Microsoft Elevate

CAPSTONE PROJECT CNCF-Architect Pro: An Agentic AI Framework for Universal Repository Analysis

PRESENTED BY

STUDENT NAME: ROUSHNI SINHA

COLLEGE NAME: PATNA WOMEN'S COLLEGE

DEPARTMENT: MASTERS OF COMPUTER APPLICATION

EMAIL ID: ROUSHNISINHA111@GMAIL.COM

Microsoft Elevate

OUTLINE:

- **Problem**
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT:

- Developers face a steep learning curve when onboarding to complex open-source repositories.
- Existing tools provide static summaries that lack real-time data on project health and active issues.
- Manual identification of "Good First Issues" for new contributors is time-consuming and inefficient.

PROPOSED SOLUTION:

- **Autonomous AI Agent:** Developed a system that doesn't just chat but uses **tools** to fetch live data.
- **Universal Analysis:** Capable of analyzing **any** public GitHub repository dynamically.
- **CNCF Alignment:** Specialized logic to identify Cloud Native patterns and project maturity.
- **Data Persistence:** Integrated **Supabase** for secure storage of repository insights and user history.

SYSTEM APPROACH:

- **Frontend:** Next.js 15 with Tailwind CSS for a modern, responsive DevEx interface.
- **Intelligence:** Google Gemini 1.5 Flash model for reasoning and synthesis.
- **Backend:** Supabase (PostgreSQL) for reliable data management.
- **Orchestration:** Vercel AI SDK to manage **Deterministic Tool Calling**.

ALGORITHM & DEPLOYMENT:

- **Algorithm Selection:**
  - **Chosen Model:** The system utilizes a **ReAct (Reasoning and Acting) Agentic Framework** powered by the **Gemini 1.5 Flash** Large Language Model.
  - **Justification:** While traditional models like ARIMA or LSTM are used for numerical forecasting, a **Transformer-based Agent** was selected because the problem—analyzing complex software architectures—requires **natural language reasoning**, unstructured data parsing (READMEs), and real-time decision-making. The ReAct pattern allows the AI to "think" before executing specific GitHub API calls, ensuring high accuracy.

ALGORITHM & DEPLOYMENT:

- **Data Input:**
  - **Real-Time Telemetry:** The algorithm consumes live JSON streams from the **GitHub REST API** as its primary data source.
  - **Input Features:**
    - **Repository Metadata:** Stars, forks, and open issues to determine popularity and community trust.
    - **Activity Metrics:** Commit frequency and "Last Updated" timestamps to measure project velocity.
    - **Documentation Features:** Raw Markdown from README.md and CONTRIBUTING.md to assess onboarding quality.
    - **Issue Labels:** Specific filters for "good first issue" and "help wanted" to detect beginner-friendliness.

ALGORITHM & DEPLOYMENT:

- **Training Process (Inference Logic):**
  - **Zero-Shot Learning:** Unlike traditional models that require retraining on specific datasets, this agent uses **In-Context Learning (ICL)** and **Zero-Shot Chain-of-Thought (CoT)** prompting.
  - **Function Calling Optimization:** The "training" phase is replaced by a structured **System Prompt** that defines deterministic tool schemas (Zod validation). This ensures the model reliably maps natural language queries to the correct GitHub API parameters.
  - **Hyperparameter Tuning:** The model's **Temperature** is set to a low value (e.g., 0.1) to ensure deterministic, factual outputs rather than creative or "hallucinated" descriptions

