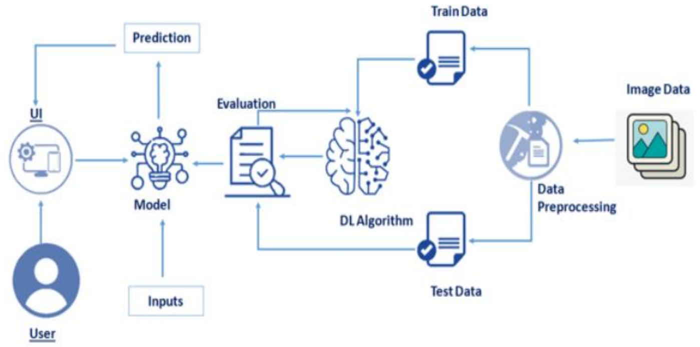
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	15 November 2022
Team ID	PNT2022TMID34634
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation
Maximum Marks	4 Marks

Technical Architecture:

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



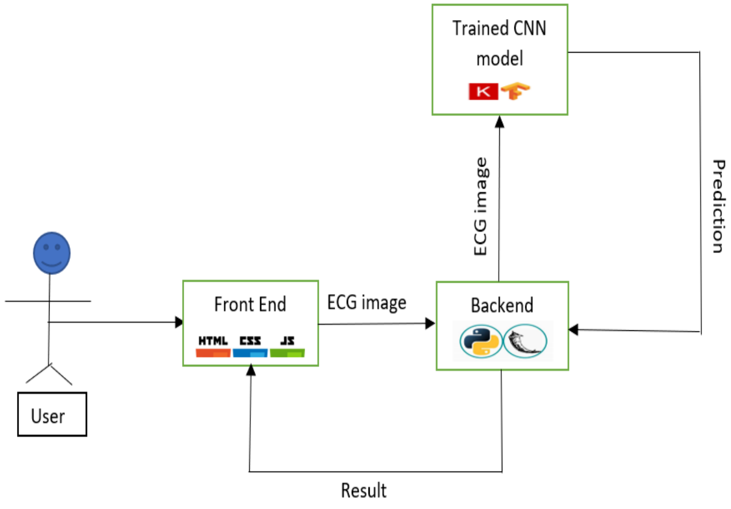


Table-1 : Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with the user interface	Anaconda, Jupyter notebook, Spyder, Python
2.	Model analysis	Once model analysed the uploaded image, the prediction is displayed	Python, OpenCV
3.	Data collection	Create the dataset	Kaggle.com, data.gov
4.	Data Preprocessing-1	Import the ImageDataGenerator library	Python, Keras ,Numpy
5.	Data Preprocessing-2	Configure ImageDataGenerator class	Python, Keras ,Numpy
6.	Data Preprocessing-3	Apply ImageDataGenerator functionality to train set	Python, Keras ,Numpy
7.	Model Building-1	Import the model building libraries and initializing	Python, Keras ,Numpy, Tensorflow
8.	Model Building-2	Adding layers and configure	Python, Keras ,Numpy, Tensorflow
9.	Model Building-3	Training and testing the model, Optimize and save the model	Python, Keras ,Numpy, Tensorflow
10.	Application Building	Purpose of create an HTML file and building python code	HTML, Python, CSS, JS
11.	Train the model on IBM	CNN development and integrate it with the flask application	IBM Watson
12.	Deployment	Deploy the application and make it available for the public to use	IBM Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Flask
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, HTTPS etc..
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	IBM cloud
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	IBM cloud
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc.	IBM cloud, Flask, Numpy