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.

Packages used:

1. Django: The web framework we used for the backend and frontend logic.
2. Django-heroku: Package used to allow our django project to interact with the Heroku cloud service for web deployment.
3. Gunicorn: Package used to connect our Django project to the webserver. It acts as an intermediary between the systems.

6. Server Infrastructure: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Set up: Setting up a Linux-based virtual machine on VirtualBox allows you to download Ubuntu and operate with multiple servers without the need to remove Windows entirely. Setting up VirtualBox and integrating Ubuntu into the virtual machine involves several sequential steps.

Steps: To download VirtualBox(https://www.virtualbox.org/), select the version compatible with your operating system; then, to download Ubuntu(https://ubuntu.com/download/desktop), then choose your preferred version, and download the ISO file. Make a bootable USB or DVD with the ISO file, start the server from the installation media, and follow the on-screen instructions to install the operating system. Once installed, update the system's package repository to ensure you have the most recent software and security updates.

- 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 was 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

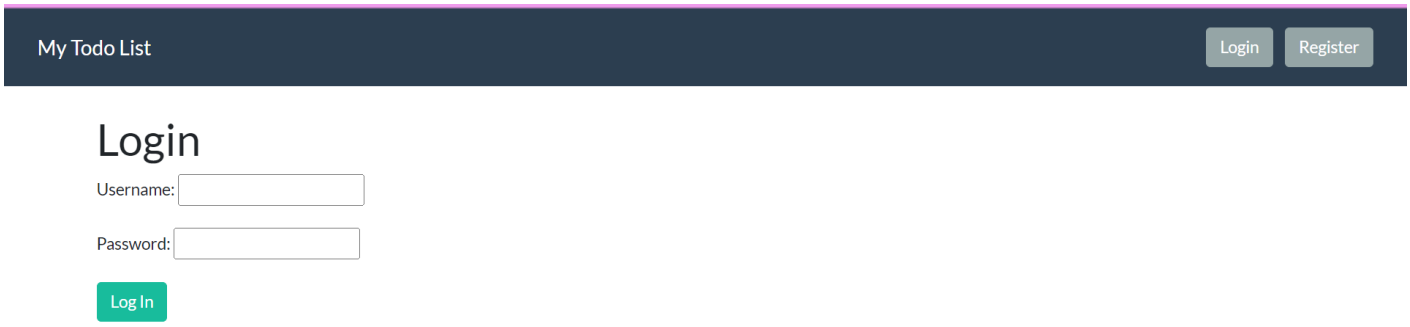
Jair: Help with writing files (creating registration and log-ins), google slides, documentation

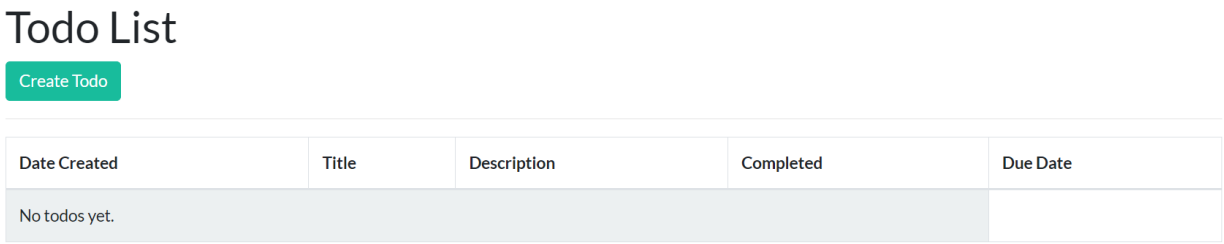
Krystal: Help writing files (edit option, date, registration), documentation, google slides

Challenges: Our team encountered numerous challenges during our collaboration. None of us had prior experience with GitHub collaborations, making the process of committing code or documents a considerable struggle. Managing versions proved to be problematic, with difficulties in overwriting codes for new versions and fully eliminating old versions. Additionally, committing posed challenges while working on Linux/Ubuntu, requiring us to navigate around system-related issues to successfully push and commit to GitHub.

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

- Screenshots:





User Interface: Using a template given from django, it is inherited to all other parts of the website application. We wanted to have a nice simple design because just like a todo list, it does not need much to stand out. The design was kept to appeal to anybody and not deter anyone or focus on any type of people. All the white space with a good blue color accent for the top bar to keep it clean for the user.

