Milestone Progress Evaluation: Milestone 1

1. Project Title: Crescent Social Guardian

2. Team Names and Emails:

- Adrianna Sauro, asauro2022@my.fit.edu
- Bhavik Patel, patelb2024@my.fit.edu
- Michael Wahoski, mwahoski2020@my.fit.edu
- Nicholas Blaisdell, nblaisdell2024@my.fit.edu

3. Faculty Advisor and Client: Dr. King, michaelking@fit.edu

4. Progress of current Milestone:

Task	Completion %	Adrianna Sauro	Bhavik Patel	Nicholas Blaisdell	Michael Wahoski	To do
1. Investigate Tools	100%	45%	18%	17%	20%	none
2. Small Demos	100%	20%	30%	50%	0%	none
3. Requirement Document	100%	25%	20%	15%	40%	none
4. Design Document	100%	40%	20%	15%	25%	none
5. Test Plan	100%	30%	15%	15%	40%	none
6. Implement, test & demo transformer model	5%	40%	60%	0%	0%	Finish implementation, then start testing and demo
7. Implement, test & demo extension	10%	0%	0%	0%	100%	Finish implementation, then start testing and demo

5. Discussion of each accomplished task (and obstacles) for the current Milestone: (at least a paragraph)

- Task 1:

  Current tools for the system have been researched and implemented into the project plan. The majority of the team has not done research projects before so an obstacle was getting everyone on the same page regarding research expectations. Furthermore, our tools interact with each other so another obstacle included making sure software engineer majors and computer science majors understood each other's tools and how they work to ensure seamless end to end functionality. Both obstacles were overcome by in-person meetings, drawing diagrams for each other, and discussions with our faculty advisor.

- Task 2:

  Small demos were created to show how our chosen tools work and that we know how to use them. An obstacle here was that we have not truly started building due to documentation requiring the majority of our focus and time dedicated to this project. We overcame this by adding extra meeting times dedicated to this task. We also struggled a bit with understanding what type of demo we were executing. We overcame this obstacle by discussing expectations with our instructor, semester 2 students, and our faculty advisor.

- Task 3:

  This task required us to identify and organize the system's architecture along with functional and nonfunctional requirements. The primary obstacle was determining the system's complete end-to-end functionality. Early in the project, the team divided responsibilities between the CSE and SWE students before everyone had developed the same overall understanding of how the components would connect. As a result, individual team members understood their assigned subsystem, but there was a disconnect between the browser extension, backend, machine-learning model, database, interpretation tools, and web application. This issue became especially apparent during a meeting with our faculty advisor, when one team member was asked to explain the complete flow of the system from chat collection through user notification. We realized that we could explain our individual sections, but we could not yet clearly explain how every part of the system worked together from beginning to end.

  This process directly influenced how we approached the remaining project documents. Rather than separating work immediately, we decided that the full team must first discuss each deliverable together until every member understands the system's end-to-end functionality. After reaching that shared understanding, team members can divide the writing or development work while maintaining consistent connections between their assigned components.

- Task 4:

  This task required us to describe the system's overall architecture, technical components, data flow, and end-to-end functionality, including how chat data moves through the system for analysis, classification, storage, reporting, and parent notification. We had an advantage in completing this task because our faculty advisor had previously asked us to create diagrams similar to those required for the design document. As a result, several of our diagrams were already nearly complete and only needed revision or additional detail to fit the final document. A major obstacle was reaching agreement on the system's full end-to-end functionality. Team members sometimes had different views regarding which features, processes, and technical components were most important or necessary for the project. Although these discussions were time-consuming, they were beneficial because they helped us clarify the project's priorities, define required functionality, and make more intentional design decisions. We overcame this obstacle by scheduling additional meetings and extending meeting times to allow for thorough discussion and debate. This gave the team enough time to resolve disagreements, finalize the necessary system-design decisions, and complete the document.
- Task 5:

  This task requires us to outline how the project and its individual components will be evaluated to ensure that the system functions as intended. This includes testing the browser extension's ability to collect authorized chat logs, verifying that the data adapter formats conversation data correctly for the model, evaluating the model's preprocessing and contextual analysis, confirming that the classifier produces the appropriate risk categories, and testing the database, notification system, and website case-management features. The testing process also considers whether the system can correctly identify concerning conversations while minimizing inaccurate classifications or unnecessary alerts. A major challenge was deciding how detailed the testing process needed to be and determining which features should be treated as essential for the first version of the project. These discussions made the document longer than originally intended, but they were valuable because they helped the team better define the project's scope and recognize the complexity of building a complete online-safety platform. By discussing possible system failures, expected results, and testing requirements, the team gained a clearer understanding of the scale of the project and the need to prioritize core features for the initial implementation.

