Code Summarizer & Flowchart Generator

AI-Powered Solution

A revolutionary web-based application that transforms complex source code into clear summaries and visual flowcharts, empowering developers, students, and educators to understand programming logic faster and more effectively than ever before.

August 2025

The Problem We Solve

Time Intensive

Developers, students, and educators spend countless hours trying to understand raw source code, leading to slower learning curves and reduced productivity.

Error Prone

Manual code reviews and debugging are highly susceptible to human error, causing missed issues and inefficient development processes.

Limited Tools

Current market lacks comprehensive tools that combine AI-driven code summarization with visual representation of programming logic and structure.

Impact Cost

Without effective solutions, teams face increased effort in code comprehension, slower onboarding processes, and reduced overall development efficiency.

Our Solution

Smart Code Analysis

A comprehensive web-based application that accepts uploaded source code, generates plain-English summaries using advanced AI and NLP technologies, and produces visual flowcharts to represent program structure, making code comprehension effortless.

Key Features

- AI Code Summarization
- Visual Flowcharts
- Web Interface
- Export Options
- Multi-language Support

90%
Time Savings

3x
Faster Code Review

100%
Browser Compatible

Development Roadmap

Planning & Design

Weeks 1-3: Requirements gathering, system architecture design, and technology stack finalization with team role assignments.

Core Development

Weeks 4-8: Parallel backend API development using FastAPI and frontend creation with React and Tailwind CSS.

AI Integration

Weeks 9-12: Integration of ML/NLP models using LangChain and flowchart generation module with Graphviz and NetworkX libraries.

Testing & Deployment

Weeks 13-18: Comprehensive testing with PyTest and Selenium, debugging, final deployment, and documentation completion.

Feasibility Analysis

Technical Feasibility

Built on proven technologies including Python FastAPI, React.js, LangChain, and Graphviz. Uses reliable open-source tools ensuring scalability and maintainability.

Market Feasibility

Browser-based accessibility requires no installation, serving global audience of learners, educators, and professionals. Minimal development costs with strong revenue potential through premium features.

Implementation Feasibility

Clear 18-week timeline with defined team roles for backend, frontend, ML/NLP, and QA. Adheres to open-source licenses supporting inclusive educational use.