

# **Recipe Browser Assessment**

**By Brandon Jim**

## **#Overview**

This project implements an application that parses recipe data from the JSON file, stores it in a SQLite database, and provides an API to access and search the data. This assessment also includes a frontend UI that gathers from the API to display the recipe information with filtering, sorting, and pagination capabilities.

## **# Backend**

- Data Parsing: Reads and parses recipe data from the JSON file called USA\_recipes.json
- Data storing: Stores the parsed data in a sqlite database
- API: /api/recipes: fetches all recipes with pagination and sorting.
- /api/recipes/search: searches recipes by title, cuisine, rating, and total time.

## **#frontend**

- Recipe table: Displays the recipes with columns for Title, Cuisine, Rating, Total Time, and serves.
- Filters each columns to refine search results.
- Responsive recipe table with filtering capabilities
- Detailed recipe view in a side drawer
- Pagination controls with customizable results per page
- Star rating visualization
- Expandable details for time information
- Nutrition information table

## # Setup and Installation

### ## Prerequisites

-Python 3.7, I used VSCODE and installed python through there.

-Flask -> Installed in vscode terminal 'pip install flask'

- Download SQLite extension in vscode

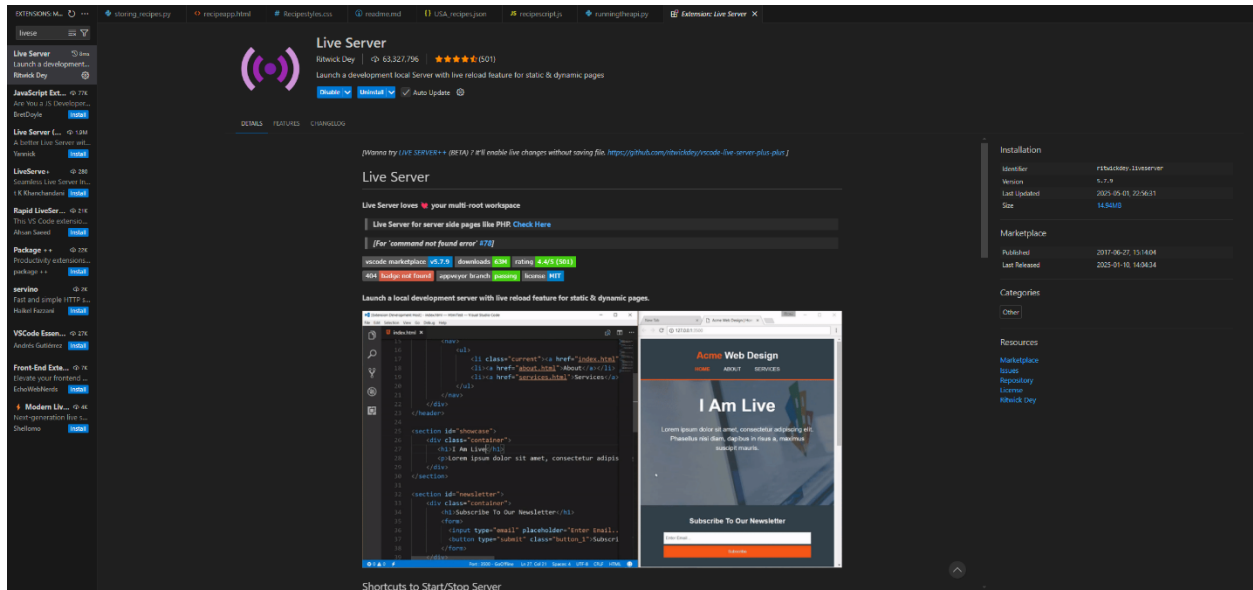
-Download liveserver extension in vscode

### ### Installation

1. Get clone Repository
2. I used vscode to run this project.
3. Install Flask in vscode terminal 'pip install flask'
4. Get vscode extensions liveserver and sqlite
5. Run 'python recipes\_storingrecipe.py' to create a database called USA\_recipes.db
6. Ctrl-c and then run 'python runningtheapi.py' to run the API to the server.
7. Then right click on recipeapp.html and click on 'Open with live server'

## Pictures

### Exension liveserver



## Extension SQLite

The screenshot shows the SQLite extension page in the Visual Studio Code marketplace. The extension is by alexcz, has 3,485,176 downloads, and a 4.5-star rating from 74 reviews. It is described as a tool to "Explore and query SQLite databases." The page includes a sidebar with a list of related extensions, a main preview area showing the extension's interface, and a right-hand panel with installation details, marketplace information, and resources.

