Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

TEAM ID : PNT2022TMID36404

TEAM LEADER : GAYATHRI VS (110819205005)

TEAM MEMBERS :BHARATHI S (110819205002)

ISWARIYA S(110819205007)

SANGEETHA S(110819205018)

RAJALAKSHMI D (110829205501)

DEPARTMENT : INFORMATION TECHNOLOGY

COLLEGE NAME : JAYA ENGINEERING COLLEGE

INDEX

1. INTRODUCTION

1.1 Project Overview :

The main objective of this project is to create a model using AI (Artificial Intelligence) which is used to predict the disease called Arrhythmia which has classification in it. The term Arrhythmia refers to a medical or biological name which means irregular heartbeat. It is first diagnosed by the ECG (Electro Cardio Gram). The Electro Cardio Gram is a recording of the heart's electrical activity. The classification is done by predicting from the images of ECG taken in the hospitals. The model is trained and tested by the dataset which contains ECG images. The classification depends on the following six types of Arrhythmia. They are "Left Bundle Branch Block", "Normal', "Premature Atrial Contraction", "Right Bundle Branch Block", "Premature Ventricular Contraction", "ventricular Fibrillation".

1.2 Purpose :

This model can be deployed for the official use in hospitals for immediate diagnosis of Arrhythmia. In Arrhythmia case most of the Arrhythmia are minor were as there are some major complications which may lead to cause of patient's life. Considering this, the AI model created to predict the Arrhythmia will be useful for the Doctors to diagnose and give treatment in such a way that it may save patient's life in earlier stage.

2. LITERATURE SURVEY :

2.1 Existing problem :

In existing, ECG are checked by the cardiac doctors and result is told orally. The ECG may vary from one patient to another. In common, if a patient wants to know the result about their ECG, they have to wait until they visit their doctors. Then the doctor diagnoses by analysing

the ECG. But this process takes a long time for a doctor. The proposed solution may save the doctor's time and help not only doctors but also the patients to know their heart's condition.

2.2 References :

SI.NO	Title	Author	Year of publishing	Reference link
1	Automated characterization of arrhythmias using non-linear features from tachycardia ECG beats	U Rajendra, Acharya, Hamido Fujita, Muhammad Adam, Oh Shu lih, Tan Jen Hong, Vidya K Sudarshan	2017	https://ieeexplore.ieee.org/abstract/document /7844294
2	Interpretation and Classification of Arrhythmia Using Deep Convolutional Network	Prateek Singh, Ambalika Sharma.	2022	ttps://www.researchgate.net/publication/363291215_Interpretation_and_Classification_of_Arrhythmia_using_Deep_Convolutional_Net work
3	A Review of Automated Diagnosis of ECG Arrhythmia Using Deep Learning Methods	Praveen kumar tyagi, Neha Rathore, Deepak Parashra, Dheeraj Agrawal.	2022	https://www.researchgate.net/publication/361597512_A_Review_of_Automated_Diagnosis_of_ECG_Arrhythmia_Using_Deep_Learning_Methods
4	Building normal ECG models	Keigi Gyohten, Shota hori, Hidehiro	2022	https://www.researc hgate.net/publication /348282231_Building_normal_ECG_model

	to detect any arrhythmias using deep learning	Ohki, Toshiya Takami.		s_to_detect_any_arr hythmias_using_deep _learning
5	ECG Classification for Heart Arrhythmia Using Deep Machine Learning	Shalin savalia, Vahid Emamian	2021	https://www.researchg ate.net/publication/35 6327202_ECG_Classific ation_for_Heart_Arrhyt hmia_Using_Deep_Mac hine_Learning

2.3 Problem Statement Definition :

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Doctor	Diagnose the Arrhythmia	ECG with variations have same problem	Arrhythmia is classified only by small variation	A fear to the prediction of wrong one
PS-2	Arrhythmia patient	Classify the Arrhythmia	Cannot diagnose clearly	With ECG classification is not accurate	Tensed

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas :

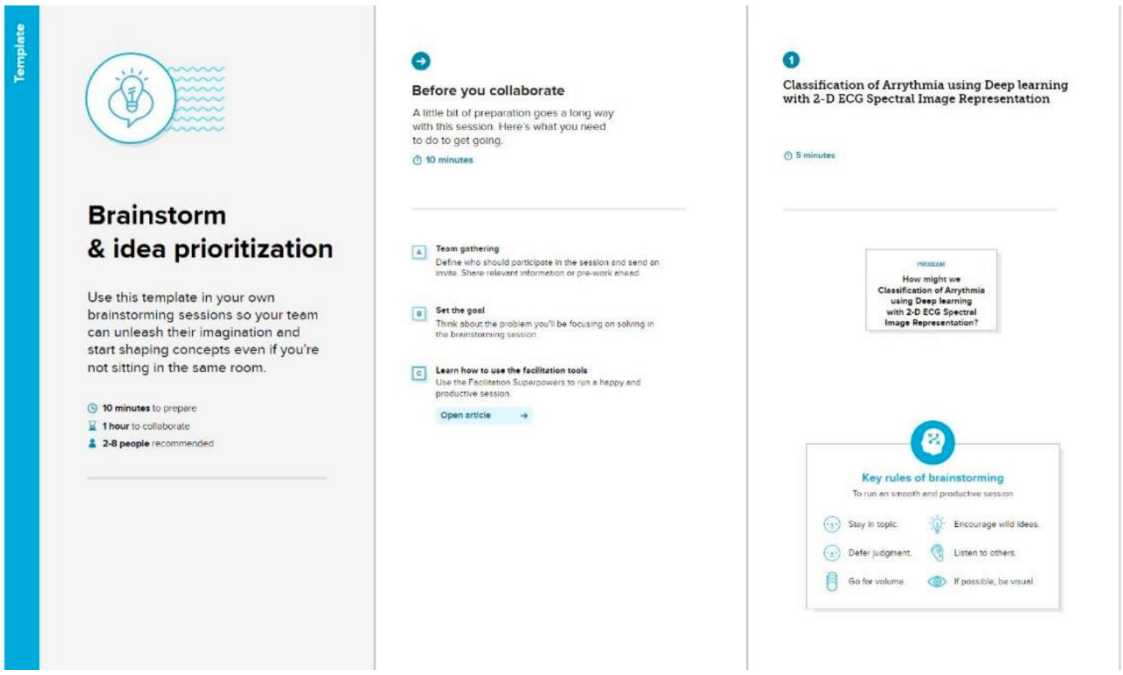
3.2 Ideation & Brainstorming :

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

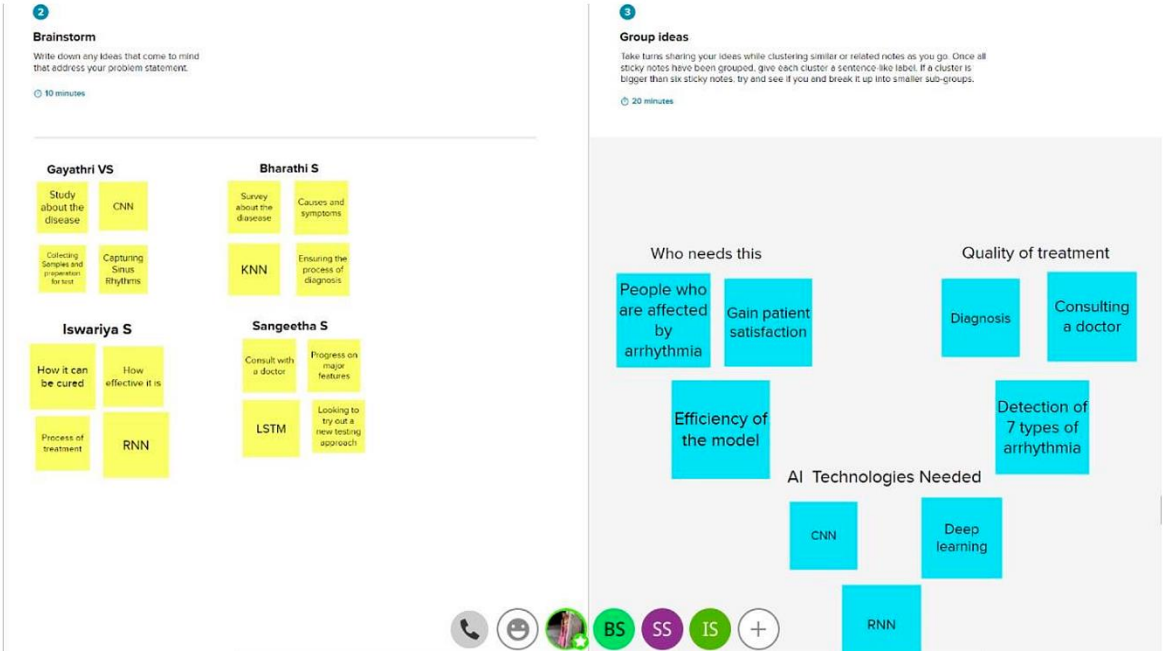
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference:https://app.mural.co/t/jayaengineeringcollege6968/m/jayaengineeringcollege6968/1664786196461/8a84c3c44c1cc06e803b821c5f012b046db1219c?sender=ud2131a8f091fbb39f9d28083

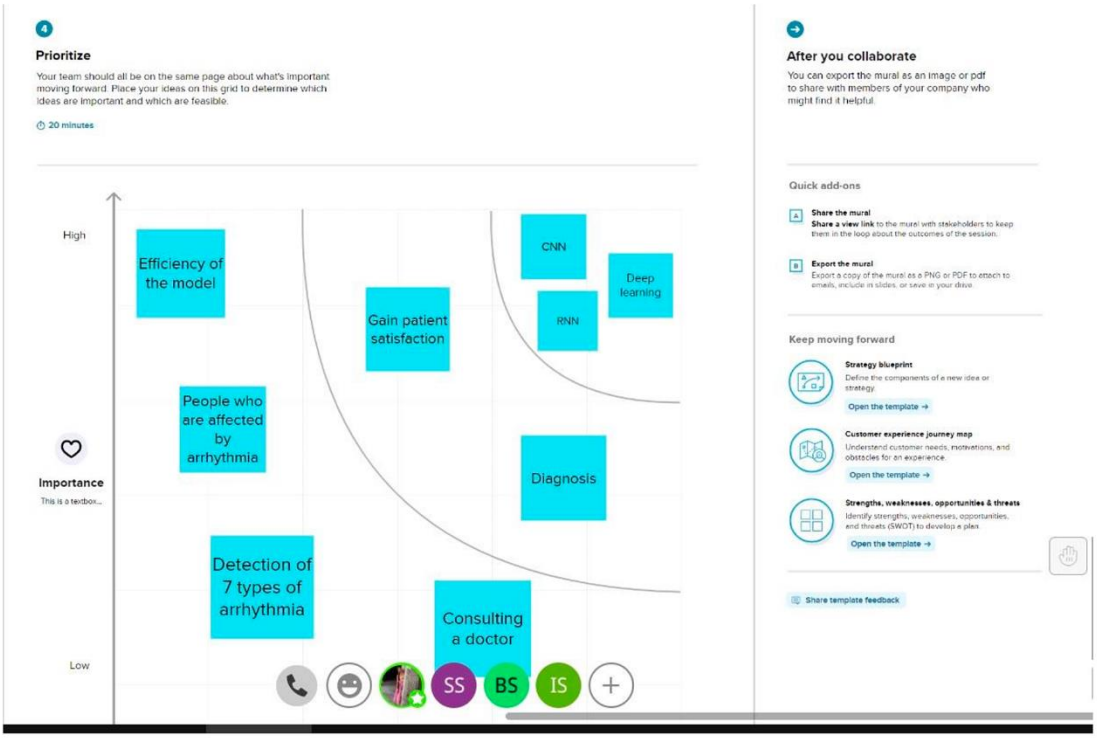
Step-1: Team Gathering, Collaboration and Select the Problem



Step-2: Brainstorm, Idea Listing and Grouping :



Step-3: Idea Prioritization :



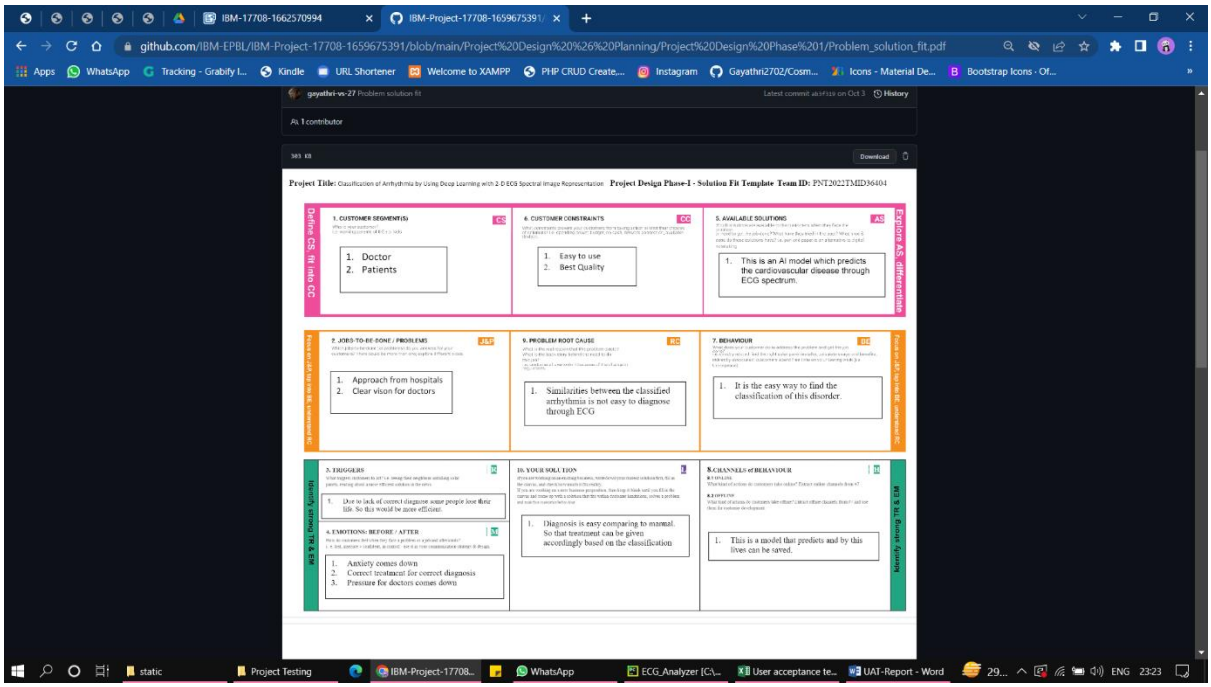
3.3 Proposed Solution :

Project team shall fill the following information in proposed solution template.

SI.NO	PARAMETERS	DESCRIPTION
1	Problem Statement (Problem to be solved)	We know that Arrhythmia is a heart related disorder and can either be curable or uncurable. These are classified into 7 types. Classifying this disorder for the ECG will the best challenge because only few dissimilar variations are there between every type of this disorder.
2	Idea / Solution description	Our Idea is to create a model using deep learning that can read the ECG and can classify

		the Arrhythmia even by those similarities
3	Novelty / Uniqueness	The classification is done by using deep learning with 2-D ECG Spectral Image representation. This helps to diagnose the difference between each and every Arrhythmia which can be differentiated by ECG.
4	Social Impact / Customer Satisfaction	This may help doctors diagnose easily and the patients don't need to panic and worry about their health.
5	Business Model (Revenue Model)	AI is one of today's most heralded technologies and this model are for health care this application is most likely to be used by money.
6	Scalability of the Solution	scalability of the model depends on the training and accuracy of the model which helps the model to train itself. More training of the model leads to a good prediction of the model.

