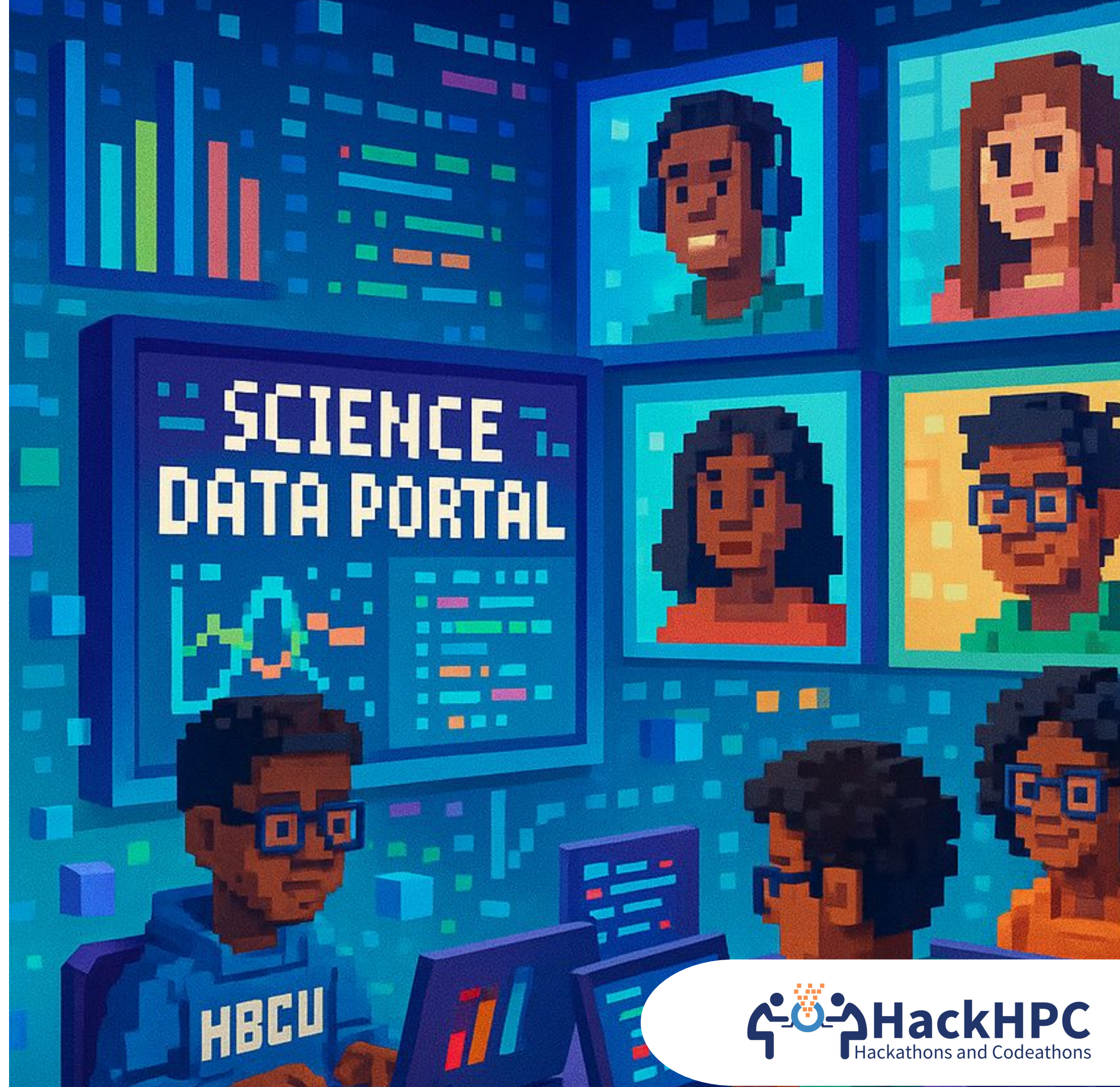


HACKHPC@ ADMI25

HACKATHON

Team Project Plans - v3

June 25, 2025





Wiresharks



Wiresharks



Team Members Name

- Auiana D'Avilar
- Ayinde Hooks
- Howard "Shiloh" Ames
- Ryan Grimes



Hands Up, Buttercup
Neon Beach

<https://app.soundstripe.com/songs/6425>

Paper Availability	Availability of Code and Software	Availability of Datasets	Computer Requirements	GPU Requirements	Documentation Quality	Ease of Setup	Reproducibility of Results	Rating
Is the paper open-access and freely downloadable? Is it behind a paywall, requiring a subscription or purchase? Or is it simply unavailable/difficult to find?	Is the code publicly available (e.g., on GitHub, GitLab)? Are there clear installation and execution instructions? Is it bundled with necessary scripts or is manual compilation/set up required?	Are datasets accessible, and is metadata provided?	What hardware and OS are needed (e.g., CPU, memory, OS compatibility, architecture)?	Are GPUs required, and what specs?	How clear and helpful the project's instructions and notes are. Easy-to-follow instructions mean good quality.	How simple it is to get the project working on your computer. If it's quick and smooth, it's easy.	Can someone else get the same answers from the project as the original creators did? If yes, it's reproducible.	1 (Impossible): Cannot be run due to critical issues or missing parts. 2 (Very Difficult): Can't run without major problems; needs expert help or significant workarounds. 3 (Doable): Can be run with some effort; requires troubleshooting or minor fixes. 4 (Mostly Smooth): Runs well with minimal effort; minor adjustments might be needed. 5 (Plug and Play): Runs perfectly by simply following the instructions; no issues.

