**Documentation**

Logical Approach:

**Understand the Requirements**

* Display a paginated list of recipes fetched from the FastAPI backend.
* Allow users to change the number of results per page dynamically.
* Provide navigation buttons to move between pages.
* Update the UI accordingly when data is fetched.

**Define Variables**

* currentPage (Tracks the current page, starts at 1).
* resultsPerPage (Stores how many results are displayed per page, default 10).

**Event Handling**

* Dropdown Change (Results Per Page)
  + Update resultsPerPage when the user selects a new value.
  + Reset currentPage to 1.
  + Fetch and display recipes.
* Previous Page Button (#prev-page)
  + If currentPage > 1, decrement it and fetch recipes.
* Next Page Button (#next-page)
  + Increment currentPage and fetch recipes.

**Fetch data from API**

* Fetch data using fetch().
* If successful, parse JSON response.
* Clear and update the table with new data.
* Update the current page number on the UI.

**Update UI Elements**

* Display fetched recipes in the table dynamically.
* Update the displayed page number (#page-info).

**Handle Errors**

* If the API call fails, log errors to the console.
* Handle cases where no recipes are returned.

**Initialize on Page Load**

* Call fetchRecipes() to load the first set of data when the page loads.

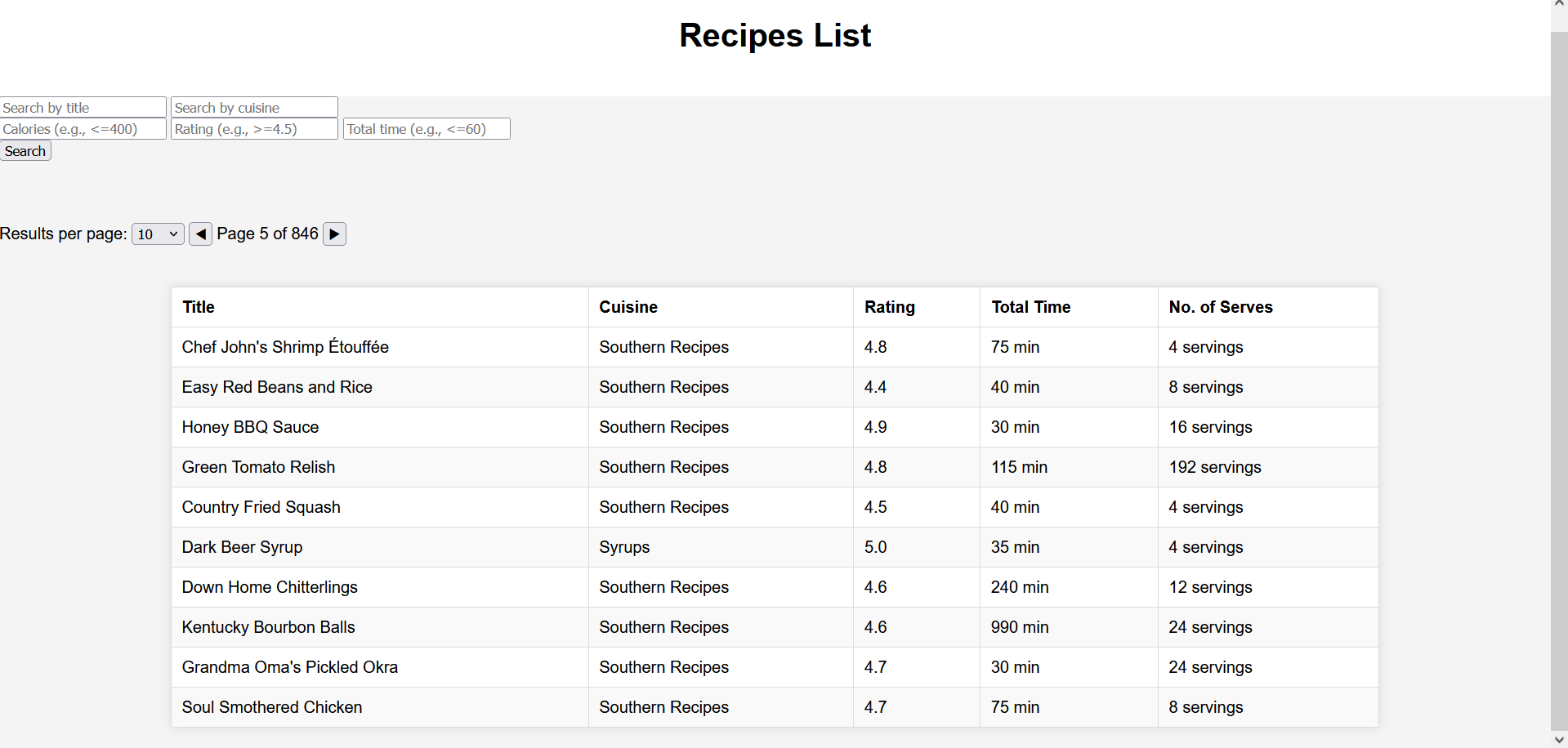
**Screenshots:**

Default Page:

