2. In-Scope Functionalities

The following functionalities are within the scope of this project:

- **Requirement Interpretation:** Convert user-defined natural language inputs into structured software requirements.
- **Automated Code Generation:** Produce modular, production-ready code including directory structures, configuration files, and reusable components.
- **Hierarchical Task Decomposition:** Apply tree and graph-based planning methods to break down software requirements into logically ordered subtasks.
- **Specialized LLM Agents:** Assign subtasks to LLMs specialized in frontend, backend, testing, or documentation to ensure domain-specific accuracy.
- **Real-Time Testing and Error Correction:** Embed testing mechanisms to validate code correctness and automatically resolve common errors.
- **Documentation Generation:** Auto-generate maintainable documentation alongside each code module for transparency and usability.
- **Scalable Architecture:** Provide a modular system design that allows seamless integration of additional programming languages, frameworks, or AI agents.

3. Intended Benefits

The project is expected to deliver the following benefits:

- **Time Efficiency:** Drastically reduce the time required to move from idea to prototype.
- **Consistency & Quality:** Maintain coding standards through automated linting, formatting, and modular structuring.
- **Reduced Cognitive Load:** Free developers from repetitive coding and allow focus on innovation and creative problem-solving.
- **Scalability:** Enable future integration with additional AI agents, tools, or frameworks without major re-engineering.
- **Accessibility:** Make professional-quality code generation available to non-experts such as students, researchers, or startups.

4. Future Scope (High-Level Vision)

While the present project focuses on building a functional prototype covering requirement interpretation, task decomposition, automated coding, testing, and documentation, future extensions may include: