Sadržaj dokumentacije projekta

## Naslovna strana

* Fakultet informacionih tehnologija, Softversko inženjerstvo i IT354.
* Luka Stojić 5355.
* AI Recipes – Web aplikacija za upravljanje receptima
* 21. Maj 2025.

## Sadržaj

Contents

[Naslovna strana 1](#_Toc198813249)

[Sadržaj 2](#_Toc198813250)

[1. Uvod 3](#_Toc198813251)

[2. Planiranje i dizajn sistema 4](#_Toc198813252)

[2.1. Definicija zahteva 4](#_Toc198813253)

[ Funkcionalni zahtevi: 4](#_Toc198813254)

[2.2. Use-case dijagrami 4](#_Toc198813255)

[3. Razvoj korisničkog interfejsa (UI) 5](#_Toc198813256)

[3.1. Opis tehnologija za stilizaciju 5](#_Toc198813257)

[3.2. Pregled ključnih stranica 5](#_Toc198813258)

[4. Implementacija funkcionalnosti 7](#_Toc198813259)

[4.1. Korisničke funkcionalnosti 7](#_Toc198813260)

[4.2. Administratorske funkcionalnosti 7](#_Toc198813261)

[5. Simulacija backend-a 8](#_Toc198813262)

[5.1. Korišćenje json-servera (ako je korišćen) 8](#_Toc198813263)

[5.2. Povezivanje frontenda i backend-a 8](#_Toc198813264)

[5.3. Alternativne tehnologije 11](#_Toc198813265)

[6. Zaključak 11](#_Toc198813266)

[7. Prilozi 11](#_Toc198813267)

# Uvod

* Opis teme projekta.

Projekat AI Recipes predstavlja web aplikaciju razvijenu korišćenjem React frameworka. Aplikacija omogućava korisnicima da pregledaju, dok administratori imaju dodatne mogućnosti kreiranja, ažuriranja i brisanja recepata. Pored velikog broja dostupnih recepata, implementiran je i AI asistent koji je zadužen za pomaganje korisnicima oko bilo kakvih problema i nedoumica.

* Cilj projekta i funkcionalnosti koje će biti implementirane.

Cilj projekta je omogućiti korisnicima da pretraže, odaberu, pregledaju bilo koji recept u cilju lakšeg kuvanja, pružajući im sve savremene alate. Funkcionalnosti uključuju prijavu/registraciju, zaštićene rute, pregled liste recepata, prikaz detaljnih informacija kroz modal prozore, kao i administraciju sadržaja.

* Kratak pregled tehnologija i alata koji će biti korišćeni.

Tehnologije i alati koji su korišćeni uključuju ReactJS, react-router-dom za navigaciju, Tailwind CSS za stilizaciju, Vite kao alat za bundlovanje i razvoj, kao i dodatne biblioteke poput react-markdown i rehype-highlight za ulepšan prikaz odgovora od strane AI asistentom, i framer-motion kako bi tranzicija izmedju stranica bila lepša.

# Planiranje i dizajn sistema

## Definicija zahteva

## Funkcionalni zahtevi:

* + - * Korisnici mogu da se registruju i prijave u sistem, čime im se omogućava pristup zaštićenim rutama unutar aplikacije.
      * Detaljan prikaz pojedinačnih recepata se obavlja putem modal prozora.
      * Razgovor sa AI asistentom.
      * Sistem omogućava Administratorima pregled, kreiranje, ažuriranje i brisanje recepata.
    - Nefunkcionalni zahtevi:
      * Aplikacija mora biti responzivna i optimizovana za rad na različitim uređajima.
      * Implementacija mora osigurati kratko vreme odziva i visoku pouzdanost prilikom obrade korisničkih zahteva.
      * Interfejs treba biti vizuelno konzistentan i jednostavan za korišćenje.

## Use-case dijagrami

* + - Prikaz aktera i slučajeva korišćenja.
    - Kratak opis dijagrama.