	Paper Availability	Availability of Code and Software	Availability of Datasets	Computer Requirements	GPU Requirements	Documentation Quality	Ease of Setup	Reproducibility of Results	Rating
BFT Detector	The paper was accessible, although the exact method (open-access vs. institutional access) wasn't explicitly logged, it was obtained for review.	The code was available on GitHub, and we successfully cloned it. However, the provided instructions were significantly outdated for modern Python environments, leading to numerous installation failures.	The documentation mentioned "test inputs" but did not provide clear links or instructions for accessing the main dataset. We couldn't locate it.	The project was built for Ubuntu 20.04 LTS (Python 3.8/3.9). Our Kali Linux WSL environment runs Python 3.13. Furthermore, an inability to enable BIOS virtualization on the host machine prevented us from running Docker, which was our primary workaround for environment compatibility.	GPU requirements were not explicitly stated in the project documentation .	The setup instructions were outdated, specifically regarding Python versions and expected system packages (libgconf-2-4 was unavailable in Kali's repos). This forced extensive troubleshooting beyond the provided guide.	Setup was extremely challenging. We faced persistent Python dependency conflicts (scikit-image, setuptools), and encountered an unresolvable hardware/system barrier (BIOS virtualization preventing Docker). Even a cloud-based alternative (Gitpod) presented its own "runner" configuration obstacles, which were beyond the scope of simple setup.	We were unable to get the project to run to completion. Due to the intractable environment and dependency issues, we could not execute any experiments or verify the results claimed in the paper.	2

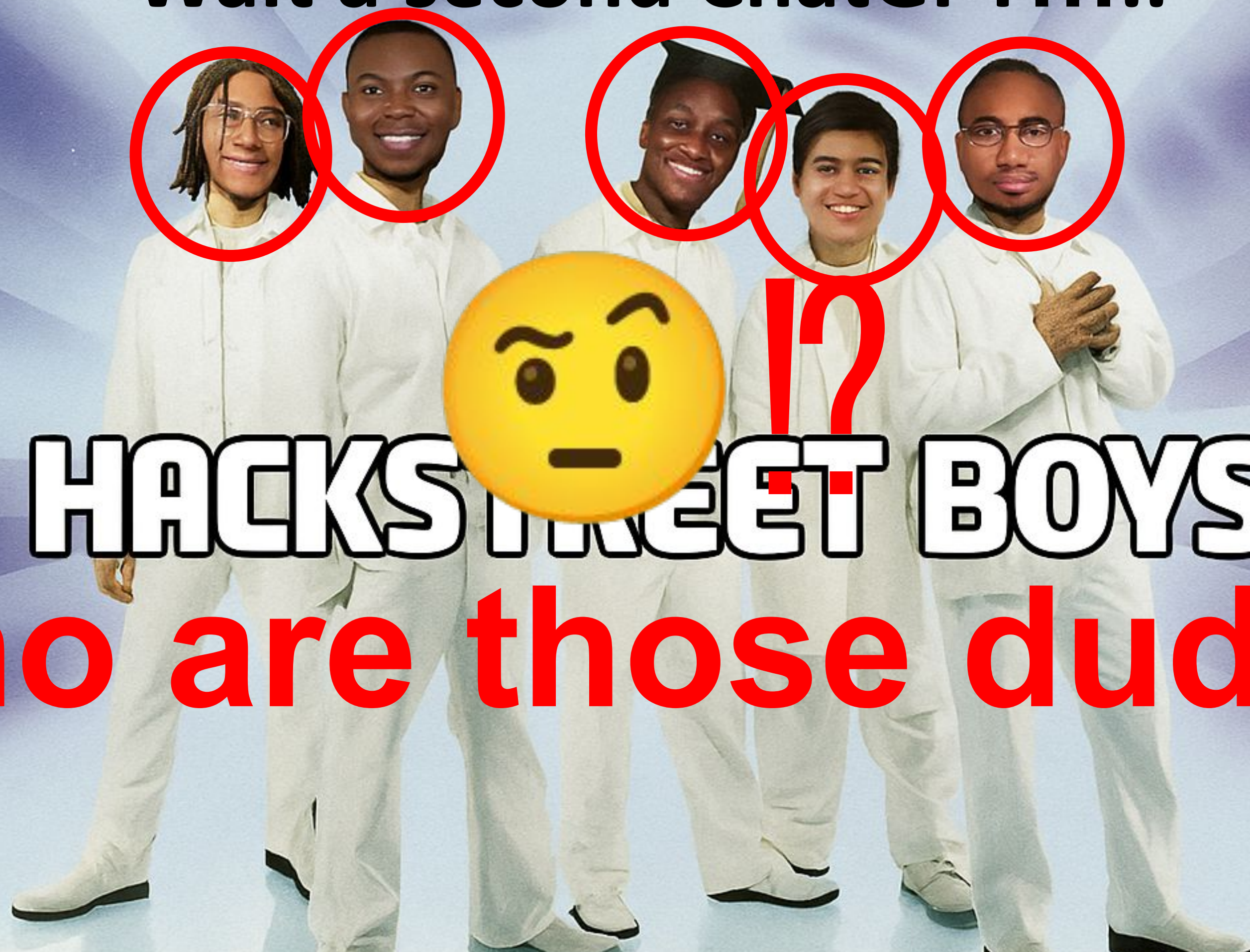
HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

HackStreet Boys



Wait a second ChatGPT.....



HACKS IN THE STREET BOYS

Who are those dudes?



YARI PETTIS

ZION PEASE

DAVE BROWN

EJAY AGUIRRE

JULIAN TOLBERT



Project Overview & Team Roles

Julian – GitHub & Documentation Lead

Sets up repo, manages README.md, folder structure, and code organization.

Ejay – Poster & Presentation Lead

Designs project poster and final slide deck; supports portal content & layout. Will be working on Flask as well.

Zion – Paper Analyst

Selects target papers, evaluates reproducibility criteria, leads scorecard writing.

Dave – Code Runner

Attempts to reproduce paper results, logs code, dataset, and hardware issues

Yari – Communications & Submission Manager

Coordinates daily check-ins, manages submission proof, team info, and final review.



5 Day Plan

Days

Focus

Key Outputs

Monday

Kickoff & Setup

Repo setup, roles assigned, paper shortlist

×

×

×

×

Tuesday

Paper Deep Dive & Planning

Paper selected, access tested, plan slides

Wednesday

Scorecard Development & Testing

JSON/CSV file, initial portal layout, graph

Thursday

Portal Build & Poster Finalization

Site live, poster PDF, submission proof

Friday

Final Presentation & Deliverables
Wrap-up

Slides PDF, final push to GitHub, rehearsal

HackStreet Boys: Project Progress

Priorities:

Our current priority is creating the datasets from the papers and a proper rubric.

We have decided to use a python script to read the scores in so that we can use the data to plot that information.

While also starting to build our website/github to host our information and the actual score cards that we are building.



HackStreet Boys

Project Plan

Update!

