Recipe Vault- LE/EECS 1012

Team: The Invincibles

Team Members:

- **Parul Mittal**
  Email: parulm75@my.yorku.ca
  Lecture- LE/EECS 1012 Section B
  Laboratory 02
  Role: Worked heavily on the basic structure of the website using HTML and CSS. Built the webpage layouts, organized content and added styles to make the pages look polished and consistent throughout. Helped in developing tests. He also collaborated with the team to ensure the design and functionality aligned with the project goals.

- **Ammar Panjwani (**
  Email: ammar110@my.yorku.ca
  Lecture- LE/EECS 1012 Section B
  Laboratory 02
  Role: Played a fundamental role in the project by troubleshooting errors and ensuring that the webpage displayed correctly across all devices, including cellphones, tablets, and larger screen sizes. Worked on all three programming languages and assisted other group members whenever they needed help. Helped create all the wireframes and contributed ideas for website features. Additionally, handled the documentation and report-writing process, ensuring that all the requirements were fulfilled and well-organized.

- **Mohammad Abbas**
  Email: mabbas25@my.yorku.ca
  Lecture- LE/EECS 1012 Section B
  Laboratory 02
  Role: Worked mostly on the JavaScript component and the backend of the page. The programming lead for the project. Supervised the code. Worked on server side API calls and responsive design. Worked on test files and overall coding milestones. Additionally, he helped other team members when they encountered any problems.

- **How to run?**

Run **npm install** followed by **npm start** in the terminal. Refer to README for more details

- **Revised User Stories**

In the process of development we realized our goals were a little too ambitious so we toned it down a bit.

- As a user, I want there to be a navigation bar so that I can navigate the site easily.
- As a verified user, I want a ‘My Profile’ page so that I can view and change my username and password or delete my account.
- As an existing user, I want a login page so that I can easily log into my account.
- As a new user, I want an option to easily register/sign up via the login page by providing my email and password.
- As an existing user, I want an option to delete certain recipes by selecting them so that I can manage my recipes easily.
- As an existing user, I want to be able to categorize recipes.
- As a user, I want the web application to have both a light and dark mode so that I choose whatever mode I prefer based on my preferences.
- As a user, I want to edit recipes by clicking on a button so that I can easily and quickly edit recipes.
- As a user, I want to be able AI generate recipes for inspiration.
- As a user, I want to be able to add bullet points, headings, and basic text formatting so I can make my recipes more legible.
- As a user, I want to be able to export my recipes to a text file or a pdf file so that I can download them and keep copies of my recipes.
- As a user, I want a log out button so that I can log out and automatically lock all my recipes.
- As a user, I want to filter recipes by category (e.g., breakfast, dessert, vegan) so that I can organize and view them efficiently.
- As a user, I want to be able to set a secure password.
- As a user, I want the website to be responsive so I can view it on multiple devices.

- **Learning outcomes:**

This project was a great collective learning outcome for us. The project itself was designed in a way that required us to apply all the knowledge gained in classes and the labs while also pushing us to learn additional skills on our own to get the desired results. Each of us gained hands on experience and programming skills greatly. The project addressed all of the learning outcomes. It required us to do extensive testing and troubleshooting any errors to ensure that we got the desired outcomes. For example, at the start, we did not test our HTML pages, as a result the pages appeared off centered and unprofessional on different screen sizes.

For JavaScript we use conditional statements such as if statement to ensure specific requirements were met. For instance, the password feature required users to create passwords between 8 and 15 characters, so we implemented an if statements to enforce this condition. Other if-else statements combined with try-catch blocks ensured the website functioned correctly and handled errors correctly. We used **RESTful API** and **Gemini API** mainly for our project. We used JSON as a makeshift database. These technologies gave us deep insight on client-server applications.

Writing programs using a given software was also a key learning outcome. All thanks to our prior lab work, we were able to easily set up the basic HTML and CSS structure, which allowed us to efficiently kick start the project. All out files followed optimal syntax, which to be really honest saved us a lot of time. Usage of correct syntax and comments made the code easy to understand. We also ensured that the code followed the principle of separation of concerns. All our code was modular. No embedded styling or embedded scripts. All code was also divided nicely into meaningful sections which made debugging a breeze. This made sure that each file had a single clear focus.

- ## Challenges:

One challenge that we faced was coordinating meetings between each of the group members since all of us had very busy and different schedules. To overcome this, however, we started using Google Meets since we could collaborate and share code remotely. Additionally, whenever we got time, we would discuss the code in labs and class and helped each other troubleshoot errors and give feedback on the code. We also decided to use Git for streamlining the process. Another challenge was that we were not taught all the relevant code structure and functions. The pacing of the course didn't match our project workload. To overcome this, we relied on resources like W3School and YouTube tutorials to understand and implement the necessary code.

Estimating the difficulty of project requirements was also challenging. Some tasks took much longer than expected, causing uneven workloads. To overcome this we made sure that if anyone needed help, we would help them just to make sure that everyone worked fairly and did their fair share of the work. Implementing the JavaScript code was particularly difficult since it required a lot of logical thinking and precise syntax to achieve the desired outcome. So, to overcome this we made sure to find tutorials on how to get the results that we wanted and we also consulted TA's for guidance. Our lab TA was particularly helpful.

- ## Use of AI:

AI is so closely linked to programming specifically these days that we hold the view not using AI would be foolish. Not adapting to technology can be catastrophic too. However, the downside obviously is the learning. The way forward is **responsible** use of AI. How'd we do that? We set boundaries. What to use AI for and what not to. To us, AI was just a

way to eliminate searching through YouTube videos and thousands of web resources to optimize our code and build up on it. It helped a lot with debugging too. We used Claude as our main chatbot- we found it to be the most detailed and accurate. AI helped us with responsiveness and server-side JS because we were introduced to RESTful API much later in the course and that was a big part of our project so AI helped us write the basic API calls. Some functionality we struggled in such as converting to pdf and AI API calls were programmed by AI but as a rule of thumb we used AI as a co-collaborator not just prompting it and copy pasting the code. Our take on AI was to use it as a learning tool that is more advanced than using say w3schools. One of the prompts I used “How do I call Gemini to generate a recipe? What prompt should I give the API?” Additionally, one of the preconditions we set before we started was to understand logically whatever AI wrote and to code similar tasks ourselves using what we learnt. AI helped us greatly take this project to the next level. Disclaimer, all the HTML was entirely us and most of the CSS was us. We only used AI for some minor debugging and optimization. The only part where we relied on AI was server-side JS and the test files.

P.S Linus Torvalds (yes, the linux guy) agrees with our take