# Razvoj korisničkog interfejsa (UI)

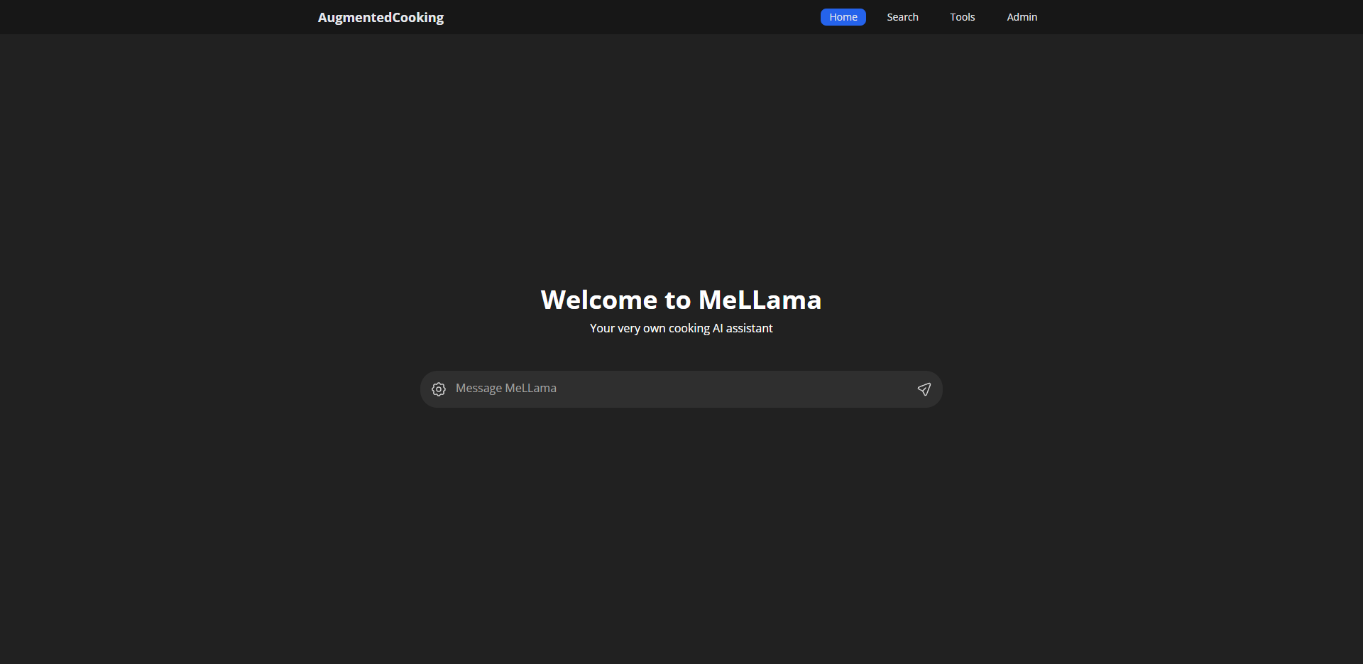
## Opis tehnologija za stilizaciju

* + - Za stilizaciju interfejsa korišćena je biblioteka Tailwind CSS zbog njene široke primene, modularnosti, pouzdanosti, kao i jednostavnosti i brzine korišćenja

