeči zadat http://localhost:8080/h2-console. Odeslání e-mailu běží asynchronně, aby neblokovalo uživatele.

Mailtrap

Pro testování odesílání e-mailů v aplikaci bylo zvoleno použit Mailtrap, protože umožňuje bezpečně simulovat a sledovat e-mailovou komunikaci. Taky poskytuje přehledné uživatelské rozhraní, ve kterém lze kontrolovat obsah i formát zpráv. Navíc se snadno

integruje se Spring Boot (což je ideální pro tenhle projekt) a dalšími vývojářskými nástroji pomocí jednoduchého SMTP nastavení.

Struktura projektu



Odeslání formuláře

Pro tenhle proces byl pomocí Visual Paradigm vytvořen "Business Process Model and Notation" a "Use Case Diagram", které ho snadně vizuálně popisují. Pro spuštění a nastavení projektu jsou násldedující kroky:

1. Uživatelský vstup (Frontend)

- Uživatel otevře webovou stránku (např. index.html), kde najde kontaktní formulář.
- Vyplní pole: jméno, e-mail a zpráva.
- Po stisknutí tlačítka Submit JS (script.js) zachytí událost odeslání formuláře.
- Místo klasického odeslání stránky se spustí AJAX volání pomocí fetch(), které vytvoří HTTP POST požadavek na backend API (http://localhost:8080/contact/submit).
- Jestli uživatel něco nezadá, tak mu pošle oznámení, že mu něco chybí.
- Data jsou odeslána ve formátu JSON (name, email, message).

2. Přijetí požadavku a zpracování (Backend - Controller)

- Backendová metoda v ContactFormController přijme JSON tělo požadavku (@RequestBody ContactForm contactForm).
- Spring automaticky namapuje JSON na objekt ContactForm.
- Zavolá se služba contactFormService.saveContactForm(contactForm).

3. Uložení do databáze (Backend - Service a Repository)

- V ContactFormService se pomocí contactFormRepository.save(contactForm) uloží nová zpráva do databáze H2.
- Databáze automaticky přiřadí unikátní ID každé zprávě.

4. Odeslání e-mailu (Backend - Service)

- Po úspěšném uložení služby pokračuje odesláním e-mailu.
- Metoda sendEmail(ContactForm) vytvoří e-mailovou zprávu pomocí SimpleMailMessage.
- E-mail je odeslán na e-mail adresu hotelu přes SMTP server Mailtrap.

5. Odpověď backendu (Controller na Frontend)

- Pokud vše proběhne v pořádku, backend vrátí „Form submitted successfully“.
- Pokud nastane nějaká chyba, vrátí so terminálu stav 500 INTERNAL SERVER ERROR s odpovídající chybovou zprávou.

6. Zobrazení výsledku uživateli (Frontend)

- JS přijme odpověď z backendu.
- V případě úspěchu zobrazí uživateli zprávu „Message sent successfully!“.
- V případě chyby zobrazí „Error sending message!“.

Registrace uživatele

1. Uživatelský vstup (Frontend)

- Uživatel navštíví stránku, kde je umístěn registrační formulář.
- Vyplní pole: jméno, příjmení, telefon, email, heslo a potvrzení hesla.
- Po kliknutí na tlačítko „Sign Up“ JS zachytí událost submit z formuláře.
- Skript provede základní validaci (např. délka hesla, shoda hesel). Pokud něco chybí nebo je špatně, uživatel dostane upozornění.
- Po úspěšné validaci se pomocí fetch() odešle HTTP POST požadavek na endpoint.

2. Přijetí požadavku a zpracování (Backend – Controller)

- Backend zachytí požadavek v metodě signup() ve třídě ZakaznikController.
- Pomocí @RequestBody se JSON automaticky převede na RegisterRequest.
- Z něj se vytvoří nový objekt Zakaznik, který se předá do ZakaznikService.

3. Uložení uživatele (Backend – Service a Repository)

- Provede se validace, zda e-mail už neexistuje.
- Heslo se ukládá v hashované podobě pomocí BCrypt, čímž je zajištěna bezpečnost údajů i při případném úniku databáze.
- Nový zákazník se uloží do databáze.

4. Odpověď backendu (Controller na Frontend)

- Pokud je registrace úspěšná, tak se zobrazí zpráva, že všechno proběhlo úspěšně.
- Pokud e-mail už existuje nebo nastane jiná chyba, vrátí se chybá zpráva.

5. Zobrazení výsledku uživateli (Frontend)

- Frontend přijme odpověď z backendu.
- Pokud je registrace úspěšná tak zobrazí hlášku „Your registration is done. Please login“ a pak přesměruje uživatele na přihlašovácí stránku.
- Pokud se registrace nezdaří tak se zobrazí chybová hláška (buď „User already exists“ nebo „Registration failed“).

API endpointy

1. POST /contact/submit

Umožňuje odeslat kontaktní formulář (z webu nebo přes klienta).

2. GET /contact/ping

Kontrola správnosti serveru, zda běží.

3. GET /contact/by-email

Vrátí všechny zprávy zadaného uživatele podle jeho e-mailu.

4. DELETE /contact/deleteByEmail

Smaže všechny zprávy z databáze podle e-mailu.

5. GET /export/pdf

Export všech zpráv do PDF souboru.

6. POST /api/zakaznici/signup

Slouží k vytvoření nového uživatele (zákazníka) v databázi.

7. POST /api/zakaznici/login

Slouží k ověření uživatelských údajů při přihlašování. Jestli uživatel existuje, tak ho přihlásí do profilu.

Testování API (přes curl - cmd)

Příkaz pro příjmutí dat formuláře a jejích zpracování:

curl -X POST http://localhost:8080/contact/submit ^ -H "Content-Type: application/json" ^ -d "{"name": "Uzivatel", "email": "uzivatel@test", "message": "Testovaci zprava z formulare."}"

Příkaz pro testování, že backend běží správně:

curl "http://localhost:8080/contact/ping"

Příkaz pro výpis všech zpráv od určítého uživatele podle e-mailu:

curl "http://localhost:8080/contact/by-email?email=uzivatel@test"

Příkaz pro odstranění všech zpržv z databáze podle e-mailu:

curl -X DELETE "http://localhost:8080/contact/deleteByEmail/uzivatel@test"

Příkaz pro export všech zpráv z databáze do formatu PDF:

curl -o message_forms.pdf http://localhost:8080/export/pdf

Nebo v prohlížeči: http://localhost:8080/export/pdf

Příkaz pro registraci uživatele:

curl -X POST http://localhost:8080/api/zakaznici/signup -H "Content-Type: application/json" -d "{\"jmeno\":\"uzivatel\",\"prijmeni\":\"test\",\"telefon\":\"123456789\",\"email\":\"uzivatel@test\",\"heslo\":\"qwerty12\"}"

Příkaz pro přihlášení uživatele:

curl -X POST http://localhost:8080/api/zakaznici/login ^

-H "Content-Type: application/json" ^

-d "{\"email\":\"uzivatel@test\",\"heslo\":\"qwerty12\"}"