Project Documentation Via Crowd-Sourcing with ChatGPT

Team members:
Valin Vigliotti, vvigliotti2022@my.fit.edu;
Jonathan Nazario Rosado, jnazariorosa2022@my.fit.edu;
Hashiim Mohammed Sheriff Sathick Batcha,
hmohammedshe2021@my.fit.edu; Braden Corkum,
corkumb2013@my.fit.edu

Advisor: Marius Silaghi, msilaghi@fit.edu
Client: Marius Silaghi - FIT Faculty

Task	**Completion**	Valin Vigliotti	Jonathan N. R.	Hashim M. S. S. B.	Braden Corkum	To do
Compare and select technical tools for website and database	100%	25%	25%	25%	25%	none
"hello world" demos	100%	25%	25%	25%	25%	none
Resolve technical challenges	33%	8%	8%	8%	8%	Need to figure out GitHub integration and collaboration filtering
Compare and select collaboration tools	100%	25%	25%	25%	25%	none
Requirements Document	100%	25%	25%	25%	25%	none

Design Document	100%	0%	50%	0%	50%	
Test Plan	100%	50%	0%	50%	0%	none

Discussion of Matrix 1 Tasks:

Compare and Contrast Tools:

Our two main tools we needed to compare and contrast were between Angular and React JS for our web application and also for our database which compared MongoDB to others such as SQL. For our Web App, we chose React due to a variety of different reasons. Although our advisor was more familiar with Angular, React was more beginner friendly with simple syntax, more flexible, and is more commonly associated with smaller applications. As for our database, we were most familiar with MongoDB, so we chose it since it would be easier to use.

"Hello World" Demos:

The goal was to create "hello world" programs by hosting a local website and seeing if the string was successfully outputted. This was done individually for both the React and Angular frameworks. The purpose was not necessarily to decide on a preference based on the result, but instead just to guarantee that the tool functions appropriately. Eventually, React was chosen out of the two frameworks.

Resolve Technical Challenges:

We resolved the technical challenge of picking and learning React. Each group member individually researched the tool and made a hello world program. The other technical challenge of including GitHub integration has not been solved yet because there is no foundation to work with. Moreover, the final technical challenge that required handling undesirable contributions has not been solved for the same reason. However, both unsolved problems are incredibly common in the world of web development so we expect to find solutions already available and greatly documented when we cross that bridge.

Compare and Select Collaboration Tools:

Our two main collaboration tools that we picked for FlowDoc are Discord and WhatsApp. As a team, we are most familiar with discord and since we all had an account, it would be easy to create a team. Our advisor is most familiar with Whatsapp so we created a group chat to easily communicate with him. Therefore, we will communicate through discord for smaller problems that don’t require our advisor and communicate through Whatsapp to communicate for problems that require our advisors input.

Requirements Document:

The Requirements Document was done ahead of the design document and the test plan by the whole team. Afterwards, the test and design documents were derived from the requirements listed in the document. It includes the functional requirements which specify the system's behavior. Secondly, it includes interface requirements made up of user, hardware, software, and communication interfaces. Lastly, it includes performance requirements that describe specific quantitative requirements for the system that can be directly tested.

Design Document:

Creating the Design Document required creating the System's Architecture, a GUI mock-up, and a database ER diagram. We used the tool Figma to do the website's mock-up and showcased multiple features of the system. The ER diagram was created using the Mermaid Live Editor browser and the result in the environment was screenshotted. Finally, the System's Architecture diagram was modeled using draw.io and was screenshotted.

Test Plan:

Creating the Test Plan for our project presented a few challenges. When creating test cases, we needed to understand every feature within the project as well as what different inputs the user could use and how the system would function for each input. This requires the knowledge of all the functions the project has. For example we need to test scenarios like overloading, logins, creations, edits, and privacy settings. Without an in-depth test plan, we may overlook important aspects in the creation of the project that could cause future problems

Discussion of Contributions

Valin Vigliotti:

Contributed to the SRS document in sections 2.3 - 2.5 regarding user characteristics, constraints, and assumptions and dependencies.. Worked on the Test Document, completing the purpose and scope, as well as the test cases from 4.1 to 4.6. This included the behaviors and functions of many of the project's key features such as user login and ChatGPT integration. I also helped contribute to the MileStone 1 Report, discussing our previous tasks and the challenges we overcame.

Jonathan N. R.:

Compared and contrasted React and Angular. Made a "hello world" program for both. Agreed with the group to choose React as the framework to use for the website. Contributed to shared tools by creating a google drive to house all the documents together. Contributed the interface requirements in the SRS document and sections 2.1 - 2.2. Contributed the ER diagram (for the database) and GUI mock-up sections of the design document.

Hashim M. S. S. B.:

Conducted research on ReactJS and Angular to determine the best framework for our project, analyzing performance, scalability, and ease of use. Contributed to sections 1.3 to 1.5 of the SRS document. Additionally, refined and formatted the functional requirements, also enhanced performance requirements. Moreover, contributed to the test documentation by including test objective and enhanced Features/Behaviours and test cases

Braden Corkum:

Did research on React and Angular. Wrote the introduction of the requirements document determined the performance requirements. Designed the architecture for the application and drew up the architecture diagram.

Milestone 2 Matrix:

Tasks (Creating Database)	Valin Vigliotti	Jonathan N. R.	Hashim M. S. S. B.	Braden Corkum
Website Foundation	25%	25%	25%	25%
Login Backend	25%	25%	25%	25%
Project Page Backend	25%	25%	25%	25%
Edit History Backend	25%	25%	25%	25%
User History Backend	25%	25%	25%	25%
Top Contributors Backend	25%	25%	25%	25%

Discussion of Matrix 2 Tasks:

Website Foundation:

Implement the website foundation by acquiring a domain, test if people can access the website through the corresponding link, demo the feature by putting a simple "hello world" message on the website that a user will see

Login Backend:

Implement the login backend into the database, test if a user's username and password both function correctly and do not leak the password, and demo the feature by successfully creating a new account username/password and logging into a dummy program all while using the database.

Project Page Backend:

Implement the website project page backend into the database, test if select users can access the page and view/edit the project documentation information, and demo the feature by outputting a sample project page onto a cmd or something if the user has access.

Edit History Backend:

Implement the algorithm for saving editing history into the database, test the feature by editing a sample project page and checking that the edit history is correct, and demo the feature by outputting the edit history of a dummy user onto a cmd or something.

User History Backend:

Implement the algorithm for tracking a user's contribution data, test the feature by having a dummy user contribute to a simple project page and checking if the number of hours spent contributing, the different project types contributed to, and the average coverage that contributions have are correct, and demo the feature by having a dummy user contribute and outputting a report of the contributions

Top Contributors Backend:

Implement the features to track the highest contributors into the database, and test if the system correctly identifies the users based on total number of contributions and hours logged. Demonstrate by using dummy user data and it should give the output of the correctly calculated and ranked leaderboard from the database onto the interface

Client Feedback

See Faculty Advisor Feedback below

Date(s) of Meeting(s)

- 02/07/2025 from 2:00pm - 3:45pm

Faculty Advisor Feedback on Each Task for the Current Milestone

placeholder

Faculty Advisor Signature: ______________________________ Date: ________

Evaluation by Faculty Advisor

Valin Vigliotti	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Jonathan N. R.	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Hashim M. S. S. B.	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Braden Corkum	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: ________________________________ Date: __________