

**order, and fallback mechanisms**, thereby creating a coherent pipeline from requirement to deployment.

- **Limited Specialization of LLM Agents for Subtask Execution**

Recent works such as Chain-of-Agents [1] and Corex [2] have laid the foundation for multi-agent collaboration among LLMs. However, in most cases, agents still lack fine-grained specialization. For example, agents responsible for frontend logic, backend logic, and documentation often share the same model weights and capabilities, leading to inconsistencies and redundant reasoning. This reveals a need for **domain-specific agent specialization**, where each LLM is tuned or selected based on its strength in a specific component.

- **Inadequate Error Recovery and Self-Correction Mechanisms**

Many frameworks lack robust self-repair mechanisms when generated code fails to compile or pass unit tests. Although some systems, such as Mixture-of-Agents [4] and AgentDropout [8], propose resilience through redundancy and dynamic pruning, **automated test-triggered correction loops** are either missing or simplistic. Incorporating feedback from real-time errors into prompt chains for iterative refinement remains an open challenge.

- **Sparse Focus on Developer Usability and Real-World Integration**

While theoretical models show promise in simulated environments [10], [19], most systems do not ensure that their outputs are aligned with real-world developer workflows, such as CI/CD pipelines, Git integration, Docker environments, or framework-specific conventions. This limits the practical applicability of the generated systems. A user-facing system like CodeCodez must close this gap by outputting **plug-and-play codebases**, complete with deployment scripts and documentation.

These gaps provide a clear motivation for the proposed work, which uniquely blends hierarchical decomposition, dynamic task graphs, agent specialization, automated testing, and contextual code packaging into a single, cohesive platform for real-world software project generation.