1. **Project title:** BrainBench (ML/LLM Models Evaluation Dashboard)

Presentation: BrainBench
Website: BrainBench Dashboard

2. **Names and email addresses of team members**
    a. Orion Powers, opowers2023@my.fit.edu
    b. Daniella Seum, dseum2023@my.fit.edu

3. **Faculty advisor:** Dr. Slhoub, kslhoub@fit.edu

4. **Client:** Dr. Slhoub

5. **Date of Meeting with the Client for developing this Plan:** 1/26/2026

6. **Goal and motivation:**
   The goal of this project is to evaluate multiple large language models (LLMs) using a standardized and reproducible testing pipeline in order to provide users with a clearer understanding of their capabilities. Rather than relying on marketing claims or inconsistent benchmarks, this project focuses on comparing models using consistent metrics such as correctness, performance, and practical usability, with an emphasis on free or locally hosted models.

   Currently, there are few widely adopted, transparent systems for comparing LLMs under identical conditions, especially for users who are constrained by cost or interested in local deployment. By developing and applying a unified evaluation framework, this project aims to present meaningful, unbiased comparisons that help users more effectively select and use LLMs based on their specific needs.

7. **Approach:**
    a. ***Categorize LLMs based on reasoning strengths:*** Our system will evaluate and categorize each LLM based on its performance across selected reasoning domains, with a primary focus on mathematical reasoning and problem-solving accuracy. Using a standardized dataset and consistent prompting, each model will be tested on the same set of tasks to ensure fair comparison. This approach allows users to clearly see how different models perform under identical conditions, such as whether one model produces more consistently correct step-by-step solutions while another struggles with complex reasoning chains. These results help

users understand which models are best suited for specific types of reasoning-heavy tasks.

   b. ***Categorize LLMs based on performance and resource considerations:*** In addition to correctness, we will evaluate practical performance metrics such as response time, consistency, and computational requirements. For locally hosted models, this includes factors like inference speed and hardware demands, while for free or limited-access models, availability and usage constraints will be considered. Presenting these metrics in a normalized format provides users with insight into real-world usability and tradeoffs beyond raw accuracy.
   c. ***Hosted website with information clearly accessible:*** The results of the evaluation will be presented through a hosted website designed for clarity and ease of use. Users will be able to compare models across selected metrics and view summarized results without needing to interpret raw data. The site will emphasize clean visualizations and straightforward explanations, making the findings accessible to both technical users and those with limited prior experience working with LLMs.

8. **Novel features/functionalities:**
   While LLM benchmarks exist, our project introduces novelty by combining performance metrics, reasoning categorization, and cost analysis into a single, user-friendly platform. Most existing comparisons are either purely technical or marketing-driven (highlighting only vendor strengths).

9. **Algorithms and tools: potentially useful algorithms and software tools**
   a. ***Evaluation Frameworks:*** HuggingFace Transformers and custom Python scripts for standardized benchmarking.
   b. ***Web Development:*** HTML/CSS/JavaScript with a framework such as Angular.
   c. ***Visualization Tools:*** Go.js for interactive performance/cost visualizations.
   d. ***Collaboration & Documentation:*** GitHub for version control, Google Docs for reports,

10. **Technical Challenges:**
   a. ***Scalability and Repeatability of Experiments:*** As the number of evaluated models and test problems increases, maintaining consistent experimental conditions becomes more complex. Managing runtime, system resources, and configuration settings while

ensuring results remain reproducible across multiple runs and environments is a key challenge for the evaluation pipeline.

b. ***Frontend Data Integration and State Management:*** Integrating dynamically generated evaluation results into the website presents challenges related to data flow, state management, and synchronization between the backend evaluation pipeline and the frontend interface. Ensuring that results are displayed accurately, updated correctly as new tests are run, and remain responsive as data volume grows requires careful frontend architecture and design decisions.

c. ***Usability and Visualization of Complex Results:*** Presenting detailed evaluation data in a way that is both informative and intuitive is a significant challenge. The website must balance technical accuracy with clarity, ensuring that users can easily interpret model comparisons, performance trends, and tradeoffs without being overwhelmed by raw data or overly complex visualizations.

