Project Phase #1
A CLI for Trustworthy Model re-use

Milestone 1

As a Boilermaker pursuing academic excellence, we pledge to be honest and true in all that we do. Accountable together – We are Purdue.

(On group submissions, have each team member type their name).

Team Members

Name	Purdue Username	GitHub

Date:

GitHub Repository Link

Include a public GitHub repository link below:

Statement on GitHub tokens & Repository creation

Replace this line to include a statement that GitHub tokens and public repository has been created have been obtained.

Tool Selection and Preparation

List all the tools, communication methods and LLMs that the team plans to use below. The below table may include anything but must meet the requirements as listed below:

- *Toolset, component selection [linter? Git-hooks? CI? Logging library? API Library? CLI Parser?]*
- *Communication mechanism(s) [Slack? Teams? Email?]*
- *Testing Framework*
- *Use of LLMs in a **responsible** manner (CoPilot, Code Llama, etc.)*

You should include a justification of any components you choose to re-use – how will you decide whether they are trustworthy? Remember the above section only outlines a few categories of tools/software that the team can re-use. Based on the design of the system additional tools may be required.

Table 1: List of Tools and Components

Tools	Justification
Testing Framework:	…
LLM:	…
Communication Mechanism:	…
…	…

Team Contract

Include a detailed team contract that will outline all responsibilities and efforts that your team shall take to ensure that the project achieves all learning outcomes as outlined in the project information documents. The team contract may cover anything from meeting times, coding etiquette and minimum communication times to possible conflict resolution guidelines. Finally, include a statement after the table that states the team agrees to abide by all guidelines.

Category	Guidelines
Coding Style/Etiquette	
Communication	
Responsiveness	
Commitment to CI/PRs/Code Review	
Conflict Resolution	
Assigned Work Delivery Timeline	
Other	

Team Synchronous Meeting Times

List all the times that your team members will meet during the week. Prof. Davis recommends at least a (one) short mid-week sync and one end-of-week longer sync to ensure that your team stays on track with the milestones and weekly reports.

Project Requirements

Insert in every row the requirements that the client, ACME, has provided in the project information document. Each row must only include one requirement and the details associated with it. Make sure the details are outlined in bullet points.

Table 2: Requirements

Requirement	Details on given Requirement

Preliminary Design

Your preliminary design must give a detailed overview of how the team plans on operationalizing each metric by providing references to and justification for the method used. Additionally, the team must also provide a formula for the NetScore. Make sure the formula is included as an equation in the table below (If you do not know how to do this, search it up!).

Metric Operationalization

Metric	Operationalization
Size	
License	
Ramp Up Time	
Bus Factor	
Available Dataset And Code Score	
Dataset Quality	
Code Quality	
Performance Claims	
Net Score	

Design Planning

As outlined in the project information document, include two UML diagrams using ***Lucid Chart****.*

- *UML Activity Diagram to depict the activities performed by your system.*
- *Simplified UML Class Diagram to depict the critical entities in the system.*

The logical flow description must follow and refer to the two diagrams included. Feel free to add additional diagrams that can aid in your description of the system that you hope to build. (More thought and planning are always better than just winging it when you write code!).

The design planning must also include:

- *Explanation of the design of the "metrics" feature so that you can accommodate Sarah's projected need to add new metrics later. What logical flow and what entity structure (e.g., class hierarchy?) did you select to improve the modularity of this portion?*
- *Explanation of the design of the "handle URLs" feature so that you can accommodate URLs of all three categories. What logical flow and what entity structure (e.g., classes hierarchy?) Did you select to improve the modularity of this portion?*
- *Explanation on how the team plans on meeting rate limiting requirements for the GitHub/Hugging Face APIs.*

Validation

Include a detailed validation plan. This may include any diagrams that you wish to use on how to test your program. The validation plan must include how you plan to use the test framework and guidelines that you will follow that give you confidence that your software performs nominally under all conditions.

What is your plan to assess whether the delivered software satisfies Sarah's requirements? What behaviors will you check? What performance metrics (if any) will you apply? Additionally, how does the team plan on promoting the testability of individual components? and how consistent the team will be to handle errors across the project?

Milestones

Each milestone should list the necessary tasks, the expected owners of those tasks, the estimated time to complete it, and how success will be measured. Any communication requirements between tasks should be noted, e.g., "Jason and Tahani need to discuss the interface involved between task A and task B."

Task #	**Description**	**Duration (Hours)**	Team Member(s)
Milestone 2 – due. *Sept 14*			
…	…	…	…
Milestone 3 – due. *Sept 21*			
…	…	…	…
Milestone 4 (Delivery) – due. *Sept 28*			
…	…	…	…

References