Design Document 3: Project Plan

3.1 Project Management/Tracking Procedures

Our project management style will be a combination of agile and waterfall. Our style will be very similar to agile in that we prioritize flexibility, constant collaboration with each other, and breaking down our project into iterative tasks that will be completed step by step. We also need to be able to change our project if any changes are required by our advisor, adding to the flexibility of the project. Our style is different because we cannot meet as consistently as we should for a proper agile management style since we only meet once a week with each other and monthly/as needed with our advisor.

Our project utilizes Git as our code repository, so we have decided it would be easiest to use Git for our project management board. We have five columns: Backlog, To Do, In Progress, Stalled, and Closed. Our backlog is for our tasks that need to be eventually worked on but are not an immediate priority. The to-do column is for tasks that must be worked on shortly to continue our project effectively. The in-progress column is for tasks currently being worked on. Stalled is for tasks waiting on another task/team member to continue working on it. Finally, the completed column is for our completed tasks. As a team, we will update this board consistently to reflect our progress on the tasks we are working on and reflect our overall progress on the project.

3.2 Task Decomposition

For the backend, multiple tasks and subtasks are needed.

First, we need to set up a basic database that can store user/admin data. To do this, we need to figure out how the WP database works if any plugins will be necessary, and how queries are done internally. We also need to set up another database for the LLM that the AI can use to pull answers from, and it needs to have interconnectivity with other aspects of the website. The first basic database will then need to be connected to the front-end login, and make sure that the ambassadors have the correct permissions to do what they need to do. For the LLM, we will need to figure out how much data to store and how the app will access it, ensuring only the right people have access.

The backend will also have to handle the authentication and validation of ambassadors trying to access user questions. Potentially using MFA.

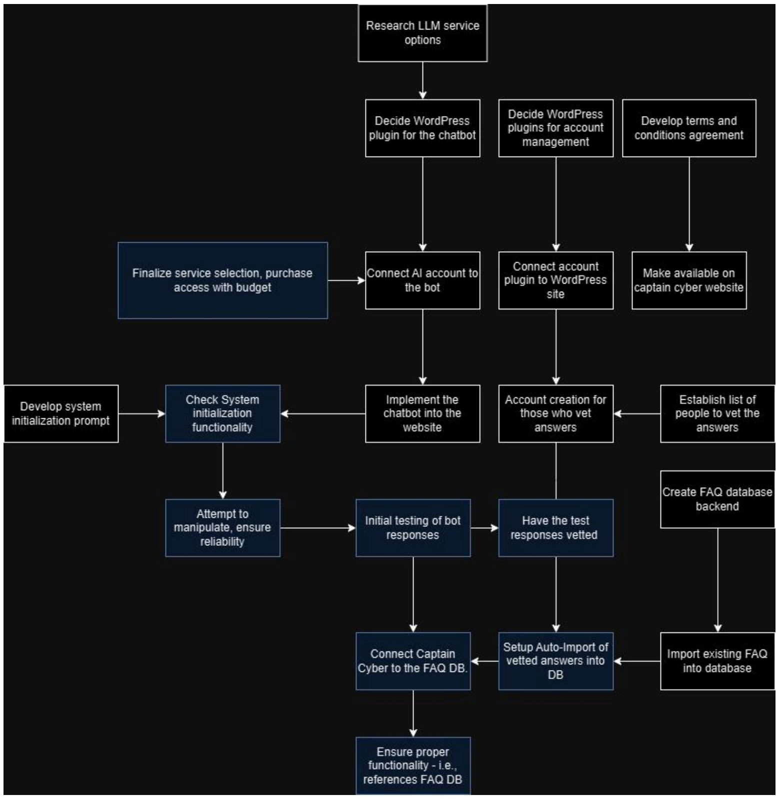

Research LLM service options
Decide WordPress plugin for the chatbot
Decide WordPress plugins for account management
Develop terms and conditions agreement
Finalize service selection, purchase access with budget
Connect AI account to the bot
Connect account plugin to WordPress site
Make available on captain cyber website
Develop system initialization prompt
Check System initialization functionality
Implement the chatbot into the website
Account creation for those who vet answers
Establish list of people to vet the answers
Create FAQ database backend
Attempt to manipulate, ensure reliability
Initial testing of bot responses
Have the test responses vetted
Connect Captain Cyber to the FAQ DB.
Setup Auto-Import of vetted answers into DB
Import existing FAQ into database
Ensure proper functionality - i.e., references FAQ DB

