

Project Development Phases – GitHub Project Generator

This document outlines the complete development lifecycle of the GitHub Project Generator. Each phase represents a logical evolution of the system, introducing new features, architectural improvements, and production-level refinements.

Phase 1: Backend Foundation

Goal: Establish the core backend infrastructure.

- Initialized FastAPI application structure and entry point.
- Configured SQLAlchemy engine and session management.
- Designed initial database models and Pydantic schemas.
- Established modular project directory structure for scalability.

Phase 2: Authentication & Security

Goal: Implement secure user access and identity handling.

- Integrated GitHub Personal Access Token verification.
- Introduced User database model for persistent identity tracking.
- Developed authentication routes for login and token validation.
- Implemented dependency-based user resolution using FastAPI injection.

Phase 3: Repository Management

Goal: Enable seamless interaction with GitHub repositories.

- Fetched authenticated user repositories using GitHub REST API.
- Implemented repository persistence using a local database.
- Designed Repository model and validation schemas.
- Created REST endpoints for repository listing and connection.

Phase 4: Project Generation & AI Integration

Goal: Implement core business logic using AI-driven README parsing.

- Integrated Ollama to run large language models locally.
- Developed README parsing service for extracting structured project metadata.
- Exposed project generation endpoints triggered via repository ID.
- Optimized inference timeout and request handling for AI workloads.

Phase 5: Frontend & Configuration

Goal: Deliver a user interface and centralize configuration.

- Developed Streamlit-based frontend for end-user interaction.
- Connected frontend workflows with backend REST APIs.
- Refactored configuration management to use environment variables.
- Added helper scripts for database inspection and server startup.

Phase 6: Polish & Documentation

Goal: Finalize the system and ensure maintainability.

- Added comprehensive docstrings and inline documentation.
- Validated imports, dependencies, and project structure.
- Improved code readability and consistency.
- Finalized technical and user-facing documentation.