11. Design: system architecture diagram

a. Components/modules of the software system:

i. ***Dataset Module:*** Loads and validates the XML problem set, including question text, expected answers, and metadata (topic/difficulty if available).

ii. ***Prompting & Execution Module:*** Standardizes prompts and runs them through each LLM (supporting locally hosted models and any free-access options used).

iii. ***Parsing & Scoring Module:*** Extracts final answers from model outputs and determines correctness using consistent rules, handling format differences across models.

iv. ***Metrics & Logging Module:*** Records runtime, success/failure states, and summary statistics; supports repeatable experiment configurations.

v. ***Results Storage Module:*** Saves outputs and computed metrics in a structured form for later retrieval and display.

vi. ***Web Dashboard Module:*** Displays results through interactive pages and visualizations, allowing users to compare models and view metric definitions and methodology.

b. Interfacing with different types of users:

i. ***General Users (Viewers):*** Can browse the dashboard, compare models, and view summarized results and explanations of metrics without needing to run tests themselves.

ii. ***Project Team/Admin Users:*** Can run or re-run evaluations, update datasets, and publish refreshed results to the dashboard. This interface is primarily through developer tools (command-line scripts and repository workflows), with the website reflecting updated outputs after results are generated.

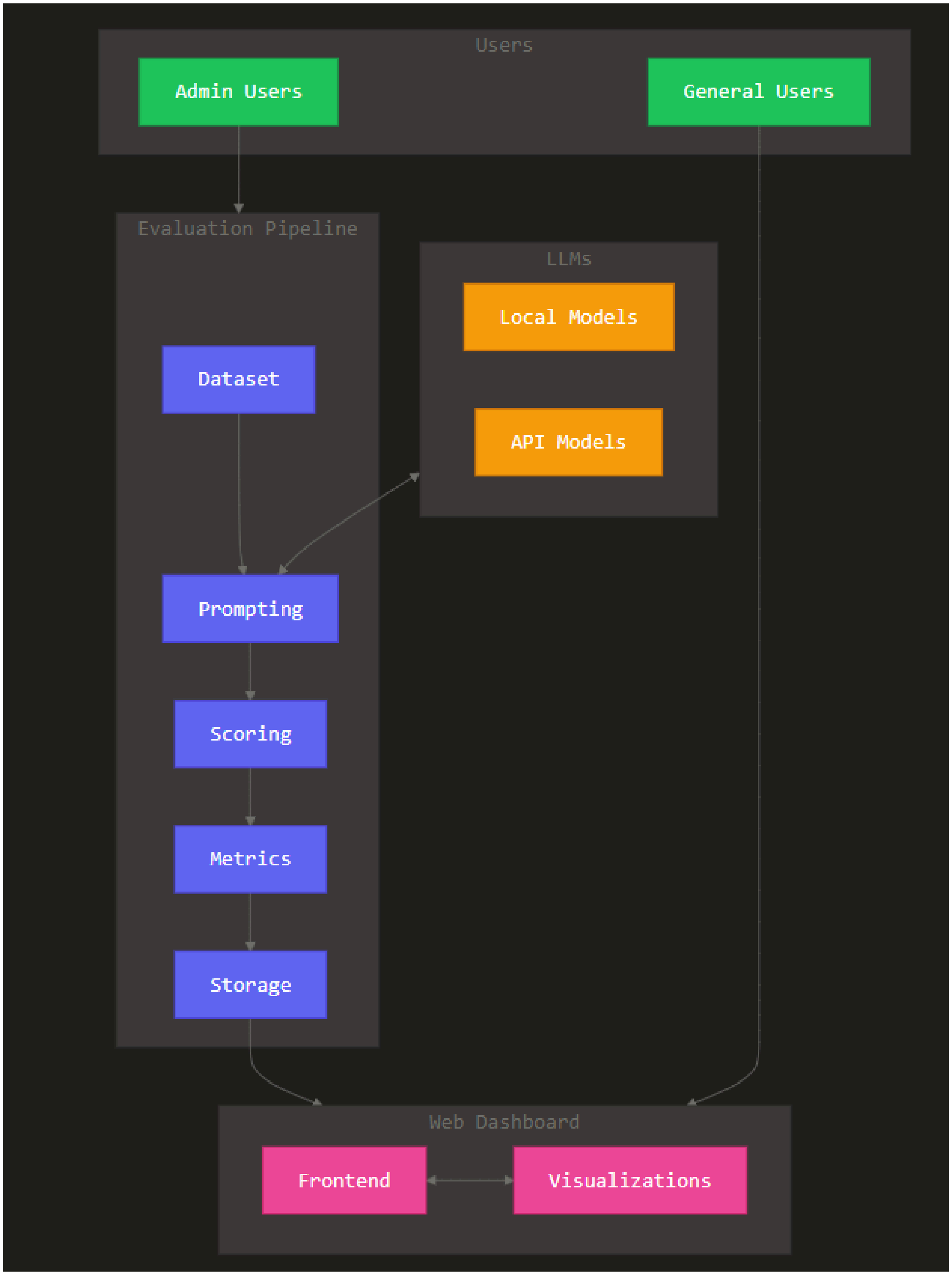
12. System Evaluation:

