- **Demonstration and Prototype**

As part of the mid and final deliverables, a working prototype will be presented showcasing end-to-end generation of at least two sample software projects. These demos will validate the accuracy, completeness, and efficiency of the developed system.

1.10 Novelty of Work

The proposed system introduces a novel methodology in the domain of automated software generation by combining hierarchical task decomposition with multi-agent collaboration among specialized Large Language Models (LLMs). Unlike conventional AI-based code generation tools that operate on prompt-level interactions or token-wise synthesis, this project implements a **category-driven and structure-aware approach** to task formulation and execution. Rather than relying solely on surface-level token prediction, tasks are **first categorized** into meaningful components such as UI, backend logic, data schema, testing, and documentation. These are then decomposed using **tree and graph-based planning strategies**, offering far more control over execution flow, parallelism, and dependency resolution.

A key innovation lies in how the system **assigns decomposed tasks to role-specific LLMs**. Each subtask is matched to an LLM agent that is either trained, fine-tuned, or contextually prompted to specialize in that specific domain. This breaks from the standard practice of using a single monolithic LLM, which often struggles with instruction drift, memory limitations, or logical inconsistency over long and multi-part generations. By using a **multi-agent orchestration layer**, the system not only distributes the cognitive load across models but also introduces **self-verification cycles**, where outputs are tested and iteratively refined before inclusion.

Furthermore, the system emphasizes **end-to-end automation**, going beyond code generation to include project scaffolding, test integration, documentation creation, and developer-friendly output formatting. This holistic approach mirrors real-world software engineering workflows and drastically reduces the manual overhead typically required in stitching together individual code components. Unlike traditional tools that offer mere suggestion-based assistance or code snippets, this project delivers **deployment-ready repositories**, including Docker support, CI/CD compatibility, and consistent API patterns.