- Task 6:

  Task 6 is implementing, testing, and preparing a demonstration of the transformer model for CRESCENT Social Guardian. This task focuses on developing a core component of the project, which will analyze chat windows for contextual patterns and assign risk classifications. A major obstacle was that the team needed to devote substantial time to completing the documentation before beginning full implementation. Although this limited the time available to start building the model, the documentation was necessary because it established the system architecture, data flow, component responsibilities, and functional requirements that guide implementation. To address this challenge, one computer science student focused more heavily on completing the documentation while the other began initial development work. This division of responsibilities was somewhat helpful; however, some implementation decisions still had to wait until design questions were discussed and resolved. The team is currently in the implementation phase of this task. Because the transformer model is one of the most important components of the project, we recognize that this work requires significant time, testing, and careful development. The software-engineering components make the model's output understandable and accessible to users, but the transformer-based classification system is the foundation of CRESCENT's ability to identify potential online safety concerns. Therefore, the team believes that taking additional time to finalize the design before proceeding with implementation was necessary and will help prevent avoidable development errors and rework later in the project.
- Task 7:

  This task is to develop the initial build of the extension. This includes finishing research into the build process and testing process. The extension will be difficult to build due to no true system pipeline developed for an input and output.. This challenge is further pushed due to the team not being well versed in DOM structure and scrapping. The team is finalizing research into how this can be accomplished and implemented. The team will address this knowledge gap with more research on how this works. To address the pipeline issue, the team currently believes Postman will solve most if not all the testing requirements for the input and output of chat services.

6. Discussion of contribution of each team member to the current Milestone: (at least a paragraph)

- Adrianna Sauro:
    - For Task 1, I conducted the majority of our tool-investigation research using Google Scholar and Elicit AI, identifying prior work in our topic area and helping the team base our decisions on established methods in existing literature. I found/decided on BERT, BART, Transformers Interpret, and Pytorch. Specific aspects of our project such as being contextual based and using sliding windows came from this literature as well.
    - For Tasks 3–5, I organized and formatted all written documents, proofread each submission before delivery, and delegated sections to team members including myself to try and ensure balanced workload and consistency. Throughout the milestone, I managed our timeline by setting goals at the beginning and end of each meeting, establishing internal deadlines ahead of official ones, and coordinating opportunities to receive feedback from our academic advisors. I also served as the primary point of communication between our team and our faculty advisor, ensuring alignment and clarity across all parties.
    - For Task 6 I worked on implementation with Bhavik.
- Bhavik Patel:
    - For Task 1, I did research on different BERT models and decided DistilBERT was our best option. For Task 2, I developed the initial demos for our preprocessing code and the initial demo for our transformer model. For Task 3, I helped write the requirements to build our DistilBERT model and our XGBoost Classifier model. For Task 4, I helped write the initial design concepts for our transformer model, including what activation functions we would pair with our BERT model and how we can test our model. For Task 5, I created the initial code for the model to show that it can accept input and create valid output.
- Nicholas Blaisdell:
    - For Task 1, I researched tools over the summer for the initial project proposal. Later research found the tool Interpreto to be very similar to Transformers Interpret, but Transformers Interpret was more simplistic. I researched how the tools would work together with BART, the summarization model. I ensured BART was able to handle the workload from the data, due to the token load of 1024. I researched the idea of the adapter, thinking of an initial way to be able to format messages universally into a specific format.
    - For Task 2, I am implementing small demos that are used to test the explainability layer, involving Transformers Interpret and BART. Captum,

being a part of Transformers Interpret, is being tested by constructing the attribution tensors and sending them to Transformers Interpret. Currently, I'm testing to see if the tensors could be used to properly format the list of tuples required to send to the formatting middleware. I am testing to see if the newly formatted list of tuples would properly send to BART.
    - For Task 3, I worked on the Non-Functional requirements and the Functional Requirements of the explainability layer and extension. Learning how to obtain chat logs through a Google Chrome extension was confusing initially, but learning about DOM Trees and nodes to be able to obtain data from HTML was very useful. Michael and I discussed the system architecture for the Interface Diagram, which had to be revised many times.
    - For Task 4, I worked on section 7 of the design document. I added the explainability layer to the requirements matrix. I also added some rationale for the small tools for the explainability layer.
    - For Task 5, I worked on the test cases for the Extension and the Explainability Layer.

