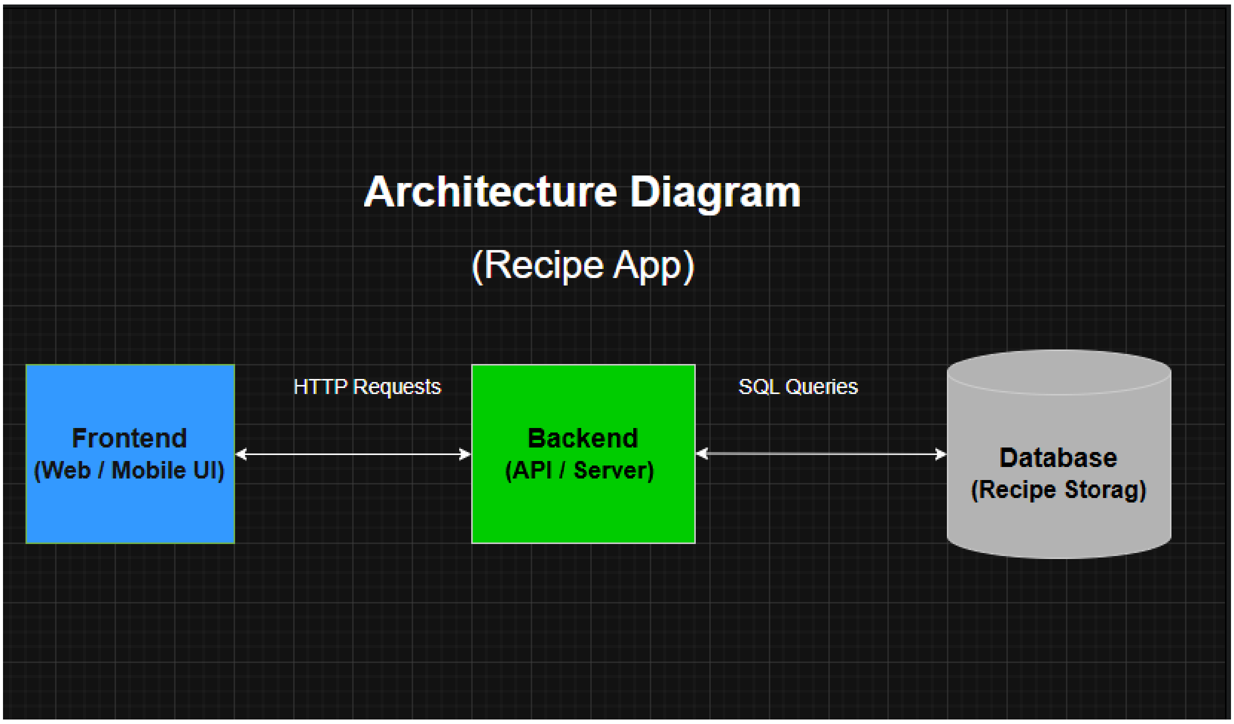
By: Benjina Paudyal

# Architecture Diagram (Recipe App)



## Component Explanations

### Frontend (UI)

- Displays all recipe lists and individual recipe details
- Sends requests to the backend to create, read, update, and delete (CRUD) recipes
- Receives responses and updates the user interface accordingly

### Backend (API Server)

- Provides endpoints
  - POST /recipes –> Add new recipe
  - GET /recipes –> Get all the recipes
  - GET /recipes/:id –> Get a recipe by ID
  - PUT /recipes/:id –> Update a recipe by ID
  - DELETE /recipes/:id –> Delete a recipe by ID
- Validates incoming data
- Communicates with database
- Returns responses back to the frontend

### Database

- Stores all recipe data: id, title, ingredients, instructions, etc.
- Supports CRUD operations requested by the backend

## Sequence Diagrams (Text Format)

### Adding a new recipe

- User → Frontend → Backend → Database
- User clicks "Add Recipe" button
- Frontend sends POST /recipes with recipe data
- Backend saves data to database
- Backend sends success response
- Frontend updates the page

### Viewing a specific recipe

- User → Frontend → Backend → Database
- User clicks on a recipe to view
- Frontend sends GET /recipes/:id
- Backend retrieves recipe data from database
- Backend sends recipe data to frontend
- Frontend displays recipe details

### Editing an existing recipe

- User → Frontend → Backend → Database
- User clicks "Edit" on a recipe
- Frontend sends PUT /recipes/:id with updated data
- Backend updates recipe in database
- Backend sends success response
- Frontend updates the page with new recipe details