**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

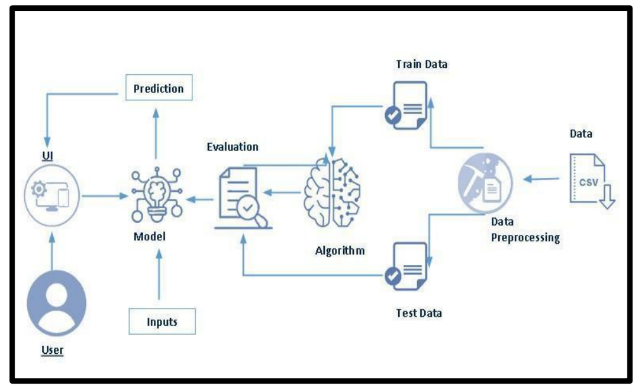
| Date | 7th November 2023 |
| --- | --- |
| Team ID | 592802 |
| Project Name | Disease Prediction Using Machine Learning |
| Maximum Marks | 4 Marks |

**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/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



**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. | Data Preprocessing | Data cleaning and preprocessing logic | Python, pandas |
| 3. | Model Training - KNN | K-Nearest Neighbors Classifier | scikit-learn - KNeighborsClassifier |
| 4. | Model Training - SVM | Support Vector Machine Classifier | scikit-learn - SVC |
| 5. | Model Training - Decision Tree | Decision Tree Classifier | scikit-learn - DecisionTreeClassifier |
| 6. | Model Training - Random Forest | Random Forest Classifier | scikit-learn - RandomForestClassifier |
| 7. | Model Evaluation | Evaluation of machine learning models | scikit-learn - accuracy\_score |
| 8. | Data Preprocessing Function | Function for data preprocessing | Defined in the code |
| 9. | Visualization - Pie Chart | Visualization of symptom distribution | Matplotlib, Seaborn |
| 10. | Visualization - Bar Chart | Visualization of symptom distribution | Matplotlib, Seaborn |
| 11. | Visualization - Swarm Plot | Visualization of symptom correlation | Seaborn |
| 12. | Data Preprocessing Function | Function for data preprocessing | Defined in the code |
| 13. | Model Training Function | Function for training machine learning models | scikit-learn |
| 14. | Model Evaluation Function | Function for evaluating machine learning models | scikit-learn |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1. | Open-Source Frameworks | List the open-source frameworks used | scikit-learn, Matplotlib, Seaborn |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | TBD (Security measures not explicitly mentioned) |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | TBD (Scalability measures not explicitly mentioned) |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | TBD (Availability measures not explicitly mentioned) |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | TBD (Performance measures not explicitly mentioned) |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/) [**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture) [**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)