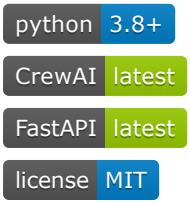
# AI Code Review Crew




A multi-agent AI system for automated code review. Five specialized agents collaborate to analyze Python code for bugs, security vulnerabilities, performance issues, and documentation quality.

## The Agents

1. **Code Analyst** — Identifies logical errors, edge cases, and exception handling
2. **Security Expert** — Scans for OWASP Top 10 vulnerabilities and security flaws
3. **Performance Optimizer** — Detects algorithmic bottlenecks and inefficient patterns
4. **Documentation Specialist** — Reviews docstrings and code comments
5. **Quality Assurance** — Compiles final report with recommendations

## Features

- **Dark Mode UI** — Modern FastAPI interface with drag-and-drop file upload
- **GitHub Integration** — Clone and analyze public repositories directly
- **Real-time Agent Status** — See which agent is currently analyzing your code
- **Detailed Reports** — Markdown reports with severity levels, line numbers, and suggested fixes
- **Multi-Agent Workflow** — Sequential task execution with context sharing

## Project Structure

```
3.Crew_AI_projects/
├── app.py                 # FastAPI application
├── src/
│   ├── crew.py           # CrewAI orchestration logic
│   ├── logger.py         # Logging configuration
│   ├── config/
│   │   ├── agents.yaml   # Agent definitions
│   │   ├── tasks.yaml    # Task definitions
│   │   └── settings.py   # Application settings
│   └── static/           # Frontend
│       ├── index.html    # Main UI
│       ├── app.js        # JavaScript
│       └── style.css     # Dark mode styling
├── examples/             # Sample files for testing
├── output/               # Generated reports
├── logs/                 # Application logs
└── requirements.txt      # Python dependencies
```

## 🚀 Getting Started

### Prerequisites

- Python 3.8+
- OpenAI or Anthropic API key

### Installation

1. Clone the repository

```
git clone <your-repo-url>
cd 3.Crew_AI_projects
```

2. Install dependencies

```
pip install -r requirements.txt
```

3. Set up environment variables

```
cp .env.example .env
# Edit .env and add your OPENAI_API_KEY or ANTHROPIC_API_KEY
```

## 💻 Usage

### Running the Application

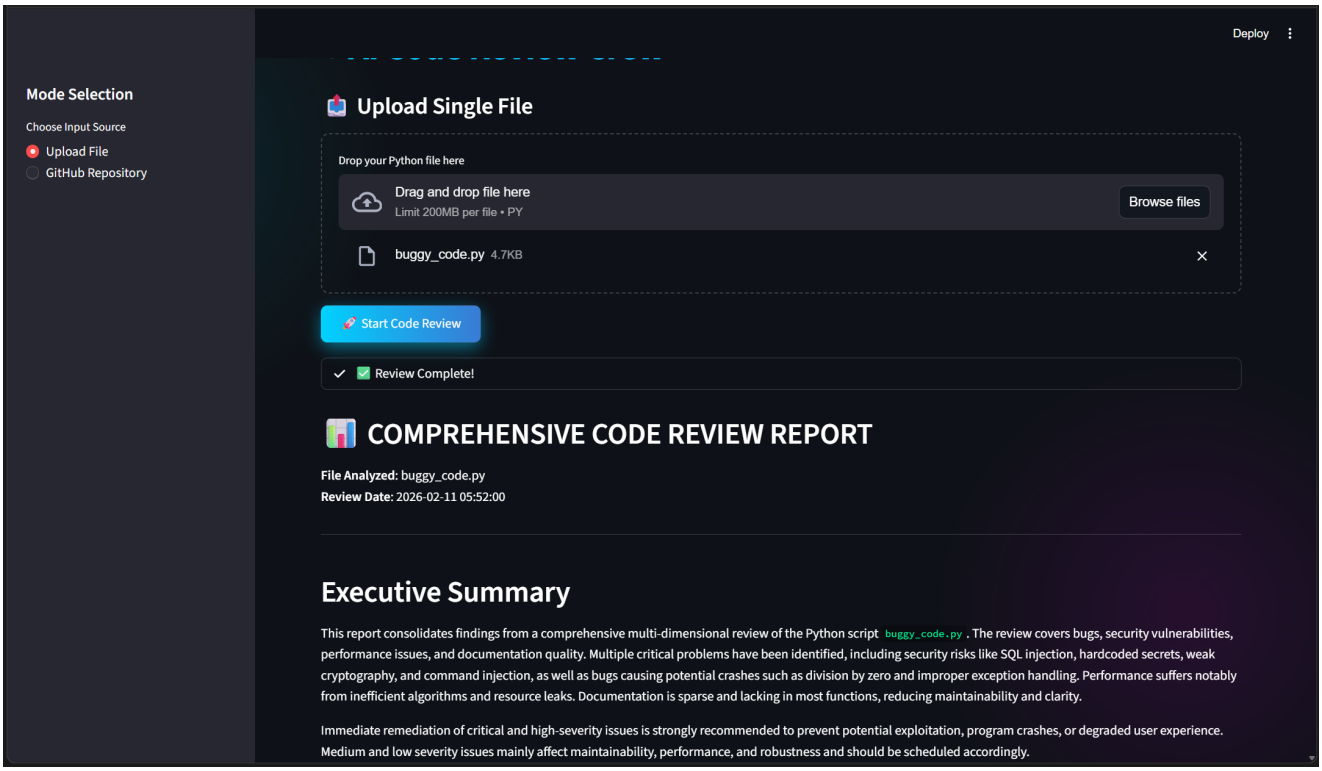
```
python app.py
```

The application will start on `http://localhost:8000`

## 📡 API Endpoints

- `GET /` - Serve main HTML page
- `GET /health` - Health check endpoint
- `POST /api/review/upload` - Review uploaded Python file
- `POST /api/review/github` - Review GitHub repository
- `GET /api/files/list` - List Python files in a GitHub repository

## 📸 Screenshots




*Code Review Interface showing code analogy*

### Upload a File

1. Select "Upload File" mode
2. Drop your `.py` file
3. Click "Start Code Review"
4. Download the generated report

### Review a GitHub Repo

1. Select "GitHub Repository" mode
2. Paste a public repo URL
3. Select files to analyze
4. Click "Analyze Selected Files"

## Docker Deployment

### Build and Run

```
docker build -t ai-code-review .
docker run -p 8000:8000 --env-file .env ai-code-review
```

## License

MIT License