Figure 1: Task decomposition chart

The next step the backend has to worry about is the AI and how the AI API will be integrated with the UI. The AI will need to take user input -> go to either the LLM->If that does not work, go to the internet-> then deliver the response to the user within a responsible amount of time. This will require good networking code and fast algorithms. We will also need to think about parallel processing. If multiple users are using the bot, it needs a way to know which user to deliver the answer to and still run fast. To do this more research will need to be done if this will be a plugin or a feature built in JavaScript. The backend will also have to identify when it is appropriate to store user questions. This could be done manually, or we could make an automated process to store user questions. The backend will also need to have some way to update the information to make sure it is up to date, but this would be a future feature.

For the front end of the website, we will be able to build the screens and then implement the backend calls through the WP React API. This will allow us to make Rest API calls to the WordPress Database from a React frontend. With this, we have varying tasks that must be completed.

First, once a user enters the chat view, we must establish a secure and stable connection to make the conversation quick and seamless. Since we will rely on ready-made answers that have been vetted already, we can quickly query pre-formatted answers. If the question has yet to be answered we will then pivot to AI answers and make it known they have yet to be vetted by an expert and offer a notification once it has been. As this happens, we will populate the chat screen to emulate the ideal experience for the user.

We will also have to build an expert dashboard where they can view, collaborate, and vet answers. This will require a secure login authorization service. Ideally, we can use a pre-made service such as Okta's Autho service, or we can use the ISU login service with Okta. In this dashboard, we will make an intuitive screen that lets expert ambassadors easily interact with the list of questions.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

There are various milestones of this project, largely consisting of:

- Webpage implementation completion - All Ask Captain Cyber web pages are up and running, with complete functionality and React implementation.
- LLM FAQ References - The LLM has proven that it can pull accurate information from the FAQ and recognizes when a prompt is unrelated to any FAQ content.
- Expert Vetting Process - The LLM-generated responses can properly flow through the expert vetting stages and make their way to the user.
- Complete Backend integration and any API calls working properly, which will be tested via Postman.
- AI is fully functional with stress tests with multiple users.

Our evaluation criteria will consist of 3 aspects.

