HACKHPC@ ADMI25 HACKATHON

Team Project Plans - Updates

June 23, 2025

SGX3
Extend. Expand. Exemplify.

Omnibond

ADMI

HackHPC
Hackathons and Codeathons

HACKHPC@
ADMI25
HACKATHON

https://hackhpc.github.io/admi25

Goals For the Next Check-in

Your team needs to add two (2) slides to the shared deck with:

- 1st Slide with just Team Name
- 2nd slide with:
  - Team Name
  - Team Members Names
  - Team Zoom Virtual Background
  - Team Theme Song
    - *Royalty Free Music*
    - *No Lyrics*

[Note: If your slide deck is made in Canva it must be exported to PPTX and then imported to the shared slide deck]

ADMI SGX3 omnibond

HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

Duplicate this slide add your team information and delete this box

Team Name

AD MI SGX3 omnibond

https://hackhpc.github.io/admi25

HACKHPC@ ADMI25 HACKATHON

Team Name

Team Members Name

- Name 1
- Name 2
- Name 3
- Name 4

Duplicate this slide add your team information and delete this box. You can change YOUR slide to use any theme you wish.

Royalty Free Song Title
Royalty Free Song Writer
Royalty Free Song Link

HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

Hard To Cache

AD MI SGX3 omnibond

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

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

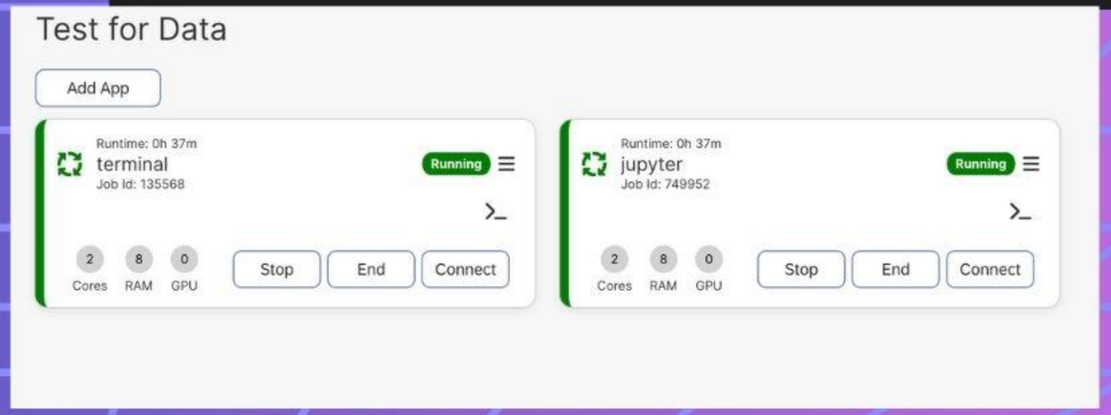
Technology/Resources in Use

- 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)

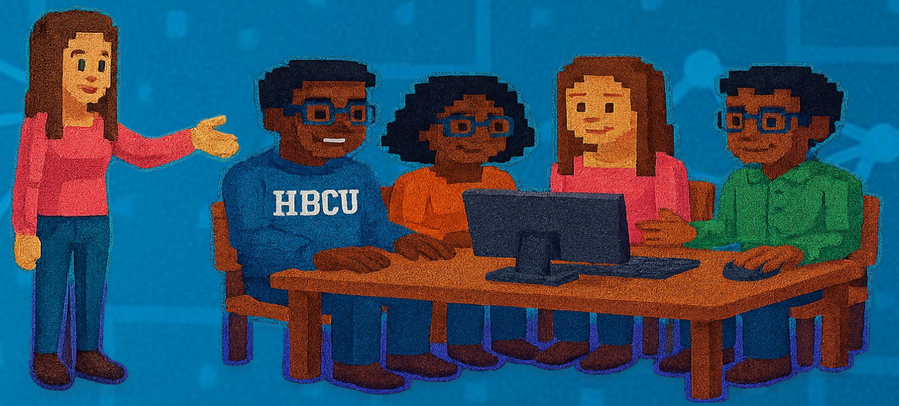
```python
import pdfplumber
import pandas as pd

# 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)
```



HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

CodeRunners



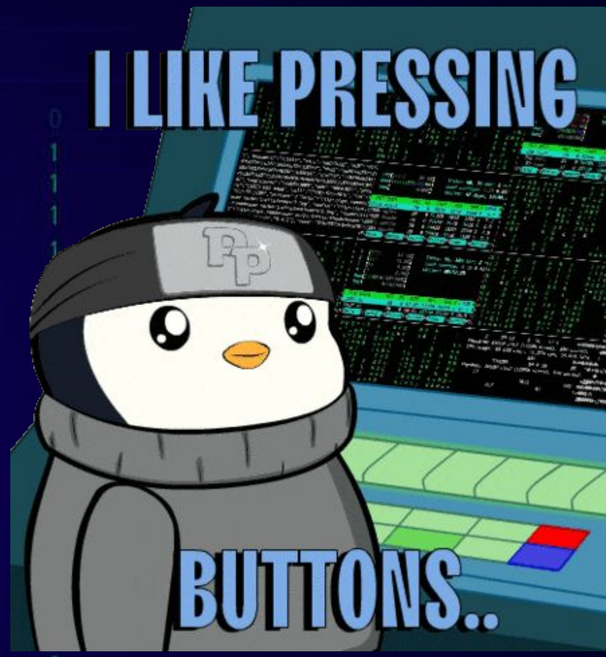
ADMI SGX3 Omnibond

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

Team Roles

& Responsibilities

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

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)

HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

HackStreet Boys

ADMI SGX3 Omnibond

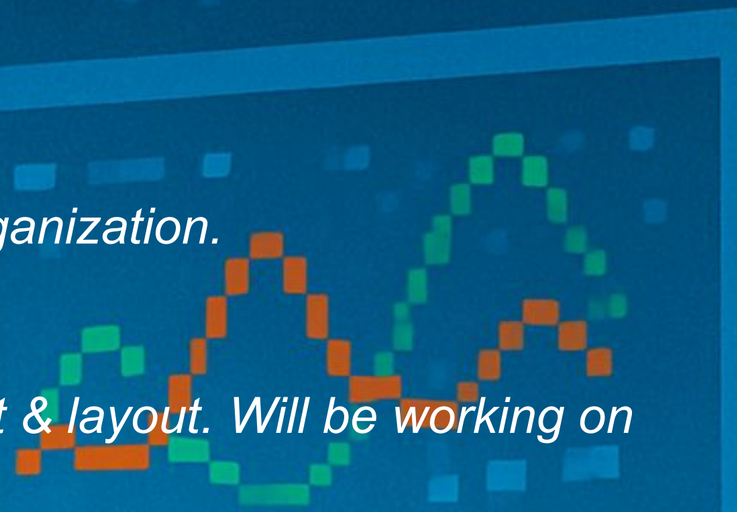
Future Design
penguinmusi
https://pixabay.com/music/future-bas

HackStreet Boys: Project Overview & Team Roles

Team Members & 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

Day	FOCUS	KEY OUTPUTS
MON	Kickoff & Setup	Repo setup, roles assigned, paper shortlist
TUE	Paper Deep Dive & Planning	Paper selected, access tested, plan slides
WED	Scorecard Development & Testing	JSON/CSV file, initial portal layout, graph
THUR	Portal Build & Poster Finalization	Site live, poster PDF, submission proof
FRI	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 set 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 were having problems with scrapping the data but we decided to approach the problem differently by scraping the data manually and then applying that data into a dataset to be read into.


