**CodeSense - Automated Code Review System**

**📌 Overview**

CodeSense is an AI-powered **automated code review system** that analyzes Python code for **quality, security, and style issues**. It uses **Pylint, Bandit, and Flake8** to detect problems and **AI (GPT-4)** to suggest improvements.

**📜 Features**

✅ Upload Python files for analysis. ✅ Check for **syntax errors, PEP8 violations, security risks**. ✅ AI-powered **code improvement suggestions**. ✅ Generate **detailed reports** (view/download as PDF). ✅ Real-time feedback & CI/CD integration.

**🛠️ Tech Stack**

**Backend:**

* Python (Flask)
* Pylint, Bandit, Flake8 (for code analysis)
* OpenAI GPT-4 API (for AI suggestions)
* SQLite (for storing reports)

**Frontend:**

* React (JavaScript, JSX, Tailwind CSS)

**Deployment:**

* Backend: Render/Heroku
* Frontend: Netlify

**Other Tools:**

* GitHub Actions/Jenkins (for CI/CD integration)
* Docker (for deployment)

**📂 Project Structure**

CodeSense/

├── backend/ # Flask API

│ ├── app.py # Main backend logic

│ ├── analyze.py # Code analysis module

│ ├── ai\_suggestions.py # AI-powered code suggestions

│ ├── models.py # Database models (SQLite)

│ └── utils.py # Helper functions

│

├── frontend/ # React App

│ ├── src/ # React components

│ ├── public/ # Static files

│ ├── package.json # Dependencies

│ ├── index.js # Entry point

│ └── App.js # Main component

│

├── uploads/ # Uploaded code files

├── reports/ # Generated reports

├── tests/ # Unit & integration tests

└── README.md # Documentation

**🚀 Installation & Setup**

**1️⃣ Clone the Repository**

git clone https://github.com/yourusername/CodeSense.git

cd CodeSense

**2️⃣ Backend Setup (Flask & SQLite)**

cd backend

python -m venv venv

source venv/bin/activate # On Windows: venv\Scripts\activate

pip install -r requirements.txt

**3️⃣ Frontend Setup (React & Netlify)**

cd frontend

npm install

npm start # Runs React app locally

**4️⃣ Set Up OpenAI API Key**

Sign up at [OpenAI](https://openai.com/) and get an **API Key**.

export OPENAI\_API\_KEY='your-api-key-here' # On Windows: set OPENAI\_API\_KEY=your-api-key-here

**5️⃣ Run the Application**

**Start Backend:**

cd backend

python app.py

Access it at **http://127.0.0.1:5000/**

**Start Frontend:**

cd frontend

npm start

Access it at **http://localhost:3000/**

**🔍 How It Works?**

1️⃣ **Upload** a Python file. 2️⃣ **Code Analysis**: Runs Pylint, Bandit, and Flake8. 3️⃣ **AI Suggestions**: GPT-4 suggests fixes & improvements. 4️⃣ **View Report**: Get detailed feedback & download as PDF. 5️⃣ **CI/CD Integration** (Optional): Auto-review code on GitHub.

**📜 API Endpoints**

**1️⃣ Upload File**

**Endpoint:** /upload

* **Method:** POST
* **Description:** Uploads a Python file for review.

**2️⃣ Analyze Code**

**Endpoint:** /analyze/<filename>

* **Method:** GET
* **Description:** Runs Pylint, Bandit, and Flake8 on the uploaded file.

**3️⃣ Get AI Suggestions**

**Endpoint:** /suggest/<error\_message>

* **Method:** GET
* **Description:** Uses GPT-4 to suggest improvements for code errors.

**🛡️ Security Considerations**

* **Sanitize file uploads** to prevent malicious scripts.
* **Limit API requests** to prevent abuse.
* **Use environment variables** for sensitive data.
* **Integrate OAuth/GitHub Authentication** for access control.

**🚀 Future Enhancements**

✅ **Multi-language support** (JavaScript, Java, C++) ✅ **VS Code/JetBrains Plugin** ✅ **GitHub Bot for PR Reviews** ✅ **Cloud Deployment (AWS/GCP)**

**📞 Contact & Contributions**

🤝 **Want to contribute?** Fork the repo and submit a PR! 📧 **Questions?** Reach out at your-email@example.com