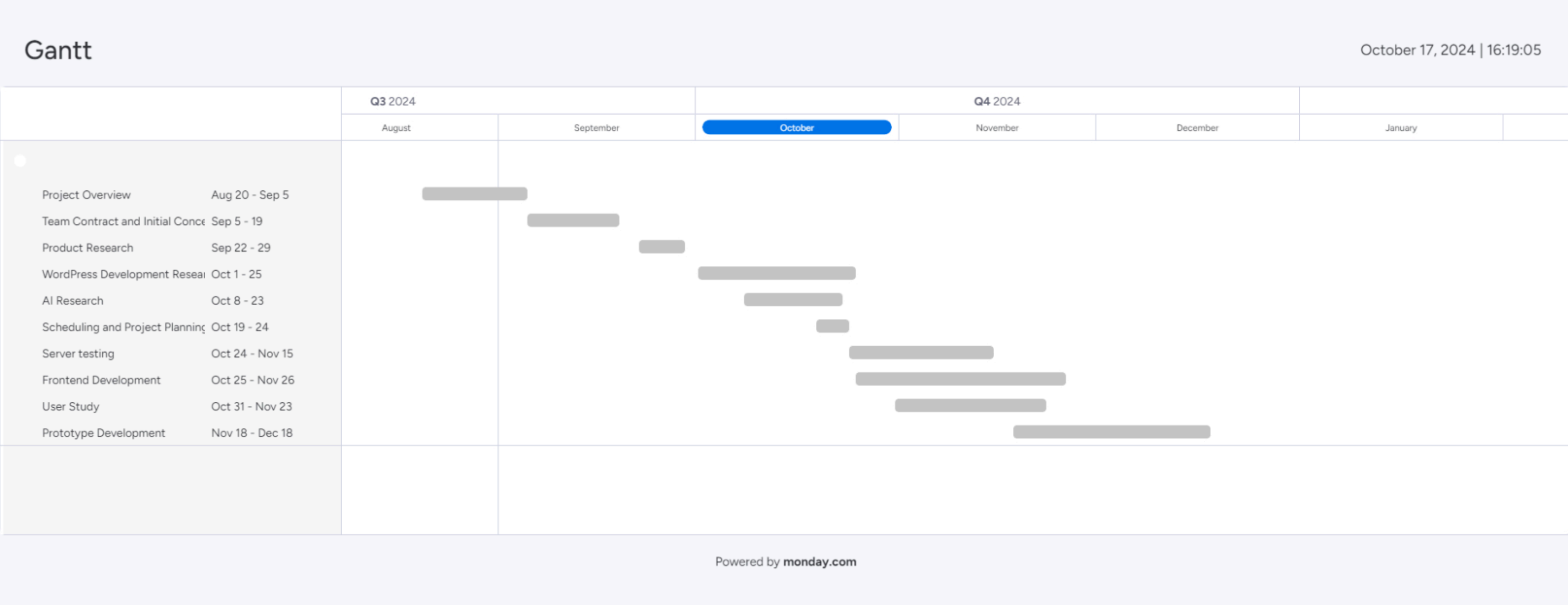
- Intuitive UI implementation - This will be evaluated via a Usability test of our UI.
- LLM FAQ References - The LLM can reference and learn from the dynamic FAQ.
- Generation Accuracy - The LLM abides by its initialization prompt restrictions and does not stray from discussing solely cybersecurity-related prompts.

For the aforementioned criteria, the metrics used to evaluate our UI will come from a usability study. We will have various users rate different aspects of the UI out of 10, with this milestone being reached when the cumulative UI evaluation reaches a rating of 7/10.

For the LLM FAQ references, we require that Ask Captain Cyber scans the database for relevant information upon every prompt. We will consider this milestone achieved when the LLM generates 80% of FAQ-related questions and does not have to be vetted by experts.

The metric used to evaluate the generation accuracy is that the LLM generates relevant information 80% of the time. This will be determined by human consideration of the relation between the user prompt and the generated response. This will also encapsulate the experts' responses to ensure they provide relevant information to the user.

3.4 Project Timeline/Schedule



Project Overview	Aug 20 - Sep 5
Team Contract and Initial Conce	Sep 5 - 19
Product Research	Sep 22 - 29
WordPress Development Resear	Oct 1 - 25
AI Research	Oct 8 - 23
Scheduling and Project Planninç	Oct 19 - 24
Server testing	Oct 24 - Nov 15
Frontend Development	Oct 25 - Nov 26
User Study	Oct 31 - Nov 23
Prototype Development	Nov 18 - Dec 18

Figure 2: Gantt Chart

The current phase of development that we just finished was product research. This is where we investigated what tools we were going to use, how we were going to structure this project, and general design work. We finished this task, and we are currently working on server testing. This testing ensures the server is up and running, and we can actively connect and start working on development. We need to ensure everyone has a local instance of WP on their device to start development. We also need to see what features the website already has.

We are also working on the initial prototype frontend and backend, which will go through November and December. Front end development will involve Login/Signup, Chatbot page, FAQ, and a backend vetting page for admins. There also needs to be some initial work done to figure out how to connect to the backend.

Backend work is being done to set up the user and LLM database, work on integrating API calls, and general security that will be needed. Integrating the LLM, vetting questions, and the front end will be the most challenging part of the project. This might take longer than usual, but we want to get some rough initial work done. We hope to get a rough setup/plan by the end of November.