HAMPTON
UNIVERSITY
HU
UNIVERSITY
HAMPTON
UNIVERSITY
HU
UNIVERSITY
HU
UNIVERSITY
Wiresharks
ai

Team Members Name

- Auiana D’Avilar
- Ayinde Hooks
- Howard “Shiloh” Ames
- Ryan Grimes

Hands Up, Buttercup
Neon Beach
https://app.soundstripe.com/songs/6425

BFTDETECTOR: Automatic Detection of Business Flow Tampering for Digital Content Service

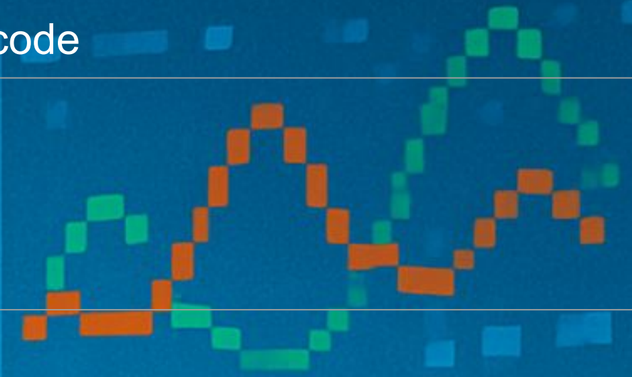
Notes: Day 1

Issue	What Happened	Why It Happened	Concern	Fix
pyhash Installation Failure	The `pyhash` program couldn't install.	It's made for older Intel computer chips, but your computer has a newer Apple Silicon chip. They're not compatible.	This program won't work on our Apple Silicon or Windows computer without special steps or a different program.	Try using a different program, or run your terminal with Apple's Rosetta 2 (Intel chip emulator).
`networkx` Missing	The main program couldn't find a tool called `networkx`.	The first attempt to install all programs didn't finish, so `networkx` was never put on your computer.	This problem should go away once the main installation is fixed.	Run: `pip install networkx`
`scikit-image` Install Failed (Python 3.13)	The `scikit-image` program failed to install.	The version (0.19.2) is old and doesn't work with newer Python (3.13) or related tools like `setuptools`. It tries to use an old method that Python 3.13 doesn't have anymore.	Your Python version is too new for this old program.	Use an older, more compatible Python version (e.g., Python 3.8 or 3.9).
`numpy` Missing for `scikit-image`	`scikit-image` couldn't install because `numpy` wasn't there first.	Some programs, like `scikit-image`, need `numpy` to be installed on your computer before they can even start their own installation.	Always put `numpy` on your computer before installing other complex science-related programs.	Run: `pip install numpy`

Day 1: Kickoff	• Assign roles • Read the paper and understand project scope • Skim the GitHub repo: what code/data is included? • Try to install dependencies, run a sample part of the code
Day 2: Environment Setup	• Run main experiments/analysis from paper • Compare any results with what the paper shows • Record any differences or blockers • Scribe documents every step

Day 3: Reproduction Attempt	• Rate reproducibility from 1–5:
	  ○ 1 = Impossible
	  ○ 3 = Doable with moderate effort
	  ○ 5 = Plug and play
	• Note challenges (e.g., outdated libraries, missing data)

Day 4: Build Deliverables	• Portal Builder sets up GitHub Pages or a clean README
Day 4b: Final Touches + Presentation	• Test the portal • Polish slides • Practice presentation