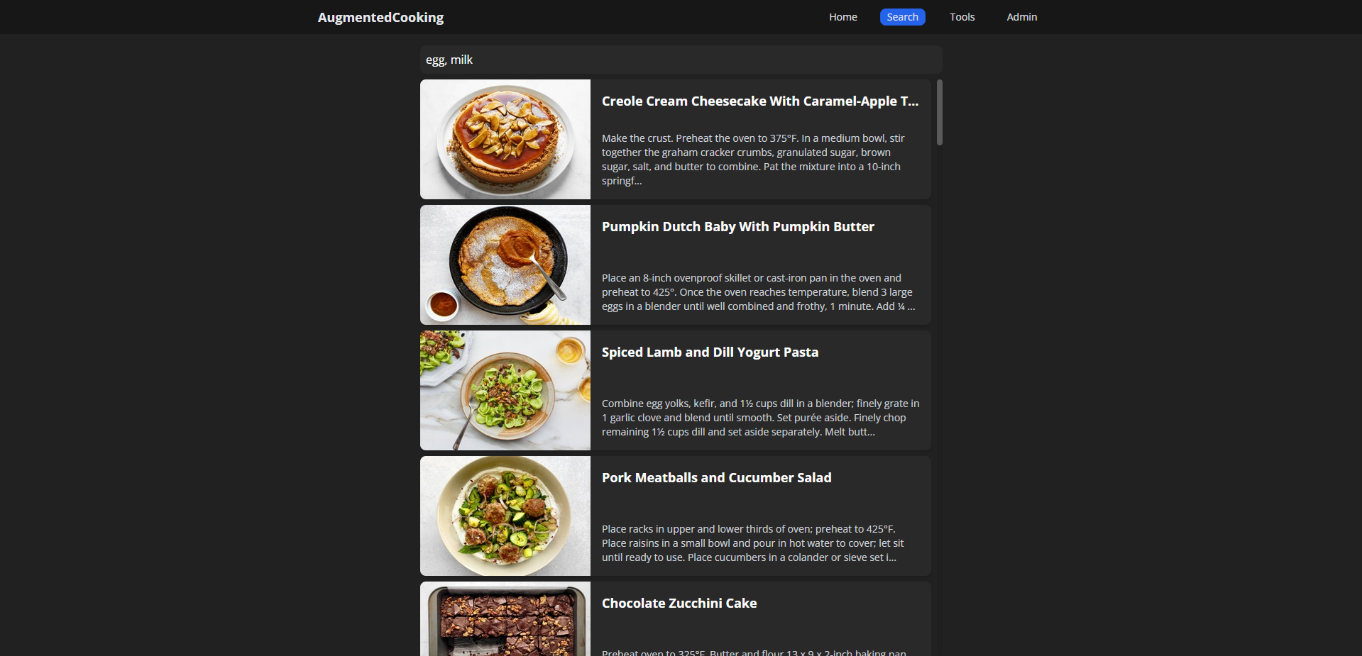
## Pregled ključnih stranica

* + - Početna stranica: Opis i screenshot.

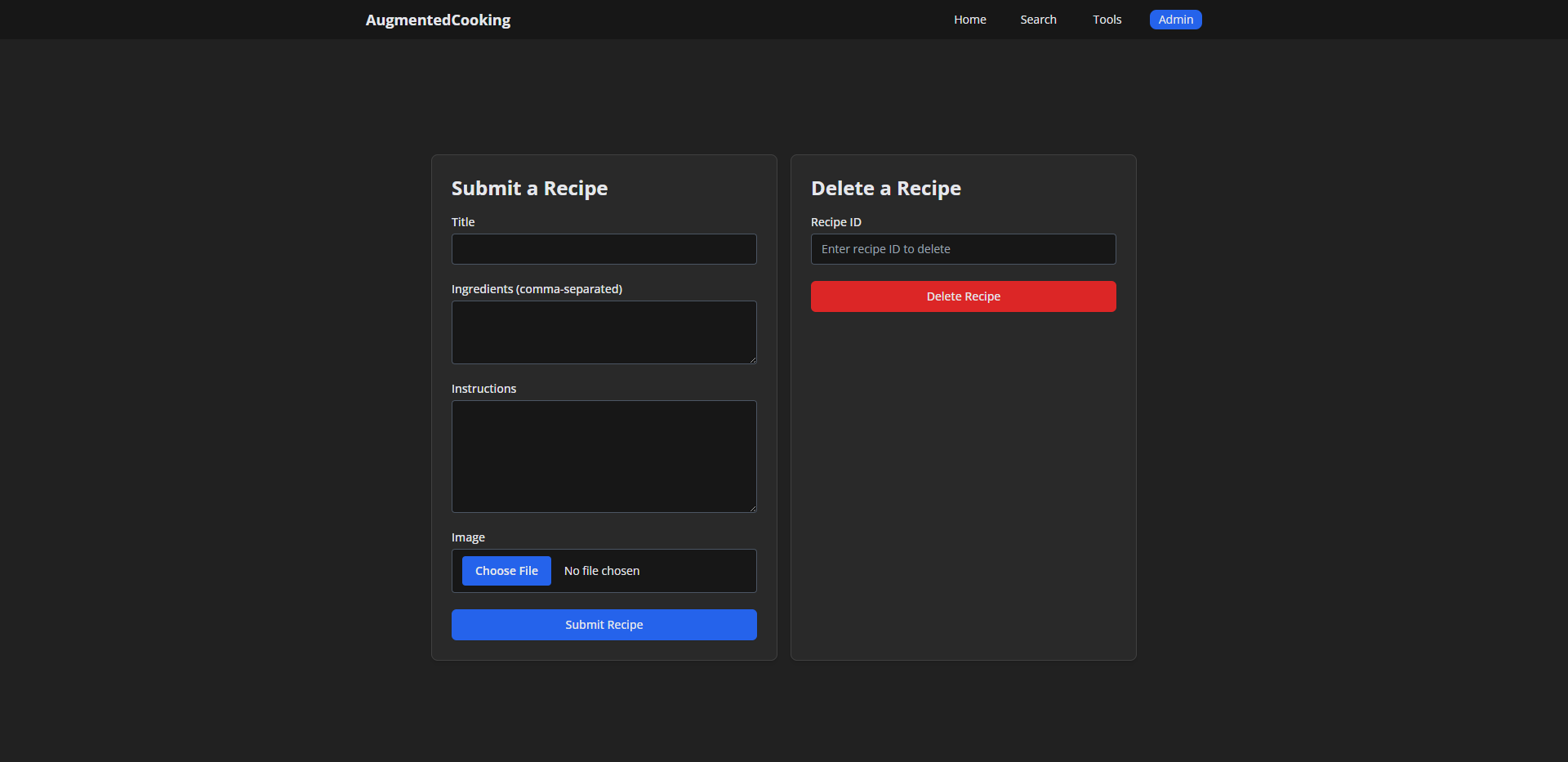
Prva stranica predstavlja interfejs AI asistenta, gde se korisnicima prikazuje dinamičan sadržaj generisan veštačkom inteligencijom.