**SQLite**  
alexcz · 3,485,176 · ★★★★★ (74)  
Explore and query SQLite databases.

**Details** · Features · Changelog

**vscode-sqlite**  
VSCode extension to explore and query SQLite databases.

**Installation**

Property	Value
Identifier	alexcz.vscode-sqlite
Version	0.14.3
Last Updated	2025-05-07 22:31:16
Size	1.13MB

**Marketplace**

Property	Value
Published	2018-06-25 10:25:17
Last Released	2025-05-04 10:28:15

**Categories**

Other

**Resources**

- Marketplace
- Issues
- Repository
- License
- alexcz

**Requirements**

Windows, MacOS: No requirement.

**Linux:** If the extension is not working out of the box, it may be necessary to install sqlite in your system (on Ubuntu: `sudo apt install sqlite3`)

**Note:** The extension includes precompiled binaries for the SQLite CLI (used to execute queries), in case the included binaries do not work (or if you want to use your own binaries) you need to provide the path/command to the sqlite CLI executable in the setting `sqlite.sqliteCli`.

**Features**

- Query SQLite databases, and view results in a table (also supports dot commands like `SQLite> .schema`, `SQLite> .exit`).

## Install for flask

```
PS C:\Brandon's stuff\New Assesment> pip install flask
Requirement already satisfied: flask in c:\users\brandon\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from flask) (3.1.0)
Requirement already satisfied: Werkzeug>=3.1 in c:\users\brandon\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from flask) (3.1.0)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\brandon\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.2 in c:\users\brandon\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\brandon\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from flask) (8.1.8)
Requirement already satisfied: blinker>=1.9 in c:\users\brandon\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from flask) (1.9.0)
```

The screenshot shows the Visual Studio Code terminal with the command `python storing_recipes.py` entered. The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the command prompt and the command being executed.

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS C:\Brandon's stuff\New Assesment> python storing_recipes.py
```

Used to parse through the USA\_recipes.json to create USA\_recipes.db

```

# runningtheapi.py
1 from flask import Flask, request, jsonify
2 import sqlite3
3 import re
4 import json
5 import os
6
7 app = Flask(__name__)
8
9 # Database connection function
10 def get_db_connection():
11     db_path = os.getenv('DB_PATH', 'USA_recipes.db') # Use environment variable for DB path
12     conn = sqlite3.connect(db_path)
13     conn.row_factory = sqlite3.Row
14     return conn
15
16 @app.route('/', methods=[GET])
17 def home():
18     """Home route that provides information about available API endpoints"""
19     return jsonify({
20         'status': 'success',
21         'message': 'Recipe API is running',
22         'available endpoints': {
23             'Get all recipes (paginated)': '/api/recipes?page=1&limit=10',
24             'Search recipes': '/api/recipes/search?keyword=italian&page=1',
25             'Filter recipes': {
26                 'By rating': '/api/recipes/search?rating=4.5',
27                 'By title': '/api/recipes/search?title=pizza',
28                 'By cuisine': '/api/recipes/search?cuisine=Italian',
29                 'By time': '/api/recipes/search?total_time=30',
30                 'By rating': '/api/recipes/search?rating=4.5'
31             }
32         }
33     })
34
35 @app.route('/api/recipes', methods=[GET])
36 def get_recipes():
37     """Fetch paginated recipes sorted by rating"""
38     try:
39         # Validate and parse query parameters
40         page = int(request.args.get('page', 1))
41         limit = int(request.args.get('limit', 10))
42         offset = (page - 1) * limit
43
44         conn = get_db_connection()
45         cursor = conn.cursor()
46
47         # Get total count of recipes
48         cursor.execute('SELECT COUNT(*) FROM recipes')
49         total = cursor.fetchone()[0]
50
51         # Get paginated recipes
52         cursor.execute('SELECT id, cuisine, title, rating, prep_time, cook_time, total_time, '
53                        'description, nutrition, serve '
54                        'FROM recipes '
55                        'ORDER BY rating DESC '
56                        'LIMIT ? OFFSET ?')
57         recipes = cursor.fetchall()
58
59         # Convert to list of dictionaries
60         recipes_list = []
61         for recipe in recipes:
62             recipes_list.append({
63                 'id': recipe['id'],
64                 'cuisine': recipe['cuisine'],
65                 'title': recipe['title'],
66                 'rating': recipe['rating'],
67                 'prep_time': recipe['prep_time'],
68                 'cook_time': recipe['cook_time'],
69                 'total_time': recipe['total_time'],
70                 'description': recipe['description'],
71                 'nutrition': recipe['nutrition'],
72                 'serve': recipe['serve']
73             })
74
75         return jsonify({'recipes': recipes_list, 'total': total, 'page': page, 'limit': limit})
76     except Exception as e:
77         return jsonify({'error': str(e)}), 500
78
79 if __name__ == '__main__':
80     app.run(debug=True)

```

Run the command `python runningtheapi.py` for the api server to initiate

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Brandon's stuff\New Assesment> python runningtheapi.py

