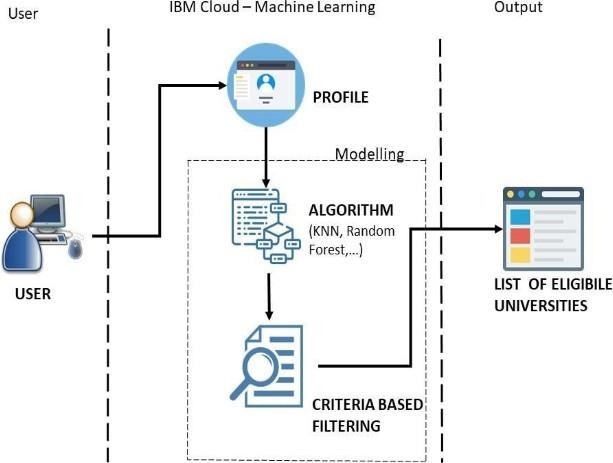
Project Design Phase-II Technology Stack (Architecture & Stack)

|  |  |
| --- | --- |
| Date | 11 Oct 2022 |
| Team ID | PNT2022TMID08090 |
| Project Name | Project - University Admit Eligibility Predictor |
| Maximum Marks | 4 Marks |

**TECHNOLOGY ARCHITECTURE**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.N**  **o** | **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 aSSISTANT |
| 4. | Database | Data Type, Configurations etc. | csv |
| 5. | External API | Purpose of External API used in the application | List of eligible Universities |
| 6. | Machine Learning Model | To predict whether a student is eligible  to get admitted in a university | Prediction Model |
| 7. | Infrastructure (Server /  Cloud) | Application Deployment on Local  System / Cloud  Local Server Configuration: Cloud Server Configuration : | Local, Cloud Foundry, |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N**  **o** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | Python for Backend purpose and flask is imported for front end purpose | Python(Flask) |
| 2. | Security Implementations | The user profile will be secure | Encryptions, IAMControls, OWASP etc |
| 3. | Scalable Architecture | The accurate list of eligible universities  name and its description will be provided | Random Forest ML Algorithm |
| 4. | Availability | Anyone and in anytime they can visit our website | IBM Load Balancer |
| 5. | Performance | The user can have a knowledge of  their eligibility for applying Universities through our website | Random Forest ML Algorithm |