

Kratak opis aplikacije

Recipe Collection je web aplikacija koja omogućava korisnicima čuvanje, uređivanje i organizovanje recepata. Putem intuitivnog korisničkog interfejsa korisnici mogu:

- Dodavati nove recepte sa nazivom, kategorijom i vremenom pripreme
- Pregledati sve sačuvane recepte
- Uređivati postojeće recepte
- Brisati recepte

Aplikacija je izgrađena kao full-stack rješenje sa Node.js backend-om i čistim HTML/CSS/JavaScript frontend-om, uz MongoDB bazu podataka za trajno skladištenje podataka.

Softverski preduslovi

Za pokretanje aplikacije potrebno je sljedeće:

- **Docker**
- **Operativni sistem:** Linux, macOS ili Windows sa WSL (Windows Subsystem for Linux)
- **Web pretraživač:** Chrome, Firefox, Safari ili Edge

Arhitektura aplikacije

Recipe Collection koristi mikroservisnu arhitekturu koja se sastoji od tri glavne komponente:

1. Frontend servis:

- HTML, CSS i JavaScript
- Pristupa API-ju za manipulaciju podataka
- Koristi Vite za development sa hot-reloading funkcionalnosti

2. Backend servis:

- REST API izgrađen sa Express.js
- CRUD operacije za upravljanje receptima
- Komunikacija sa MongoDB bazom podataka

3. MongoDB baza podataka:

- NoSQL baza podataka za trajno čuvanje recepata
- Čuva podatke u Docker volume-u

Opis Docker komponenti

Servisi

1. notes-frontend:

- Web aplikacija dostupna na portu 8080
- Kontejner ima pristup lokalnom direktoriju **frontend/** za development
- Koristi Vite za hot-reload funkcionalnost

2. notes-backend:

- API servis dostupan na portu 3000
- Kontejner ima pristup lokalnom direktoriju **backend/** za development
- Komunicira sa MongoDB bazom podataka

3. notes-mongodb:

- MongoDB instanca koja radi na standardnom portu 27017 (interno)
- Podaci se čuvaju u Docker volume-u za perzistenciju

Mreže

- **notes-network:** Privatna Docker mreža koja povezuje sva tri servisa i omogućava komunikaciju među njima

Volumes

- **mongodb_data:** Docker volume koji služi za trajno čuvanje podataka iz MongoDB baze, nezavisno od životnog ciklusa kontejnera

Uputstva za pokretanje

Aplikacija dolazi sa skriptama koje olakšavaju upravljanje njenim životnim ciklusom:

Priprema aplikacije

Skripta **setup.sh** obavlja pripremu aplikacije:

- Kreira Docker mrežu
- Gradi Docker image-e za frontend i backend
- Povlači MongoDB Docker image
- Kreira Docker volume za perzistenciju podataka

`./scripts/setup.sh`

Pokretanje aplikacije

Skripta `start.sh` pokreće sva tri servisa:

- Pokreće MongoDB kontejner
- Pokreće backend API servis
- Pokreće frontend aplikaciju

`./scripts/start.sh`

Zaustavljanje aplikacije

Skripta `stop.sh` zaustavlja sve servise aplikacije:

- Zaustavlja i uklanja frontend kontejner
- Zaustavlja i uklanja backend kontejner
- Zaustavlja i uklanja MongoDB kontejner

`./scripts/stop.sh`

Brisanje aplikacije

Skripta `delete.sh` potpuno uklanja aplikaciju:

- Zaustavlja sve servise (poziva `stop.sh`)
- Briše Docker image-e
- Briše Docker mrežu
- Opciono može obrisati i MongoDB volume (trenutno zakomentarisano da bi se podaci sačuvali)

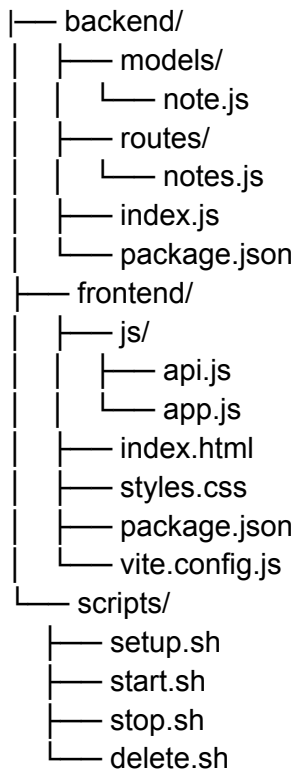
`./scripts/delete.sh`

Pristup aplikaciji

Nakon pokretanja aplikacije, možete joj pristupiti na sljedeći način:

- **Frontend aplikacija:** `http://localhost:8080`
- **Backend API:** `http://localhost:3000/notes`

Struktura projekta



Napomene za razvoj

- Frontend koristi Vite za brzi razvoj sa hot-reload podrškom
- Backend koristi nodemon za automatsko ponovno pokretanje nakon promjena
- Svi Docker kontejneri imaju pristup lokalnim folderima projekta, što omogućava izmjene koda bez ponovne izgradnje image-a

Sve skripte su prethodno pripremljene i nalaze se u **scripts/** direktoriju. Prije pokretanja skripti osigurajte da imaju izvršna prava (**chmod +x scripts/*.sh**).