HACKHPC@
ADMI25
HACKATHON

HACKHPC@
ADMI25
HACKATHON

NSF

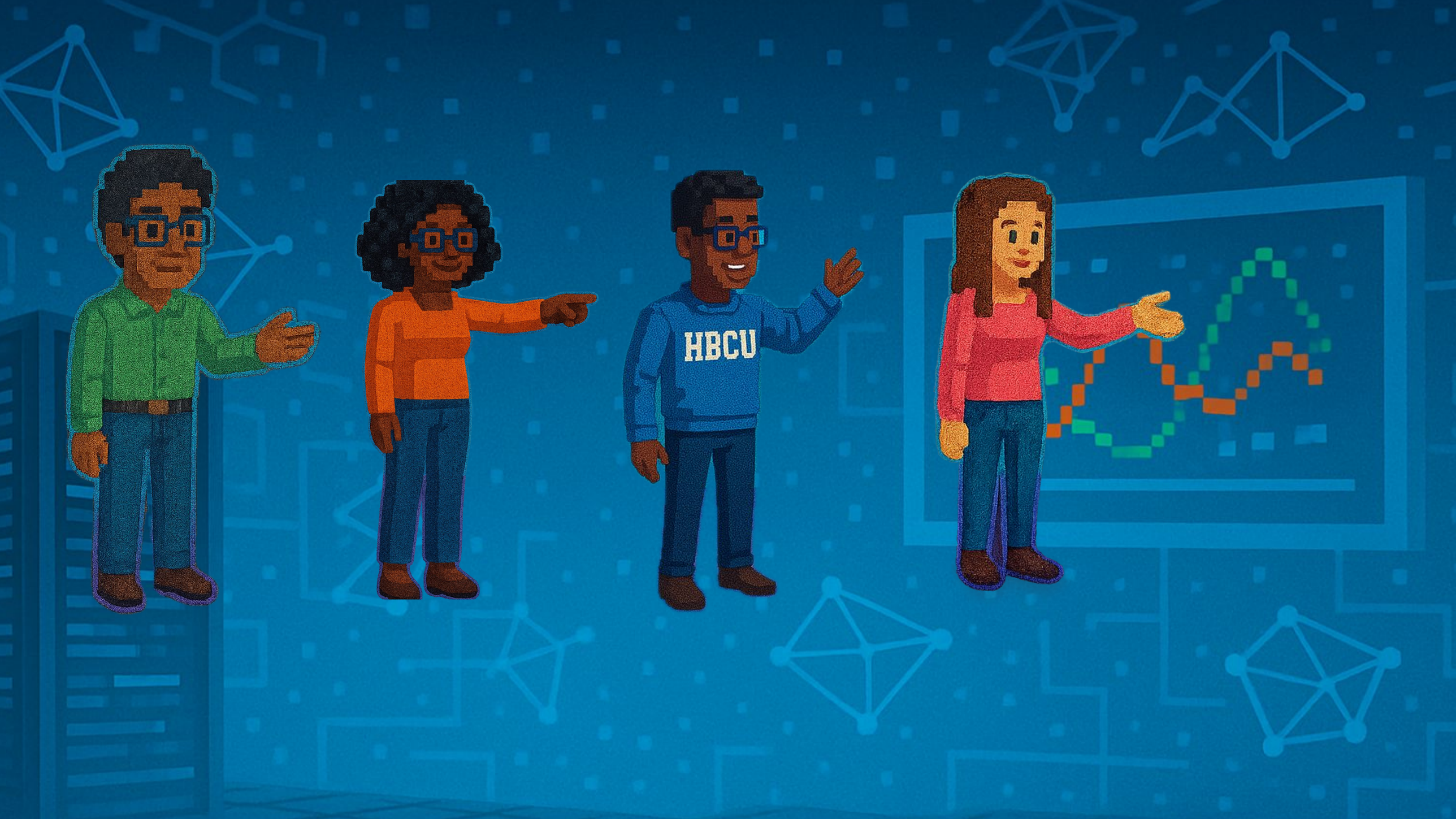
ADMI

SGX3
Extend. Expand. Exemplify.