3.4 Problem Solution fit :



4. REQUIREMENT ANALYSIS :

4.1 Functional requirement :

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Registration Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through Form Login through Gmail
FR-4	User Login Confirmation	Redirect to Home page

FR-5	Upload ECG Graph Pictures	Capture the ECG and upload Directly upload from Files
FR-6	Prediction by Model	Result is based on the classification of the Arrhythmia disease read from the prediction

4.2 Non-Functional requirements :

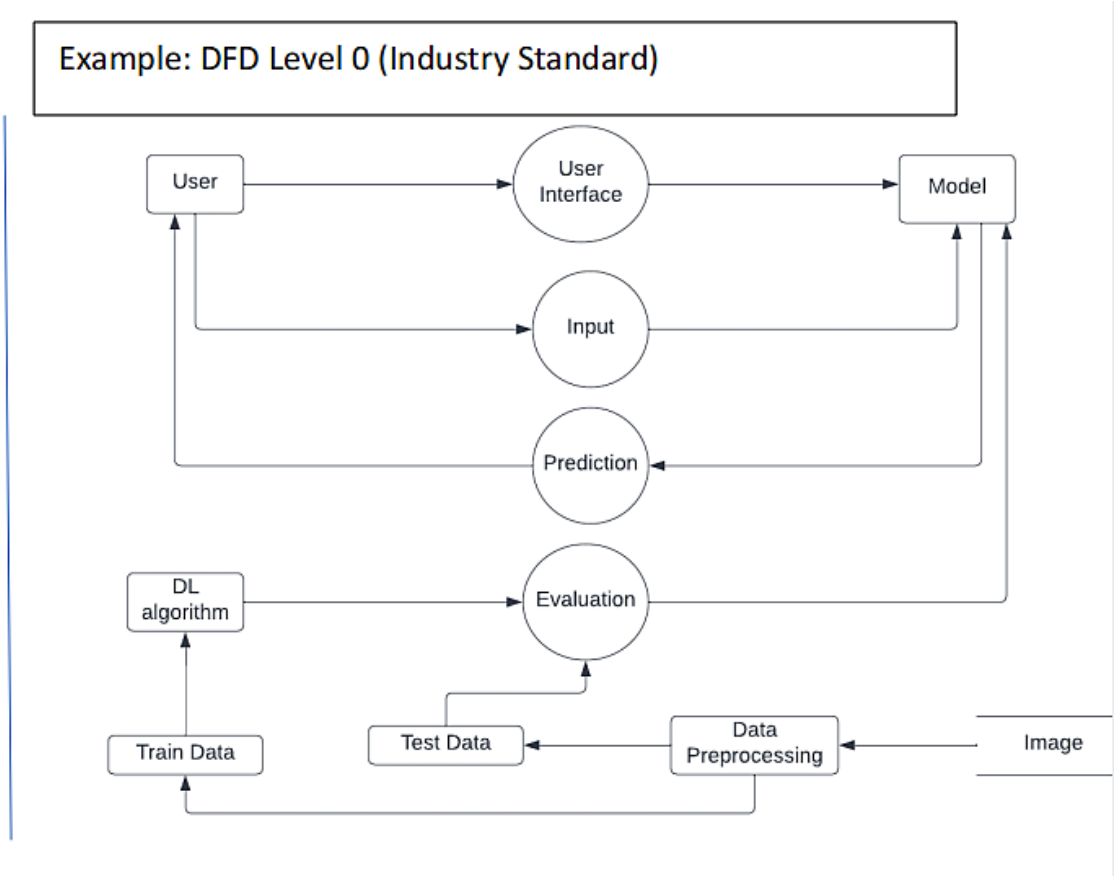
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The UI is designed in such a way that the user can easily use it and easy to understand
NFR-2	**Security**	Since the model is hosted online, security is little less but the security is high when the web app is made as a mobile application.
NFR-3	**Reliability**	The model is reliable in such a way that it can identify the classification between the arrhythmias
NFR-4	**Performance**	Since it is used online, the performance is based on the internet speed connected to the device
NFR-5	**Availability**	Since it is hosted online, it is available on all devices like PC, Laptop, Mobile phones etc.

NFR-6	**Scalability**	The scalability of the model is excepted likely to be high as the performance of the model is good.

5. PROJECT DESIGN :

5.1 Data Flow Diagrams :

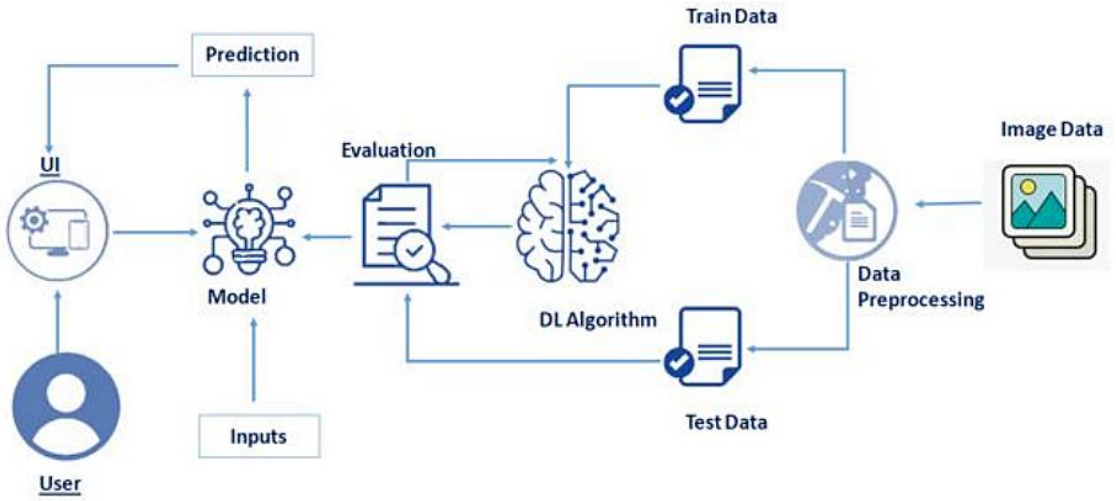
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture :

According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the number one cause of death today. Over 17.7 million people died from CVDs in the year 2017 all over the world which is about 31% of all deaths, and over 75% of these deaths occur in low and middle-income countries. Arrhythmia is a representative type of CVD that refers to any irregular change from the normal heart rhythms. There are several types of arrhythmia including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia. Although a single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. In this project, we build an effective electrocardiogram (ECG) arrhythmia classification method using a convolutional neural network (CNN), in which we classify ECG into seven categories, one being normal and the other six being different types of arrhythmia using deep two-dimensional CNN with grayscale ECG images. We are creating a web application where the user selects the image which is to be classified. The image is fed into the model that is trained and the cited class will be displayed on the webpage.