A screenshot of a recipe list

AI-generated content may be incorrect.

Limit = 10 and Page = 5

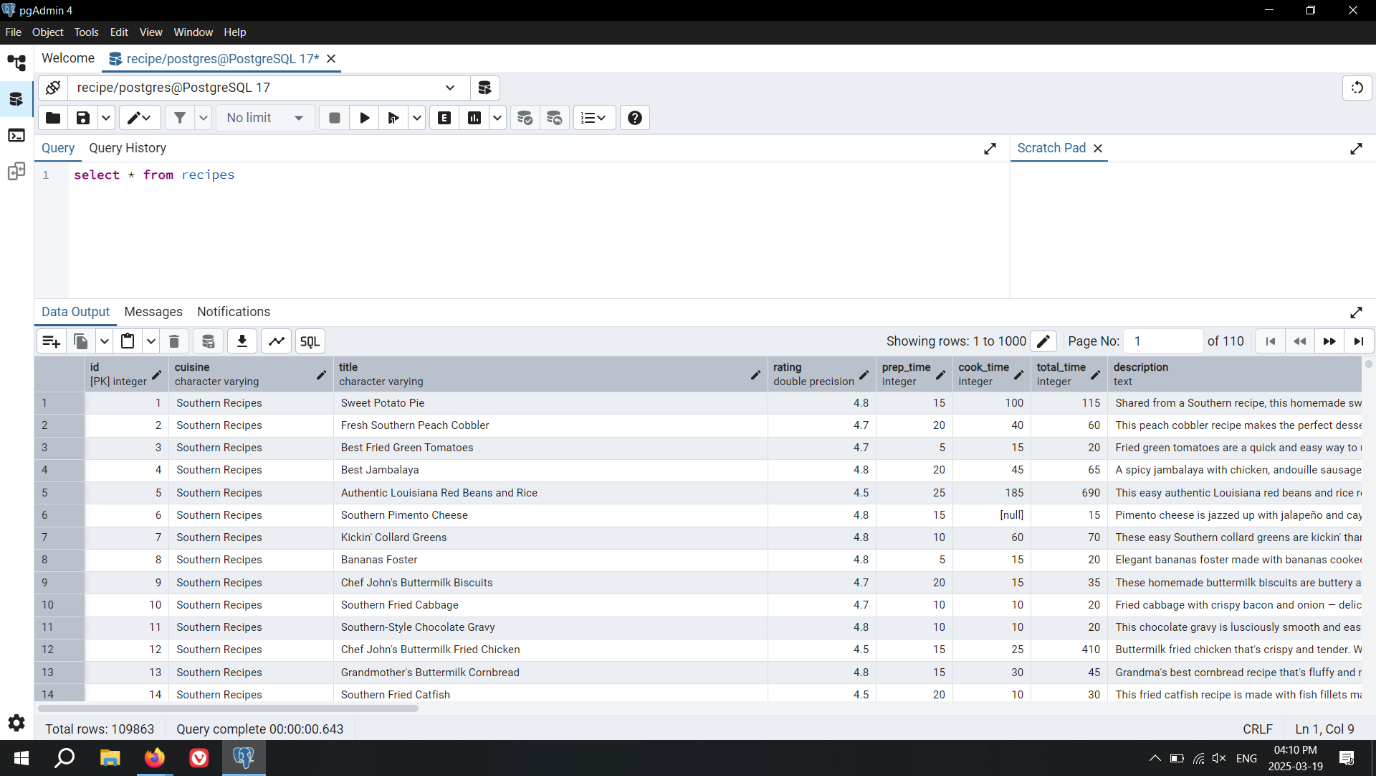