9. Open Source Principles: Open Source Software: Open source software is a development approach that prioritizes transparency, collaboration, and community involvement. Essentially, it entails making a software project's source

code freely accessible to the public, allowing users not just to utilize the software but also to inspect, modify, and share its code. The fundamental principles of open source encompass the free distribution of software, open access to source code, collaborative community efforts, the use of open licenses, a merit-based development model, continual improvement through iterative processes, adherence to open standards, security achieved via transparency, and empowering users. These principles have nurtured a global environment where diverse contributors work together to build, refine, and sustain high-quality software that is often available at no cost, fostering innovation, adaptability, and user empowerment.

Project incorporation: In the realm of open-source concepts, collaborative teamwork has become significantly more streamlined. With the convenience of platforms like GitHub, collaborating as a team has become more efficient, particularly with features like the option to push commits and share links. This makes the process of gathering and consolidating code much more straightforward. To say as a team, we were able to work on certain parts of the code and gathered up all together to push commits. Since we used Pycharm, it was much easier to retrieve small deletions or mistakes if files were backed up or saved.

10. Agile Practices:

Agile methods: Agile methods help projects progress better because they are flexible and adaptable. Teams can easily change task priorities, adjust timelines, and accommodate modifications, making the development process more responsive and resilient. Agile methods are known for their effectiveness in improving collaboration, communication, and the overall advancement of projects. It also supports clear visibility of project progress using tools like burndown charts and task boards. This transparency helps find and address problems early in the development cycle, making it easier to track progress and manage risks.

Meetups: The group was unfortunately unable to meet up in person, although working with pycharm and discord, we were able to work together without being physically present. A notable challenge arose due to our varying work schedules, making it difficult to find mutually available meeting times. To address this hurdle, we either convened late at night on Discord to complete tasks or individually finished portions in advance, committing them for the next meeting.

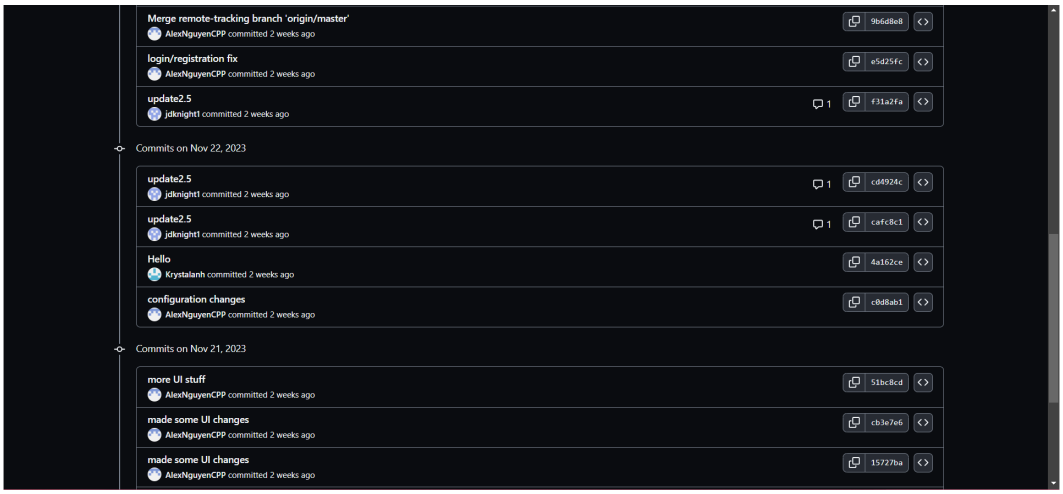
Agile practice: To say as a team, we were able to adjust our time and work without being physically present and able to gather everything once we met up. Communication was very crucial within our team as we needed to

acknowledge which part to work on and when we could commit together to run the code. Additionally, we collaborated using PyCharm and GitHub, which streamlined the process of sharing code instead of manually sending it. Given our inability to meet in person, Discord communication played a crucial role, providing options for screen sharing and chat at our convenience.

11. Challenges and Learning:

Challenge:

Github commit issues



Reflection: These challenges highlighted the importance of improving our communication within the team, emphasizing that it's acceptable to ask questions and seek clarification. Addressing the significant issue of GitHub commit challenges, we found a solution by designating Alex as the admin responsible for verifying commits made by Jair and Krystal. Additionally, in response to code overwriting problems, we recognized the need to consistently maintain backup codes to mitigate any commitment issues that may arise.

