**Project Design Phase-II**

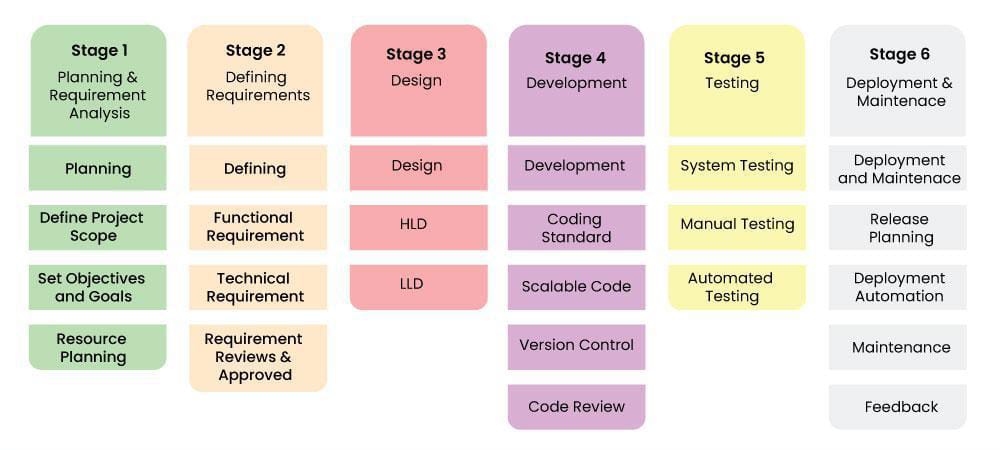
**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 17 June 2025 |
| Team ID | LTVIP2025TMID29152 |
| Project Name | SmartSDLC – AI-Enhanced Software Development Lifecycle |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The technical architecture of a Smart SDLC (Software Development Life Cycle) involves integrating various technologies and methodologies to streamline the development process, enhance collaboration, and improve software quality. It leverages tools and practices from Agile, DevOps, and cloud computing to create a more responsive and efficient development environment.

**Architecture Layout:**

****

**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | How user interacts with the system(e.g.  Web, Mobile App, Chatbot.) | HTML/CSS,JavaScript / React Js. |
|  | Application Logic-1 | Core logic:Authentication,role based access,session handling. | Python,Node.js(FastAPI),JWT. |
|  | Application Logic-2 | Chat-bot interaction and AI query handling. | OpenAI GPT,Langchain, |
|  | Application Logic-3 | PDF classification,sentiment analysis,summarization logic. | Python(transformers,NLTK,scikit-learn). |
|  | Database | User data,feedbback,session logs. | postgreSQL,mongoDB. |
|  | Cloud Database | Scalable managed database service. | Firebase firestore,AWS RDS,google could SQL. |
|  | File Storage | File upload for PDfs,code,test cases. | AWS S3,google cloud storage,firebase storage. |
|  | External API-1 | For 0Auth integration's(google,LinkedIn). | Google 0Auth API,LinkedIn API. |
|  | External API-2 | Feedback and sentiment analysis API. | open-AI API,GitHub API,REST APIs. |
|  | Machine Learning Model | Code summrizer,bug fixer,test case generator. | GPT-based condex,T5,huggingface transformers. |
|  | Infrastructure (Server / Cloud) | Deployment and scaling across platforms. | Docker,kubernetes,,GCP app enginel,github actions. |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | List the open-source frameworks used for frontend,backend,and AI models. | React.js,node.js,python(fastAPI),hugging face transformers,tensorflow,flask |
|  | Security Implementations | All access control and security measures used. | JWT,0Auth2,HTTPS. |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Cloud run,microservices,API gateway. |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | AWS/GPT load balancer. |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Lazy loading,framer motion,cloudfront. |