```

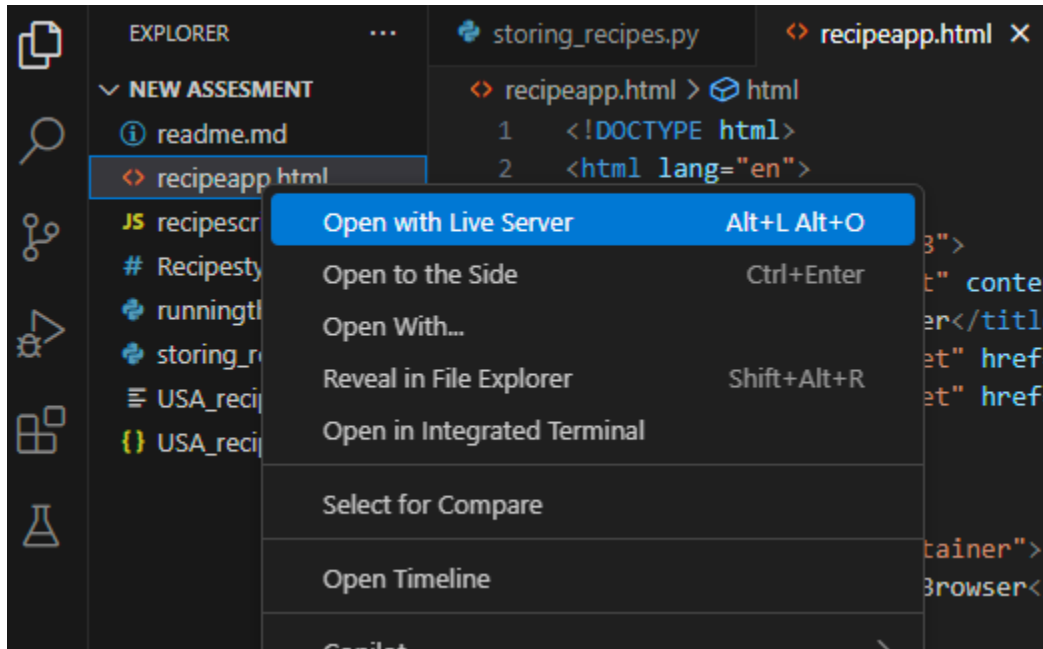
This is the result of the command running perfectly

```

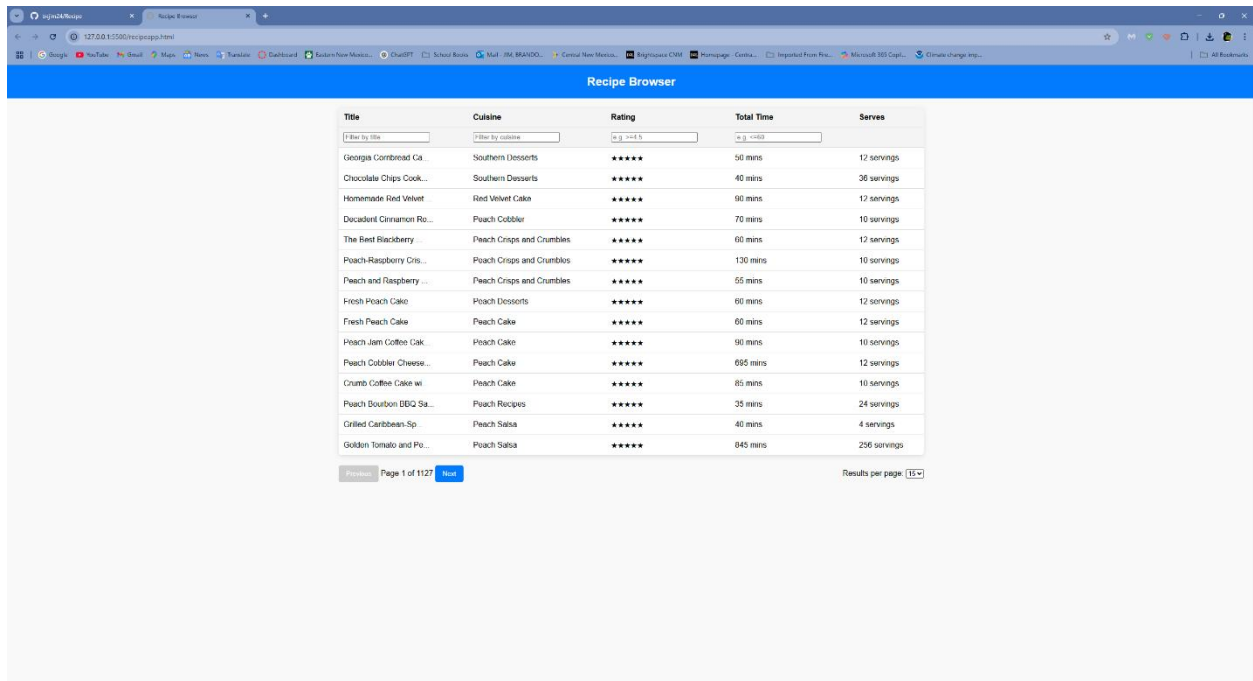
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Database created and populated successfully!
* Debugger is active!
* Debugger PIN: 402-002-812
PS C:\Brandon's stuff\New Assesment> python runningtheapi.py
* Serving Flask app 'runningtheapi'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 402-002-812