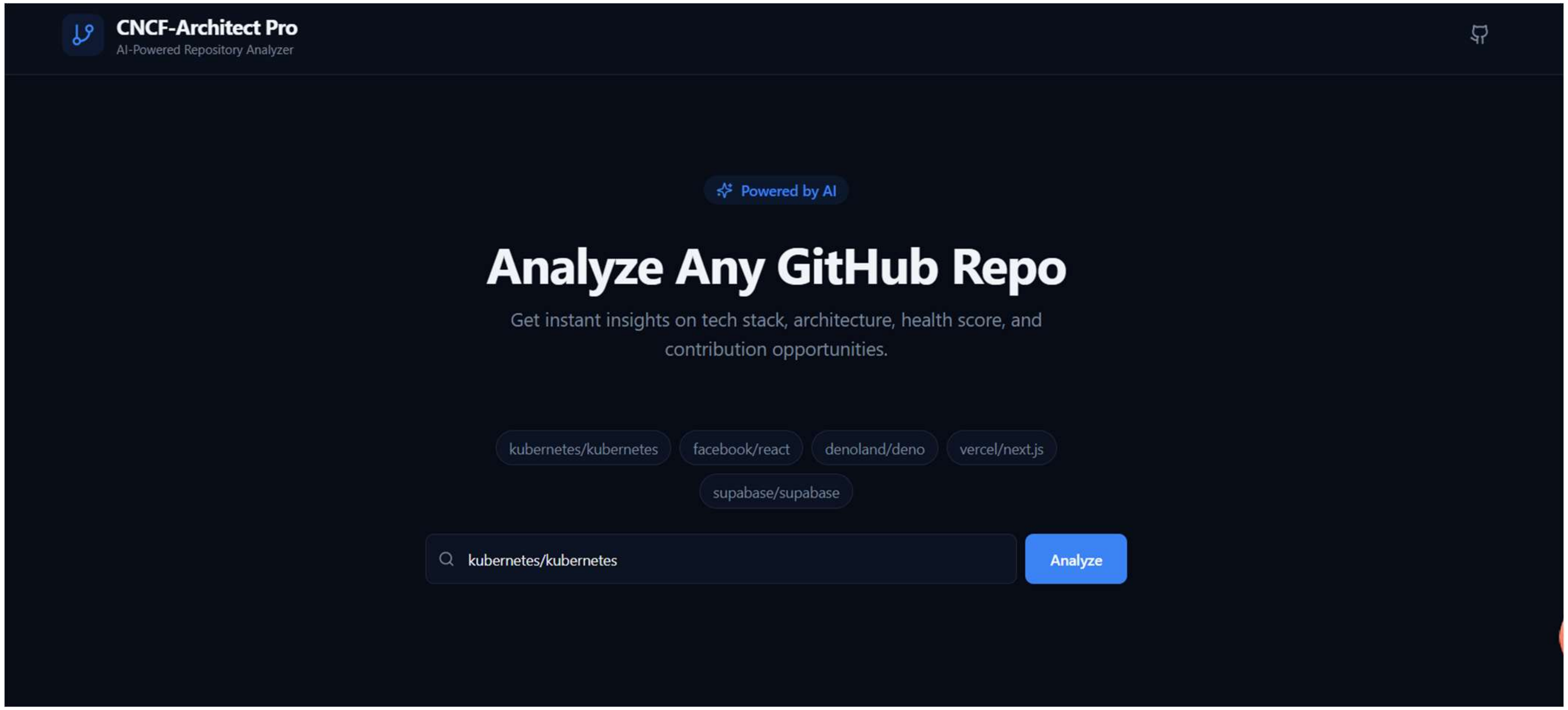
ALGORITHM & DEPLOYMENT:

- **Prediction Process (The Reasoning Loop):**
  - **Step 1 (Plan):** The agent parses the user's repository URL and identifies the required data points.
  - **Step 2 (Execution):** The agent triggers three parallel tools: analyzeRepo, fetchIssues, and fetchRepoStats.
  - **Step 3 (Synthesis):** The model synthesizes the gathered telemetry against **CNCF Graduation Criteria** to predict a **Maturity Score (0-100)**.
  - **Step 4 (Output):** The final prediction includes a structured **Health Score**, a detected **Architecture Pattern**, and a prioritized **Onboarding Roadmap** for the user.