Architecture:



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release

Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email & password		High	Sprint-1
	Uploading Pictures	USN-4	As a user, I can upload the ECG pictures or capture the picture I can see the result		High	Sprint-2

Customer (Web user)	Registration	USN-5	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-6	As a user, I can register for the application nthrough	Gmail	Medium	Sprint-1
	Login	USN-7	As a user, I can log into the application by entering email & password		High	Sprint-1
	Uploading Pictures	USN-8	As a user, I can upload the ECG pictures or capture the picture I can see the result		High	Sprint-2

Administrator	Login	USN-9	As an admin, I can login with some given credentials for limited access of customer details	I can access the admin account	Medium	Sprint-1

6. PROJECT PLANNING & SCHEDULING :

6.1 Sprint Planning & Estimation :

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Story Points	Team Members
Sprint-1	Download The Dataset	USN-1	We can download the Dataset contains Six classes	**4**	low	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint 1	Import The ImageDataGenerator Library	USN-2	We can import ImageDataGenerator	**4**	low	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-1	Configure ImageDataGenerator class	USN-3	We can configure the ImageDataGenerator class	**6**	Medium	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D

Sprint-1	Apply the ImageDataGenerator functionality to Train Set and Dataset	USN-4	We can apply ImageDataGenerator to train dataset	**6**	Medium	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-2	Import Libraries	USN-5	We can import required Libraries	**1**	low	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-2	Initialize the Model	USN-6	Initializing the Image recognition model	**2**	Medium	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-2	Adding CNN layer	USN-7	We can add Convolutional Neural Network(CNN) used for image/object recognition and classification	**4**	High	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-2	Adding Dense Layer	USN-8	We can add Dense Layer in which each neuron receives input from all the neurons of previous layer	**4**	High	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-2	Configure The Learning Process	USN-9	We can configure The Learning process which is a method, mathematical logic or algorithm that improves the network's performance and/or training time.	**4**	High	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D

Sprint-2	Train the Model	USN-10	We can train our model with our image dataset. fit generator functions used to train a deep learning neural network	**4**	High	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-2	save the Model	USN-11	We can save The model with .h5 extension	**2**	Medium	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-2	Train the Model	USN-12	We can Test the model through Loaded necessary libraries, the saved model	**2**	Medium	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-3	test the Model	USN-13	We use HTML to create the front-end part of the web page.	**8**	High	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-3	Build Python code	USN-14	We build the flask file ‘app.py’ which is a web framework written in python for server-side scripting.	**8**	High	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-3	Run the App	USN-15	We can run the App	**4**	Medium	Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D
Sprint-4	Register IBM Cloud	USN-16	We can register IBM Cloud	**8**	Medium	Gayathri VS Bharathi S Iswariya S Sangeetha S

Sprint-4	Train the model on IBM	USN-17	We can Train Out model on IBM	**12**	High	Rajalakshmi D Gayathri VS Bharathi S Iswariya S Sangeetha S Rajalakshmi D

Project Tracker, Velocity & Burn down Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	8 Days	24 Oct 2022	30 Oct 2022	20	31 Oct 2022
Sprint-2	23	7 Days	30 Oct 2022	04 Nov 2022	20	05 Nov 2022
Sprint-3	20	7 Days	06 Nov 2022	11 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	1420Nov 2022	20	18 Nov 2022

Velocity:

To calculate the team's **average velocity (AV)** per iteration unit

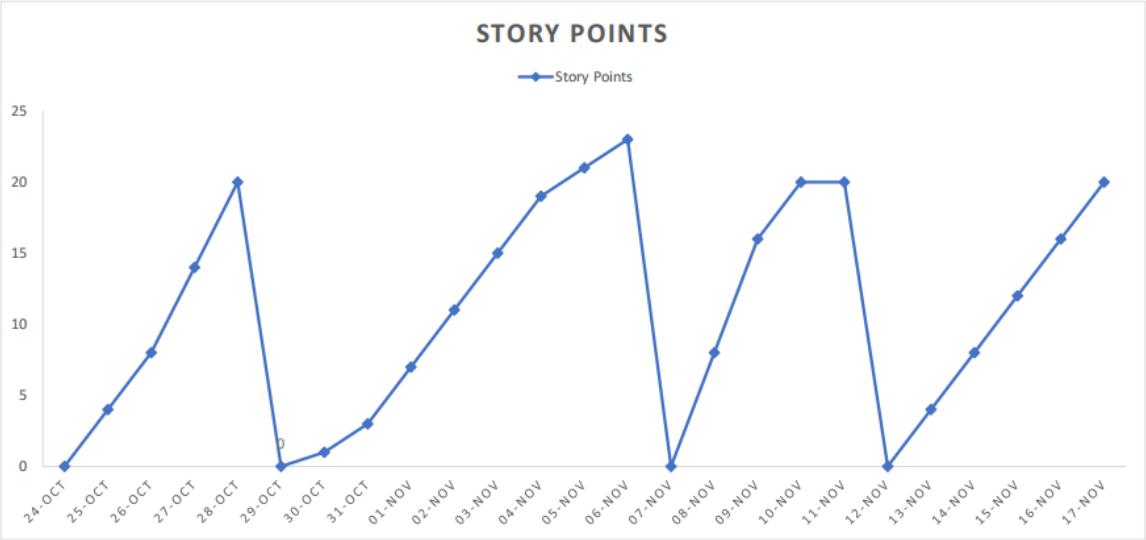
Av = $\frac{Velocity}{Sprint\ Duration}$

Average Velocity - Story points per day
Sprint duration - Number of days (Duration) for Sprints

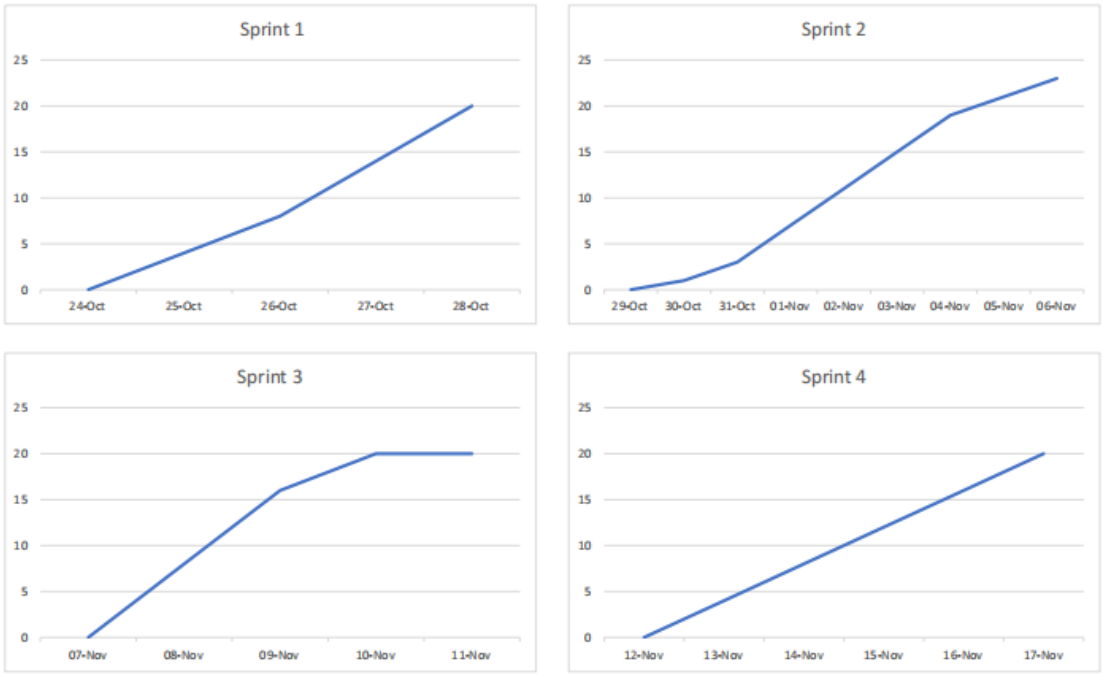
Velocity

Av = 20 =2.8 7

Average Velocity is **2.8** points per Sprint



Burn-down Chart:



6.2 Sprint Delivery Schedule :

Product Backlog, Sprint Schedule, and Estimation :

S.NO	MILESTONE	ACTIVITIES	DATE
1.	Preparation Phase	Pre-requisites	24 Aug 2022
		Prior knowledge	25 Aug 2022
		Project Structure	23 Aug 2022
		Project Flow	23 Aug 2022
		Project Objectives	22 Aug 2022
		Registrations	26 Aug 2022
		Environment Set-up	27 Aug 2022
2.	Ideation Phase	Literature Survey	29 Aug 2022 - 03 Sept 2022
		Empathy Map	05 Sept 2022 - 7 Sept 2022
		Problem Statement	08 Sept 2022 - 10 Sept2022
		Ideation	12 Sept 2022 - 16 Sept 2022
3.	Project Design Phase -I	Proposed Solution	19 Sept 2022 - 23 Sept 2022
		Problem Solution Fit	24 Sept 2022 - 26 Sept 2022
		Solution Architecture	27 Sept 2022 - 30 Sept 2022
4.	Project Design Phase -II	Customer Journey	03 Oct 2022 - 08 Oct 2022
		Requirement Analysis	09 Oct 2022 - 11 Oct 2022
		Data Flow Diagrams	11 Oct 2022 - 14 Oct 2022
		Technology Architecture	15 Oct 2022 - 16 Oct 2022
5.	Project Planning Phase	Milestones & Tasks	17 Oct 2022 - 18 Oct 2022
		Sprint Schedules	19 Oct 2022 - 22 Oct 2022

