****

**PROGRAMMING FOR AI (LAB)**

**Lab Task No 07&08**

**Submitted To :- Sir Rasikh Ali**

**Submitted By :- Aiman Ijaz**

**Roll No :- SU92-BSAIM-F23-010**

**Department :- Software Engineering**

**Section :- BSAI-4A**

**Flask and API Based Recipe Finder App**

**1. Introduction**

The **Recipe Finder App** is a web-based application developed using **Flask** for the backend and **HTML/CSS** for the frontend. It allows users to search for recipes by name and view detailed instructions, ingredients, and images. The app fetches real-time data from the **MealDB API** and presents it in an interactive and user-friendly way.

**2. Purpose of the App**

The app is designed to help users find recipes easily by simply entering a dish name. It eliminates the need to manually search for recipes online by providing instant results with **ingredients, instructions, and images** from an external API.

**3. How the App Works**

**Backend (Flask) Functionality**

1. **Fetching Data from API**
   * The Flask app sends a request to the **MealDB API** using requests.get().
   * The API responds with **data** containing recipe details.
   * The app extracts relevant information (dish name, ingredients, instructions, and image).
2. **Routing in Flask**
   * The app has a main route (/) that displays the search form.
   * When a user submits a search query, Flask processes it and fetches the corresponding recipe from the API.
   * The results are then passed to an HTML template (index.html).

**4. User Interaction & Frontend Functionality**

**Step-by-Step User Flow**

1. **Home Page**
   * The user visits the homepage and sees a **search bar**.
   * They enter the name of a dish (e.g., "Pasta") and click the **Search** button.
2. **Fetching & Displaying Data**
   * The app sends a request to the API.
   * The API returns the recipe details.
   * The app dynamically updates the page with:
     + The **recipe name**
     + A **high-quality image** of the dish
     + A **list of ingredients**
     + Step-by-step **cooking instructions**
3. **Responsive & Interactive Design**
   * The frontend uses **HTML, CSS, and JavaScript** for a smooth user experience.
   * The layout adjusts based on screen size (mobile, tablet, or desktop).

**5. Technologies Used**

| **Technology** | **Purpose** |
| --- | --- |
| **Flask** | Backend framework to handle API requests & routing |
| **HTML/CSS** | Frontend design and structure |
| **Requests** | Fetching data from the **MealDB API** |
| **Bootstrap** | Styling & responsiveness |