The project plan before is still in progress but we have started to move somethings to be done in tandem with other goals, like the beginning to code our flask and properly setting up the information held in papers/websites. While also including more flexibility due to time zones and some skill levels.

— — .

Technology/Problems

We have been using AI like manus, chatgpt and gemini prompting it to give us ideas for rubric and helping debug our code. While also applying our skills in virtual environments like colab to test out our python code.

We created a python script that scrapes a specified number papers, feeds them into Gemini AI, which then provides feedback on some scoring metrics and stores them in a csv file.





HackStreet Boys

We were able to create a python script to evaluate papers and put them into a dataframe to be graphed. We also have started to get our flask app up and running including our team/background information

```
# 2. Iterate through each paper and get LLM evaluation
for i, paper in enumerate(papers_for_evaluation):
    paper_title = paper.get("Title", "Untitled Paper").replace('\n', ' ')
    print(f"\n--- Evaluating Paper {i+1}: '{paper_title}' ---")

    prompt = generate_llm_prompt_for_paper(paper, RUBRIC)

    try:
        response = llm_model.generate_content(prompt)
        evaluation_result_text = response.text.strip()

        print(evaluation_result_text)
        all_raw_evaluations_text += f"\n--- Evaluation for Paper {i+1}: '{paper_title}' ---\n"
        all_raw_evaluations_text += evaluation_result_text
        all_raw_evaluations_text += "\n" + "="*80 + "\n"

        # Parse the LLM's text response directly into a dictionary
        parsed_scores = parse_llm_response_to_dict(evaluation_result_text, paper_title)
        structured_evaluations_list.append(parsed_scores)

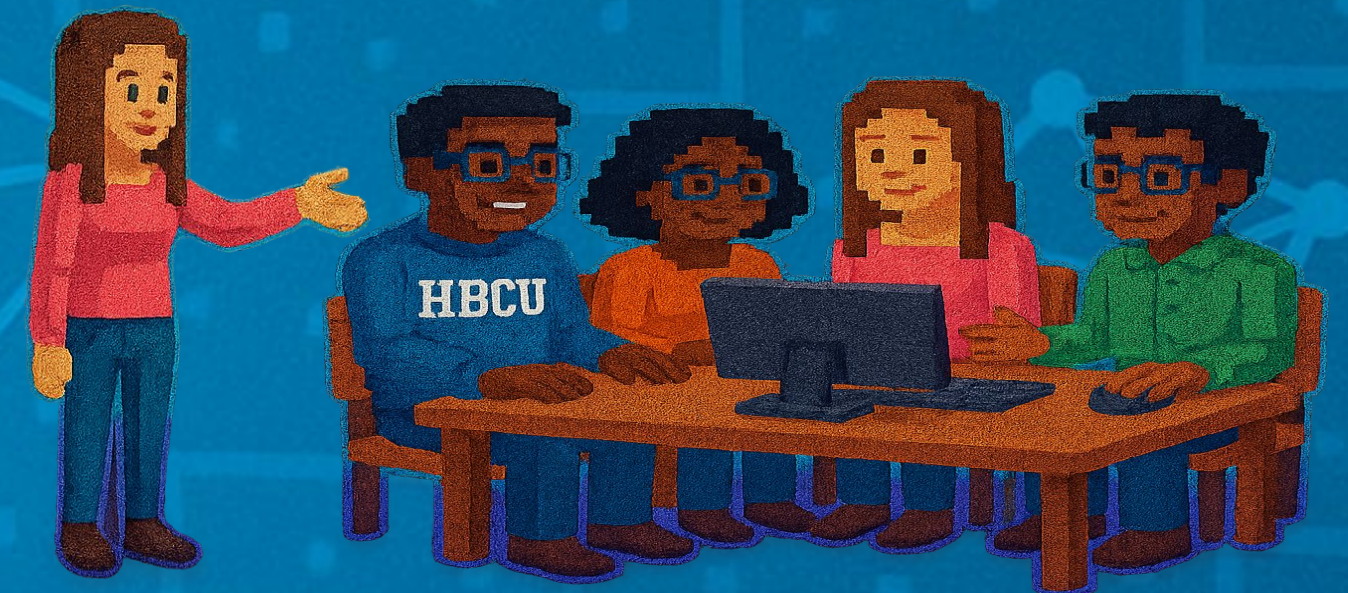
    except Exception as e:
        print(f"Error evaluating paper '{paper_title}' with Gemini: {e}")
        error_message = f"LLM evaluation failed for this paper: {e}"
        all_raw_evaluations_text += f"\n--- Evaluation for Paper {i+1}: '{paper_title}' (FAILED) ---\n"
        all_raw_evaluations_text += error_message
        all_raw_evaluations_text += "\n" + "="*80 + "\n"
        structured_evaluations_list.append({"Paper Title": paper_title, "Evaluation Status": "Failed"