a. ***Speed:*** The system will be considered successful if it can complete a full evaluation run across all selected LLMs and test problems within a reasonable and repeatable time frame. Metrics such as average runtime per problem and total evaluation time per model will be recorded and used to compare system performance across runs.

b. ***Model Evaluation Accuracy:*** Success will be measured by the system's ability to accurately evaluate LLM responses against known correct answers. The correctness metric will be reported as a percentage of correctly solved problems per model. Consistent scoring across repeated runs using the same dataset and configuration will indicate accurate and stable evaluation behavior.

c. ***Reliability:*** The system will be evaluated on its ability to execute repeated benchmark runs without failure. Reliability will be measured by the proportion of successful, complete runs out of multiple attempts under identical conditions, including correct result generation and storage without crashes or data loss.

System architecture diagram:



13. Progress Summary:

Module	**Completion %**	**To-Do**
Dataset Module	100%	N/A
Prompting & Execution Module	100%	N/A
Parsing & Scoring Module	95%	Improve handling of edge cases in model outputs.
Metrics & Logging Module	85%	Extend metric collection (e.g., aggregated statistics), improve logging clarity, and finalize experiment configuration tracking.
Results Storage Module	75%	Finalize data format, support versioned results, and ensure integration with the web dashboard.
Web Dashboard Module	0%	Design, create, implement, and test the hosted website.

14. **Milestone 4 (Feb. 23rd): itemized tasks:**
    a. Run all datasets through all LLM's three times.
    b. Interpret correctness results across runs.
    c. Collect and interpret statistics of different runs and across different LLMs.
    d. Begin website development and integration.

15. **Milestone 5 (Mar 30th): itemized tasks:**
    a. Finalize website.
    b. Conduct overall system evaluation and analyze results.
    c. Create a poster for the Senior Design Showcase.

16. **Milestone 6 (Apr 20th): itemized tasks**:
    a. Finalize tests for all features.
    b. Test/demo of the entire system.
    c. Conduct overall system evaluation and analyze results.

d. Create a user/developer manual.
e. Create a demo video.

17. Task matrix for Milestone 4:

Task	Daniella	Orion
Run all datasets through all LLM's three times.	50%	50%
Interpret correctness results across runs.	50%	50%
Collect and interpret statistics of different runs and across different LLMs.	50%	50%
Begin website development and integration.	50%	50%

18. Description (at least a few sentences) of each planned task for Milestone 4:

a. ***Run all datasets through all LLMs three times:*** Each selected dataset will be executed against all chosen LLMs across three separate runs using identical configurations. Repeating the evaluations allows the team to capture variability in model behavior and system performance while ensuring that results are not influenced by system or runtime conditions.
b. ***Interpret correctness results across runs:*** Correctness outcomes from repeated runs will be analyzed to identify consistency and variance in model performance. This analysis will highlight whether models reliably produce correct results or show fluctuations across executions, providing a better assessment than single-run evaluation.
c. ***Collect and interpret statistics across runs and LLMs:*** Statistics such as average accuracy, variance, and runtime distributions will be computed across runs and models. These statistics will be used to compare overall model performance and stability, forming the basis for meaningful cross-model comparisons presented in the final results.
d. ***Begin website development and integration:*** Development of the web dashboard will begin, focusing on core layout, navigation, and integration

with the evaluation output data. Initial efforts will prioritize displaying summarized metrics and establishing a data pipeline between the evaluation system and the website to support ongoing result updates.

19. **Approval from Faculty Advisor:**

"I have discussed with the team and approved this project plan. I will evaluate the progress and assign a grade for each of the three milestones."
Signature: __Dr Khaled Slhoub_____________ Date: __01/26/2026____