6.	Project Development Phase	Sprint-1	24 Oct 2022 - 30 Oct 2022
		Sprint-2	30 Oct 2022 - 05 Nov 2022
		Sprint-3	06 Nov 2022- 12 Nov 2022
		Sprint-4	13 Nov 2022 - 18 Nov 2022

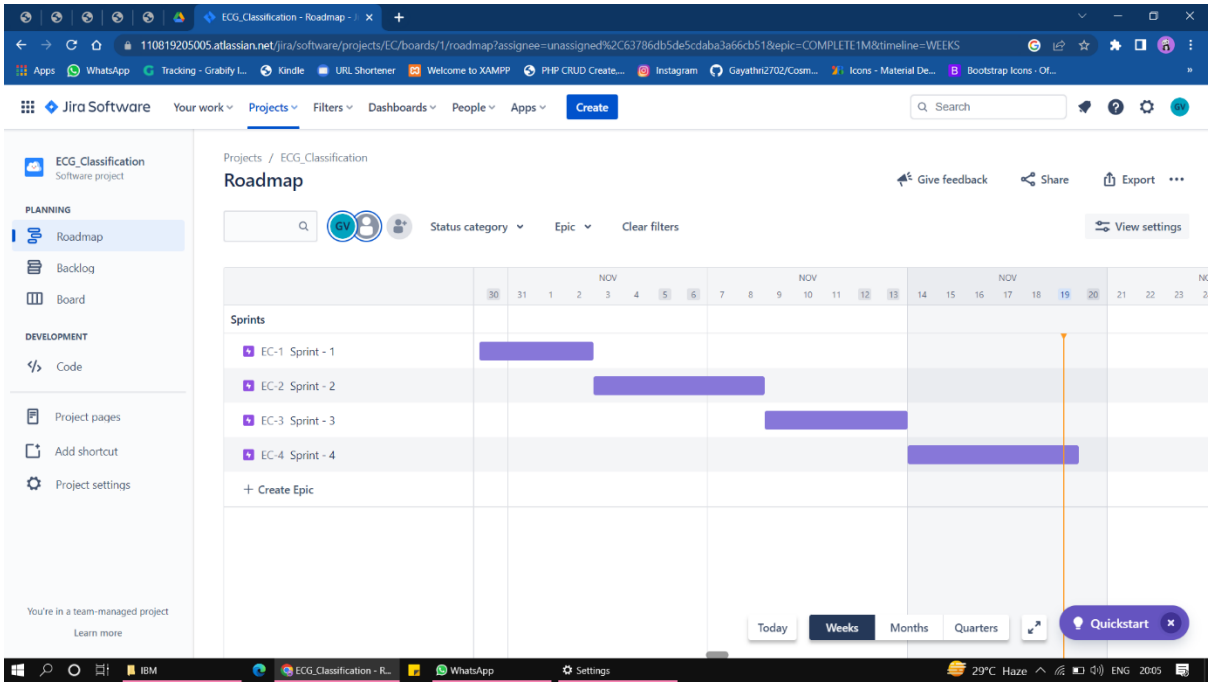
Sprint	Sprint-1	• Download the Dataset • Import ImageDataGenerator • Library • Configure • ImageDataGenerator • class • Import Libraries • Initialize the Model	24 Oct 2022 – 30 Oct 2022

	Sprint – 2	• Register IBM Cloud • Apply ImageDataGenerator functionality to Trainset and • Dataset • Test the model	30 Oct 2022 – 05 Nov 2022

	Sprint – 3	• Train the model on IBM • Create Html files • Train the Model	06 Nov 2022 –11 Nov 2022

	Sprint – 4	• Configure the Learning Process • Build Python code • Adding Dense Layer • Adding CNN layer	13 Nov 2022 –18 Nov 2022

6.3 Reports from JIRA :



7. CODING & SOLUTIONING

The two main features of the application are

- ECG classification
- Prediction of P,Q,R,S,T

7.1 Feature 1: ECG classification :

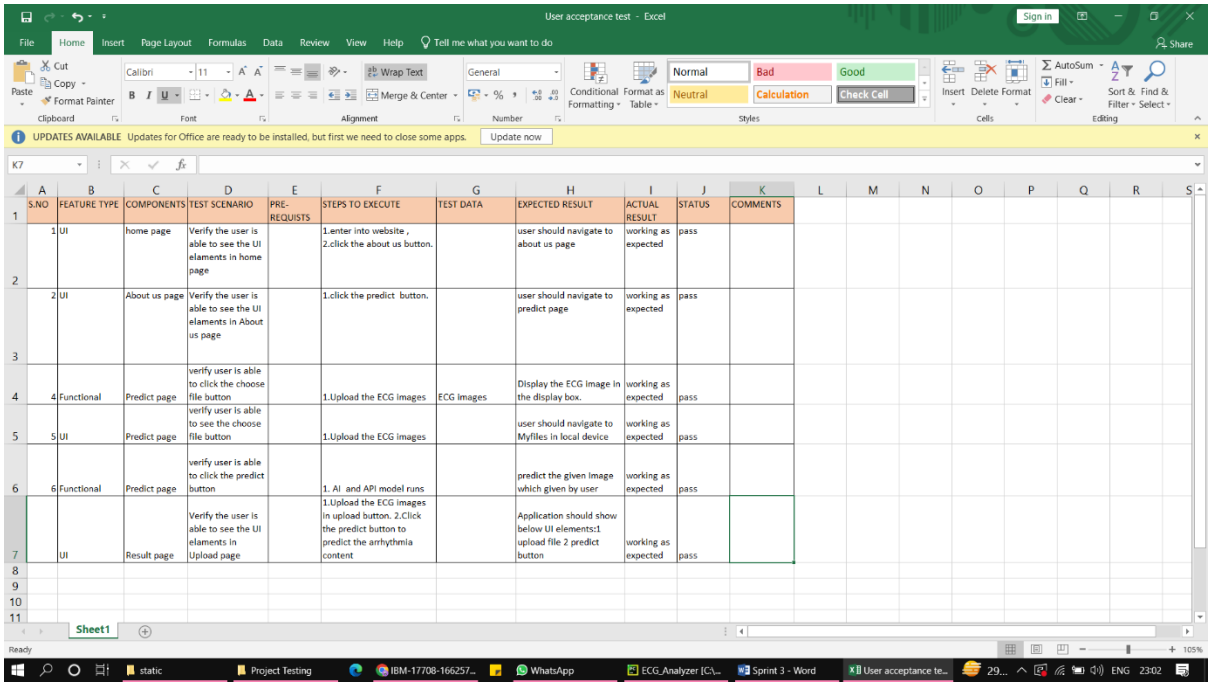
The main foremost feature of the application is classifying the ECG graph from the image which is uploaded by the user. The accuracy of the model is 90%.

