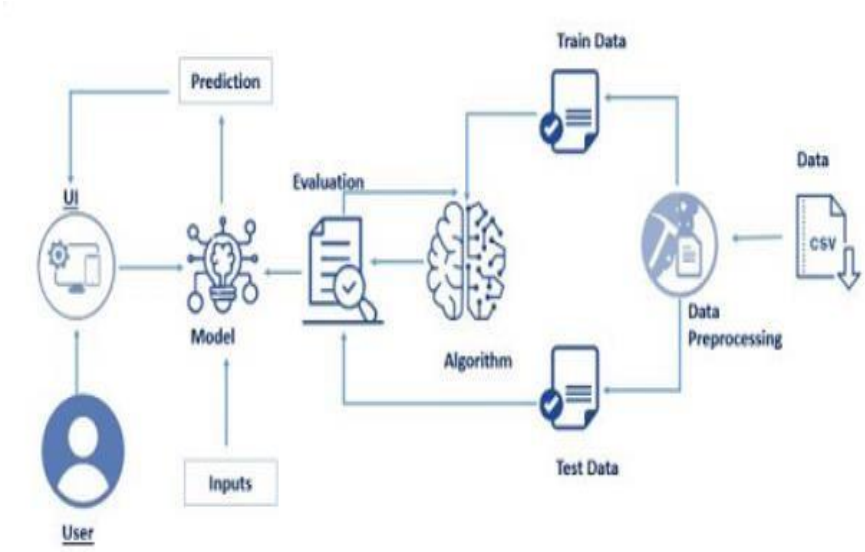
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	03October 2022
Team ID	PNT2022TMID10940
Project Name	Machine Learning based Vehicle Performance Analyzer
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



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	The user has a better experience and may visit the website easily with the aid of web UI.	HTML, CSS
2.	Application Logic-1	Customer can give their vehicle details .	IBM Watson STT service
3.	Application Logic-2	The performance of their car and the vehicle itself may be checked by the customer following aservice.	IBM Watson Assistant
4.	Cloud Database	Cloud database service	IBM DB2, IBM Clouding, etc.
5.	File Storage	Storage needs for files	Cloud Object Storage
6.	Machine Learning Model	To create model for Prediction.	Random Forest Regressor
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	IBM Cloud Services.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used FLASK	The technology of Opensource framework PYTHON
2.	Security Implementations	By improving your car's performance your car will live for long	-
3.	Scalable Architecture	Justify the architecture's capacity to scale (3 – tier, Micro- services) Because application servers may be installed onseveral computers, scalability is improved. The database only needs connections from a smallernumber of application servers, so it doesn't have to establish lengthier connections with every client.	Presentation Layer – FLASK (HTML, CSS) Application Layer – Flask (Python) Data Layer – IBM DB2
4.	Availability	Justify the availability of applications (e.g. use of load balancers, distributed servers, etc.)	-
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	-