1.5 Assumptions and Constraints

S. No.	Assumptions and Constraints
1	It is assumed that users will provide clear and unambiguous natural language or structured input specifications. The system's accuracy depends significantly on the quality of these inputs. Additionally, it is assumed that users have a basic understanding of software project architecture, which aids in interpreting and validating the generated outputs.
2	It is assumed that API access to the LLMs (such as OpenAI, Gemini, etc.) will remain stable and uninterrupted during the generation process. If access limits or token quotas are reached, fallback handling will be triggered. It is also assumed that the API keys are correctly configured and valid throughout the project duration.
3	Since this is a multi-agent LLM-based architecture, it is assumed that each LLM used is properly optimized or fine-tuned for its specific role, such as frontend, backend, documentation, or testing. A constraint here is the limitation of using general-purpose LLMs for highly domain-specific code which may lead to occasional inaccuracies.
4	It is assumed that the underlying hardware (local GPU or cloud-based compute) will be capable of handling concurrent LLM inferences. In scenarios where resources are limited, the performance may degrade. Hence, compute efficiency and memory optimization techniques have been considered as constraints.
5	A key constraint is the dependency on external libraries, frameworks, or tools used by the generated code (e.g., React, Flask, Docker). It is assumed that the target environment has these dependencies pre-installed or can install them using standard package managers.
6	It is assumed that users will test the generated project outputs using the integrated test framework before deploying in production. While basic unit tests and integration tests will be auto-generated, exhaustive edge-case testing remains a user responsibility.
7	One constraint is that not all programming languages or frameworks are supported in the MVP version of the system. Initially, only mainstream stacks (e.g., Python, JavaScript) will be available, and support for additional languages will depend on community or user feedback.
8	It is assumed that version control systems like Git are available on the user's system. The platform assumes the user can initialize repositories and commit generated code when required. Integration with GitHub/GitLab remains optional but encouraged.

TABLE 1: Assumptions and Constraints