```


HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

CodeRunners

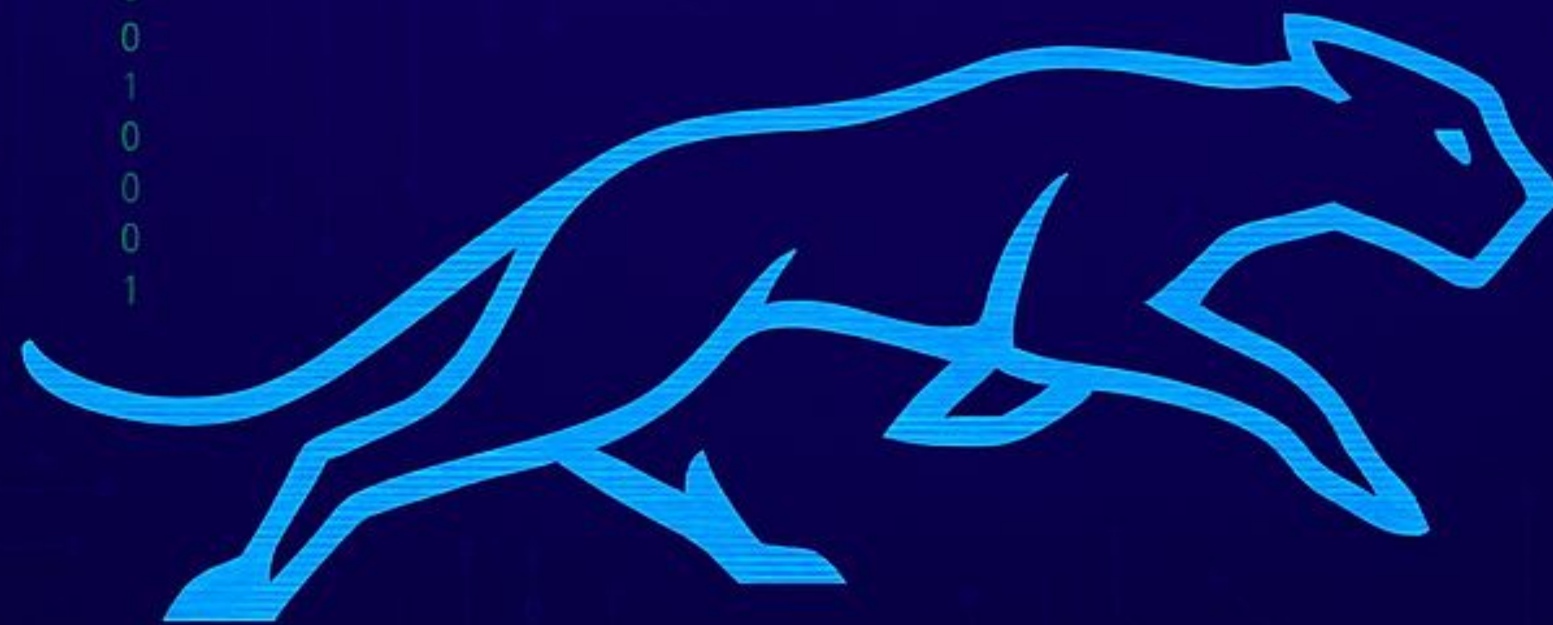


CodeRunners

CODE RUNNERS

Team Members Name

- Iyana Jones
- Arghavan Noori
- Aaliyah Lockett
- Copernic Mensah
- Holy Agyei



<https://on.soundcloud.com/u1r553T8KodM0Lj0j>



CodeRunners Key Milestones

01

Team formation, paper selection,
and role assignment

Deliverables: Intro slide,
README.md, GitHub repo with
paper list and goals

02

Define reproducibility
metrics and evaluation
criteria.

Deliverables:
Reproducibility scorecard
(template), test plan

03

Evaluate reproducibility
across multiple papers
(ICSE/SC24)

Deliverables: Scorecards,
logs, Python scripts for
automated scoring

04

Build comparison
dashboard

Deliverables:
Streamlit/Flask portal
with visual metrics for all
papers

05

Submit final poster and
presentation

Deliverables: Final poster,
presentation slides, portal
link, updated repo

**Iyana | Lead**

Tracks goals, edits README, manages daily progress, ensures overall project alignment.

**Arghavan | Model Analyst**

Compares model outputs, analyzes results, and scores reproducibility gaps.

**Copernic | Presenter**

Creates compelling visuals for the poster and presentation slides.

**Aaliyah | Experiment Engineer**

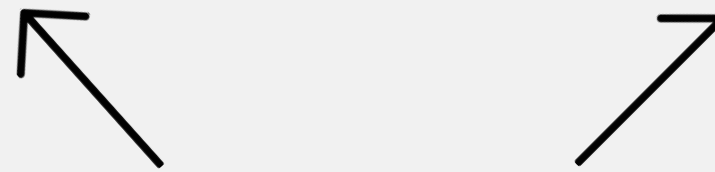
Sets up tasks, configures environments, and runs models for evaluation.

**Holy | Portal Builder**

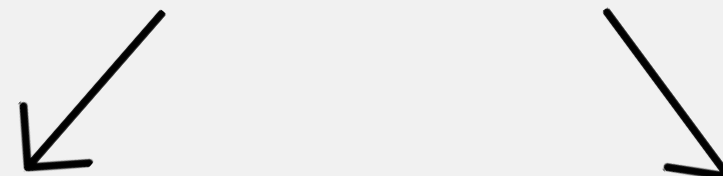
Develops the interactive dashboard or website for the reproducibility scorecard and visualizations.

**Github**

<https://github.com/SGX3CodeRunners/RealWorldBugs.git>



Team Roles & Responsibilities





CodeRunners

Project Overview and Goals

Objective:

Evaluate and compare reproducibility across multiple ICSE 2023 & SC24 papers focused on large language models (LLMs) for code understanding.

Goals:

- Score each paper using a standardized reproducibility framework.
 - Build a public portal to visualize comparative results.
 - Summarize findings in a Gateways 2025 poster.
- 

CodeRunners

Progress

- Expanded from single paper to multi-paper comparative reproducibility study
- Designed and implemented a reproducibility scorecard (100-point framework)
- Currently generating Python code to automate scoring from paper content
- Challenge: Missing GitHub links in some papers limits full artifact scoring
- Streamlit/Flask portal under development to visualize paper scores
- All updates align with the revised project plan (Comparative Repro Study)

```
Paper ID: 18
Title: Validating SMT Solvers via Skeleton Enumeration Empowered by Historical Bug-Triggering Inputs
Score: 15
Artifact URL: https://github.com/CGCL-codes/HistFuzz
DOI URL: https://doi.org/10.1109/ICSE48619.2023.00018
Notes:
- Code available on GitHub (assumed open-source license).
- Docker/Containerization: Requires manual check of the repository.
- Dependency Management: Requires manual check of the repository.
- Build Instructions: Requires manual check of the repository README.
- Specialized Hardware Support: Requires manual check of the repository.
- CI/CD Pipelines: Cannot be inferred from URL. Requires manual check.
- Version Control: Assumed via GitHub.
- Comprehensive README: Requires manual check of the repository.
- API/Data Schema Docs: Requires manual check of the repository.
- Reproducibility Badge: Cannot be inferred from URL. Requires manual check.
- Runtime Instructions: Requires manual check of the repository.
- Result Validation: Requires manual check of the repository.
- Public Dataset Links: Data accessibility uncertain from URL.
- Data Preprocessing: Requires manual check of the repository.
- Model Weights: Requires manual check of the repository.
- Issue Tracking: Assumed via GitHub.
- Discussion Forum: Cannot be inferred from URL. Requires manual check.
```

