ABSTRACT

Traditional code generation tools have increasingly integrated AI models, particularly Large Language Models (LLMs), to assist developers. However, these tools often generate isolated and context-limited code snippets, lacking modularity, structure, and real-world project completeness. Developers are still required to perform manual scaffolding, debug outputs, and ensure the alignment of generated code with business logic and architectural needs. Furthermore, the absence of integrated documentation, testing, and error-handling pipelines leads to inefficiencies and delays during the software development lifecycle.

Our project, **CodeCodez**, aims to fill these gaps by introducing an **end-to-end AI-powered software project generation platform**. By leveraging advanced LLMs, CodeCodez takes detailed user specifications or prompts and transforms them into **fully structured, production-ready projects**. The platform automatically generates:

- organized directory structures,
- configuration files,
- modular and context-aware code,
- unit and integration test cases,
- as well as dynamic documentation.

Additionally, it integrates error detection and correction logic to refine code during generation, reducing post-generation maintenance.

It seamlessly fits into modern **CI/CD pipelines**, enabling developers to plug the tool into their existing workflows without disruption. No dedicated CLI is built; instead, integration is achieved through modular hooks and API-based interaction within common development environments like VS Code, GitHub, or web-based editors.

By automating redundant processes and streamlining workflows, CodeCodez is expected to **enhance developer productivity, reduce human error, and accelerate the transition from idea to deployment**. The long-term goal is to establish a flexible, scalable platform that empowers teams to prototype and deliver software more efficiently while maintaining quality, modularity, and maintainability.