Microsoft Elevate

RESULT:

Live application Url : https://universal-repo-explorer-ai.vercel.app/



CNCF-Architect Pro

AI-Powered Repository Analyzer

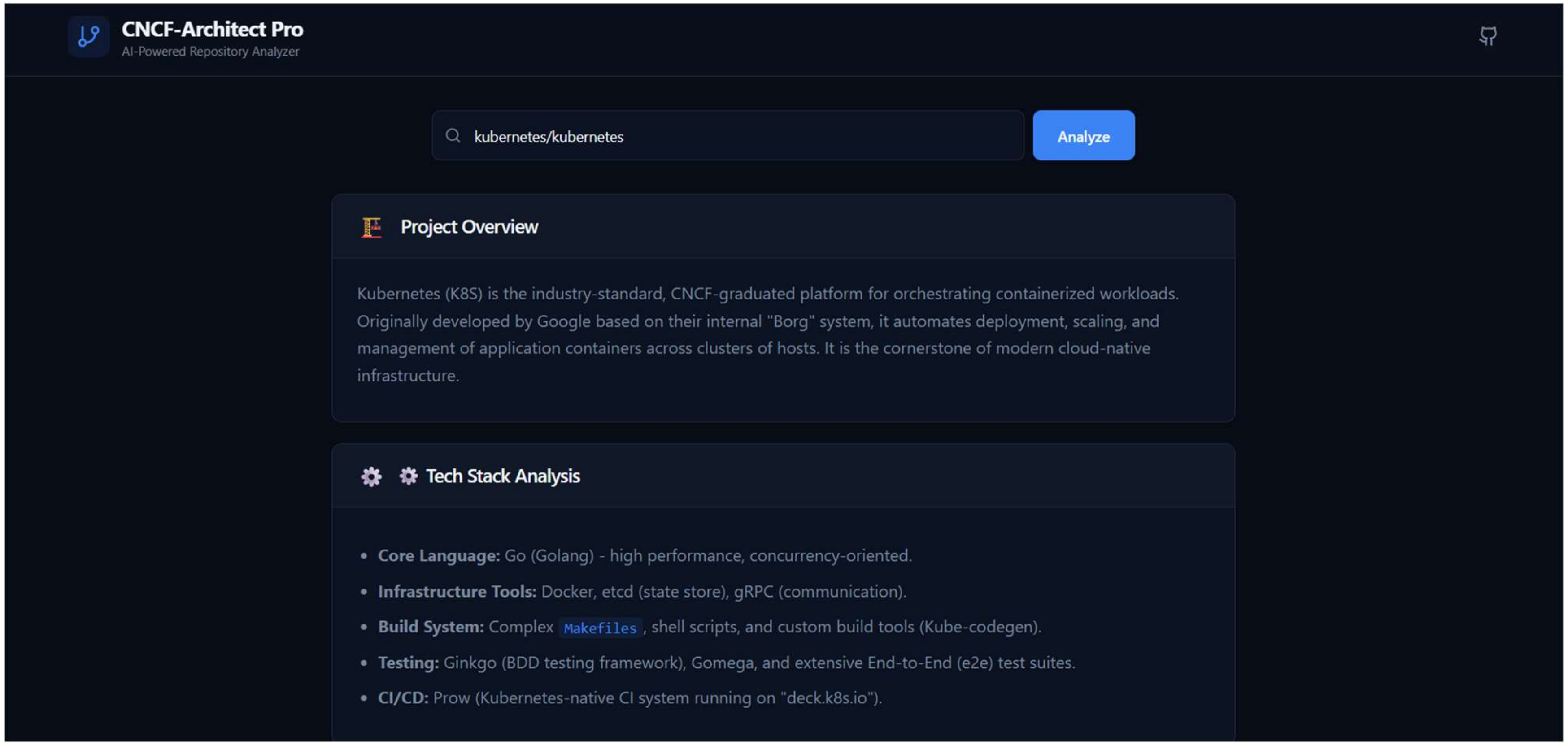
Powered by AI

Analyze Any GitHub Repo

Get instant insights on tech stack, architecture, health score, and contribution opportunities.

RESULT:

Live application Url : https://universal-repo-explorer-ai.vercel.app/



Project Overview

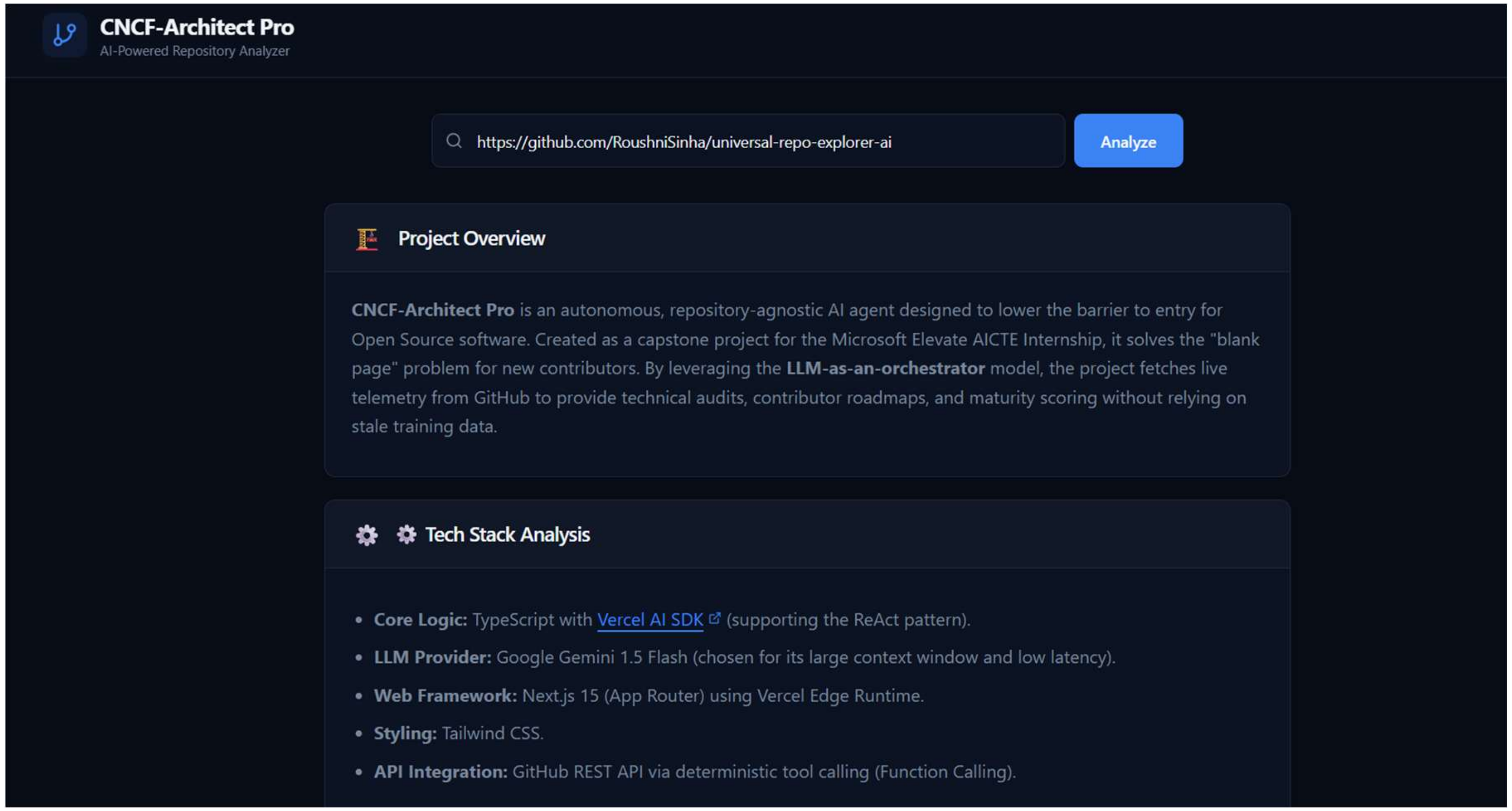
Kubernetes (K8S) is the industry-standard, CNCF-graduated platform for orchestrating containerized workloads. Originally developed by Google based on their internal "Borg" system, it automates deployment, scaling, and management of application containers across clusters of hosts. It is the cornerstone of modern cloud-native infrastructure.

⚙️ Tech Stack Analysis

- **Core Language:** Go (Golang) - high performance, concurrency-oriented.
- **Infrastructure Tools:** Docker, etcd (state store), gRPC (communication).
- **Build System:** Complex `Makefiles`, shell scripts, and custom build tools (Kube-codegen).
- **Testing:** Ginkgo (BDD testing framework), Gomega, and extensive End-to-End (e2e) test suites.
- **CI/CD:** Prow (Kubernetes-native CI system running on "deck.k8s.io").