- Using chatgpt and manus ai, we created a python script in Google Colab that was able to run all of the papers through the scorecard. The issues we came across was it repeatedly listed all papers with a score of 13-15 unless we manually checked the Github repository.
- New approach: Semi-Manual (Hybrid) Approach (Recommended for Efficiency)

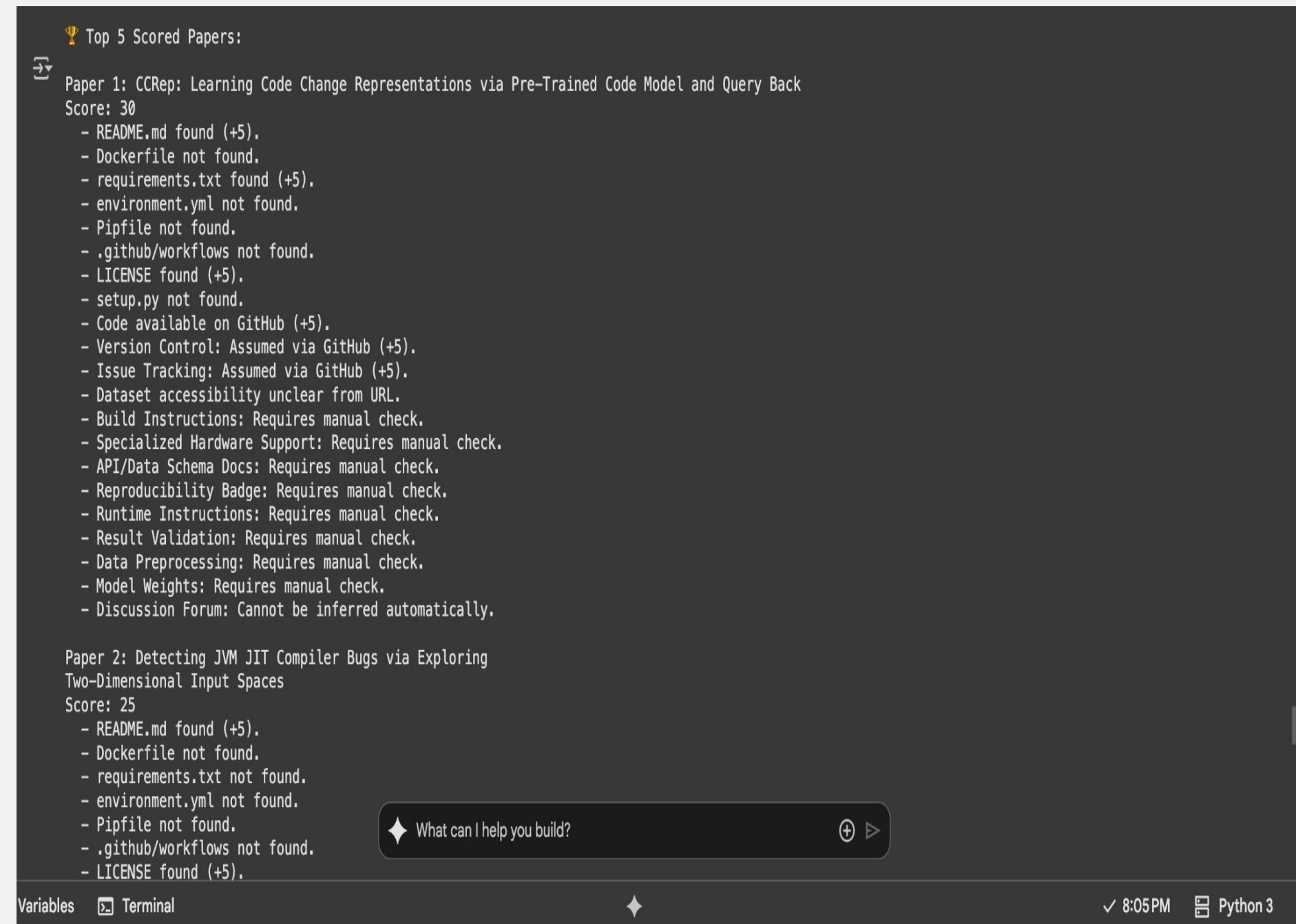
CodeRunners Progress

- Designed and implemented a reproducibility scorecard (100-point framework)
- Changed the code so that more pages are automatically scored
- Currently generating Python code to automate scoring from paper content
- Challenge: Missing GitHub links in some papers limits full artifact scoring
- Streamlit/Flask portal under development to visualize paper scores
- Started building the project portal
-

```
🔗 Paper: One Adapter for All Programming Languages?  
Adapter Tuning for Code Search and Summarization  
README.md: Found  
Dockerfile: Not found  
requirements.txt: Not found  
environment.yml: Not found  
Pipfile: Not found  
.github/workflows: Not found  
LICENSE: Not found  
setup.py: Not found  
-----  
Paper: CCRep: Learning Code Change Representations via Pre-Trained Code Model and Query Back  
README.md: Found  
Dockerfile: Not found  
requirements.txt: Found  
environment.yml: Not found  
Pipfile: Not found  
.github/workflows: Not found  
LICENSE: Found  
setup.py: Not found  
-----  
Paper: Keeping Pace with Ever-Increasing Data: Towards
```


CodeRunners Progress

- Using chatgpt and manus ai, we created a python script in Google Colab that was able to run all of the papers through the scorecard. The issues we came across was it repeatedly listed all papers with a score of 13-15 unless we manually checked the Github repository.
- New approach: Semi-Manual (Hybrid) Approach (Recommended for Efficiency)
- Challenges where that for some papers you had to put it in manually and it was not showing the scores.
- We used Manus ai to get a code that would do all the papers automatically and give us the scores.



The screenshot shows a CodeRunner interface with a dark background. At the top, it says "Top 5 Scored Papers:". Below this, it lists two papers with their scores and a list of checks for each.

