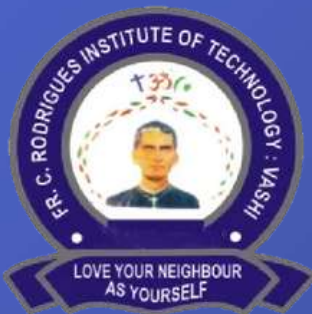# HACKQUINOX 2.0

HACK THE UPSIDE DOWN

Team Name : It Works !
Team ID :
Domain : Open Innovation
Problem Statement : Engineering teams lack a centralized and intelligent system to manage projects, track tasks, and monitor progress efficiently.
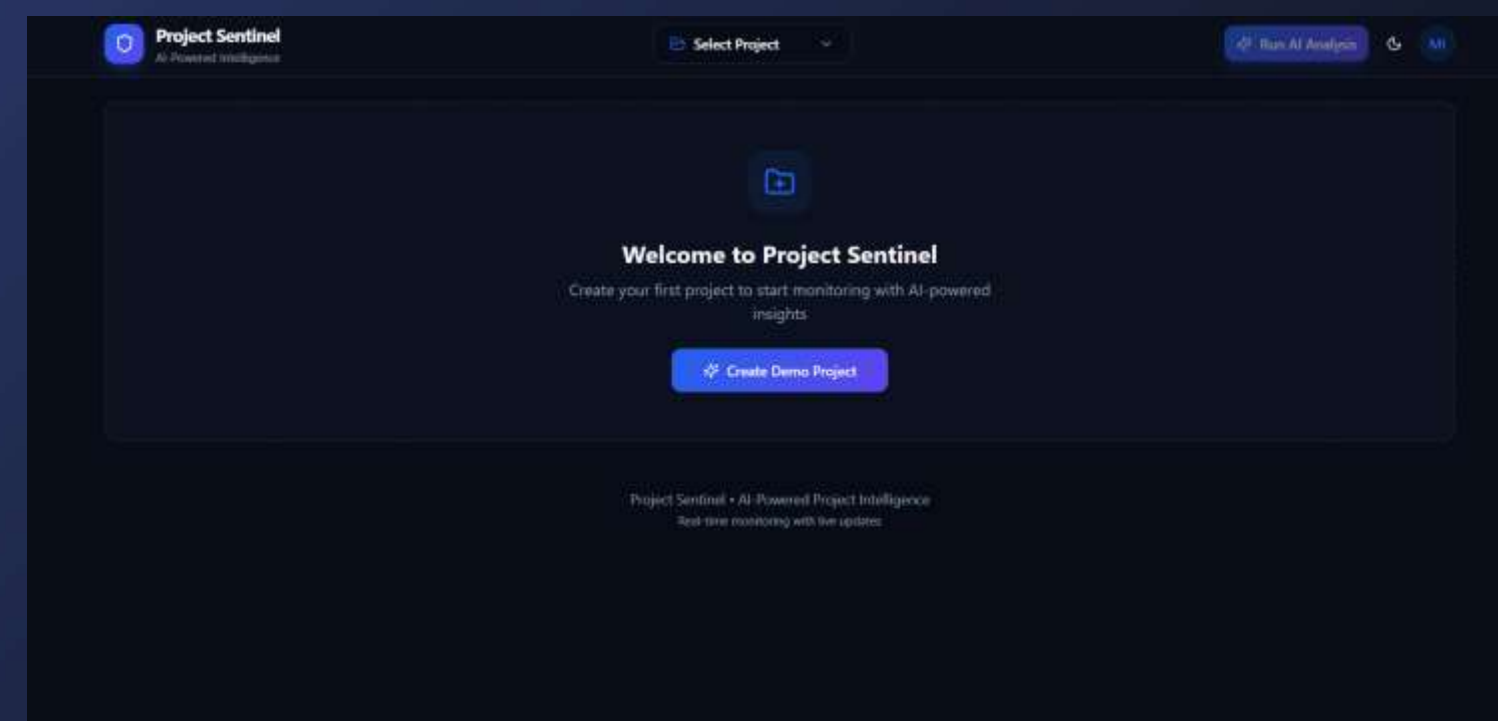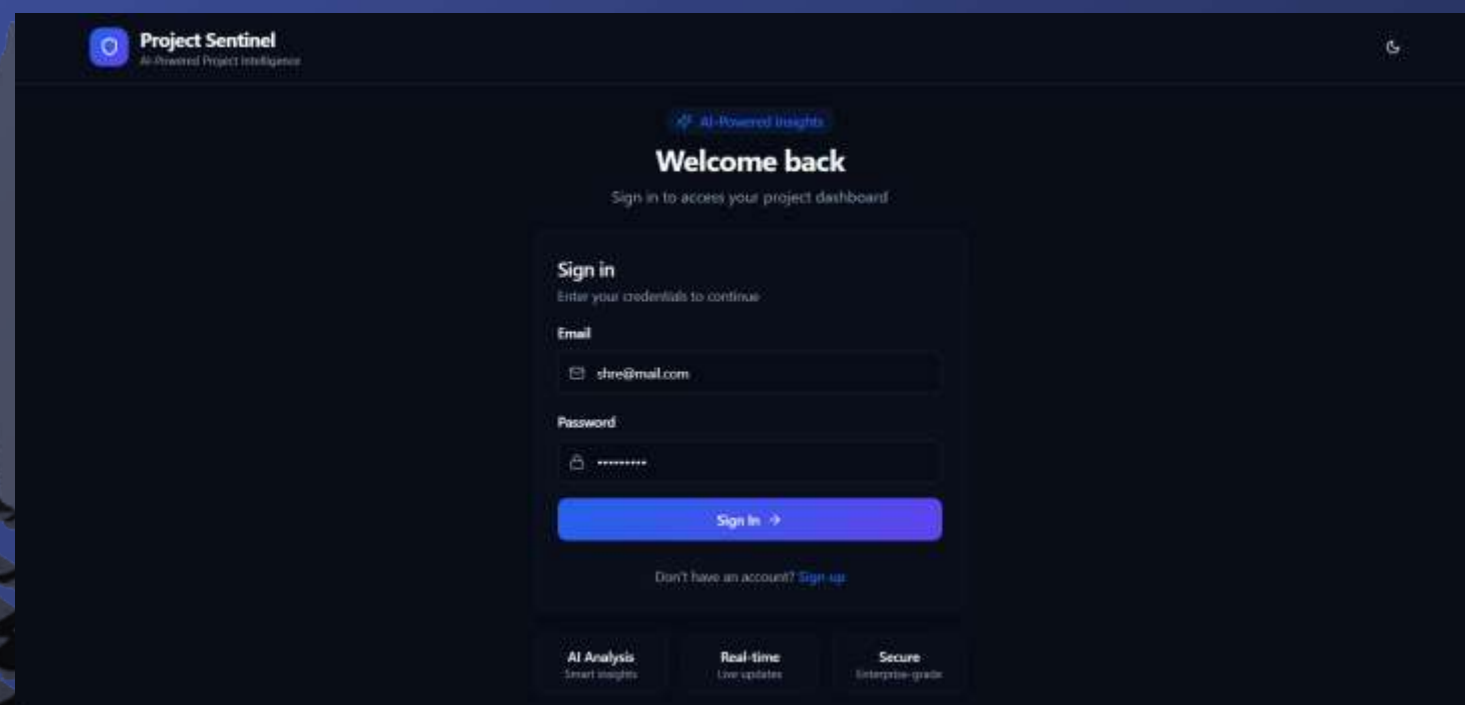
SOLUTION