7.2 Feature 2: Prediction of P,Q,R,S,T time :

The another important feature of the application is classifying (P,Q,R,S,T) easily . so the application was more Faster to Predict

8. TESTING

8.1 Test cases



S.NO	FEATURE TYPE	COMPONENTS	TEST SCENARIO	PRE-REQUISTS	STEPS TO EXECUTE	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS	COMMENTS
1	UI	home page	Verify the user is able to see the UI elaments in home page		1.enter into website , 2.click the about us button.		user should navigate to about us page	working as expected	pass	
2	UI	About us page	Verify the user is able to see the UI elaments in About us page		1.click the predict button.		user should navigate to predict page	working as expected	pass	
4	Functional	Predict page	verify user is able to click the choose file button		1.Upload the ECG images	ECG images	Display the ECG image in the display box.	working as expected	pass	
5	UI	Predict page	verify user is able to see the choose file button		1.Upload the ECG images		user should navigate to Myfiles in local device	working as expected	pass	
6	Functional	Predict page	verify user is able to click the predict button		1. AI and API model runs		predict the given image which given by user	working as expected	pass	
	UI	Result page	Verify the user is able to see the UI elaments in Upload page		1.Upload the ECG images in upload button. 2.Click the predict button to predict the arrhythmia content		Application should show below UI elements:1 upload file 2 predict button	working as expected	pass	

8.2 User Acceptance Testing

Acceptance Testing
UAT Execution & Report Submission

Date	17 November 2022
Team ID	PNT2022TMID36404
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation
Maximum Marks	4 Marks

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	7	3	24
Duplicate	0	0	3	0	3
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	23	9	14	24	70

9. RESULT

9.1Performance Metrics

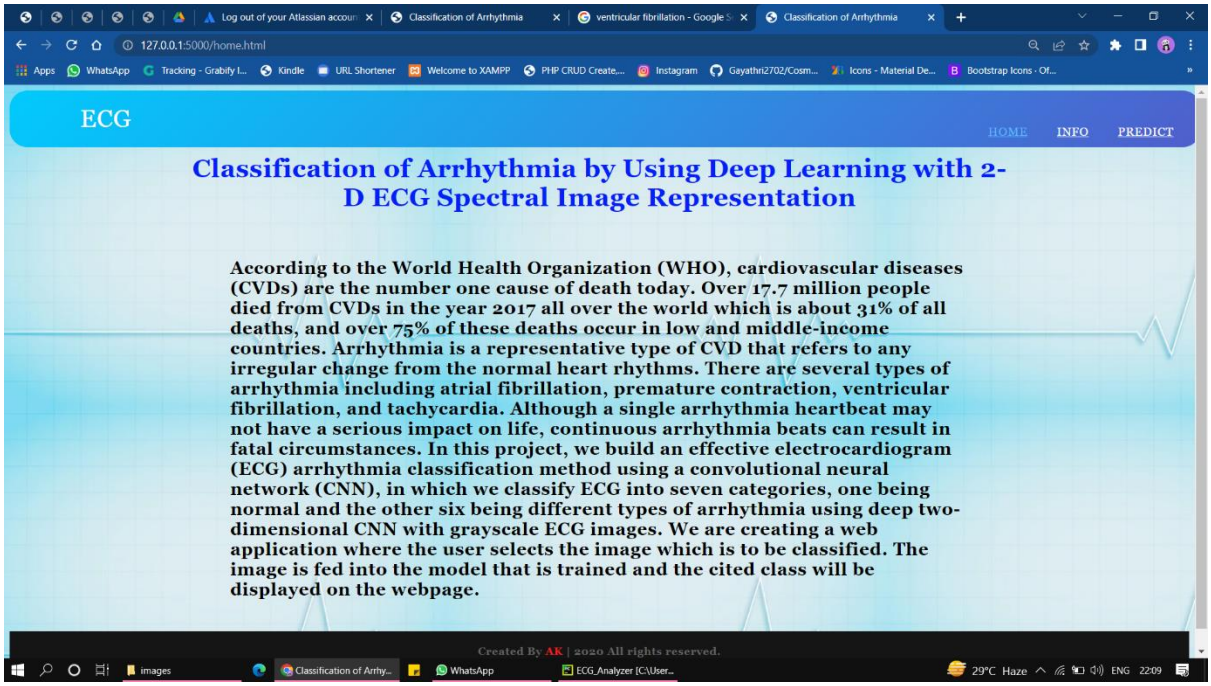
Accuracy

```
[8] print(f"the accuracy is {metrics.accuracy_score(test_data['label'].values, test_data['model_preds'].values)}")

    the accuracy is 1.0
```

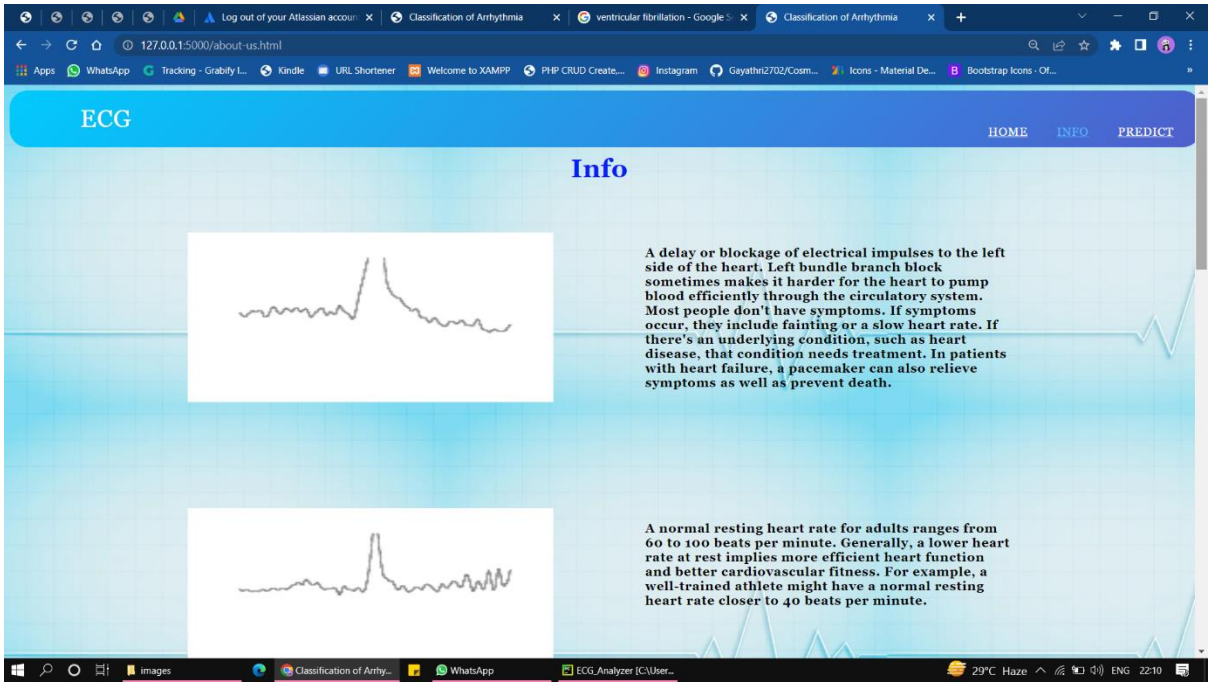
Precision

```
[11] print(f"the precision is {metrics.precision_score(test_data['label'].values, test_data['model_preds'].values, average = 'weighted')}")

     the precision is 1.0
```