Limit = 50 and Page = 1

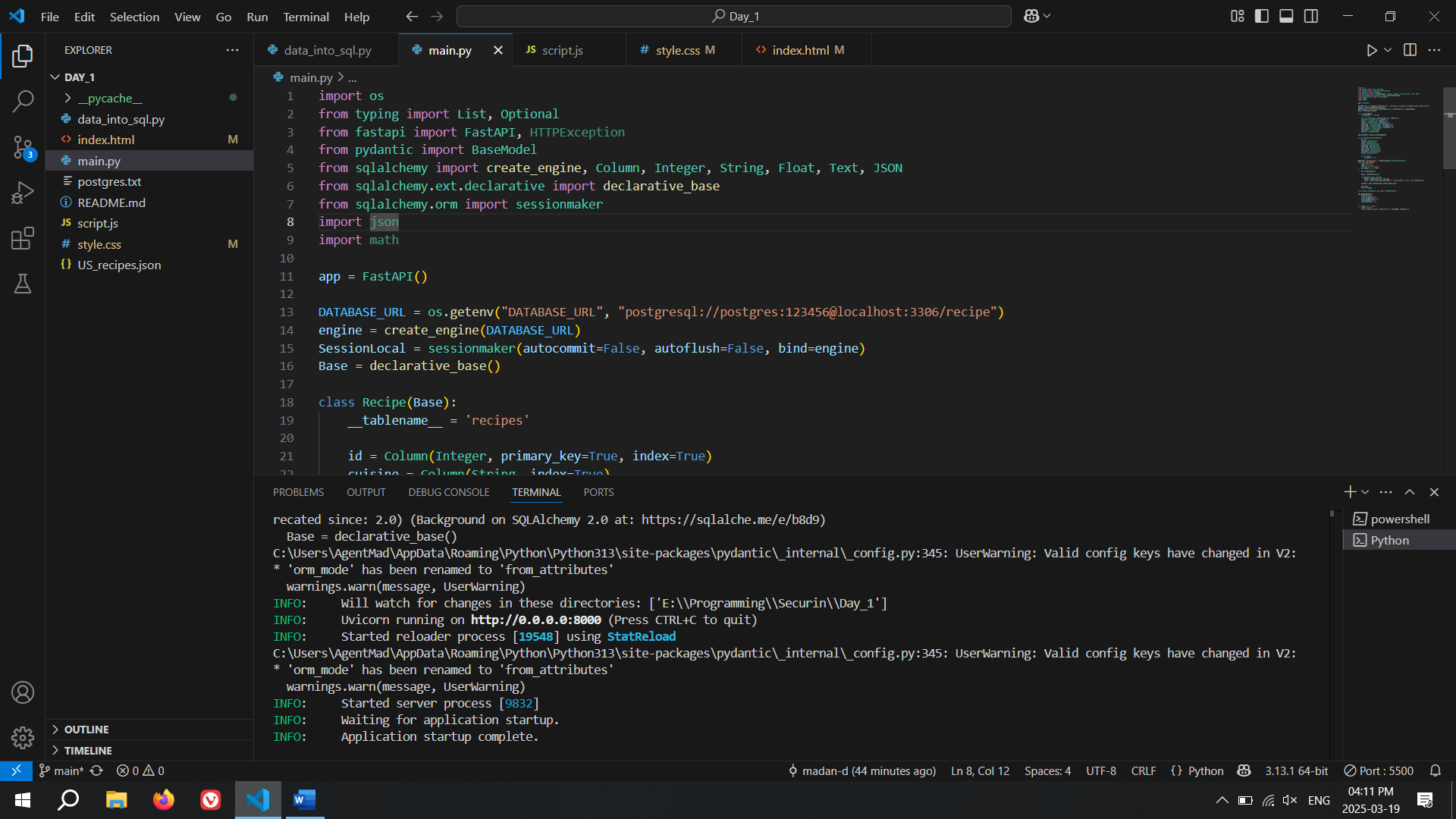
A screenshot of a recipe list

AI-generated content may be incorrect.