Idea

- Build a centralized project management system.
- Combine "frontend, backend, and database" into one platform.
- Provide visibility into project tasks, status, and progress.

Solution

- Web-based application with:
- User authentication.
- Project & task management.
- Status tracking and summaries.
- Backend handles:
- Data storage.
- Business logic.
- API communication.

Prototype

TECHNICAL APPROACH

How the System Works

- Users interact with the web dashboard
- Frontend sends requests to backend APIs
- Backend processes logic and connects to database
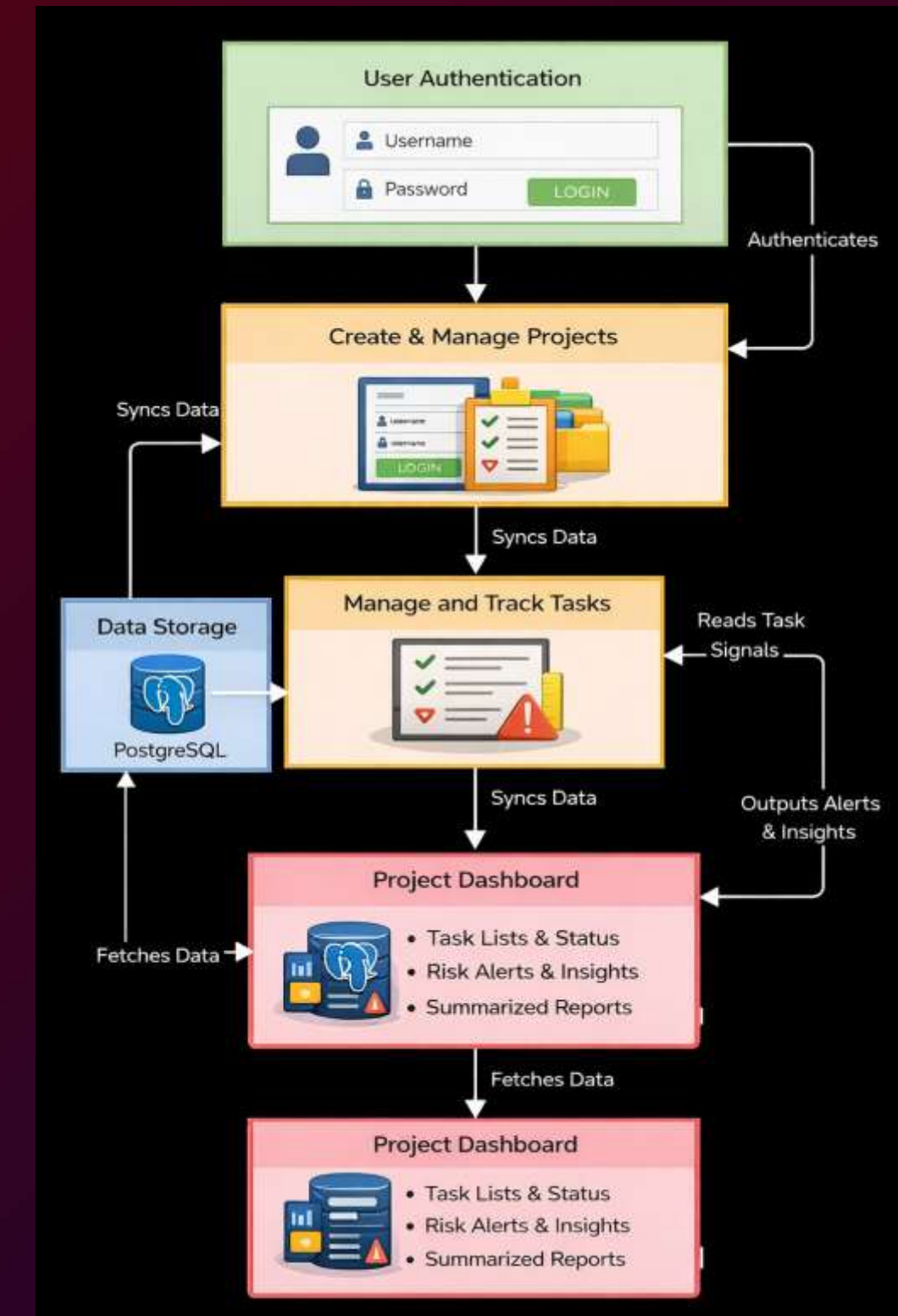- Data is stored and retrieved securely
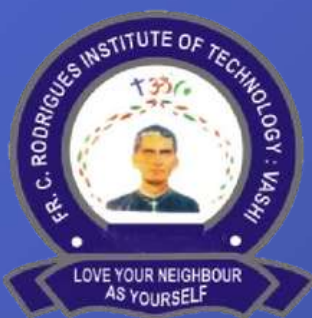
Design Focus

Clean separation of:
- UI
- Logic
- Data
- Scalable architecture for future features