* + - Stranice za registraciju i prijavu: Opis i screenshot.

Druga stranica omogućava pregled liste recepata. Korisnik ima opciju pregleda svih recepata, kao opciju pretrage po sastojcima, a detaljan prikaz pojedinačnih recepata se ostvaruje putem modal prozora klikom na željeni recept. Tokom listanja recepata korisniku se dodatni podaci dinamički učitavaju.

* + - Administrativni panel sa funkcijama pregleda i manipulacije podataka: Opis i screenshot.

Četvrta stranica je administrativni panel, dostupan isključivo administratorima, gde se omogućava dodavanje, ažuriranje i brisanje podataka.  


# Implementacija funkcionalnosti

## Korisničke funkcionalnosti

* + - Registracija i prijava korisnika (opis sa kodom).

Autentikacija je implementirana korišćenjem React konteksta `[useAuth](https://github.com/stojic-luka/faks-projekat/tree/master/Clients/web/src/contexts/useAuth)`. Korisnici se prijavljuju putem `[SignInPage](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/pages/signIn.tsx)` komponente, a pristup zaštićenim rutama osigurava se putem `react-router-dom`.

* + - Pregled podataka i interakcija sa njima. (opis sa kodom)

Komponenta `[RecipeList](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/components/recipeDisplay/recipeList.tsx)` dinamički učitava podatke recepata korišćenjem hook-a `[useFetchRecipes](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/hooks/recipe/useFetchRecipes.tsx)`. Klikom na određeni recept, otvara se detaljan prikaz u okviru `[RecipeModal](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/components/recipeDisplay/recipeModal.tsx)`.

* + - Dodavanje, brisanje i/ili ažuriranje podataka. (opis sa kodom)

Administratori mogu da kreiraju ili ažuriraju recepte pomoću forme unutar komponente `[SubmitRecipe](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/components/tools/admin/submitRecipe.tsx)`, zatim se podaci obrađuju i šalju ka backend-u koristeći hook `[useRecipeSubmit](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/hooks/recipe/useRecipeSubmit.tsx)`. Što se tiče brisanja recepata, administratori to mogu učiniti unutar `[DeleteRecipe](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/components/tools/admin/deleteRecipe.tsx)`, zatim se podaci obrađuju i šalju ka backend-u koristeći hook `[useRecipeDelete](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/hooks/recipe/useRecipeDelete.tsx)`.

## Administratorske funkcionalnosti

* + - Dodavanje novih elemenata.

Administratorima je omogućen pristup postavljanju novih recepata preko administrativnog panela.

* + - Ažuriranje postojećih elemenata.

Administratorima je omogućen pristup ažuriranja postojećih podataka preko administrativnog panela.

* + - Brisanje podataka.

Administratorima je omogućen pristup brisanju recepata preko administrativnog panela.

# Simulacija backend-a

## Korišćenje json-servera (ako je korišćen)

* + - Opis json-server alata i njegovog postavljanja.
    - Primer JSON datoteke sa podacima.