```

Then right click and open with live sever to initiate the GUI



The recipe url on localhost



127.0.0.1:5500/recipeapp.html

GoogleYouTubeGmailMapsNewsTranslateDashboardEastern New Mexico...ChatGPTSchool BooksMail - JIM BRANCO...Central New MexicoBrightspace CHMHomepage - Centra...Imported From Fire...Microsoft 365 Copi...Climate change imp...

Recipe Browser

Title	Cuisine	Rating	Total Time	Serves
<input type="text" value="Filter by title"/>	<input type="text" value="Filter by cuisine"/>	<input type="text" value="e.g. &gt;=4.5"/>	<input type="text" value="e.g. &lt;=60"/>	
Fresh Peach Cake	Peach Pie	★★★★★	60 mins	12 servings
Peach Crumble Pie	Peach Pie	★★★★★	65 mins	8 servings
Peach Tarts with ...	Peach Pie	★★★★★	32 mins	18 servings
Spiced Peach Pie	Peach Pie	★★★★★	205 mins	8 servings
Refreshing Peach Cup...	Peach Pie	★★★★★	45 mins	12 servings
Banana Pudding with ...	Banana Pudding	★★★★★	145 mins	24 servings
Banana Pudding with ...	Banana Pudding	★★★★★	30 mins	8 servings
Broche Bread Puddin...	Banana Pudding	★★★★★	160 mins	16 servings
Oat Pudding with Man...	Banana Pudding	★★★★★	40 mins	2 servings
Chocolate Pudding Po...	Banana Pudding	★★★★★	190 mins	4 servings
Banana Coconut Puddl...	Banana Pudding	★★★★★	20 mins	8 servings
Vegan Chocolate Bana...	Banana Pudding	★★★★★	5 mins	2 servings
Mini Croissant Crust...	Pecan Pie	★★★★★	75 mins	12 servings
Derby Pie	Pecan Pie	★★★★★	88 mins	8 servings
Chef John's Chocolat...	Pecan Pie	★★★★★	180 mins	8 servings

[Previous](#)Page 2 of 1127[Next](#)

Results per page: 15

Fresh Peach Cake  
(Peach Pie)

Description:  
This is a quick and easy peach cake made with fresh peaches. Perfect when you're getting a case of peaches from the market.

Total Time:  
60 mins

Serves:  
12 servings

Nutrition Information

Nutrient	Amount
calories	183 kcal
carbohydrateContent	25 g
cholesterolContent	51 mg
fatContent	9 g
fiberContent	0 g
proteinContent	2 g
saturatedFatContent	5 g
sodiumContent	85 mg
sugarContent	17 g
unsaturatedFatContent	0 g

## Data Flow

