



AI Explainer and Code Generation Platform

Convert ideas into clean code — and code into clear understanding.

Quick Links




Category	Link
 Demo Video	<i>Coming Soon</i>
 Docker Support	Yes
 Supported Languages	Python · JavaScript · SQL



Table of Contents

- About the Project
- Problem Statement & Motivation
- Key Features
- Architecture
- Tech Stack
- Models Used
- Project Structure
- Installation & Setup
- Usage Guide
- Admin Controls

- History & Analytics
 - Screenshots
 - Roadmap
 - Team
 - License
-

About the Project

CodeGenie is an AI-powered software development assistant designed to help beginners and engineers write efficient code and understand complex programming concepts.

-  Developed as part of **Infosys Springboard Internship**
 -  Ideal for: Students, developers, trainers, and IT professionals
-

Problem Statement & Motivation

Programming can be hard for beginners:







- Understanding code logic takes time
- Writing correct and optimized code requires expertise

 CodeGenie removes these barriers by:

- Generating runnable code from plain English
 - Explaining any code in a simple, structured manner
 - Providing feedback & learning insights
-

Key Features

User Capabilities

Module	Description
 Secure Authentication	JWT login, refresh tokens, OTP recovery
 Text → Code Generator	Converts ideas to runnable Python / JS / SQL
 Code → Explanation	Breaks down code into logic + complexity
 Ratings & Feedback	User-powered model evaluation system
 Activity History	Track, view & revisit all past operations
 Profile Settings	Secure password updates with re-auth

Admin Features

- Manage User Roles (Max 2 Admins)
 - Analytics Dashboard:
 - Language popularity
 - Model usage heatmaps
 - Top contributors leaderboard
 - Feedback-focused insights (charts + word clouds)
 - Global search across:
 - Code, explanations, users, interactions
-

Architecture

- ✓ Modular Backend
- ✓ Secure ML integration
- ✓ Token-based user isolation

📌 Diagrams (To be added)

- System Architecture
- ER & Database Schema

Tech Stack

Component	Technology
Frontend	Streamlit
Backend	Python · FastAPI
NLP Models	Hugging Face Transformers
Database	SQLite (local backend)
Authentication	JWT + bcrypt
Email Services	SMTP with Gmail
Deployment	Docker

Models Used

Model	Use Case	Notes
CodeLlama	Code generation + refinement	Meta AI model
DeepSeek-Coder	Multi-language generation	Efficient code patterns
StarCoder2	Code rewriting & improvements	Robust token understanding

AST Parsing

Python explanation engine

Structural understanding

Tokenizers

JS & SQL code explainability

Custom tokenizer pipelines

Project Structure

CodeGenie/

```
|
|— app.py                # Streamlit UI
|— backend/
|   |— auth.py           # JWT auth + role management
|   |— generator.py      # Text-to-code engine
|   |— explainer.py      # Code explanation logic
|   |— models.py         # SQLite database
|   |— history.py        # Logs & saved records
|   |— feedback.py       # Rating + analytics
|   └— admin.py          # Admin-only routes
|
|— requirements.txt
|— Dockerfile
|— .env.example
└— docs/
    |— architecture.png
    |— flow_diagram.png
    └— screenshots/
```

Installation & Setup

Prerequisites

- Python 3.10+
- Git
- (Optional) Docker

Clone & Install

```
git clone <repo-link>
cd CodeGenie
pip install -r requirements.txt
```

Configure Environment Variables

Create `.env`:

```
JWT_SECRET_KEY=your_key_here
SMTP_EMAIL=your_mail_here
SMTP_PASSWORD=your_app_pass_here
```

Run Project

```
streamlit run app.py
```



Usage Guide

- 1 Sign up / Login
- 2 Choose module:
 - *Generate Code*
 - *Explain Code*
 - 3 Select Language
 - 4 Submit query
 - 5 Review AI output
 - 6 Rate feedback 👍
 - 7 Access previous results anytime in **History**

Screenshots included below 📌



History & Analytics

- Every session stored with:
 - Code snippet
 - Language
 - Rating
 - Timestamp
 - User identity
 - Admin dashboard visualizes:
 - Growth metrics
 - Performance chart
 - Word cloud from feedback
-



Screenshots

(To be added with real UI images)

Suggested image list:

- Login Page
 - Dashboard
 - Code Generation Module
 - Code Explanation Module
 - Admin Analytics Page
 - History Log UI
-

Team

Name	Role	Responsibilities
<i>Add Names</i>	ML Engineer	Model integration + optimization
<i>Add Names</i>	Backend Developer	Auth, DB, API
<i>Add Names</i>	Frontend Developer	UI/UX with Streamlit
<i>Add Names</i>	Documentation	Report, README, PPT

License

MIT License — Free to reuse with attribution.