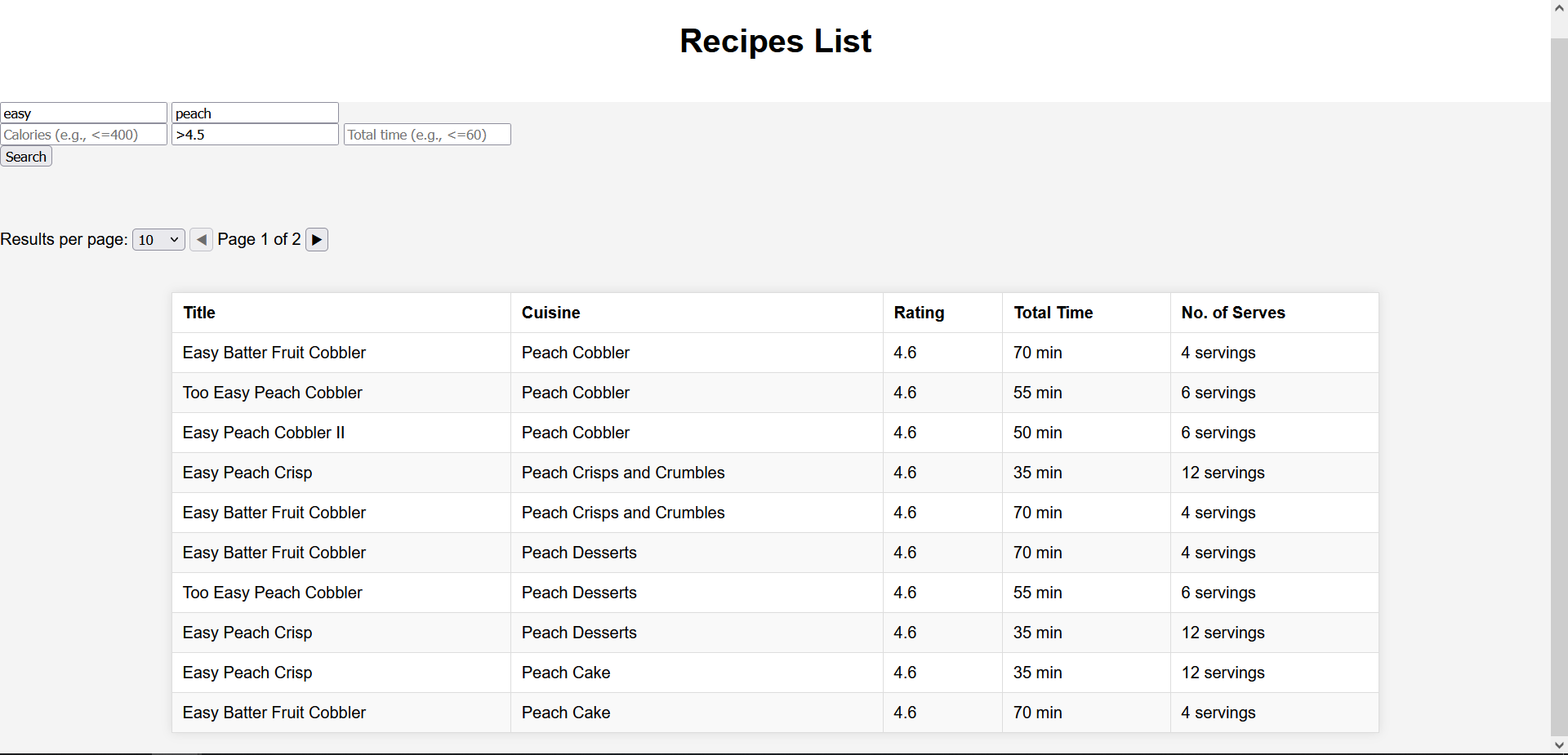
Postgresql Output:



FastAPI



Seach result: (title = easy, cuisine = peach, rating = > 4.5)



**Challenges encountered and the solutions implemented:**

**Pagination Edge Cases**

* **Challenge:** Navigating beyond available pages or handling empty results.
* **Solution:**
  + Disabled the "Previous Page" button when currentPage == 1.
  + Implemented a check to prevent incrementing currentPage when the API returns no data.

**Dynamic Results Per Page Handling**

* **Challenge:** Changing the dropdown value should immediately update results while resetting pagination.
* **Solution:**
  + Attached an event listener to detect dropdown changes.
  + Reset currentPage to 1 before calling fetchRecipes().

**Ensuring Table Updates Correctly**

* **Challenge:** Old data persisted in the table when fetching new results.
* **Solution:**
  + Cleared the <tbody> of the table before appending new rows.

**Maintaining Page State on UI**

* **Challenge:** Keeping track of the current page visually.
* **Solution:**
  + Updated #page-info dynamically after each fetch.

**Handling Missing Data in API Response**

* **Challenge:** Some recipes might not have all fields (e.g., missing rating or cuisine).
* **Solution:**
  + Used fallback values ('N/A') for missing attributes in table rows.

**Github link:** https://github.com/madan-d/recipe-info.git