🏆 Top 5 Scored Papers:

Paper 1: CCRep: Learning Code Change Representations via Pre-Trained Code Model and Query Back
Score: 30

- README.md found (+5).
- Dockerfile not found.
- requirements.txt found (+5).
- environment.yml not found.
- Pipfile not found.
- .github/workflows not found.
- LICENSE found (+5).
- setup.py not found.
- Code available on GitHub (+5).
- Version Control: Assumed via GitHub (+5).
- Issue Tracking: Assumed via GitHub (+5).
- Dataset accessibility unclear from URL.
- Build Instructions: Requires manual check.
- Specialized Hardware Support: Requires manual check.
- API/Data Schema Docs: Requires manual check.
- Reproducibility Badge: Requires manual check.
- Runtime Instructions: Requires manual check.
- Result Validation: Requires manual check.
- Data Preprocessing: Requires manual check.
- Model Weights: Requires manual check.
- Discussion Forum: Cannot be inferred automatically.

Paper 2: Detecting JVM JIT Compiler Bugs via Exploring Two-Dimensional Input Spaces
Score: 25

- README.md found (+5).
- Dockerfile not found.
- requirements.txt not found.
- environment.yml not found.
- Pipfile not found.
- .github/workflows not found.
- LICENSE found (+5).

At the bottom of the interface, there is a search bar with the text "What can I help you build?" and a search icon. The bottom status bar shows "Variables", "Terminal", a star icon, and the time "8:05 PM" along with "Python 3".

HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

Hard To Cache

SGX3@Hackathon-25 ~ \$: <charli_brooks> <silas_erving> <chante_ray> <seth_mack>

HAD TO CHECK

<song_title>: flames
<writer>: van xo vibes
<song_link>: <https://soundcloud.com/van-xo-vibes/flames>

Day 3 Team Progress Check-In

Progress Priorities

- Getting the scraper running correctly with all added files
- Further develop the GitHub HTML website
- Finalize the scorecard and summarization
- Finalize the web design and color scheme

Updated Project Plan

- Priority shift to working on a web-scraping program
- Reorganized workflow to work in parallel
- Re-access the usability of the found files
- Reorganizing so that members who are on travel duties are now the first priority to ensure we can continue the work

