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

Date	18 October 2022
Team ID	PNT2022TMID12347
Project Name	Project - 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
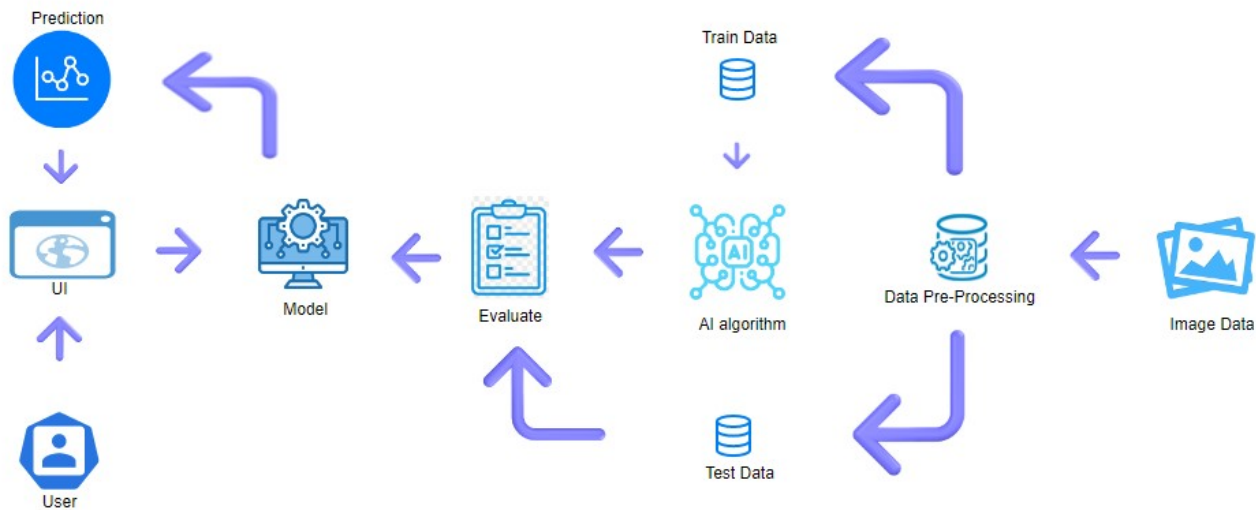
Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with User interface to upload image	Anaconda, jupyter, spyder, python. ext
2.	Model analyses	Once model analyses the uploaded image, the prediction is showcased on the UI	Kaggle.com, data. gov, UCI
3.	Data collection	Create the dataset	Python, keras, numpy
4.	Data Preprocessing-1	Import the ImageDataGenerator library	Python, keras, numpy
5.	Data Preprocessing-2	Configure ImageDataGenerator class	Python, numpy, keras
6.	Data Preprocessing-3	Apply ImageDataGenerator functionality to Trainset and Testset	Python, numpy, keras
7.	Model Building-1	Import the model building libraries and Initializing The model	Python, numpy, keras
8.	Model Building-2	Adding layers and configure	Python, numpy, keras
9.	Model Building-3	Training and testing the model, Optimize and save the model	Python, numpy, keras
10.	Application Building	Purpose of create an HTML file and Building Python code	HTML, python
11.	Train the model on IBM	CNN Development and integrate it with the flask Application	IBM Watsont

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open source software is that by which the source code or the base code is usually available for modification or enhancement.	Flask(python)
2.	Security Implementations	By placing a filtration barrier between the targeted server and the attacker, the WAF is able to protect against attacks like cross site forgery, cross site scripting and SQL injection.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Does not affect the performance even though used by many users.	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used