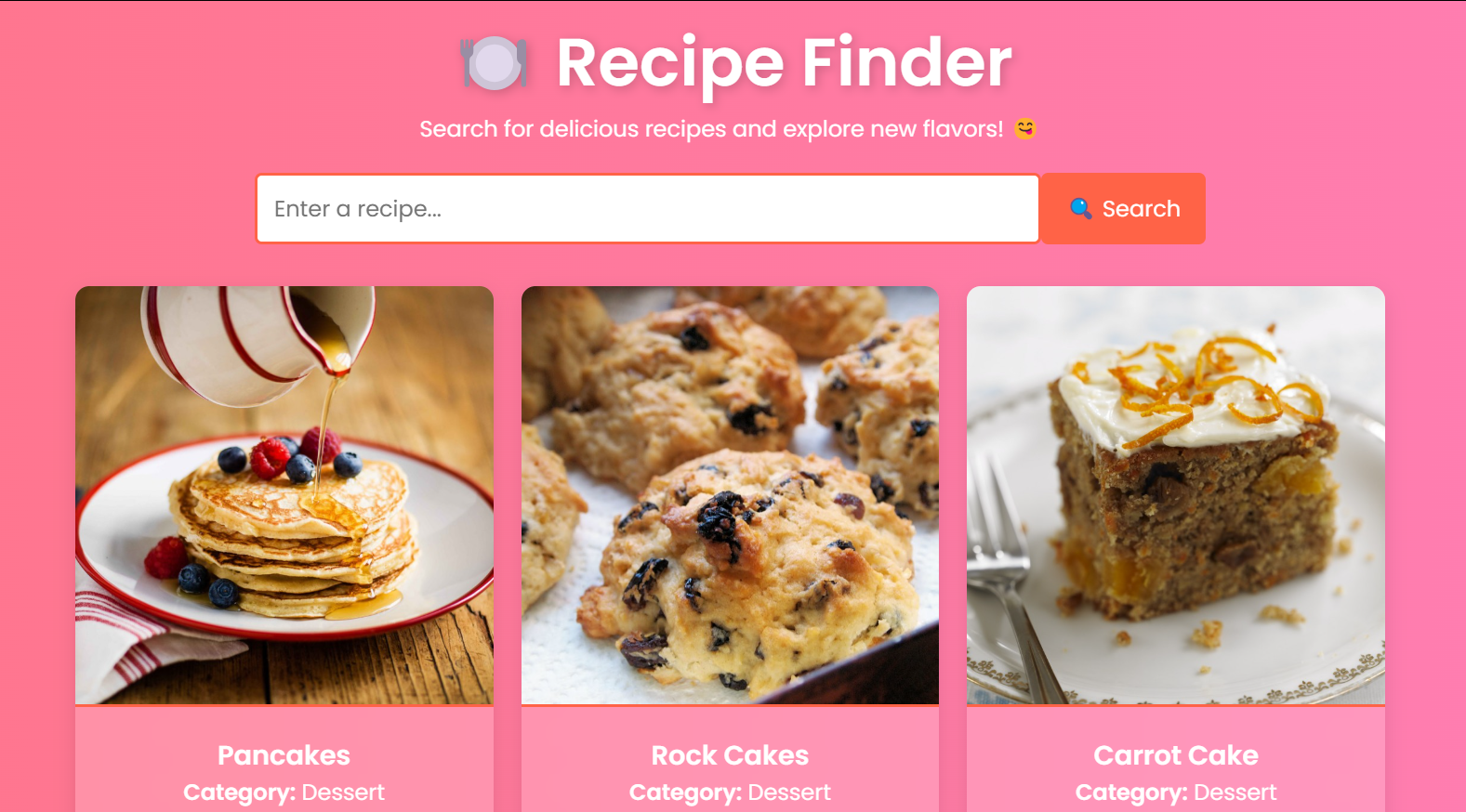
**6. Future Enhancements**

🔹 **Filter by Cuisine Type** (e.g., Italian, Indian, Chinese)  
🔹 **Advanced Ingredient Search** (Find recipes based on available ingredients)  
🔹 **Save Favorite Recipes** (User authentication & database integration)  
🔹 **Voice Search** (Search for recipes using voice input)

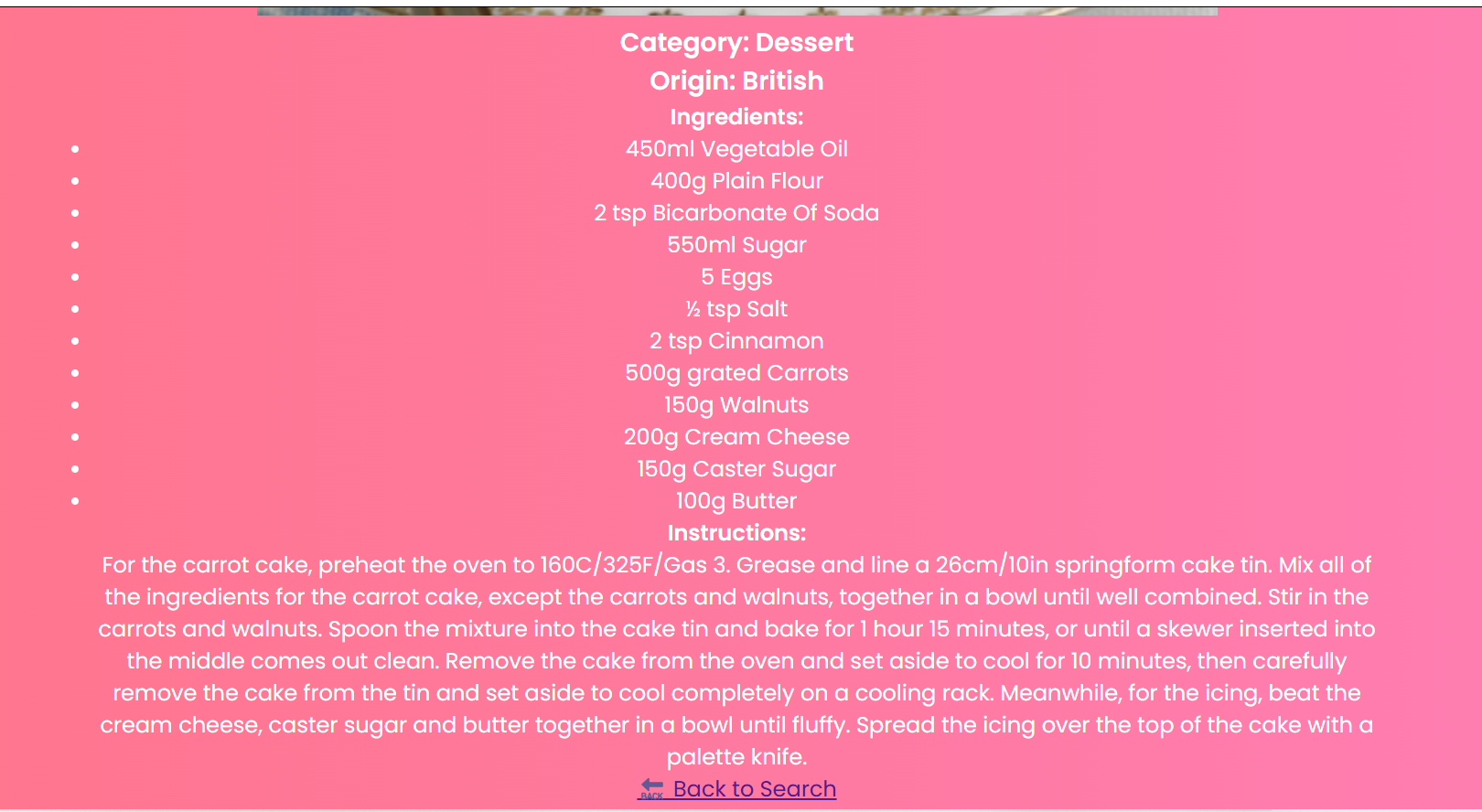
**7. Conclusion**

The **Recipe Finder App** provides a simple and efficient way for users to find and explore new recipes. By leveraging the **MealDB API**, the app delivers real-time and accurate recipe data, making cooking more accessible and enjoyable.

🎯 **Key Takeaways:**  
✅ **Easy to use** – Just enter a dish name & get recipes instantly  
✅ **Live API integration** – Always up-to-date data  
✅ **Attractive UI** – Clean and responsive design

****

****

****