RESULT:

Live application Url : https://universal-repo-explorer-ai.vercel.app/



CNCF-Architect Pro
AI-Powered Repository Analyzer

https://github.com/RoushniSinha/universal-repo-explorer-ai

Analyze

Project Overview

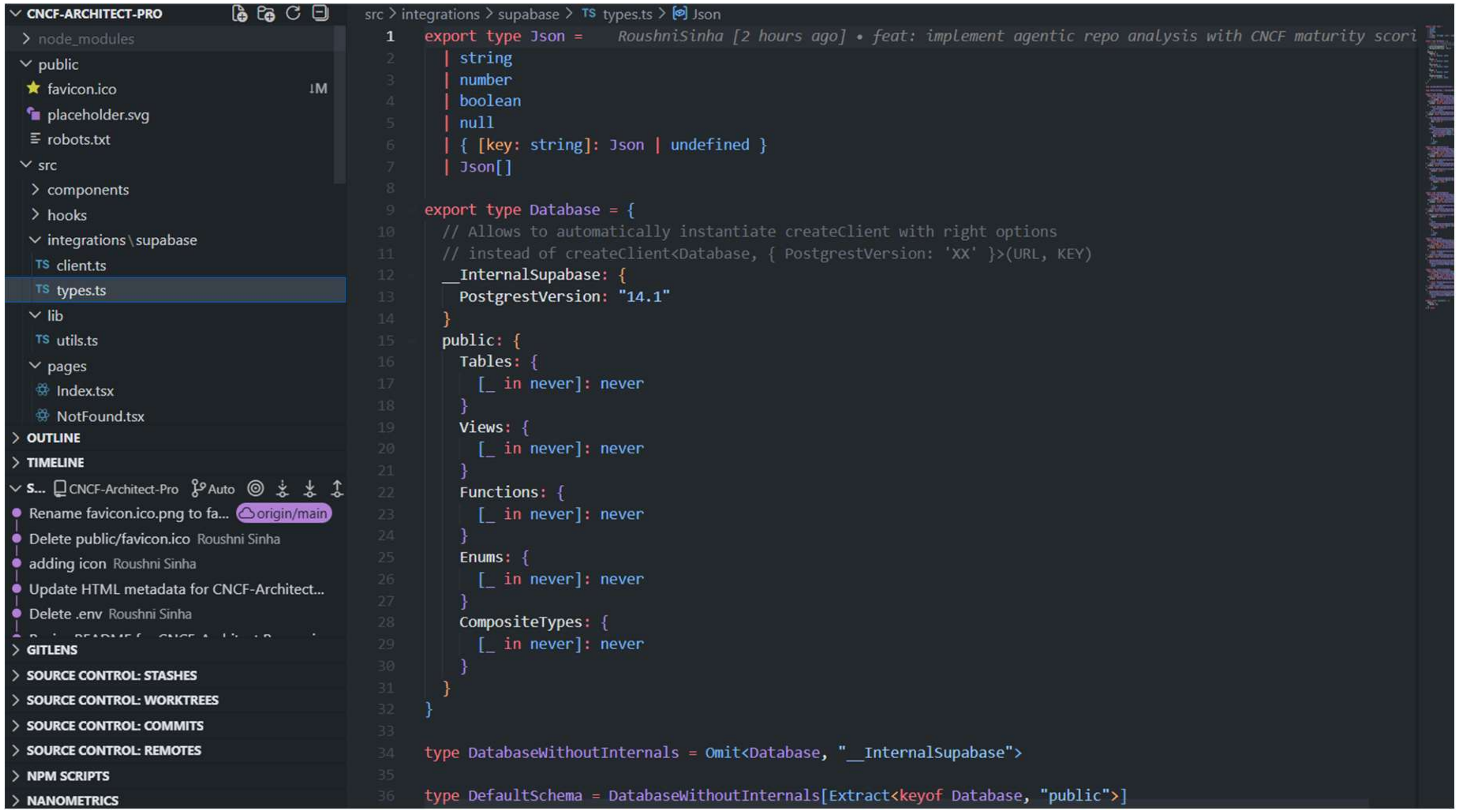
CNCF-Architect Pro is an autonomous, repository-agnostic AI agent designed to lower the barrier to entry for Open Source software. Created as a capstone project for the Microsoft Elevate AICTE Internship, it solves the "blank page" problem for new contributors. By leveraging the **LLM-as-an-orchestrator** model, the project fetches live telemetry from GitHub to provide technical audits, contributor roadmaps, and maturity scoring without relying on stale training data.

