**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 29/JUN/2025 |
| Team ID | LTVIP2025TMID32012 |
| Project Name | Citizen-ai |
| Maximum Marks | 4 MARKS |

### Technical Architecture:CIT-AI is a web-based citizen engagement platform connecting citizens to Indian government services.

### The architecture includes:

**Frontend:** HTML, CSS, JavaScript for the web UI and chatbot interface

**Backend:** Python (Flask) for routing, business logic, and integrating AI services

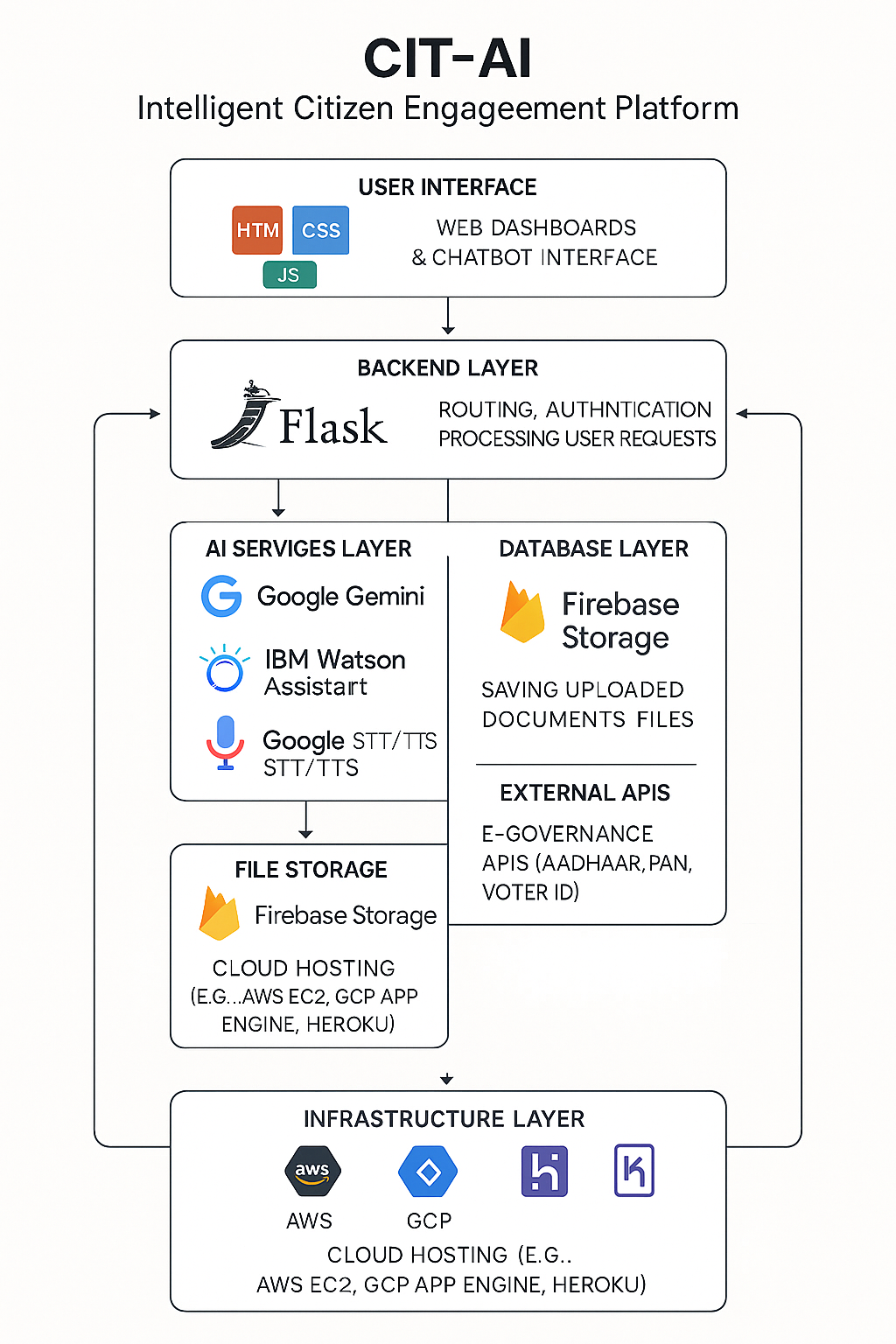
**AI Services:** Google Gemini or IBM Watson for NLP, translation, and speech services

**Database:** Firebase Realtime Database to store user info, chat logs, and form data

**External APIs:** Government e-governance APIs for services like Aadhaar, PAN

**Cloud Deployment:** Hosted on cloud servers for scalability and high availability

**Technical Architecture:**

****

**Table-1: Components & Technologies**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1 | User Interface | Web UI, chatbot interface, dashboards | HTML, CSS, JavaScript |
| 2 | Application Logic-1 | Handles routes, user auth, and request processing | Python (Flask) |
| 3 | Application Logic-2 | Voice processing: speech-to-text and text-to-speech | Google STT/TTS APIs, IBM Watson STT/TTS |
| 4 | Application Logic-3 | Chatbot conversation handling and NLP | Google Gemini, IBM Watson Assistant |
| 5 | Database | Stores user data, chat logs, form details | Firebase Realtime Database |
| 6 | Cloud Database | Managed database for scalability | Firebase (Google Cloud) |
| 7 | File Storage | Stores uploaded documents and generated files | Firebase Storage |
| 8 | External API-1 | Provides translation for multilingual chatbot | Google Translate API |
| 9 | External API-2 | Interfaces with government services | Government e-Governance APIs |
| 10 | Machine Learning Model | Natural language understanding, intent detection | Google Gemini, IBM Watson NLU |
| 11 | Infrastructure | Hosting and deployment on cloud or local servers | AWS EC2, GCP App Engine, Heroku, etc. |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | Frameworks used for backend, frontend, and integration | Flask (Python), Bootstrap, jQuery, Firebase SDK |
| 2 | Security Implementations | Encryption, secure communication, authentication, API key handling | HTTPS, JWT, Firebase Authentication, Google IAM |
| 3 | Scalable Architecture | Scalable microservices-based architecture with cloud resources | Flask microservices, Firebase Cloud |
| 4 | Availability | High availability ensured through cloud load balancing and redundancy | Cloud Load Balancers, Multi-region deployment |
| 5 | Performance | Fast responses, caching, CDN for static assets | Firebase caching, CDN, optimized API calls |