Workable Modules/Products

From Fig. 1, the project will yield the following **workable modules/products**:

1. **Requirement Specification Document** – outcome of requirement gathering.
2. **System Architecture and Design Blueprint** – defines technical backbone.
3. **Backend API Services** – APIs and middleware enabling communication.
4. **AI Task Decomposition Engine** – core agent responsible for breaking down natural language requirements.
5. **Subtask Generator and Orchestrator** – module assigning subtasks to specialized LLMs.
6. **Command Line Interface (CLI)** – for terminal-based project creation.
7. **Testing Framework** – comprehensive suite for unit, beta, and feedback testing.
8. **Logging and Monitoring Module** – automated error and output tracking system.
9. **Integrated Frontend + Backend** – consolidated application layer for usability.
10. **Deployment Package** – final product with GitHub export and delivery-ready format

3.4 Tools and Technology Used

The *CodeCodez* project leverages a combination of modern programming languages, frameworks, and AI tools to enable end-to-end automation of software development. The selection of tools and technologies has been guided by scalability, ease of integration, and compatibility with industry standards.

1. Programming Languages

- **Python –** Primary language for AI agent integration, orchestration, and backend logic. Python's extensive ecosystem (LangChain, FastAPI, Flask) makes it ideal for LLM workflows.
- **Shell/Bash –** For deployment scripts, command execution, and automation in terminal environments.

2. AI and Machine Learning Tools

- **Large Language Models (LLMs) –** GPT-based models and other open-source LLMs are used for requirement interpretation, task decomposition, and code generation.