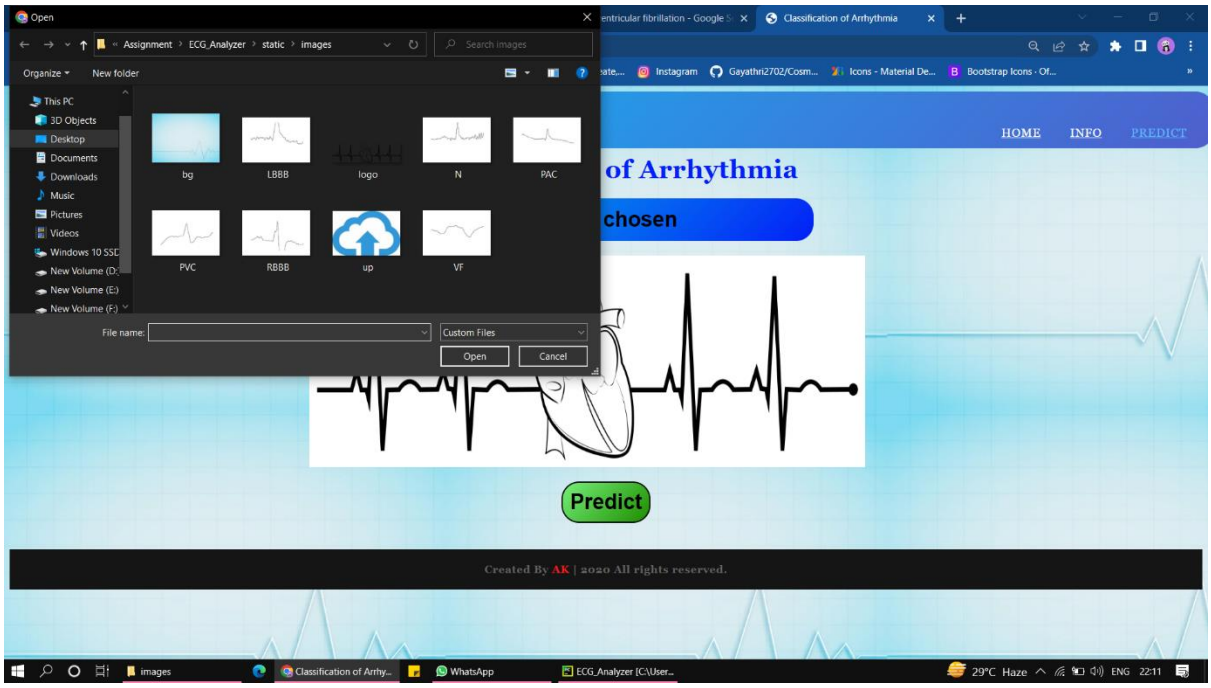
Home.html

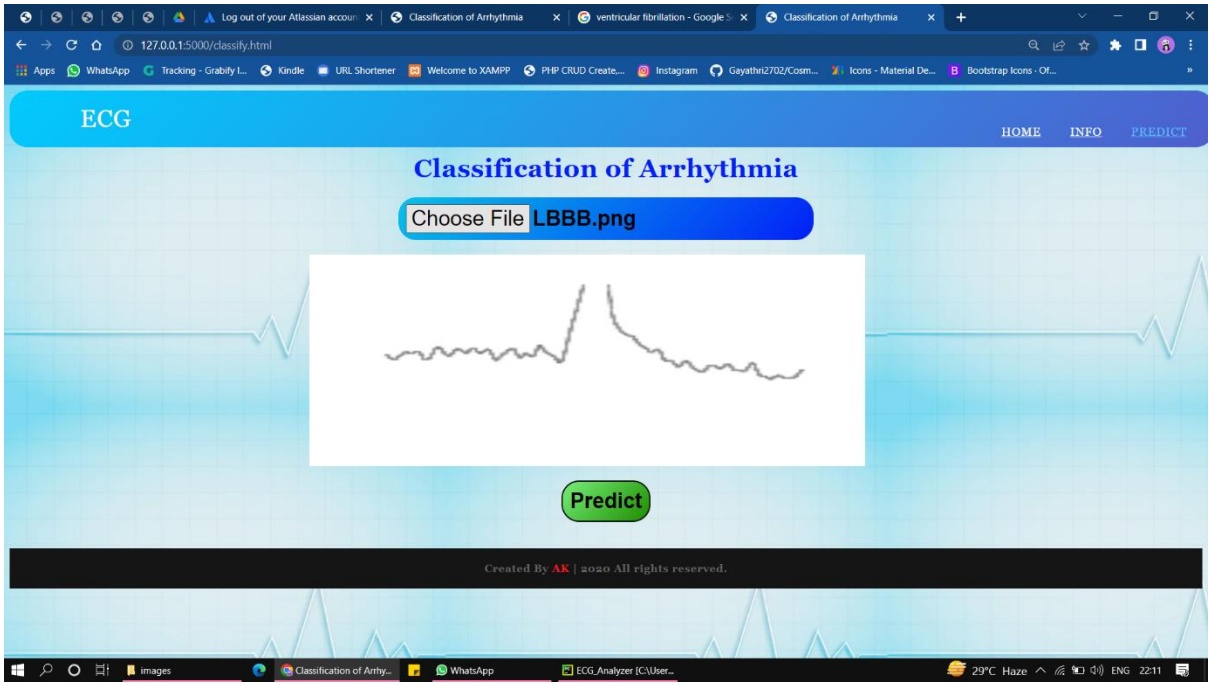


Info.html

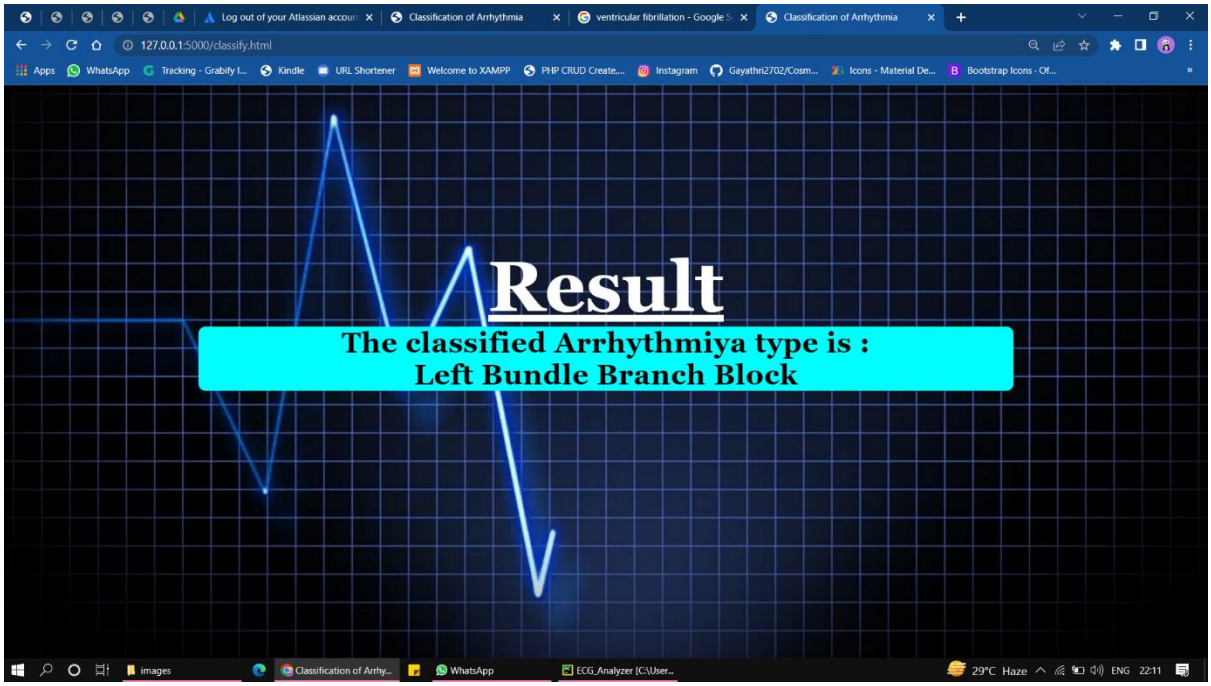


predict.html





result.html



10. CONCLUSION

CNN (Convolutional Neural Network) helps in the process of predicting the image of ECG to the doctors. There are many such technologies developed for the enhancement especially in Hospital. This AI based technology will help the doctors to diagnose the problem in the heart and cure it. Not only doctors but also the patients can able to know their current heart status.

11. FUTURE SCOPE

The future enhancement of this project may include to deploy this application on the public cloud for the world users and an Electro cardio Gram which not only records the heart's rhythms but also predicts the result and gives the result in the graph sheet. This future enhancement will not stress the process of uploading the ECG images in the application.

