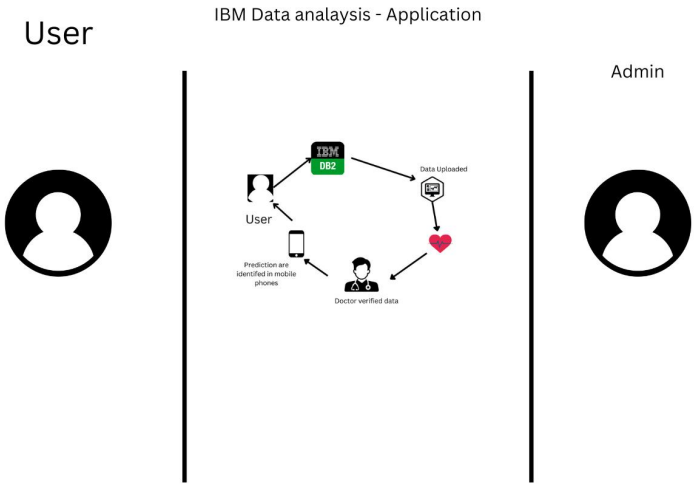
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	17.10.2022
Team Id	PNT2022TMID10855
Project Name	Visualizing and predicting heart diseases
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. nclude 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

Contents:

SNO:	Components	Description	Technology
1	User Interface	By Web UI, the user interacts with the web application and fulfil the user requirements with good user experience	HTML, CSS, JavaScript / React Js etc
2	Application Logic-1	Customer and patient register themselves and once logged in, given with various features.	UsePython
3	Application Logic-2	Once the patient need to check their physical they can contact through the web	IBM Watson Service
4	Application Logic-3	The patient can get the notification from the doctors.	IBM Watson
5	Database SQL Data Type MySQL	SQL Data Type	Mysql
6	Cloud Database	Database service on cloud	IBM DB2
7	File Storage		
8	External API-1	To validate the patient health.	Register Id Apl
9	External API-2		
10	Machine Learning Model		
11	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

SNO	Characteristics	Description	Technology
1	Open-Source Frameworks	List the open-source frameworks used REACT JS EXPRESS JS	Technology of Opensource framework JAVASCRIPT and PYTHON
2	Security Implementations	List all the security / access controls implemented, use of firewalls etc. SHA-256 to protect user details	SHA-256
3	Scalable Architecture	Justify the scalability of architecture (3 – tier) This improves scalability, because application servers can be deployed on many machines. The database does not make longer connections with every client – it only requires connections from a smaller number of application servers	Presentation Layer – React JS (HTML, CSS , JS) Application Layer – Flask (Python) Data Layer – IBM DB2
4	Availability	Justify the availability of application (use of load balancers, lets you evenly distribute network traffic to prevent failure caused by overloading a particular resource. This strategy improves the performance and availability of applications, websites, databases, and other computing resources)	
5	Performance	Design the application carefully to be component based and encapsulated. This can help in creating a scalable application providing flexibility in deployment and making it possible to partition the application and substitute other component implementations during deployment.	