Aplikacija nije moguće biti testirana uz pomoć json-servera

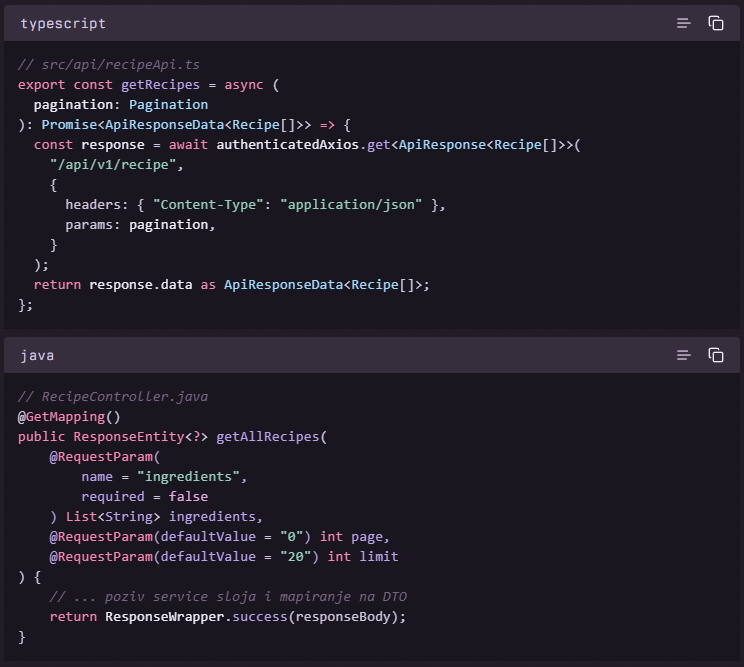
## Povezivanje frontenda i backend-a

* + - Korišćeni API pozivi (GET, POST, PUT, DELETE).
    - Primera kod-a za svaki API poziv.

Primeri korišćenih HTTP metoda i ruta iz `recipeService.ts` i `RecipeController.java`:

* GET /api/v1/recipe
* GET /api/v1/recipe/random
* GET /api/v1/recipe?ingredients=…&page=…&limit=…
* POST /api/v1/recipe
* PUT /api/v1/recipe?id={id}
* DELETE /api/v1/recipe?id={id}

