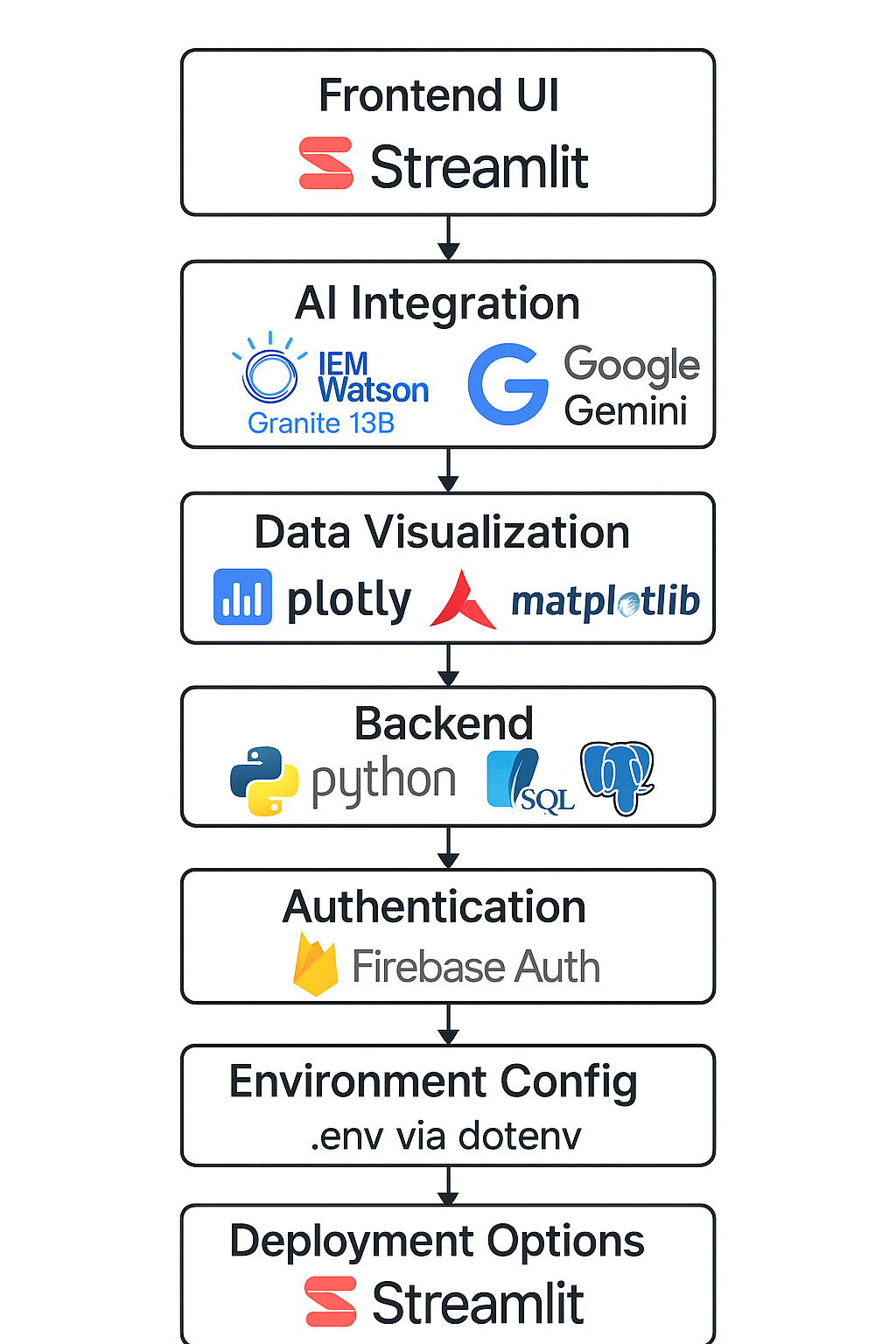
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 27 June 2025 |
| Team ID | LTVIP2025TMID59561 |
| Project Name | HealthAI: Intelligent Healthcare Assistant Using IBM Granite |
| Maximum Marks | 4 Marks |

**Technical Architecture:**



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology / Tools** |
| 1 | User Interface | Web App & Mobile App (chat + dashboard) | Streamlit (Web), Flutter (Mobile), HTML, CSS |
| 2 | Application Logic-1 | Patient Chat logic (NLP processing) | Python (NLTK, transformers), Streamlit |
| 3 | Application Logic-2 | Speech-to-text transcription | IBM Watson STT / Google Speech API |
| 4 | Application Logic-3 | Conversational assistant logic | IBM Watson Assistant |
| 5 | Database | Store patient queries, reports, interaction history | SQLite / Firebase Realtime DB / NoSQL |
| 6 | Cloud Database | Cloud-based scalable health data storage | IBM Cloudant / Firebase Cloud Firestore |
| 7 | File Storage | Store audio inputs, ML models, PDF reports | IBM Cloud Object Storage / Firebase Storage |
| 8 | External API-1 | For disease info enrichment | Wikipedia Medical API / HealthMap API |
| 9 | External API-2 | Patient authentication / government records | Aadhaar Verification API (optional) |
| 10 | Machine Learning Model | Disease prediction using symptoms | Custom-trained ML/DL models (Scikit-learn, TensorFlow) |
| 11 | Infrastructure | Cloud deployment and scaling | IBM Cloud / GCP with Kubernetes / Firebase Hosting |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology Used** |
| 1 | Open-Source Frameworks | UI and backend frameworks | Streamlit, Flask, Scikit-learn, TensorFlow, Flutter |
| 2 | Security Implementations | Data encryption, role-based access, authentication | SHA-256, OAuth2.0, Firebase Authentication, IAM Policies |
| 3 | Scalable Architecture | Microservice-ready, Firebase scaling, container deployment supported | Firebase Functions / Kubernetes-based scaling |
| 4 | Availability | 99.9% uptime via managed cloud infrastructure | IBM Cloud Load Balancer / Firebase Multi-region Hosting |
| 5 | Performance | Use of CDN for static content, async API, model inference <3 sec response | Firebase CDN, Model Optimization, Streamlit Caching |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)