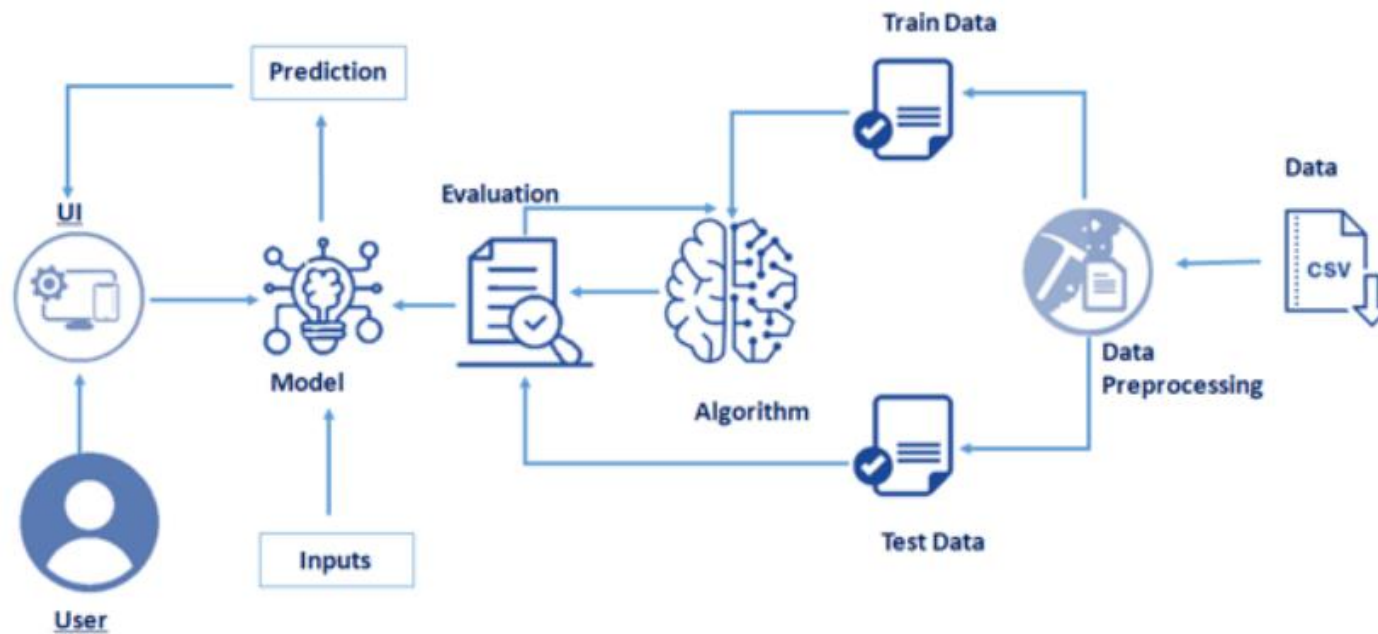


## Project Design Phase-II Technology Stack (Architecture & Stack)

Date	03 October 2022
Team ID	PNT2022TMID32680
Project Name	Project - University Admit Eligibility Predictor
Maximum Marks	4 Marks

### Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



**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, Mobile App, Chatbot etc.	HTML, CSS, JavaScript 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.	CSV
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	List of Universities
9.	Machine Learning Model	Purpose of Machine Learning Model	KNN, Decision tree, Random Forest, Object Recognition Model, etc.
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud etc.

**Table-2: Application Characteristics:**

<b>S.No</b>	<b>Characteristics</b>	<b>Description</b>	<b>Technology</b>
1.	Open-Source Frameworks	List the open-source frameworks used	Python (Flask)
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Random Forest ML architecture
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	IBM Load Balancer
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Random Forest ML architecture