Tech Stack Analysis

- **Core Logic:** TypeScript with Vercel AI SDK (supporting the ReAct pattern).
- **LLM Provider:** Google Gemini 1.5 Flash (chosen for its large context window and low latency).
- **Web Framework:** Next.js 15 (App Router) using Vercel Edge Runtime.
- **Styling:** Tailwind CSS.
- **API Integration:** GitHub REST API via deterministic tool calling (Function Calling).

RESULT:

Live application Url : https://universal-repo-explorer-ai.vercel.app/



Rename favicon.ico.png to fa...
Delete public/favicon.ico Roushni Sinha
adding icon Roushni Sinha
Update HTML metadata for CNCF-Architect...
Delete .env Roushni Sinha

```
export type Json =    RoushniSinha [2 hours ago] • feat: implement agentic repo analysis with CNCF maturity scori
  | string
  | number
  | boolean
  | null
  | { [key: string]: Json | undefined }
  | Json[]

export type Database = {
  // Allows to automatically instantiate createClient with right options
  // instead of createClient<Database, { PostgrestVersion: 'XX' }>(URL, KEY)
  __InternalSupabase: {
    PostgrestVersion: "14.1"
  }
  public: {
    Tables: {
      [_ in never]: never
    }
    Views: {
      [_ in never]: never
    }
    Functions: {
      [_ in never]: never
    }
    Enums: {
      [_ in never]: never
    }
    CompositeTypes: {
      [_ in never]: never
    }
  }
}

type DatabaseWithoutInternals = Omit<Database, "__InternalSupabase">

type DefaultSchema = DatabaseWithoutInternals[Extract<keyof Database, "public">]
```

Microsoft Elevate

CONCLUSION:

CNCF-Architect Pro successfully demonstrates how **Agentic AI** can bridge the gap between complex software ecosystems and new developers, transforming real-time telemetry into a clear, actionable roadmap for the next generation of cloud-native contributors.

FUTURE SCOPE: – Feature Expansion & Deep Analysis

•**Comparative Repository Benchmarking:** Implement a side-by-side comparison engine to evaluate two or more repositories simultaneously. This will allow developers to compare **Project Velocity**, **Maturity Scores**, and **Issue Resolution Times** across competing CNCF projects.
•**Deep Code Intelligence with RAG:** Integrate a **Vector Database** (such as Supabase Vector) to perform **Retrieval-Augmented Generation (RAG)** on the entire codebase. This moves the agent beyond README analysis to answering deep technical questions about specific code implementations.
•**Automated Contribution Drafting:** Evolve from analysis to action by implementing **AI-driven Pull Request (PR) generation**. The agent will not only identify "Good First Issues" but also suggest initial code patches or documentation fixes to lower the entry barrier for new contributors.

FUTURE SCOPE: – Ecosystem & Community Impact

•**CNCF Landscape Categorization:** Automate the classification of repositories into specific **CNCF Landscape categories** (e.g., Runtime, Orchestration, Observability) to provide better context for cloud-native developers.
•**Governance & Security Auditing:** Enhance the scoring algorithm to check for critical community health files like GOVERNANCE.md, ADOPTERS.md, and **Security Policies**, which are required for CNCF project graduation.
•**Developer Experience (DevEx) Personalized Dashboards:** Implement user accounts using **Supabase Authentication** to allow developers to save their "Onboarding Roadmaps" and track their contribution progress across multiple organizations.
•**Predictive Health Alerts:** Use historical telemetry data to predict project "burnout" or identifying when a repository might become unmaintained based on declining commit frequency and rising issue counts.

REFERENCES:

List and cite relevant sources, research papers, and articles that were instrumental in developing the proposed solution. This could include academic papers on bike demand prediction, machine learning algorithms, and best practices in data preprocessing and model evaluation.

GitHub Link: https://github.com/RoushniSinha/universal-repo-explorer-ai

Microsoft Elevate

Thank You