■ AI-Powered Repository Analyzer

1. Problem Statement

Understanding large or unfamiliar GitHub repositories is a time-consuming and error-prone process. Developers spend hours reading through documentation, scripts, and code. Legacy codebases or poorly documented repos slow down onboarding. Hackathon teams often waste time setting up and understanding the repo instead of building features.

2. Proposed Solution

Our tool — AI-Powered Repository Analyzer — leverages Gemini 2.5 Pro to automatically analyze GitHub repositories and provide structured insights.

■ Repo Analysis – Paste a GitHub repo URL and instantly get structured analysis.
■ Code Understanding – Explain what the repo does in plain English.
■ Modernization Suggestions – AI suggests migration paths, testing improvements, and restructuring.
■ Error & Issue Detection – Identify weak spots, outdated practices, or dependency issues.
■ Feature Proposals – Suggest where and how new features can be integrated.
■ Visualization – Auto-generate architecture & flow diagrams.

3. Why This Idea is Beneficial

Time Saver: Speeds up repo understanding during hackathons & real-world projects.
Learning Aid: Helps junior developers and students quickly grasp new codebases.
Collaboration: Team members can ask repo-specific questions.
Future Potential: Can evolve into an AI Dev Assistant for onboarding, audits, and code reviews.

4. Architecture

High-Level Flow:
1. User enters GitHub Repo URL
2. Repo metadata & contents are fetched
3. Data is passed into Gemini 2.5 Pro model
4. Model generates structured insights & diagrams
5. Streamlit UI displays results in Summary, Expandable Sections, Visual Diagrams

5. Demo Workflow

1. Open the Streamlit app.
2. Paste a GitHub repo URL.
3. Ask a repo-related question (e.g., 'How can I modernize this repo?').
4. Get AI-powered insights in a structured, easy-to-read format.

6. Example Use Case

Repo: GEF-Legacy (https://github.com/hugsy/gef-legacy)
Languages: Python 2, Shell, C
Purpose: Debugging & Exploit Development Toolkit
Modernization Suggestion: Migrate to Python 3, adopt pytest, containerize with Docker.
Architecture Suggestion: Split monolithic gef.py into modular package.

7. Future Roadmap

■ Fine-tuned repo-specific AI assistant
■ Dockerized deployment
■ More advanced diagrams (interactive flowcharts)
■■ Integration with VS Code / JetBrains IDEs

8. Conclusion

The AI-Powered Repository Analyzer is not just a hackathon tool but a step towards AI-driven developer productivity. It reduces time wasted on repo onboarding and enables teams to focus on building features that matter.