**Project Design Phase**

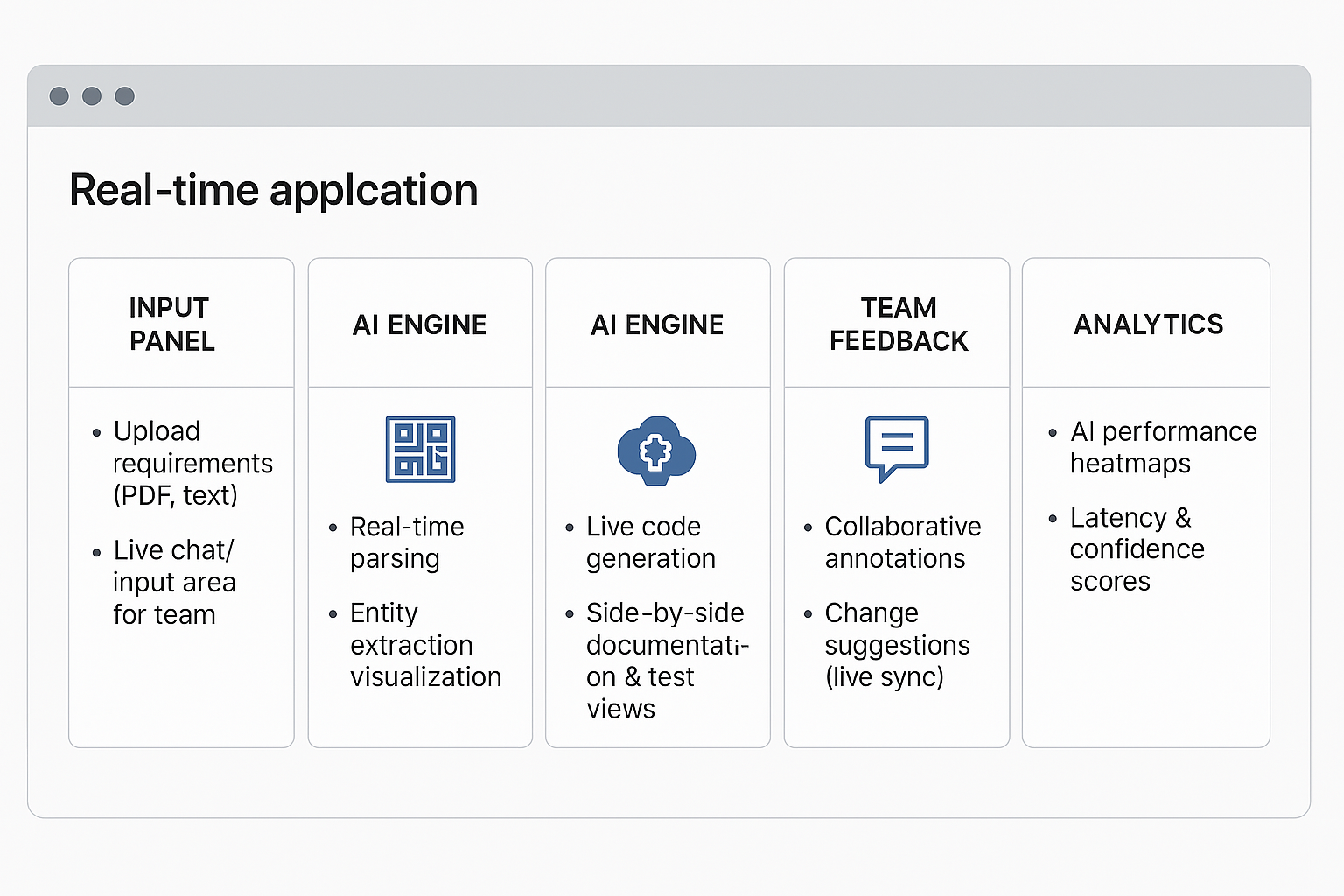
**Solution Architecture**

|  |  |
| --- | --- |
| Date | 27 June 2025 |
| Team ID | LTVIP2025TMID35544 |
| Project Name | smartsdlc – AI-enhanced software development lifecycle |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

SmartSDLC is built on a modular, cloud-native, and API-first architecture that intelligentlyautomates each phase of the SDLC. It begins with multi-channel user inputs—like PDFs, natural language prompts, or API calls—which are processed through an NLP layer to extract structured requirements. These are then routed through an AI orchestration engine that generates code, resolves bugs, writes test cases, and summarizes code contextually. The output is served through web UIs, IDE plugins, or CI/CD hooks, and all interactions feed into a knowledge layer for continuous improvement. Its scalable, plug-and-play design enables seamless integration across diverse teams and workflows.

