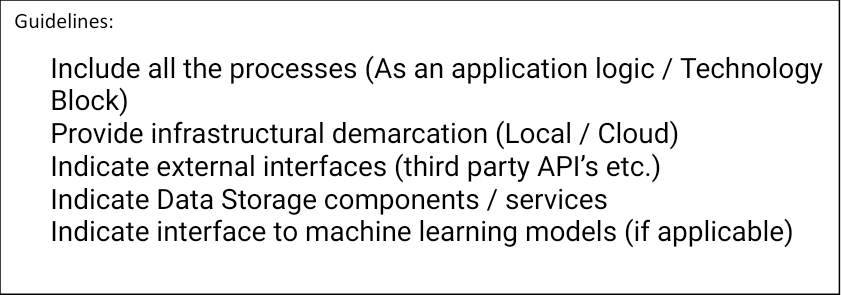
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| Date | 20 May 2023 |
| --- | --- |
| Team ID | NM2023TMID13113 |
| Project Name | Reducing-the-Environmental-Footprint-of-Food-A-Comprehensive-Management-system |

**Technical Architecture:**

**Architecture:**

**The architecture of the "Reducing the Environmental Footprint of Food: A Comprehensive Management System" can follow a typical web-based client-server architecture. The client-side can be implemented as a web application or a mobile application, while the server-side can consist of various components for data processing, storage, and analysis.**

**Client-Side:**

**Web Application: HTML, CSS, JavaScript**

**Frameworks: React, Angular, Vue.js**

**UI/UX Design: Bootstrap, Material-UI**

**or**

**Mobile Application:**

**iOS: Swift, Objective-C**

**Android: Java, Kotlin**

**Cross-platform: React Native, Flutter**

**Server-Side:**

**Programming Languages:**

**Java, Python, or Node.js**

**Web Frameworks:**

**Java: Spring Boot, JavaServer Faces (JSF)**

**Python: Django, Flask**

**Node.js: Express.js**

**Database:**

**Relational: PostgreSQL, MySQL, Oracle**

**NoSQL: MongoDB, Cassandra, CouchDB**

**Data Processing and Analysis:**

**Python Libraries: Pandas, NumPy, Scikit-learn**

**R, Apache Spark**

**Authentication and Authorization:**

**OAuth 2.0**

**JSON Web Tokens (JWT)**

**API Development:**

**RESTful APIs**

**Cloud Services and Hosting:**

**Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP)**

**Additional Tools and Technologies:**

**Version Control: Git, GitHub, Bitbucket**

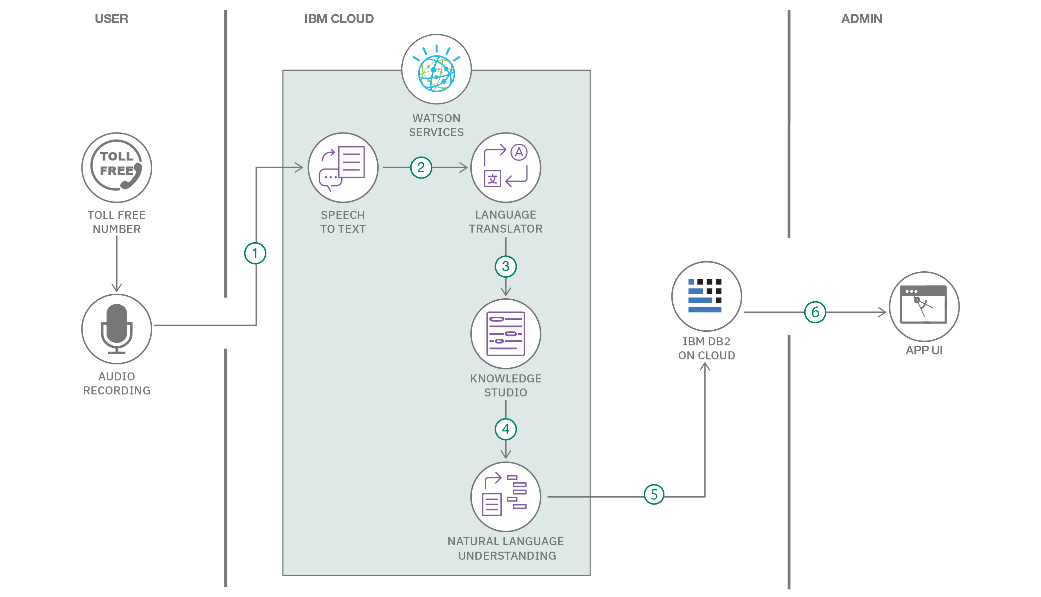
**Deployment: Docker, Kubernetes**

**Testing: Unit testing frameworks (JUnit, PyTest), API testing (Postman, Swagger)**

**Logging and Monitoring: ELK Stack (Elasticsearch, Logstash, Kibana), Prometheus, Grafana**

**It's important to note that the choice of technology stack may vary based on factors such as project requirements, team expertise, scalability needs, and budget considerations**

**Example: Order processing during pandemics for offline mode**



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

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | user interacts with application e.g.  Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript / Angular Js / React Js etc. |
|  | Application Logic-1 | Logic for a process in the application | Java / Python |
|  | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
|  | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
|  | Database | Data Type, Configurations etc. | MySQL, NoSQL, etc. |
|  | Cloud Database | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
|  | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
|  | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
|  | External API-2 | Purpose of External API used in the application | Aadhar API, etc. |
|  | Machine Learning Model | Purpose of Machine Learning Model | Object Recognition Model, etc. |
|  | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud  Local Server Configuration:  Cloud Server Configuration : | Local, Cloud Foundry, Kubernetes, etc. |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | List the open-source frameworks used | Technology of Opensource framework |
|  | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Technology used |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Technology used |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Technology used |