

Project Design Phase-II Technology Stack (Architecture & Stack)

| | |
|---------------|--|
| Date | 28 June 3035 |
| Team ID | LTVIP2025TMID60708 |
| Project Name | Sustainable Smart City Assistant Using IBM Granite LLM |
| Maximum Marks | 4 Marks |

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

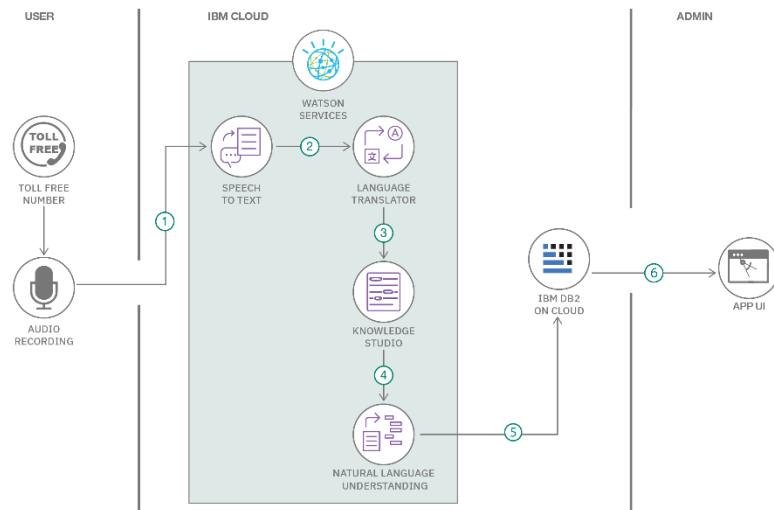


Table-1 : Components & Technologies:

| S.No | Component | Description | Technology |
|------|---------------------------------|---|--|
| 1 | User Interface | How users interact with the application (Web UI, Mobile App, Chatbot, etc.) | HTML, CSS, JavaScript / React.js / Angular.js |
| 2 | Application Logic-1 | Core backend logic for processing user inputs and managing flow | Python / Java |
| 3 | Application Logic-2 | Handles speech-to-text conversion | IBM Watson Speech-to-Text (STT) Service |
| 4 | Application Logic-3 | Manages chatbot interaction and dialogue flow | IBM Watson Assistant |
| 5 | Database | Stores user profiles, queries, and eco tips | MySQL / MongoDB / PostgreSQL |
| 6 | Cloud Database | Scalable cloud-hosted data storage | IBM DB2, IBM Cloudant |
| 7 | File Storage | For storing uploaded documents or generated reports | IBM Block Storage / IBM Object Storage / Local |
| 8 | External API-1 | To fetch real-time weather and environmental updates | IBM Weather API |
| 9 | External API-2 | For identity verification or authentication (if needed) | Aadhaar API (UIDAI), DigiLocker API |
| 10 | Machine Learning Model | To analyze city data and suggest smart sustainability actions | IBM Granite LLM / Object Recognition Model |
| 11 | Infrastructure (Server / Cloud) | Platform where the app is deployed and runs | Local Server, IBM Cloud Foundry, Kubernetes |

Table-2: Application Characteristics:

| S.No | Characteristics | Description | Technology |
|------|--------------------------|--|--|
| 1 | Open-Source Frameworks | List of open-source tools used in development | React.js / AngularJS / Node.js / Flask / Django |
| 2 | Security Implementations | Implementation of access control, encryption, and protection against threats | SHA-256, SSL/TLS, JWT, OAuth 2.0, IAM Roles, OWASP Practices |
| 3 | Scalable Architecture | Modular design supporting future growth (horizontal/vertical scaling) | 3-Tier Architecture, RESTful APIs, Microservices, Docker, Kubernetes |

| S.No | Characteristics | Description | Technology |
|------|-----------------|---|--|
| 4 | Availability | Ensuring app is always accessible through load balancing and redundancy | Load Balancer (e.g., NGINX), Distributed Cloud Servers, Auto-scaling |
| 5 | Performance | Fast response with efficient handling of high traffic using optimization techniques | Caching (Redis), CDN (Cloudflare/Akamai), Asynchronous APIs |