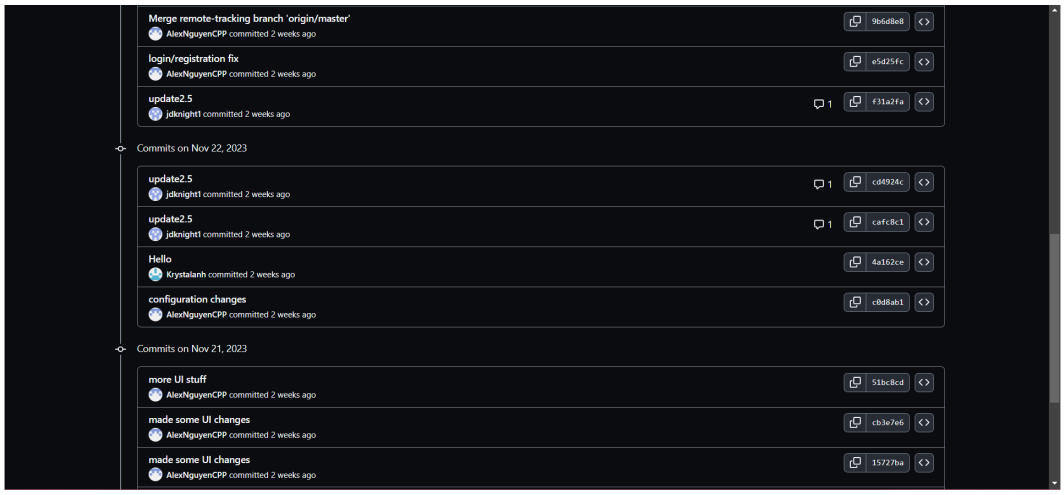
Challenge: Meet up time

Reflection: With the set up we had, we were able to get the code done within 4 days of a few hours of discord calling. It was a lot easier to meet up late at night since all of us work and schedules were complicated to work with. With the use of discord, we were able to communicate with each other whenever we wanted, able to ask questions, share screens in case of issues brought up without physically being there, and so much more. Although it was difficult to find time, we were able to compromise some of our time to work together. We also had times where we would work on parts of the code on our own and just push commits whenever we met up.

12. Group Collaboration:

Group collaboration: The team collaboration was effective as everyone actively participated by asking questions and responding promptly. There was a balanced effort from each member in terms of communication; no one ignored or intentionally delayed the progress of the project. Our primary communication platform is Discord, a tool commonly used by all team members. The project was completed swiftly because we efficiently communicated any issues on Discord and collaboratively resolved them.

Contribution:



Explanation of task division: Each team member installed the required software, and rather than having one person solely responsible for the code, we utilized PyCharm to collectively contribute to different sections. In summary, everyone downloaded PyCharm, utilized GitHub collaboration features, and had Linux installed. Each team member also tested the code and interacted with the website to identify and address any potential conflicts. Each member was able to communicate well as we talked often and had a good environment to work with. Throughout the project, we opted for Discord and voice calls instead of in-person meetings. This approach facilitated seamless communication, and any minor inconveniences were quickly resolved as the entire team collaborated effectively.

13. Git Repository and Project Links:

GitHub link: https://github.com/AlexNguyenCPP/TodoList.git

Website link: https://todolistproj-0226060662b2.herokuapp.com/accounts/login/?redirect_to=/

14. Conclusion:

Recap/Summarize: This project is a full-stack web app on Heroku, linked to its open-source Postgresql database. Python's Django manages backend tasks, while Bootstrap, HTML, CSS, and Javascript handle the frontend. Using Django improved our understanding of open source concepts by simplifying development with features like user authentication. Its organized file structure kept the project well-ordered. GitHub version control expanded our knowledge of collaborative software development, showing how contributors can work together effectively using the website's collaboration tools.

Conclusion Reflection: The project provided an opportunity for us to enhance our understanding skills as we shared and leveraged each other's knowledge. This project was a rich learning experience, encompassing the use of PyCharm, Linux, GitHub collaboration, communication, teamwork, and various other skills. The implementation of Agile practices facilitated a smooth project workflow with minimal conflicts in work ethics. Beyond coding simple websites, we gained valuable insights from our mistakes, such as avoiding code overwrites and managing different versions effectively.

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.