- Michael Wahoski:
    - I had previous knowledge I pulled in along with researching current tools for accomplishing the system designing concepts and high level build understanding. For task one, I researched and pulled in some useful tools that we may be able to use since I knew testing was going to be a difficult roadblock. I also knew we were going to need a database for many of the pieces. We also needed a way to display results and explained to the team a web application is possibly the best route for controlling data views through role based authentication and authorization. For Task 3, I led the development of the document to include structure, reasoning, and implementation. It felt like it was a system engineering perspective that I wanted to learn more about and felt confident enough to start the process. I took the time to explain some of the reasons why this document was important and used it as a starting point for our project due to requirements being one of the first steps to starting. Task 4 I created most of the diagrams as well as filling in sections pertaining to extension, adapter, database, and web application. For Task 5, I implemented extension, adapter, and web application/ User interface information. Task 7 is building the extension.

7. Plan for the next Milestone

Task	Adrianna Sauro	Nicholas Blaisdell	Michael Wahoski	Bhavik Patel
1. Implement, test & demo classification model	50%	0%	0%	50%
2. Implement, test & demo extension and adapter	0%	50%	50%	0%

8. Discussion of each planned task for the next Milestone

- Task 1: For task 1 of the next milestone, we are going to implement, test, and demonstrate our XGBoost Classification model. In order to implement the model, we first need to write some code to restructure the output of the DistilBERT Transformer model to fit the input requirements of the XGBoost model. We will be attempting to use unsupervised learning, so there will be no need to specify the output other than labels. Once that is done, we can begin training the model to make accurate predictions from the data. During this phase, we need to ensure that the model does not overfit to our data or else we will get inaccurate results during the validation and testing phases. For our testing phase, we will take the preprocessed version of the pan-12 testing data and feed it into the transformer model to produce the output to be fed into the XGBoost model. From there, we will run the transformer model's output through the XGBoost preprocessing code to ensure that the preprocessing code is properly restructuring the output to fit the needs of the model. Once that is complete, we will run the restructured output through our XGBoost model to ensure that the model is providing the correct output for the incoming data.
- Task 2: For task 2 of the next milestone, we will implement, test, and demonstrate the extension and adapter. To work on the adapter, we must evaluate progress on the extension, ensuring the preconditions of the adapter are properly met. The test website will be built and tested using Postman to create mock chat logs that are inserted into the extension, and eventually feed to the adapter.

9. Dates of meetings with Client during the current milestone:

See Faculty Advisor meetings below

10. Client feedback on the current milestone:
    See Faculty Advisor Feedback below

11. Dates of meeting with Faculty Advisor during the current milestone:
    - Friday, September 11th, 2026: 12:00 - 12:50 P.M.
    - Friday, September 25th, 2026: 12:00 - 12:50 P.M.

12. Faculty Advisor feedback on each task for the current Milestone
    - Task 1: Jiv would be interesting to look into for future research
    - Task 2: Ensure you can access and compile tools
    - Task 3: Updating diagrams, ensure child and parent users are both involved. UI Diagrams should be set up in a way that is end-to-end and more understandable.
    - Task 4: Temporal visualization of how chats are developing would be beneficial towards understanding the entire chat's progression. Further visualization would be very beneficial for data analytics and parent/guardian interaction. Learn how long we've been talking to the user. A parent would want to be able to see that. Add visualization on the case card when we make the case.
    - Task 5: A chance for overfitting with XG Boost, a lot of statistical analysis will be necessary with this route. Understanding that not every message, just based on correlation, is predatory. The transformer would push the responsibility into XG Boost. How are we planning to account for the context window with a conversation that could possibly go for months? The main concern would be forgetting conversations. After analysis, it is a discretized model.
    - Task 6: Figure out appropriate ways to evaluate the model so the predatory phrases are property assessed, but not just deemed predatory due to correlation. Free cloud hosting like CloudFare would help. Want to build scalable, but not overly scaled. Could set up a local host, capture live data to see what's happening. This would show how the model is operating without the need of extremely careful planning.
    - Task 7: Look at network traffic for the extension to take into account. Ensure verified data for structure of websites. See if we can handle invisible element. When software changes, ensure the extension can handle the invisible changes. LLMs that run on the browser itself for instruction-tuned models that can perform very specific tasks. The version we have currently will need a lot of system analysis and fine-tuning. (look into where we may get access) Apple has a CSAM that should be looked into.

Faculty Advisor Signature: _____ Date: _____