1. GET all recipes (paginacija)



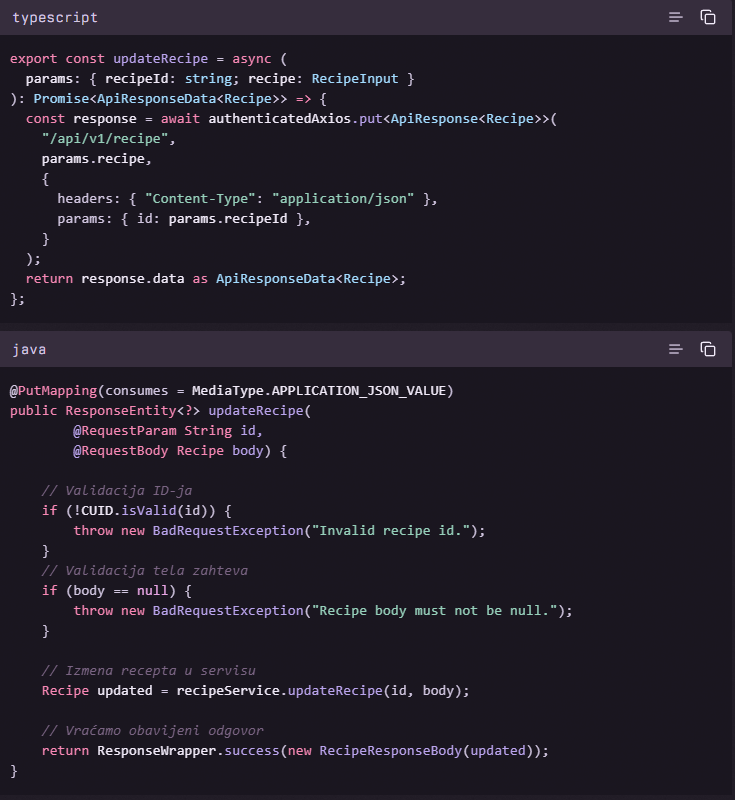
1. GET random recipe



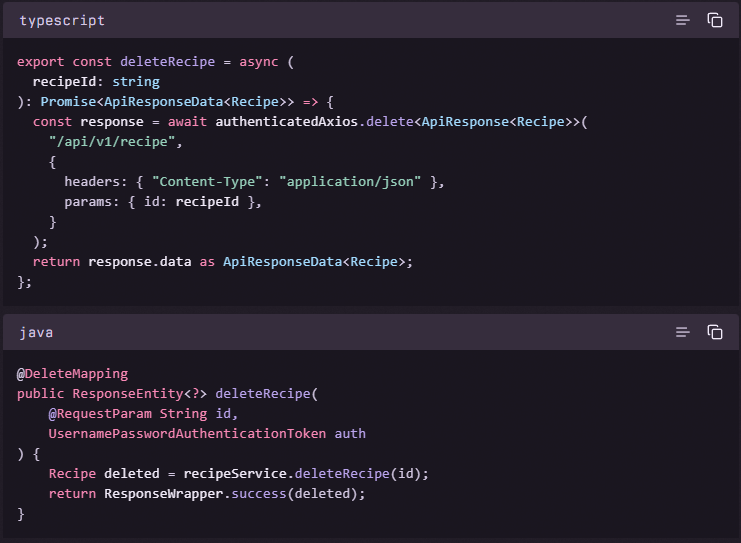
1. GET recipes by ingredients



1. POST add recipe



1. DELETE recipe



## Alternativne tehnologije

* + - Backend tehnologija:

U projektu je korišćena Java sa Spring Boot-om. Spring Boot je izabran zbog jednostavnosti konfiguracije, odlične podrške za skaliranje, kao i širokog izbor dostupnih biblioteka koje olakšavaju razvoj ove aplikacije. Takođe, Java je bila neophodna za drugi predmet, što je dodatni razlog za njen izbor.

* + - AI servis:

Za AI servis je korišćen Python jer je jednostavan za korišćenje i po moći u oblasti veštačke inteligencije. Python takodje ima veliki broj dostupnih biblioteka i framework-ova (kao što su TensorFlow, PyTorch, NumPy i podrška za CUDA softverskim slojem ko omogućava direktan pristup GPU) koji omogućavju brzu i efikasnu implementaciju AI modela.

# Zaključak

* + Pregled postignutih rezultata.

Uspostavljena je komunikacija između frontenda i backenda korišćenjem REST API-ja (GET, POST, PUT, DELETE), kao što je prikazano u fajlovima `[recipeService.ts](https://github.com/stojic-luka/faks-projekat/blob/master/Clients/web/src/services/recipeService.ts)`.

Implementirana je chat funkcionalnost sa mogućnošću dobijanja odgovora u celosti ili streamovanog odgovora, što je realizovano u klasi `[AiMessageService.java](https://github.com/stojic-luka/faks-projekat/blob/master/Server/src/main/java/com/augmentedcooking/Services/Chat/impl/AiMessageService.java)`.

Integrisan je RabbitMQ sistem za asinkronu obradu poruka, sa konfiguracijom definisanom u `[RabbitMQConfig.java](https://github.com/stojic-luka/faks-projekat/blob/master/Server/src/main/java/com/augmentedcooking/Config/RabbitMQ/RabbitMQConfig.java)`, što omogućava efikasnu obradu zahteva i odgovora.

* + Izazovi i prepreke tokom rada na projektu.

Dodavanje opcije da korisnik bira između streamovanja odgovora i dobijanja celog odgovora odjednom, čija logika se nalazi u fajlu `[AiMessageService.java](https://github.com/stojic-luka/faks-projekat/blob/master/Server/src/main/java/com/augmentedcooking/Services/Chat/impl/AiMessageService.java)`.

Konfiguracija i integracija RabbitMQ sistema bila je izazovna zbog prvog susreta sa ovom tehnologijom bez predhodnih iskustava. Konfiguracioni parametri zahtevali su detaljno razumevanje kako bi komunikacija izmedju Spring servera i AI servisa tekla bez problema.

* + Mogućnosti za buduća unapređenja aplikacije.

Dodavanje enkriptovane sinhronizacije sa ostalim aplikacijama, što bi omogućilo sigurnu komunikaciju i laku komunikaciju preko svih uredjaja.

Implementacija sistema za enkritovani backup poruka kako bi se povećala pouzdanost i sigurnost podataka.

# Prilozi

* + <https://github.com/stojic-luka/faks-projekat/tree/master/Clients/web>