HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

Hard To Cache

AD MI SGX3 omnibond


HARD
TO
CACHE
Veo

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

HARD TO CACHE

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

Day 3 Afternoon Team Progress Check-In

Progress Priorities

- Finished Biography section of the website
- Changed the GitHub repo to be more accessible
- Continued website development

Updated Project Plan

- Creating an automation process for scoring the papers
- Creating a template for the poster
- GitHub will be the main platform for our website/project

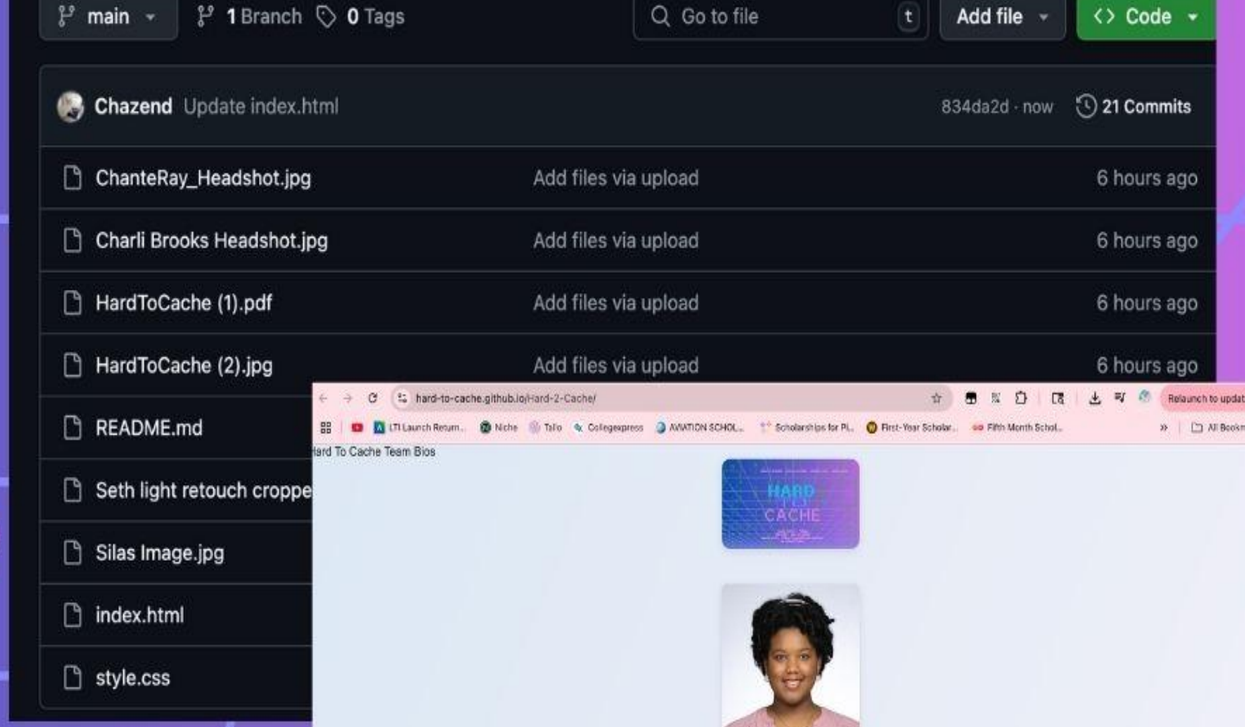
Progress Check-In

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 metics and summarizing each article through Google Sheets

Bottlenecks / Issues / Concerns

Transferring work from the local machine to the GitHub repo.
Deciding which personal scraper works best



Paper ID	er Availab	oftware A	set Availat	er Requir	Requirem	entation	se of Setu	Reproduc	erage Sco	s / Explanation
										Open-access with functional code repo; missing data
119	5	4	1	2	1	3	2	3	2.625	
49	5	4	3	3	3	3	2	4	3.375	Strong artifact paper with accessible code, datasets,
39	5	5	4	4	2	4	3	4	3.875	One of the strongest papers so far. Fully reproducible
100	5	4	4	4	4	3	2	3	3.625	Very strong experimental paper with public datasets
191	5	4	4	3	4	4	2	2	3.5	One of the stronger papers—code, data, and results
179	5	1	5	4	1	4	3	3	3.25	strong data/artifact availability, but setup complexity
										Strong reproducibility overall. Some small technical/
23	5	5	5	3	1	4	3	4	3.75	
99	5	5	5	3	0	4	3	4	3.625	slightly above average.. reproducible with moderate
13	5	5	5	4	4	4	3	4	4.25	could be reproducible, with only modest technical or
63	5	5	5	4	4	4	2	4	4.125	moderate reproducbility overall. Minor documentati