**Example - Solution Architecture Diagram:**



**Reference:** [**https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/**](https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/)

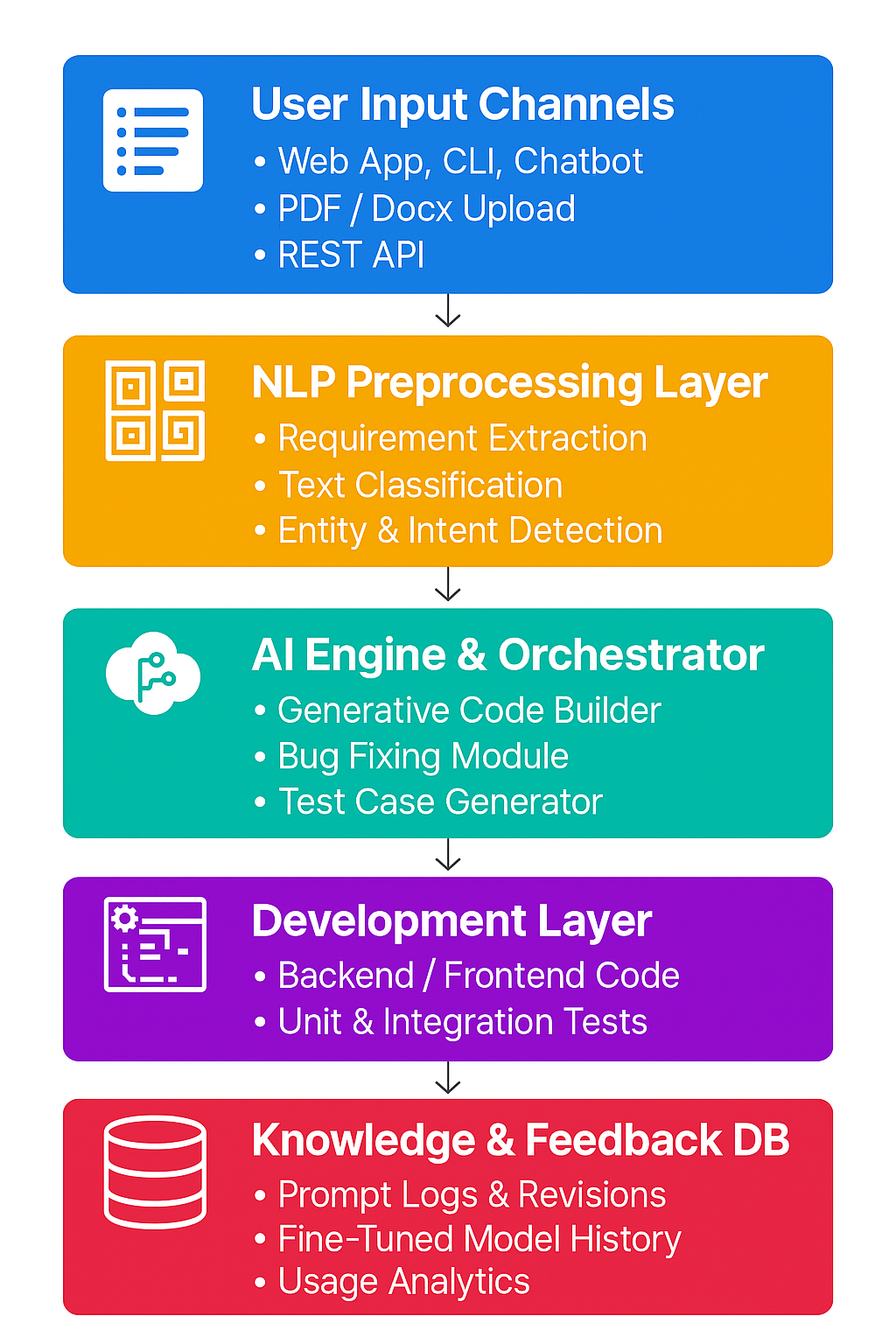
SmartSDLC – Architecture Summary :

SmartSDLC is architected as a modular, AI-first platform that intelligently automates the entire software development lifecycle. It begins by capturing inputs from multiple channels—such as natural language prompts, file uploads, chat interfaces, or API integrations—making it accessible to both technical and non-technical users. These inputs flow into a robust NLP layer, which extracts, classifies, and structures requirements using advanced entity recognition and intent parsing.

Once structured, the data is handed off to a powerful AI orchestration engine that coordinates various tasks: generating clean, contextual code; identifying and fixing bugs on the fly; automating the creation of test cases; and summarizing code logic in plain language. This process runs in real time and supports adaptive workflows based on project context.

A real-time collaboration layer empowers teams to review, annotate, and iterate on generated assets collectively. This includes in-browser editors, versioning controls, and role-based access. Outputs are routed to familiar tools—like IDEs, CI/CD pipelines, and project management platforms—making it easy to integrate with existing ecosystems.

Crucially, every interaction is logged into a central knowledge and feedback system that continuously improves model performance through fine-tuning, user input, and analytics. The entire stack is scalable, API-driven, and deployable across cloud, on-prem, or hybrid environments—ensuring flexibility, resilience, and enterprise-grade scalability.

****