- **Task Decomposition:** Parsing natural language specifications and breaking them down into subtasks using tree-based and graph-based methods (e.g., DAGs) for effective orchestration.
- **Multi-LLM Coordination:** Assigning domain-specific subtasks (frontend, backend, testing, documentation) to specialized LLMs and coordinating their outputs through an orchestration layer.
- **Built-in Testing and Validation:** Auto-generating test cases and performing iterative error correction based on test results.
- **Auto-generated Documentation:** Creating markdown-based documentation for each module to improve developer experience and maintainability.
- **Plug-and-Play Outputs:** Delivering ready-to-run projects that can be directly deployed on cloud environments or local machines, compatible with Git and CI/CD systems.

Out of Scope (Current Phase):

- **Realtime UI feedback or GUI-based prompt engineering tools** (CLI will be used).
- **Training new LLMs from scratch** — The system will utilize pre-trained LLMs through APIs.
- **Complex multi-repository management** — Current focus is on generating single-repo modular projects.
- **Dynamic runtime debugging or monitoring tools** — Not included in this phase.
- **Cross-language orchestration** — Initial scope focuses on Python, JavaScript (React/Node.js) and their ecosystem.