

## Recipe/Cuisine Search WebApp

You can develop this application to help users discover and search for recipes based on ingredients, cuisine categories, or dietary restrictions. Users can input data on available ingredients and get matching recipes.

### Key Features:

- Users can search recipes by keywords, type ahead search
- Users can save their favourite recipes/create lists
- Users can rate and review recipes
- Get personalized recommendations

### Applications in real life:

- Recipe sharing platform
- Food blogging platforms
- Healthy recipe website

### Skills Gained:

- Data Scraping
- Programming languages like Python, CSS, and JavaScript
- Database systems like SQL
- API integration: Fast APIs
- Docker: Containerization and deploying app locally or on Heroku
- **FastAPI**: Building high-performance APIs with async support
- **React.js**: Creating dynamic frontend with API integration
- **MySQL & SQLAlchemy**: Structuring relational data and using ORM
- **Containerization**: Dockerizing applications for scalability
- **CI/CD(Optional)**: Automating deployment with GitHub Actions/Azure DevOps

### Challenges:

- Extracting data from websites: Web scrapping
- Designing user-friendly interface
- Personalizing user experience

- **CORS Issues in Frontend-Backend Communication:** Configuring FastAPI to handle CORS when communicating with the React frontend.
  - **Recipe Data Collection:** Deciding between APIs vs. self-managed dataset
  - **Performance Optimization:** Handling complex search queries efficiently
  - **Scaling:** Ensuring smooth performance with growing traffic
  - **Security:** Preventing SQL injection, API abuse, and securing authentication
1. **Set Up Project Repo:** Create a GitHub repo and initialize FastAPI, React, and MySQL
  2. **Schema:** Design MySQL schema for recipes and users.
  3. **Build Backend APIs:** Implement FastAPI endpoints for searching and retrieving recipes.
  4. **Develop Frontend UI:** Design and implement React-based UI for search functionality
  5. **Integrate Database & Frontend:** Connect React frontend with FastAPI backend
  6. **Dockerize & Deploy:** Containerize and deploy the application to the cloud.

## 1. Frontend (User Interface)

- **Components:**
  - Search bar for ingredients, cuisines, and dietary preferences
  - Recipe list and details page
  - User authentication (optional)
  - Filters (e.g., vegetarian, vegan, gluten-free, low-carb)
- **Tech Stack:** React.js (preferred), TailwindCSS for styling, Axios for API calls

## 2. Backend (API & Business Logic)

- **Components:**
  - FastAPI-based REST API
  - Endpoints for searching, filtering, and retrieving recipes
  - Authentication (JWT-based if user profiles are needed)
  - Caching for frequently searched recipes (Redis or FastAPI in-memory caching)
- **Tech Stack:** FastAPI, Pydantic for request validation, SQLAlchemy for ORM

### 3. Database Layer

- **Components:**
  - MySQL database for storing recipes, ingredients, user preferences
  - Tables for recipes, cuisines, ingredients, and dietary restrictions
  - Indexing for faster search queries
- **Tech Stack:** MySQL, SQLAlchemy for ORM, Alembic for migrations

### 4. Data Ingestion & Processing

- **Components:**
  - Scraping or integrating third-party recipe APIs (e.g., Spoonacular, Edamam)
  - Data cleaning and preprocessing
  - Batch ingestion for recipes and ingredients
- **Tech Stack:** Python, BeautifulSoup/Scrapy for scraping, Celery for background jobs

### 5. Containerization & Deployment

- **Components:**
  - Docker for containerizing the application
  - Docker Compose for managing multiple services (API, DB, frontend)
  - CI/CD pipeline for automated deployment
  - Load balancing and scaling
- **Tech Stack:** Docker, Docker Compose, GitHub Actions/Azure DevOps, AWS/GCP/Azure for hosting

