# Project Design Phase-II

## Technology Stack (Architecture & Stack)

Date:03.09.2025

Team ID: [NM2025TMID08924]

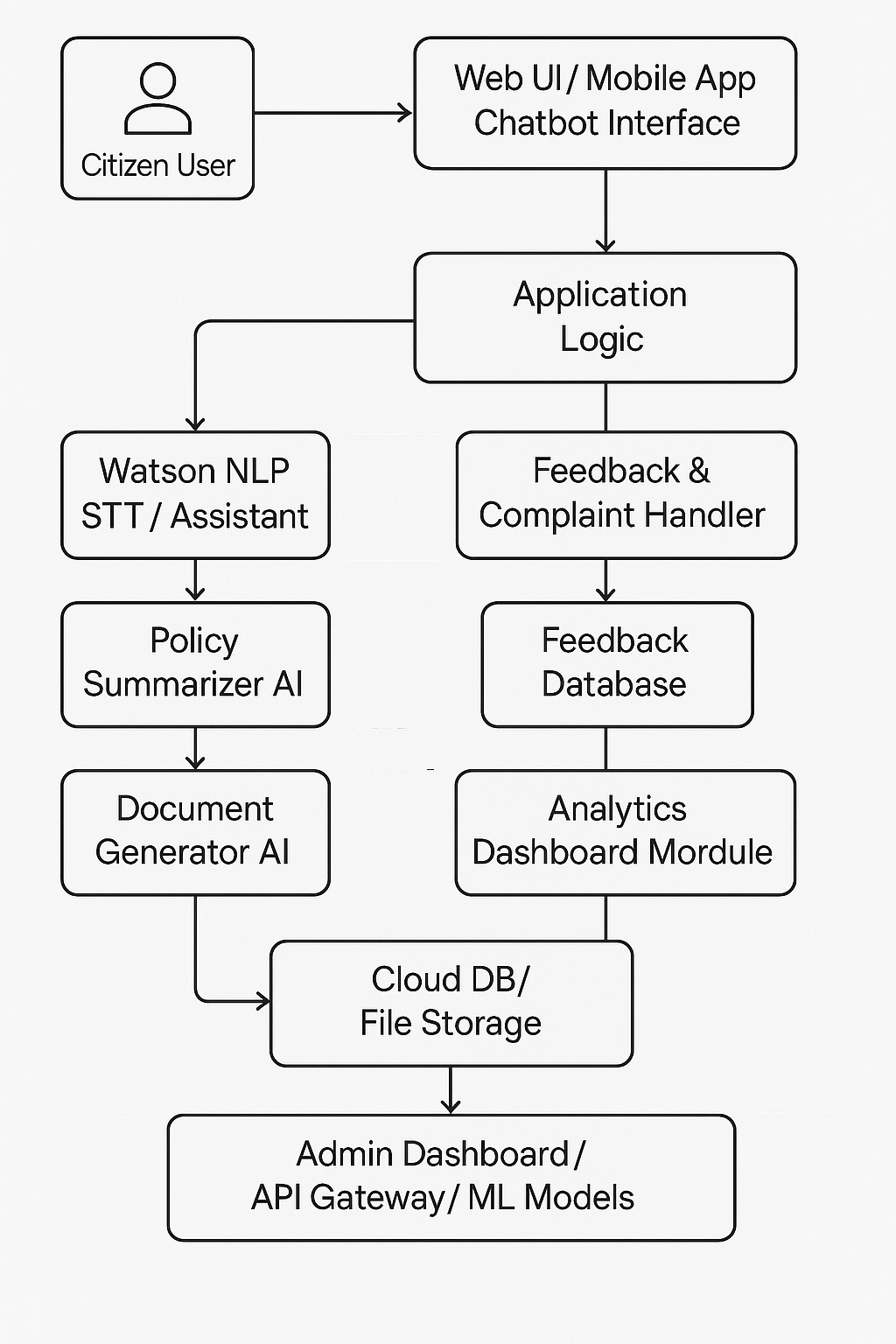
Project Name: Citizen AI – Intelligent Citizen Engagement Platform

Maximum Marks: 4 Marks

## Technical Architecture

The deliverable includes the architectural diagram and information in Table 1 & Table 2.  
  
Example Use Case: Grievance redressal, policy help, feedback tracking, and citizen assistance through AI chatbot.

## Citizen AI Architecture Diagram



## Table-1: Components & Technologies

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology Used |
| 1 | User Interface | User interaction via web/mobile/chatbot | HTML, CSS, React.js / Angular, Flutter, Bot UI |
| 2 | Application Logic-1 | Chatbot / NLP for citizen Q&A | IBM Watson Assistant / Google Dialogflow |
| 3 | Application Logic-2 | Speech-to-text conversion | IBM Watson STT / Google STT |
| 4 | Application Logic-3 | Document generation for services (e.g. ID, NOC) | Python/Node.js with docx/pdf generators |
| 5 | Database | Store user profiles, complaints, policies | MySQL / MongoDB / PostgreSQL |
| 6 | Cloud Database | Cloud-based DB for scaling | IBM Cloudant / Firebase Realtime DB |
| 7 | File Storage | File/document uploads and storage | IBM Cloud Storage / AWS S3 / Local FS |
| 8 | External API-1 | Weather / emergency APIs | IBM Weather API / OpenWeather |
| 9 | External API-2 | ID verification API (for Aadhaar, PAN, etc.) | Aadhaar API, DigiLocker API |
| 10 | Machine Learning Model | Sentiment analysis, intent recognition | TensorFlow / PyTorch-based NLP model |
| 11 | Infrastructure | Cloud-native deployment | IBM Cloud / Kubernetes / Docker / Cloud Foundry |

## Table-2: Application Characteristics

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology Used |
| 1 | Open-Source Frameworks | Use of open-source UI/NLP/backend tools | React.js, Node.js, TensorFlow |
| 2 | Security Implementations | Security for user data and access control | JWT, OAuth 2.0, SHA-256, IAM, OWASP |
| 3 | Scalable Architecture | Based on microservices and cloud scalability | Kubernetes, Load Balancer, Serverless Functions |
| 4 | Availability | Highly available with failover, backups, and monitoring | Distributed DB, Auto-scaling, Uptime Monitor |
|  |  |  |  |
| 5 | Performance | High-performance via caching, CDN, efficient API use | Redis Cache, CDN, Indexed DBs, Async APIs |