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

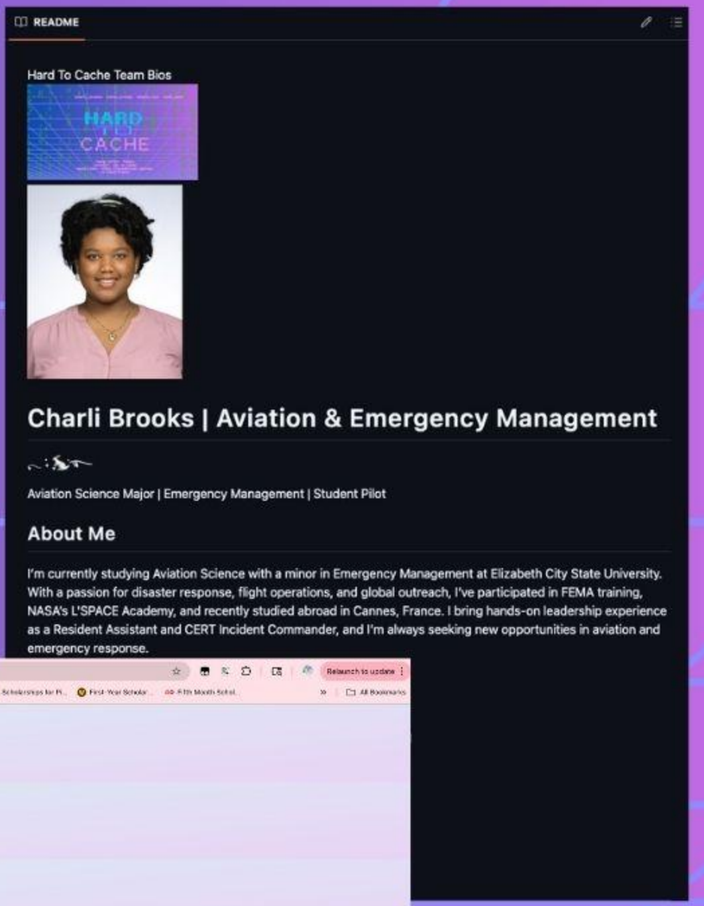
Day 3 Team Progress Check-In

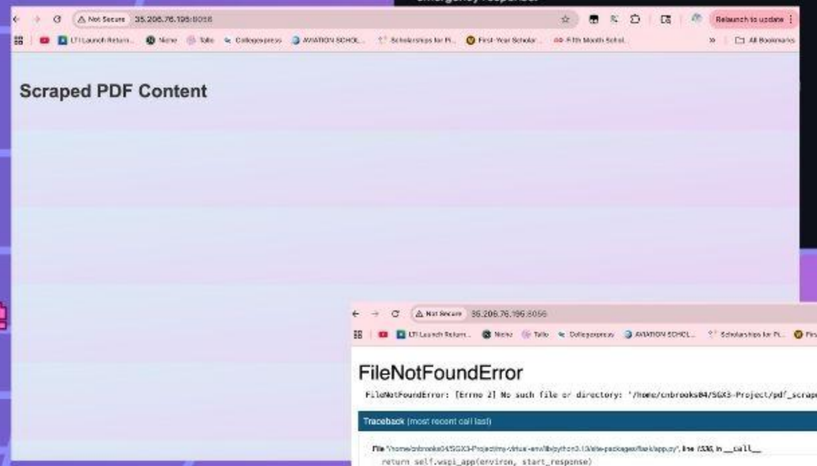
Technology/Resources in Use

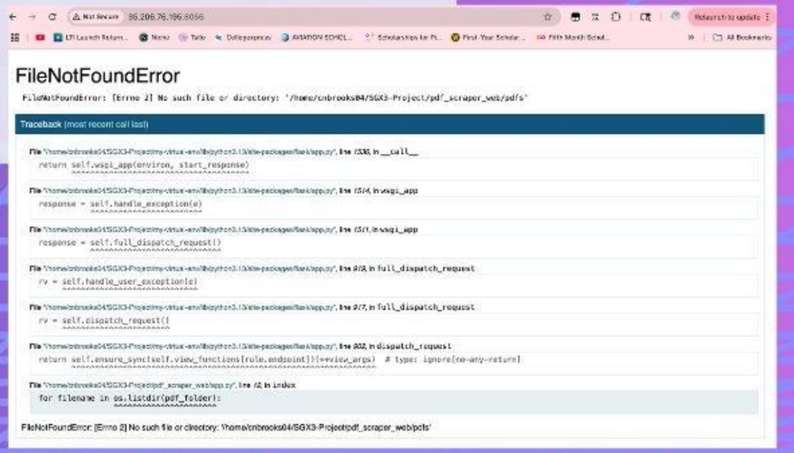
- Performed several data scrapes in a virtualenv within Macbook's terminal with a running HTML page.
- Set up a Github live website with Team Bios and Headshots
- Made moderate progress on the scorecard with Google Sheets
- Installed PyMUPDF to import files to the terminal easily

Bottlenecks / Issues / Concerns

- Maintaining the HTML websites and the continuous use of all files
- FileNotFound 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.

1

Paper Analysis

Silas: Score metrics 1-5, justifications.

2

Code Testing

Chante: Environment setup, execution logs.

3

Website/Poster

Charli: Embed scorecard visuals.

4

Presentation

Seth: Consolidate deliverables.

5

Final Submission

Team: GitHub repo, Gateways proof.

Timeline: Kickoff ▶ Testing ▶ Scoring ▶ Polish ▶ PRESENT

🎨 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

DateTask

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

Purpose	Tool(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 artifcles)

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
- socring metics 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)

```python
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)
```