Project Plan for Research Paper

1. Project Overview

- Project Name: Reproducing Big Code Benchmarks
- Objectives: Evaluate the reproducibility of results reported in the IEEE paper using large language models (LLMs) on multiple code understanding tasks, including code summarization, code completion, and translation.
- Scope:
  - Build a GitHub repo with experiment tracking, setup, documentation, and results.
  - Reproduce at least 2–3 major tasks and compare model performance (CodeT5, StarCoder, GPT-3.5/4).
  - Create a web portal with a reproducibility scorecard and interactive visualizations.
  - Submit a poster summarizing results to Gateways 2025.
- Timeline: 5 days (SGX3 hackathon)

2. Key Milestones and Deliverables

Milestone	Description	Due Date	Deliverables
Day 1 – Project Kickoff	Team formation, IEEE paper review, role assignment, GitHub setup with README and project goals	Day 1	Intro slide, README.md, team roles, initial repo push
Day 2 – Artifact Setup	Identify code/data, test reproducibility environment, begin small-scale runs	Day 2	Artifact notes, initial tests (e.g., code summarization benchmark), logging results
Day 3 – Scorecard Drafting	Define reproducibility metrics, evaluate 2–3 model-task pairs, log gaps or bugs	Day 3	Reproducibility scorecard (draft), run logs, annotated test outputs
Day 4 – Portal +	Build website/dashboard	Day 4	Streamlit/Flask portal with charts,

Poster	to display reproducibility analysis and polish poster		poster draft (PDF/Canva/Slides)
Day 5 – Final Submission	Submit poster, deliver final presentation, and commit all documentation	Day 5	Final poster + presentation deck, updated repo, portal link or local deployment

3. Team Roles

- Aaliyah/Experiment Engineer – Sets up tasks and runs models for evaluation
- Arghavan/Model Analyst – Compares model outputs and scores reproducibility gaps
- Holy/Portal Builder – Develops interactive dashboard or site
- Copernic/Presenter – Creates visuals for the poster and slides
- Iyana/Lead – Tracks goals, edits README, manages daily progress

4. Resource Requirements

- People: 5 members with experience in Python, LLMs, Hugging Face, or Docker
- Tools:
  - Python, PyTorch, Hugging Face Transformers
  - Google Colab, Kaggle Kernels, or a cloud GPU (if needed)
  - Flask/Streamlit for portal
  - Canva or Google Slides for poster
- Communication:
  - Slack, Discord, or Teams for messaging
  - Zoom for check-ins every day at 7

5. Risks & Mitigation

- Missing/incomplete benchmarks – Use archived copies, reconstruct from examples, or generate test data
- Long model runtimes – Use small samples or pretrained results, rely on hosted APIs if needed
- Tool mismatch – Use virtual environments or Docker to isolate dependencies

- Time bottlenecks – Prioritize 2 core tasks and scale from there
- Poster rejection or late submit – Submit early, check formatting guidelines, screenshot confirmation

6. Conclusion

Team CodeRunners will explore and evaluate the reproducibility of LLM-based code understanding models as presented in Big Code is a Big Deal. By validating benchmarks, testing models, and scoring artifacts on reproducibility, the team will produce a clear, visual, and open-source summary of results, contributing to the broader reproducibility movement in software engineering research.