AI Dev Team – Project Overview

Neotron Productions

1. What is AI Dev Team?

- **AI-powered development crew** that:
  - Reads GitHub issues
  - Analyzes your codebase
  - Proposes implementation plans and solutions
- **Built on**: CrewAI + GitHub integration
- **Goal**: Make it easy to point an "AI dev team" at **any** GitHub repo and start getting structured help.

---

2. Core Capabilities

- **Issue analysis**
  - Understands requirements and context from GitHub issues
  - Identifies related code, PRs, and discussions
- **Codebase awareness**
  - Uses semantic GitHub search to find relevant files and patterns
  - Leverages project context (README, tech stack, project structure)
- **Actionable output**
  - Produces clear implementation plans
  - Suggests code changes and file locations

---

3. High-Level Architecture

- **Scripts layer (`scripts/`)**
  - `main.py`: single entrypoint for new users
  - `github_crew.py`: full-featured multi-agent crew for GitHub issues
  - `example_github_issue.py`: minimal example using a single agent
  - Setup and test helpers for GitHub and Projects V2
- **Core logic (`crew_runner/`)**
  - Tools for applying changes, Git operations, coverage, logging, and safety
- **Documentation (`docs/`)**

  - Quickstart, usage guides, context guide, dashboard guide, troubleshooting

---

4. Onboarding Flow for New Users

4.1 Prerequisites

- Python 3.10+ (recommended)
- GitHub account
- GitHub Personal Access Token with:
  - `repo`
  - `read:org`

4.2 Clone & Install

```
git clone git@github.com:NeotronProductions/ai-dev-team.git
cd ai-dev-team
python3 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

---

5. One-Command Initialization (Main Entry)

5.1 Run the main initializer

```
cd ai-dev-team
source .venv/bin/activate
python3 scripts/main.py
```

The `main.py` script will:

- **Prompt for**:
  - `GITHUB_TOKEN`
  - `GITHUB_REPO` (e.g., `owner/repo`)
  - Optional `GITHUB_PROJECT_ID` (GitHub Projects V2)
- **Write values to** `.env` in the repo root
- **Run a simple GitHub issue analysis crew** against your repository

---

6. Configuration (.env)

Example `.env`:

```
GITHUB_TOKEN=ghp_your_token_here
GITHUB_REPO=owner/repo
```

```
GITHUB_PROJECT_ID=PVT_your_project_v2_id   # optional
OPENAI_API_KEY=your_openai_key_here        # optional
```

- `.env` is **git-ignored** – safe to store local secrets
- `GITHUB_PROJECT_ID` lets the crew prioritize issues from a specific Project V2 board via prompt instructions

---

7. Main Execution Modes

7.1 Simple one-agent issue analysis

- Script: `scripts/example_github_issue.py`
- Use when you want:
  - A quick analysis of a single issue
  - Or a scan of top open issues

```
source .venv/bin/activate
python3 scripts/example_github_issue.py owner/repo 123
# or, analyze open issues
python3 scripts/example_github_issue.py owner/repo
```

7.2 Multi-agent GitHub crew

- Script: `scripts/github_crew.py`
- Agents:
  - **GitHub Issue Manager** – reads and structures the problem
  - **Code Analyst** – analyzes code and proposes implementation plans

```
source .venv/bin/activate
python3 scripts/github_crew.py 123 owner/repo
```

---

8. GitHub Projects (Optional)

- You can connect a **GitHub Projects V2 board** via `GITHUB_PROJECT_ID`
- Benefits:
  - Prioritize work based on project fields (status, priority, etc.)
  - Keep the AI crew focused on issues that are actually on your board
- Helper scripts:
  - `scripts/test_project_access.py` – check accessible projects
  - `scripts/test_graphql_projects.py` – inspect Projects V2 via GraphQL
  - `scripts/test_project_fields.py` – see project fields/options

---

9. Safety and Workflow Support

- **Path and Git safety**
  - Utilities to avoid destructive changes and ensure safe paths
- **Patch-based changes**
  - Changes are typically expressed as patches, easy to review
- **Coverage & testing helpers**
  - Scripts and docs to help verify changes and run tests
- **Rich documentation**
  - Troubleshooting guides, dashboard usage, workflow continuation

---

10. Typical Demo Scenario

1. Clone `ai-dev-team` and install dependencies
2. Run `python3 scripts/main.py`
3. Enter:
   - GitHub token
   - Target repo (e.g., `your-org/your-app`)
   - Optional Projects V2 board ID
4. Provide an issue number when prompted
5. Review the generated:
   - Issue analysis
   - Code context
   - Implementation plan

This makes it easy for others to **try the project on their own GitHub repo** with minimal setup.