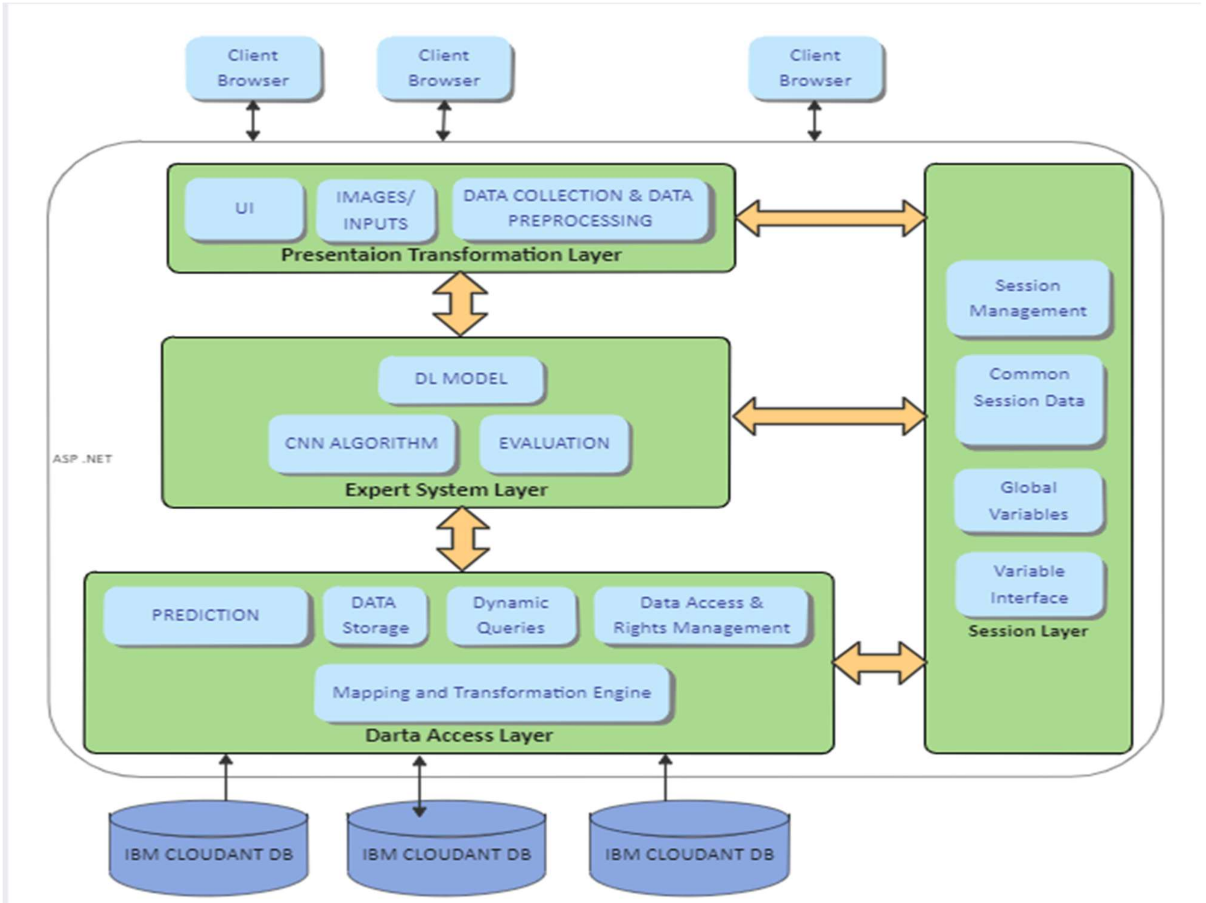
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	21 October 2022
Team ID	PNT2022TMID00519
Project Name	Project - - Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy
Maximum Marks	4 Marks

Technical Architecture:

The Technical Architecture has the following blocks:

- Data Collection.
    - Create a Train and Test path.
- Data Pre-processing.
    - Import the required library
    - Configure ImageDataGenerator class
    - Apply ImageDataGenerator functionality to Trainset and Testset
- Model Building
    - Pre-trained CNN model as a Feature Extractor
    - Adding Dense Layer
    - Configure the Learning Process
    - Train the model
    - Save the Model
    - Test the model
- Cloudant DB
    - Register & Login to IBM Cloud
    - Create Service Instance
    - Creating Service Credentials
    - Launch Cloudant DB
    - Create Database
- Application Building
    - Create an HTML file
    - Build Python Code



TECHNICAL ARCHITECTURE FOR
Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interacts with application through Web UI.	HTML, CSS, JavaScript, Flask, Python
2.	Application Logic-Creating an account	The user interacts with the Web UI to create an account. The account details are stored in a secured manner in IBM CLOUDANT DB	Flask, IBM CLOUDANT DB
3.	Application Logic- Logging in	The user logs into the application through the Web UI. The user details are verified by cross-checking with data available in IBM CLOUDANT DB	Flask, IBM CLOUDANT DB
4.	Application Logic-Getting input from the user	The user interacts with the Web UI to get input from the user.	Flask, IBM CLOUDANT DB
5.	Application Logic-Data pre-processing	The input data collected from the user is pre-processed..	Flask, IBM CLOUDANT DB
6.	Application Logic-Run the model on the data	The input data collected from the user is sent to the model and prediction is made.	Flask, IBM CLOUDANT DB
7.	Application Logic-Storage of Data	The result is stored in the IBM CLOUDANT DB .	Flask, IBM CLOUDANT DB
8.	Application Logic-Display the result	The result is retrieved from the IBM CLOUDANT DB and displayed to the user through the Web UI.	Flask, IBM CLOUDANT DB
9.	Database	The data types will be user dependent as the application is made to be customizable and is also a cloud based database.	IBM CLOUDANT DB
10.	Cloud Database	The database mentioned above is a cloud based database.	IBM CLOUDANT DB
11.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
12.	External API	It will be used to alert users of various notifications etc as defined by the user.	SendGrid Service
13.	Machine Learning Model	The pre-trained model Xception is used which is one of the convolution neural net (CNN) architectures which is considered as a very good model for Image classification.	Xception(CNN)
14.	Deployment	: Application Deployment on Local System / Cloud	Local, Cloud Foundry, Kubernetes

		Local Server Configuration: The application will run on the local server/client side to allow user to interact with Web UI. Cloud Server Configuration: The application will be hosted on the cloud for the user to user. This is done through containerization of the application stored in the container registry.	

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Anaconda is an open source framework used in the project.	Anaconda
2.	Security Implementations	Some encryption methodology can be used to protect the application from security attacks.	SHA-256, Encryptions.
3.	Scalable Architecture	The architecture can be scaled to include the micro services and a detailed description of the implementation of the application logic.	Micro services
4.	Performance	We can use Automatic Verification Datasets ,Manual Verification Datasets, Manual k-Fold Cross Validation to evaluate the performance metrics	Nil