**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

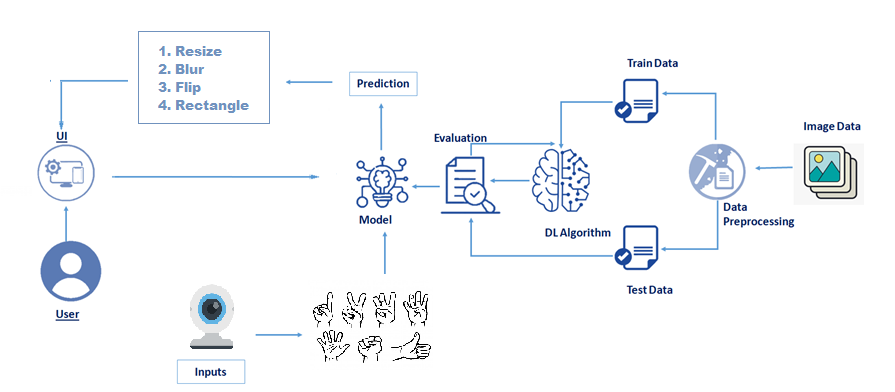
|  |  |
| --- | --- |
| Date | 13 October 2022 |
| Team ID | PNT2022TMID15152 |
| Project Name | A gesture based tool for sterile browsing of radiology images |
| Maximum Marks | 4 Marks |

**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**

**Reference: <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>**



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | How 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 | 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 | Anaconda Navigator, Tensor flow, Keras, Flask |
|  | 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) | Response time, Throughput, CPU and network usages, etc. |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | All kind of users. |
|  | 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**.** |

**References:**

**<https://c4model.com/>**

**<https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>**

**<https://www.ibm.com/cloud/architecture>**

**<https://aws.amazon.com/architecture>**

**<https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>**