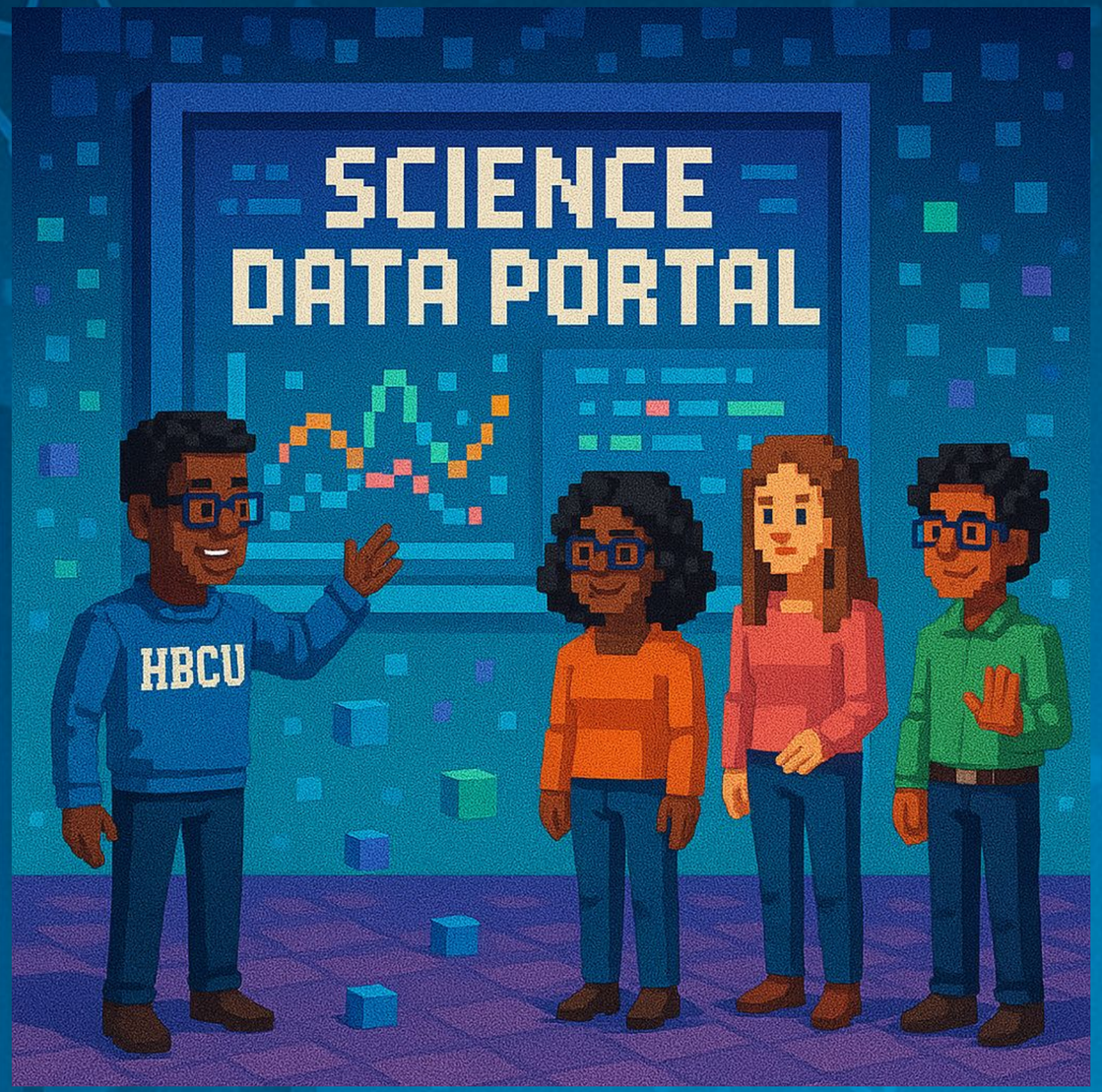
Omnibond

projectEUREKA!

HackHPC
Hackathons and Codeathons


HBCU


HBCU
HBCU
HBCU
HBCU


SCIENCE
DATA PORTAL
HBCU


SCIENCE
DATA PORTAL
HBCU