Current Outputs

- Task creation & updates
- Project status overview
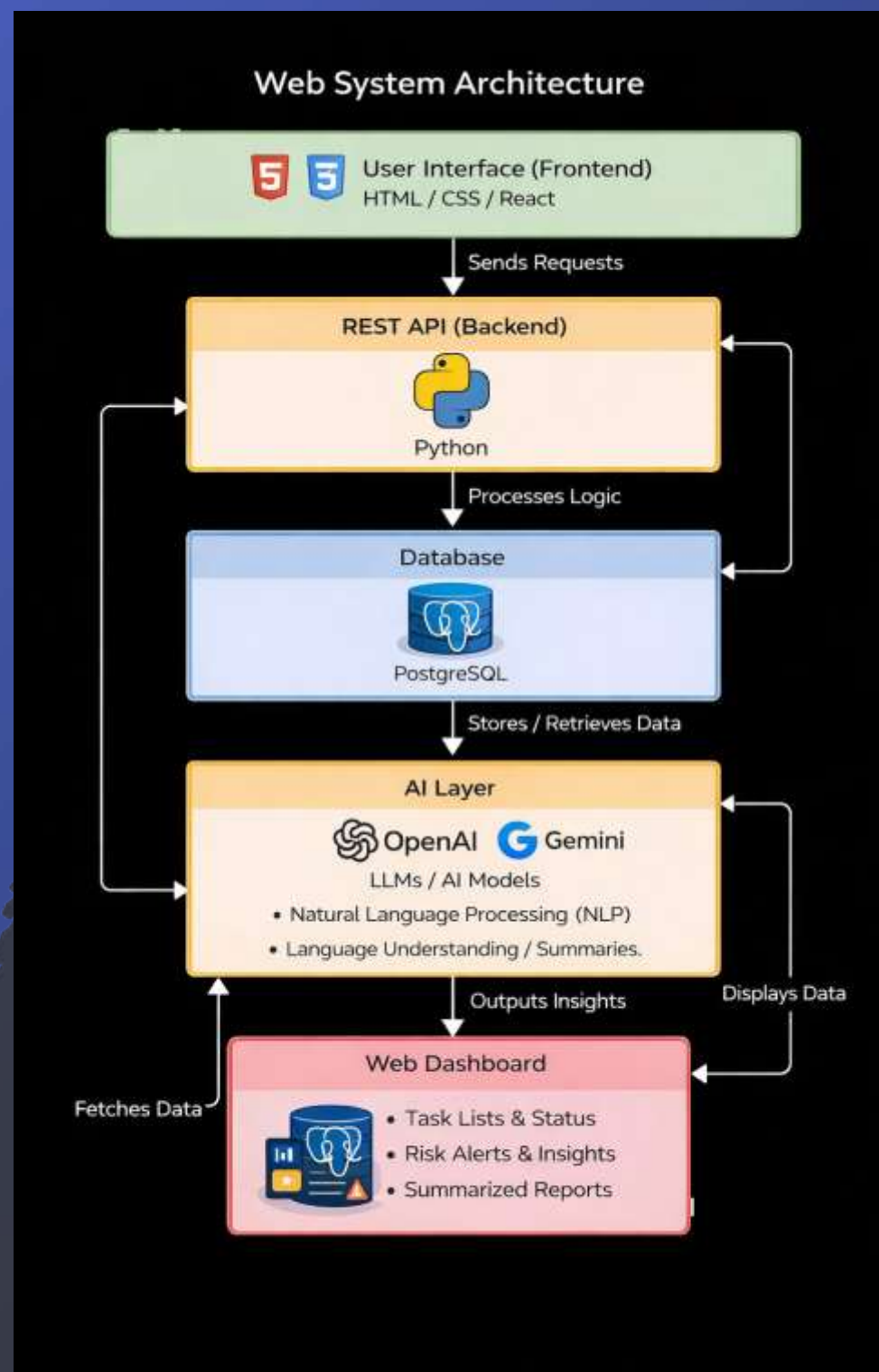- Centralized project data

FEASIBILTY

Architecture Flow

Technology Stack

- Frontend:
- HTML, CSS, React
- Backend:
- Python
- Database:
- Postgres

Feasibility

- Uses commonly available technologies
- Low infrastructure requirement
- Easy to deploy and maintain
- Suitable for small to medium teams

Why Our Solution

- Reduces manual effort
- Improves transparency
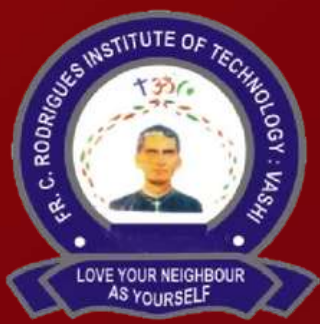- Strong base for future AI features

Comparative Analysis

Traditional Manual Tracking
- Data scattered across files/tools
- No real-time updates
- Tasks tracked manually
- Difficult to scale for large teams

Our Web-Based System
- Centralized project & task data
- Real-time updates via dashboard
- Automated task tracking
- Highly scalable architecture

IMPACT ANALYSIS

Impact on Teams

- Better task visibility
- Reduced confusion & duplication
- Improved coordination
- Clear ownership of tasks and responsibilities
- Reduced dependency on frequent meetings

Concepts Used

- MVC / layered architecture
- CRUD operations
- Authentication & routing
- Separation of concerns
- RESTful API design
- Modular and scalable system design

Impact on Organizations

- Organized project data
- Faster decision-making
- Improved productivity
- Better monitoring of project progress
- Reduced delays and operational inefficiencies

Relevant SDGs

- SDG 8:Decent Work & Economic Growth- Enhances productivity and efficient work practices
- SDG 9: Industry, Innovation & Infrastructure- Promotes digital transformation and scalable software systems

REFERENCES

Web Development & Architecture
•MDN Web Docs – HTML, CSS, JavaScript
https://developer.mozilla.org/
•React Official Documentation
https://react.dev/
•REST API Design Best Practices
https://restfulapi.net/

Backend & Database
Python Official Documentation
https://docs.python.org/3/
PostgreSQL Official Documentation
https://www.postgresql.org/docs/

AI & LLMs
•OpenAI API Documentation
https://platform.openai.com/docs
•Google Gemini Overview
https://ai.google.dev/
•Introduction to Natural Language Processing
https://www.ibm.com/topics/natural-language-processing

Software Engineering Practices
•Separation of Concerns (Software Design Principle)
https://en.wikipedia.org/wiki/Separation_of_concerns
•MVC Architecture Pattern
https://www.geeksforgeeks.org/mvc-design-pattern/