**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

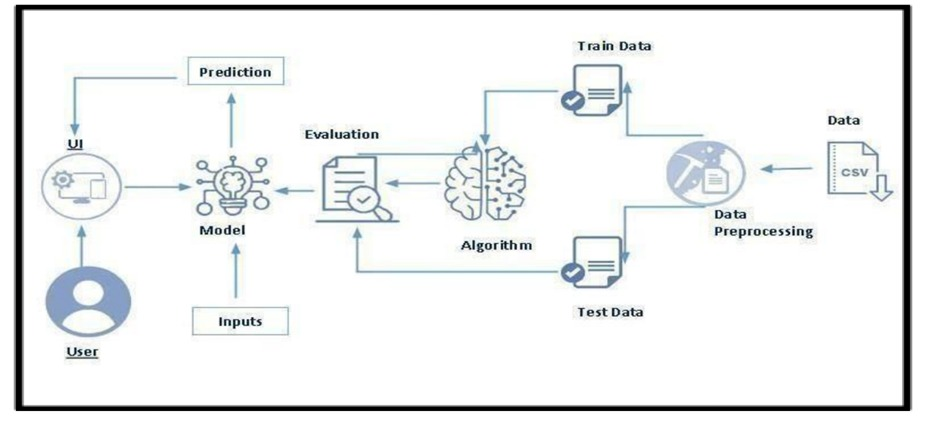
|  |  |
| --- | --- |
| Date | 9 October 2023 |
| Team ID | PNT2023TMID592830 |
| Project Name | Project – Travel Insurance Prediction |
| Maximum Marks | 4 Marks |

**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/)



Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API’s etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | How user interacts with application e.g. Web UI | HTML, CSS, JavaScript, Python |
| 2. | Application Logic-1 | Logic for a process in the application | Java / Python |
| 3. | Database | Data Type, Configurations etc. | MySQL, NoSQL |
| 6. | Cloud Database | Database Service on Cloud | None |
| 7. | File Storage | File storage requirements | Local File System |
| 8. | External API-1 | Purpose of External API used in the application | None |
| 9. | External API-2 | Purpose of External API used in the application | None |
| 10. | Machine Learning Model | Purpose of Machine Learning Model | Recognition Model |
| 11. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud Local Server Configuration:  Cloud Server Configuration: | Local |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | List the open-source frameworks used | Anaconda, Jupyter, HTML, CSS, JavaScript, Python |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | None |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | None |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Single Source Server(Local Host 8000) |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Spyder |