PROJECT DESIGN PHASE-II

TECHNOLOGY STACK (ARCHITECTURE & STACK)

|  |  |
| --- | --- |
| DATE | 16 October 2022 |
| TEAM ID | PNT2022TMID50687 |
| PROJECT NAME | Fertilizer recommendation system for  disease prediction |
| MAXIMUM MARK | 4 Marks |

**Technical Architecture:**

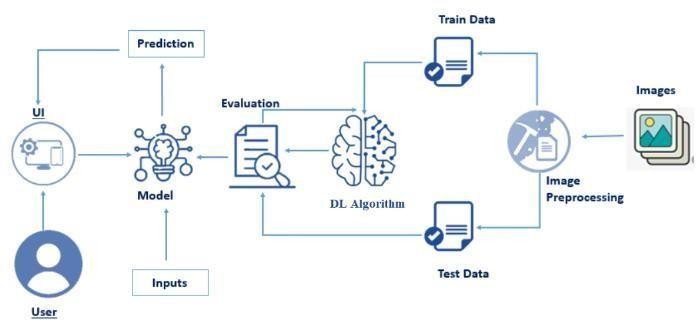


Table-1: Components & Technologies:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | How user interacts with application e.g. Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript / Angular Js / React Js etc. |
| 2. | Application Logic-1 | Logic for a process in the application | Python |
| 3. | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
| 4. | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
| 5. | Database | Data Type, Configurations etc. | MySQL, NoSQL, etc. |
| 6. | Cloud Database | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
| 7. | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
| 8. | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
| 9. | External API-2 | Purpose of External API used in the application | Aadhar API, etc. |
| 10. | Machine Learning Model | Purpose of Machine Learning Model | Object Recognition Model, etc. |
| 11. | 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** |
| 1. | Open-Source Frameworks | List the open-source frameworks used | Anaconda Navigator, Tensor flow, Keras, Flask |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier,  Micro-services) | Response time, Throughput, CPU and  network usages, etc. |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | All kind of users. |
| 5. | Performance | Design consideration for the performance of the  application (number of requests per sec, use of Cache, use of CDN’s) etc. | Predicting disease, image processing,  Visual similarity, rules, machine learning techniques, etc**.** |