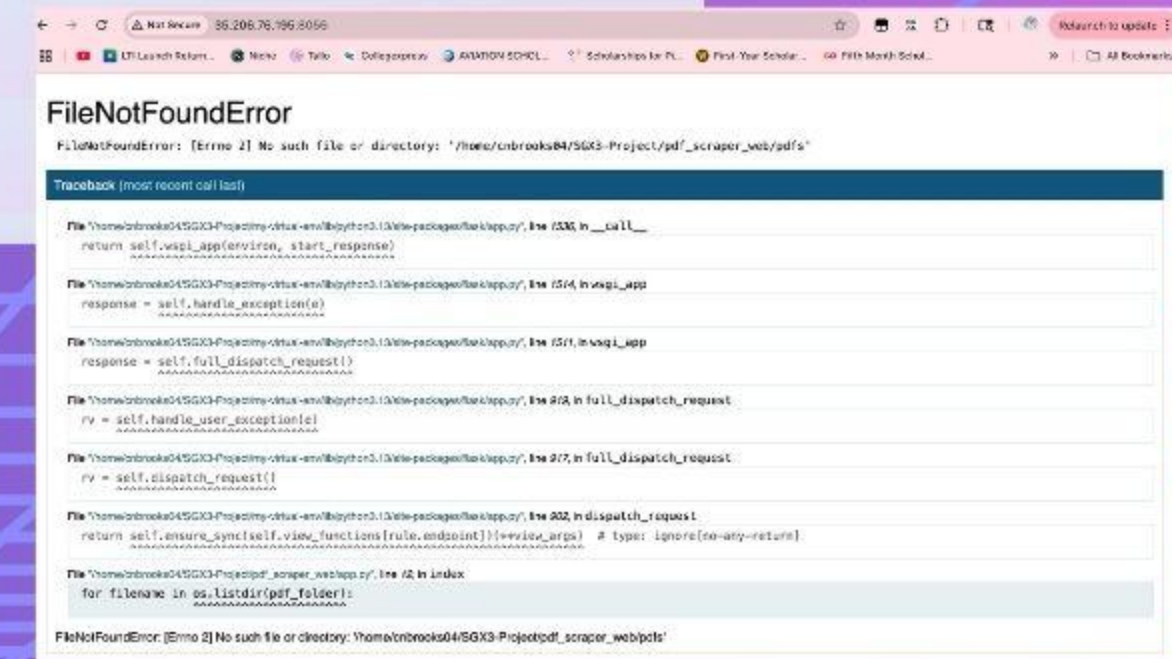
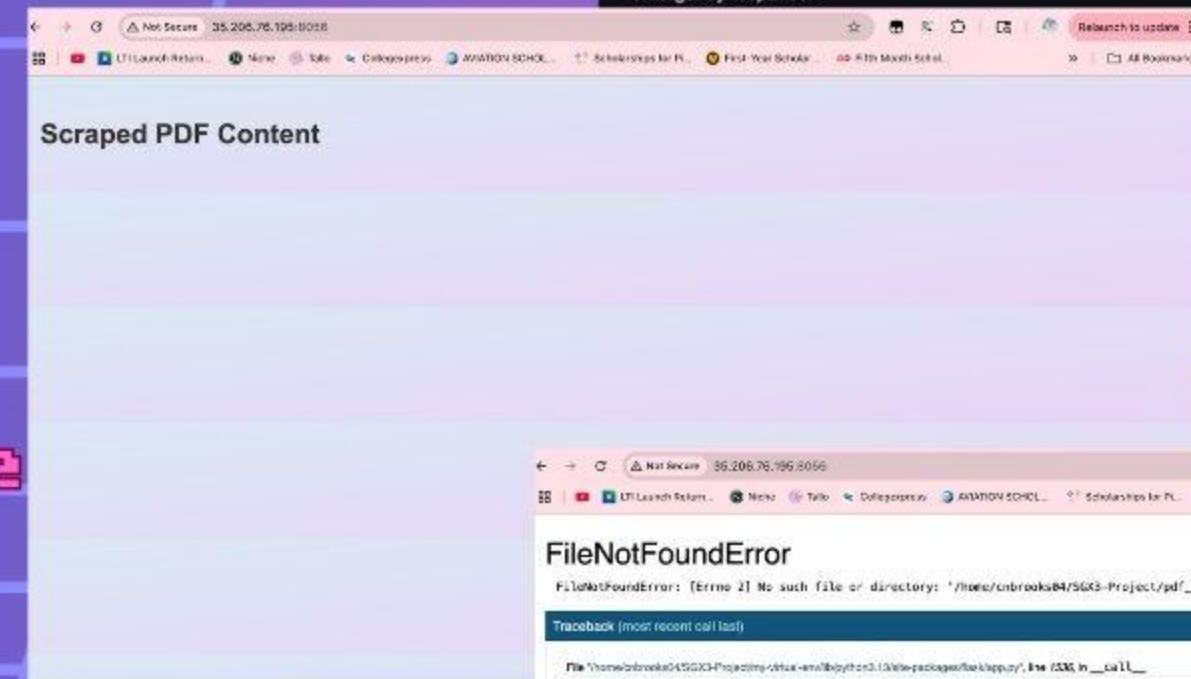
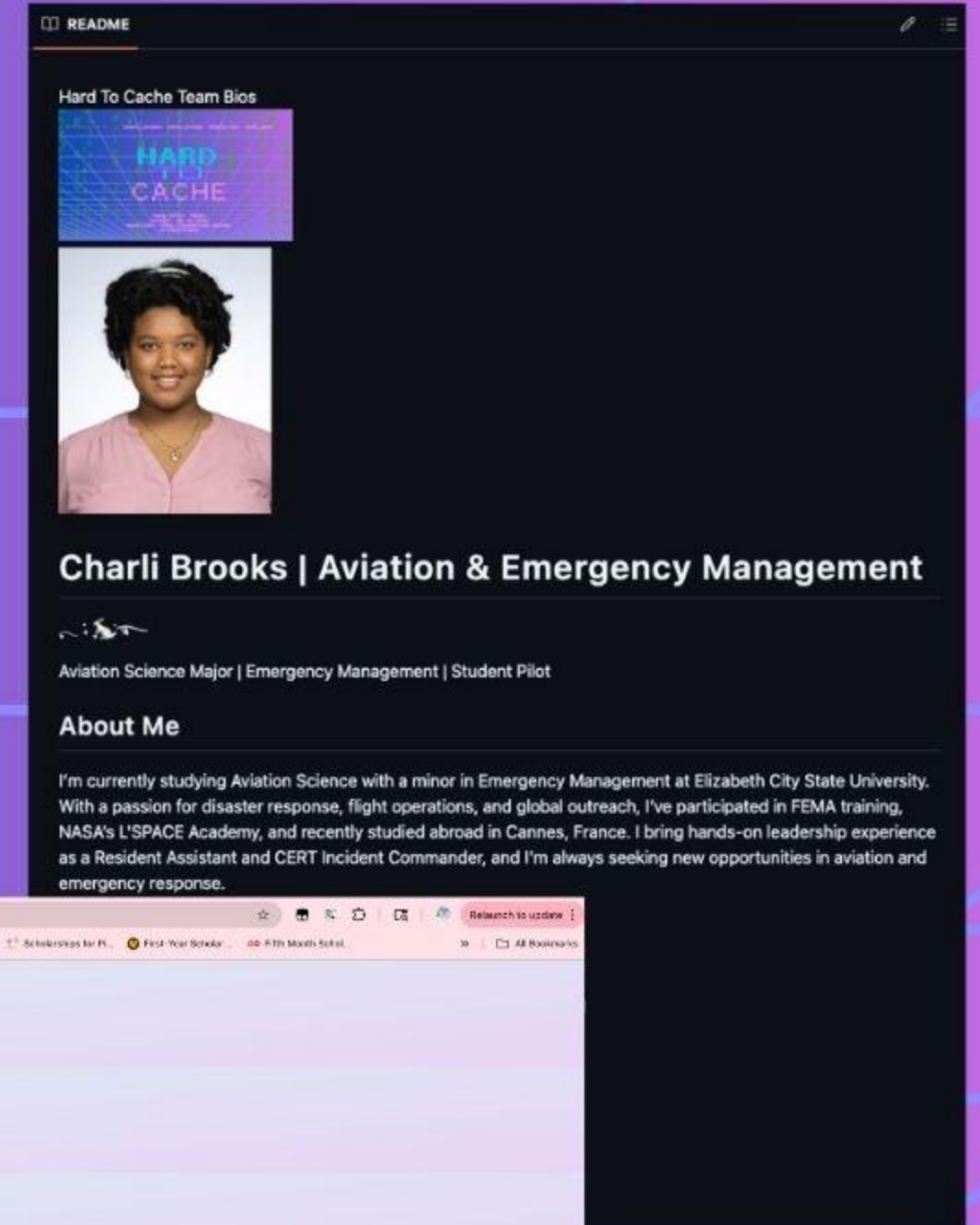
Day 3 Team Progress Check-In

Technology/Resources in Use

- Performed several data scrapes in a virtualenv within Macbook's terminal with a running HTML page.
- Set up a Github live website with Team Bios and Headshots
- Made moderate progress on the scorecard with Google Sheets
- Installed PyMUPDF to import files to the terminal easily

Bottlenecks / Issues / Concerns

- Maintaining the HTML websites and the continuous use of all files
- FileNotFoundError errors in the terminal
- Linking the GitHub HTML and terminal virtualenv
- Trying to make each group member an admin in the GitHub repository.



Team "Hard To Cache"

