CPSC 254 Project Documentation

Members: Alexander Nguyen, Krystal Phan, Jair De Orta

Project Title:

To-do List Web Application

Project Overview:

This project is a full-stack web application hosted online via Heroku. The project interfaces with Heroku's open source Postgresql database for data storage, utilizes Python's Django framework for backend logic, and utilizes the Bootstrap framework with HTML, CSS, and Javascript for the frontend. A description of the project's functionality can be found in the Project Description section below. This project contributes to our understanding of open source concepts because of its extensive use of the Django framework for python. The framework was used to simplify the process of creating the full-stack web application with its various built-in features like user authentication and database management. It also kept the project well-organized because of its standardized file structure. Github version control strengthened our understanding of collaborative software development and how multiple contributors can work together on a single project efficiently with the website's various mechanisms that facilitate cooperation.

Learning Objectives:

As mentioned in the project overview, this project's use of the open source python framework, Django, contributed to our understanding of open-source software and its importance in streamlining project development. Github version control showed us the importance of having efficient version control systems to avoid conflicts with group cooperation on projects. Discord was also used for video conferencing and screen sharing so that group members could efficiently communicate with each other without having to commute to a common location. An agile software development approach was taken with this project and various features/website changes were added regularly in an informal manner.

Project Description:

The purpose of this project is to give users an easy-to-use to-do list tool to help keep track of any important dates or activities they may have coming up. This is a full-stack web application hosted on Heroku that allows users to create, edit, and delete items on a to-do list. Users have the option to register accounts, log in, and log out. When creating tasks, users can input a title, description, as well as the due date. A to-do list item will be created in the database and displayed on the user's to-do list page with the creation time, title, description, due date, and completion status. User accounts are also stored on Heroku's Postgresql database. The Django framework for Python handles the backend logic as well as database management with its built-in Object-Relational Mapping (ORM) feature, streamlining database interactions. The Bootstrap framework for HTML, CSS, and Javascript streamlined the frontend development process and helped us create visually appealing and simple user interfaces. This project helped strengthen our understanding of full-stack web development and how data flows through the project backend, database, and frontend.

Technology Stack:

The front-end of the project was handled with HTML, CSS, and Javascript along with the Bootstrap framework. The backend and database management was handled with Python and the Django framework. Heroku, a linux-based cloud service, was used to host the web–application online and its Postgresql was used for the database. Git was used for version control.

- List the packages used. **LEFT OFF HERE. I WILL USE PIP FREEZE COMMAND TO OBTAIN A LIST OF PACKAGES LATER**

6. Server Infrastructure:

- Describe the setup and configuration of the Linux-based server/machine.

The set up...

- Explain the package management process, detailing the installation of web servers, databases, and programming languages.

- Include information on automation using shell scripts for server tasks.

7. Collaboration and Teamwork:

Collaboration: The team primarily utilized Discord for text and voice communication during the project. We used screen sharing to collaborate on each other's work and address code-related issues efficiently.

Divided Tasks: The team were divided to work on certain files and tasks.

Alex: Created the repositories on github, writing files (website, to-do list, date, home), documentation, google slides

Jair: Help with writing files (creating registration and log-ins), google slides, documentation

Krystal: Help writing files (edit option, date, registration), documentation, google slides

Challenges: - Highlight the challenges faced during collaboration and how your team addressed them.

8. User Interface (UI) and User Experience (UX) Design (For Web Applications):

- Provide screenshots or wireframes of the user interface.

- Explain the design principles and user experience considerations implemented in your Application.

9. Open Source Principles:

- Discuss your understanding of open source software and its core principles.

- Explain how your project incorporates open source concepts, such as sharing code, collaboration, and community involvement.

10. Agile Practices:

- Explain the Agile practices and methodologies your group employed during the development process.

- Discuss your daily stand-up meetings, highlighting the progress updates, challenges discussed, and resolutions made.

- Explain how you handled changes and adaptations in project requirements during the

development cycle if any.

- Reflect on the effectiveness of Agile practices in improving collaboration, communication, and overall project progress.

11. Challenges and Learning:

- Document any challenges faced during the project, especially those related to understanding open source principles and collaboration.
- Reflect on what you learned from these challenges and how they contributed to your growth as a software developer.

12. Group Collaboration:

- Describe how your group collaborated on the project.
- Explain the division of tasks, version control systems used, and communication methods employed within the group.

13. Git Repository and Project Links:

- Provide the link to your Git repository where the project code is hosted.
- Include any additional project-related links or resources such as references, blogs etc.

14. Conclusion:

- Summarize the key takeaways from the project.
- Reflect on how the project enhanced your understanding of open source, collaboration, Agile practices, and software development principles.

Submission Instructions:

Compile your documentation into a clear and concise report in PDF format. Focus on explaining your understanding of open source principles, collaboration experiences, Agile practices, and the learning outcomes. Include the Git repository link and any related resources. Submit your documentation by 8th December 2023 via Canvas.

If you have any questions or need clarification on the documentation requirements, please reach out to me via Email at snathani@fullerton.edu.