Finally we will spend the rest of the semester from November 18th working on getting a final prototype working. This includes all pages being up, AI/LLM partially working (answering questions), relatively fast speeds, and good security practices in place that are up to ISU standards. Once Git is appropriately set, we will assign issues for team members to work on.

3.5 Risks And Risk Management/Mitigation

Project Overview

Risk: Poorly stated project requirements will lead to poorly configured systems down the line.

Probability: 0.4

Severity: Moderate

Mitigation: Keep open contact with how individual portions of the project are going. Keeping everyone on the same page will prevent any communication-based misconfigurations.

Team Contract

Risk: A team contract that doesn't cover a wide range of possibilities will lead to teammates abusing loopholes.

Probability: 0.3

Severity: Low

Mitigation: Make clear rules that can't be misused. Be professional.

Product Research

Risk: Researching can consume a large portion of time.

Probability: 0.5

Severity: High

Mitigation: Set time limits and make sure people stay on target with what research is needed and what shouldn't be researched.

WordPress Development

Risk: Unfixable bugs could lead to complete code scraps.

Probability: 0.6

Severity: High

Mitigation: Debug as you code, separating parts, allowing coders to focus and find bugs quicker.

AI Research

Risk: Filling the database might require too much work for the total time we can spend on the project.

Probability: 0.7

Severity: High

Mitigation: Only fill the database with enough data to test the project.

Server Testing

Risk: Achieving high performance on the server will take a significant amount of time.

Probability: 0.5

Severity: Moderate

Mitigation: Make sure we have enough performance for the essential operation of the project. Speed is out of scope.

Frontend Development

Risk: Some user interaction features may not consistently work

Probability: 0.2

Severity: Moderate

Mitigation: We will rigorously stress test all user interaction features alongside the usability test

User Study

Risk: Low user participation will lead to poor test case coverage.

Probability: 0.4

Severity: Moderate

Mitigation: Work with the student body and client to provide reasons for users to use the platform to increase traffic.

Prototype Development

Risk: We won't have enough testers, or it will go poorly

Probability: 0.4

Severity: Moderate

Mitigation: Work with Doug to ensure we have enough testers and stay on schedule for our tasks.

3.6 Personnel Effort Requirements

Table with the amount of hours contributed by each person on each task.

Task->	Project Overview	Team contract	Product research	WordPress Development	AI Research	Server Testing	Frontend Development	User Study	Prototype Development
Casper	6	3	8	6	4	5	5	4	3
Ethan	7	3	5	6	7	5	4	4	3
Steven	5	3	7	6	4	5	5	5	4
Alex E.	6	5	5	5	5	5	6	4	3
Alek K.	5	4	6	6	6	4	5	4	4
Caden	6	4	6	5	6	4	4	5	4

3.7 Other Resource Requirements

As our project is a website hosted on Iowa State servers, we won't need any physical parts or materials. We will rely on the expertise of our team to get the project where it needs to be. We have worked tirelessly to learn about proper backend and frontend practices. These resources will pay off as we continue developing to ensure that Ask Captain Cyber works as intended. That said, we will use several resources, such as libraries and API, to bring Ask Captain Cyber to its full capability.

The main thing that we will need to request is the licensing to the Co-Pilot. This will require a payment to receive our keys when answering questions. We also will be utilizing WordPress to host which will require constant power and connection on the ISU servers, enabling our backend engineers to develop plugins to handle all of the events and queries from Ask Captain Cyber. On the front end, we will require libraries to style the website intuitively. We will likely use a third-party multifactor authentication system to log in, or we can develop our own if time and resources allow. Overall, our required resources, parts, and materials are relatively low. We must use our education and expertise to use our resources to their full potential.

The final resource is that Doug and the ISU department will need to gather Cyber Ambassadors to test the website and vet answers. This is more of an issue that regards our advisor and is out of scope for our team.