Silas Erving: Research & Scorecard Lead

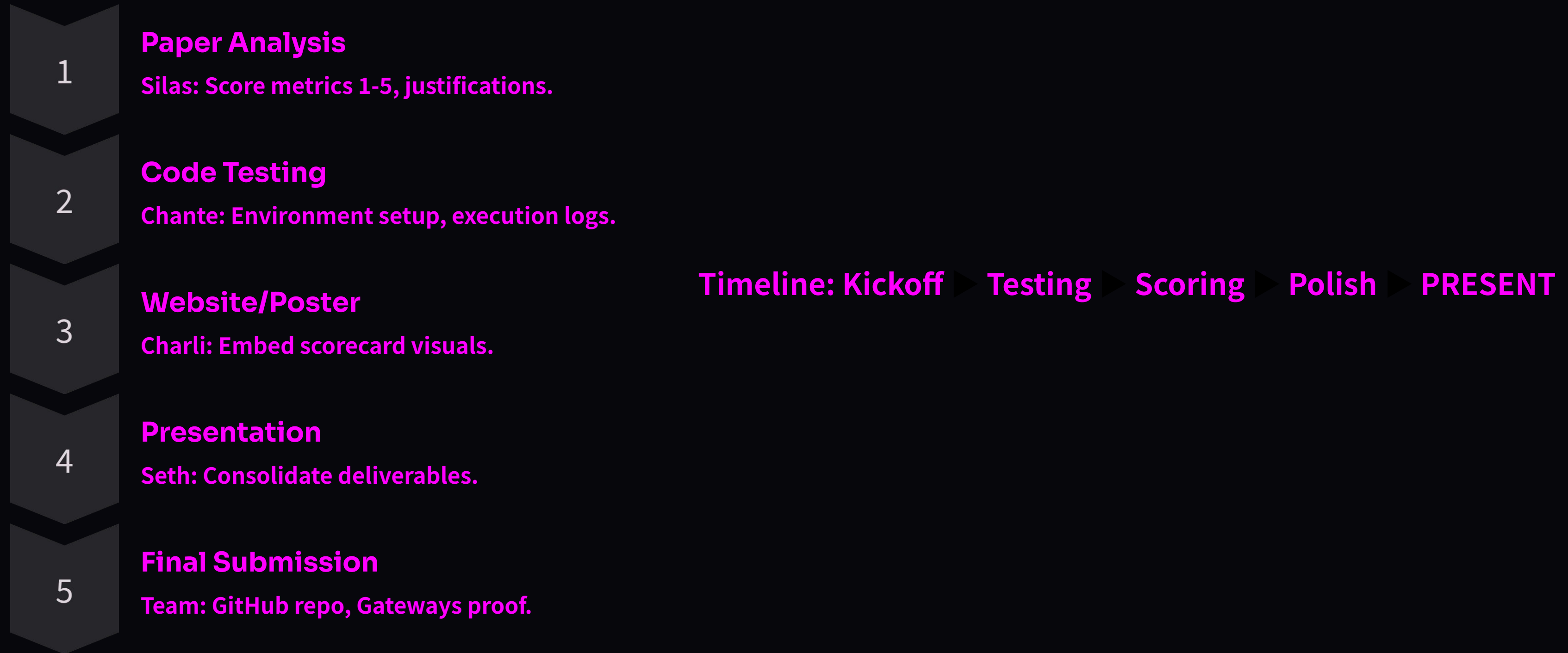
Chante: Code & Reproducibility Engineer

Charli: Web & Poster Designer

Seth: Presentation & Project Coordinator

Project Execution Plan

Evaluate reproducibility of 2023 ISCE + 2024 Supercomputing papers by June 27, 2025.





Charli – Web & Poster Designer

Role Focus: Craft the visual identity of our project across web and print platforms.

Core Responsibilities

- Design and launch a project website with embedded scorecard visuals
- Create a print-ready poster with aligned color scheme, font hierarchy, and content structure
- Collect and format team bios, photos, and profile links for inclusion
- Visualize and embed the Scorecard Prototype (see below) in both the website and poster
- Maintain version control and visual consistency across deliverables



Personal Timeline

Date**Task**

June 23

Attend kickoff, receive scorecard format & branding notes

June 24

Draft website layout, collect team photos and profile links

June 25

Design poster mockup, start embedding early scores and graphics

June 26

Finalize site and poster, QA design, ensure alignment with team plan

June 27

Support visual elements in final presentation and Q&A



Tools and Resources

PurposeTool(s)

Website: GitHub Pages, Terminal (Python3), HTML/CSS,
Google Sites

Poster Design: Canva or Google Slides (PDF export)

Visuals & Embeds: Google Sheets (scorecard
screenshots)

Team Info Collection: Google Forms or Google Docs

Day 2 Team Progress Check-In

Progress Priorities

- Set up program on Github/Collab for automation
- Further Develop GitHub HTML website
- Started to score metrics on our Scorecard sheet (finished 2 articles)

Updated Project Plan

- Priority shift to working on a web-scraping program
- Reorganized workflow to work in parallel

Day 2 Team Progress

Check-In

Technology/Resources in Use

- Compiled data from papers into readable content
- Created a path in Collab to scrape data
- Created HTML website on terminal with first stages of data scrape
- scoring metrics and summarizing each article through Google Sheets

Bottlenecks / Issues / Concerns

- Crashing my poor Macbook
- Apps stalling on Eureka/unable to handle data load
- Losing several SSH keys
- struggling to view some articles (limited access)

```
import pdfplumber
import pandas as pd

# Automatically use the uploaded file
pdf_path = list(uploaded.keys())[0]

with pdfplumber.open(pdf_path) as pdf:
    for i, page in enumerate(pdf.pages):
        print(f"\n📄 --- Page {i+1} Text ---")
        print(page.extract_text())

# Try extracting tables
tables = page.extract_tables()
for t_index, table in enumerate(tables):
    print(f"\n📊 --- Page {i+1} Table {t_index+1} ---")
    df = pd.DataFrame(table[1:], columns=table[0])
    print(df)
```

Test for Data

Add App

 Runtime: 0h 37m
terminal
Job Id: 135568

Running



2Cores

8RAM

0GPU

Stop

End

Connect

 Runtime: 0h 37m
jupyter
Job Id: 749952

Running



2Cores

8RAM

0GPU

Stop

End

Connect

HACKHPC@
ADMI25

HACKATHON

HACKHPC@
ADMI25

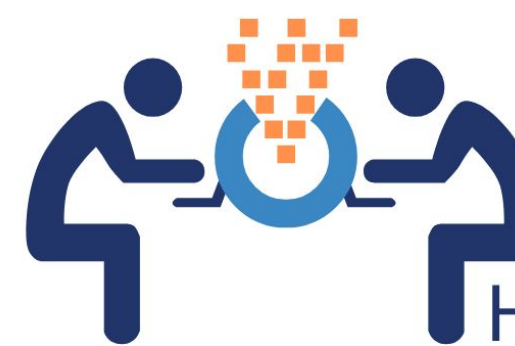
HACKATHON



SGX3
Extend. Expand. Exemplify.

 **omnibond**[®]

project **EUREKA!**

 **HackHPC**
Hackathons and Codeathons