12. APPENDIX

Github link:

https://github.com/IBM-EPBL/IBM-Project-17708-1659675391

Demo Link:

Source code:

home.html:

```
<html>
<head>
   <title>Classification of Arrhythmia</title>
<link rel="stylesheet" href="/static/csss/style.css">
</head>
<body>


<div class="BG">
         <nav>
           <input type="checkbox" id="check">
           <label for="check" class="checkbtn">
             <i class="fas fa-bars"></i>
           </label>
           <label class="logo">ECG </label>

           <ul>
             <li><a class="active" class="active" href="#">Home</a></li>
           <li><a  href="about-us.html">Info</a></li>
             <li><a  href="classify.html">Predict</a></li>

           </ul>
         </nav>
```

```
      </div>



<h1>Classification of Arrhythmia by Using Deep Learning with 2-D ECG
Spectral Image Representation</h1>
<div id="home" class="container">

      <p>According to the World Health Organization (WHO), cardiovascular
diseases (CVDs) are the number one cause of death today. Over 17.7 million
people died from CVDs in the year 2017 all over the world which is about
31% of all deaths, and over 75% of these deaths occur in low and middle-
income countries. Arrhythmia is a representative type of CVD that refers to
any irregular change from the normal heart rhythms. There are several types
of arrhythmia including atrial fibrillation, premature contraction,
ventricular fibrillation, and tachycardia. Although a single arrhythmia
heartbeat may not have a serious impact on life, continuous arrhythmia
beats can result in fatal circumstances. In this project, we build an
effective electrocardiogram (ECG) arrhythmia classification method using a
convolutional neural network (CNN), in which we classify ECG into seven
categories, one being normal and the other six being different types of
arrhythmia using deep two-dimensional CNN with grayscale ECG images. We are
creating a web application where the user selects the image which is to be
classified. The image is fed into the model that is trained and the cited
class will be displayed on the webpage.</p>
</div>
<footer>

  <div class="bottom">
    <center>
      <span class="credit">Created By <a href="">AK</a> | </span>
      <span class="far fa-copyright"></span><span> 2020 All rights
reserved.</span>
    </center>
  </div>
</footer>
</body>
</html>
```

info.html:

```
<html>
<head>
   <title>Classification of Arrhythmia</title>
<link rel="stylesheet" href="/static/csss/style.css">
</head>
<body >



<div class="BG">
        <nav>
          <input type="checkbox" id="check">
          <label for="check" class="checkbtn">
            <i class="fas fa-bars"></i>
          </label>
          <label class="logo">ECG </label>

          <ul>
```

```
            <li><a  href="home.html">Home</a></li>
          <li><a class="active"  href="#">Info</a></li>
            <li><a  href="classify.html">Predict</a></li>

          </ul>
        </nav>

      </div>



<h1>Info</h1>
<div id="about" class="container">
   <div class="content">
      <img src="static/images/LBBB.png" alt="bg-img">
   </div>
   <div class="content">
      <p>A delay or blockage of electrical impulses to the left side of the
heart.
Left bundle branch block sometimes makes it harder for the heart to pump
blood efficiently through the circulatory system.
Most people don't have symptoms. If symptoms occur, they include fainting
or a slow heart rate.
If there's an underlying condition, such as heart disease, that condition
needs treatment. In patients with heart failure, a pacemaker can also
relieve symptoms as well as prevent death.</p>
   </div>

</div>
<div id="about" class="container">
   <div class="content">
      <img src="static/images/N.png" alt="bg-img">
   </div>
   <div class="content">
      <p>A normal resting heart rate for adults ranges from 60 to 100 beats
per minute. Generally, a lower heart rate at rest implies more efficient
heart function and better cardiovascular fitness.
         For example, a well-trained athlete might have a normal resting
heart rate closer to 40 beats per minute.</p>
   </div>

</div>
<div id="about" class="container">
   <div class="content">
      <img src="static/images/PAC.png" alt="bg-img">
   </div>
   <div class="content">
      <p>Premature atrial contractions (PACs) are extra heartbeats that
start in the upper chambers of your heart.
         When the premature, or early, signal tells the heart to contract,
         there may not be much blood in the heart at that moment.
         That means there's not much blood to pump out.</p>
   </div>

</div>
<div id="about" class="container">
   <div class="content">
      <img src="static/images/PVC.png" alt="bg-img">
   </div>
   <div class="content">
      <p>Premature ventricular contractions (PVCs) are extra heartbeats
```

```
that begin in one of the heart's two
         lower pumping chambers (ventricles). These extra beats disrupt the
regular heart rhythm,
         sometimes causing a sensation of a fluttering or a skipped beat in
the chest.</p>
   </div>

</div>
<div id="about" class="container">
   <div class="content">
      <img src="static/images/RBBB.png" alt="bg-img">
   </div>
   <div class="content">
      <p>In RBBB, the left ventricle is activated normally, thus the early
part of the QRS complex correlating to septal depolarisation is unchanged
There is delayed activation of the right ventricle as depolarisation
originates from the left ventricle across the septum. This produces a
secondary R wave (R') in the precordial leads, and a wide, slurred S wave
in lateral leads
Normal activation of the left ventricle means that cardiac axis remains
normal in isolated RBBB</p>
   </div>

</div>
<div id="about" class="container">
   <div class="content">
      <img src="static/images/VF.png" alt="bg-img">
   </div>
   <div class="content">
      <p>Ventricular fibrillation (VF) is a rapid, life-threatening heart
rhythm starting in the bottom chambers of the heart. It can be triggered by
a heart attack.
Because the heart doesn't pump adequately during ventricular fibrillation,
sustained VF can cause low blood pressure, loss of consciousness or death.
Emergency treatment includes immediate defibrillation with an automated
external defibrillator (AED) and cardiopulmonary resuscitation (CPR).
         Long-term therapy includes implantable defibrillators and
medications to prevent recurrence.</p>
   </div>

</div>
<footer>

  <div class="bottom">
    <center>
      <span class="credit">Created By <a href="">AK</a> | </span>
      <span class="far fa-copyright"></span><span> 2020 All rights
reserved.</span>
    </center>
  </div>
</footer>
</body>
</html>
```

predict.html:

```
<html>
<head>
   <title>Classification of Arrhythmia</title>
```

```
<link rel="stylesheet" href="/static/csss/style.css">
</head>
<body>


<div class="BG">
        <nav>
          <input type="checkbox" id="check">
          <label for="check" class="checkbtn">
            <i class="fas fa-bars"></i>
          </label>
          <label class="logo">ECG </label>

          <ul>
            <li><a  href="home.html">Home</a></li>
          <li><a  href="about-us.html">Info</a></li>
            <li><a class="active" href="#">Predict</a></li>

          </ul>
        </nav>

      </div>


<h1>Classification of Arrhythmia</h1>
<div id="predict" >
   <form action="/classify.html" id="upload-file" method="post"
enctype="multipart/form-data">
   <input type="file" name="file" id="file" class="btn1" value="Choose
File" accept="image/png, image/jpeg ,image/jpg">
   <input type="image" id="img" class="img" src="/static/images/logo.png" >
   <input type="submit" id="btn2" class="btn2" value="Predict">
   </form>
</div>



<footer>

  <div class="bottom">
    <center>
      <span class="credit">Created By <a href="">AK</a> | </span>
      <span class="far fa-copyright"></span><span> 2020 All rights
reserved.</span>
    </center>
  </div>
</footer>
<script>
const chooseFile = document.getElementById("file");
const imgPreview = document.getElementById("img");
file.addEventListener("change", function () {
  getImgData();
});
function getImgData() {
  const files = chooseFile.files[0];
  if (files) {
    const fileReader = new FileReader();
    fileReader.readAsDataURL(files);
    fileReader.addEventListener("load", function () {
      console.log(imgPreview.src);
```

```
      imgPreview.src =this.result;
    });
  }
}
</script>

</body>
</html>
```

result.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Classification of Arrhythmia</title>
    <link rel="stylesheet" href="/static/csss/style.css">
</head>
<body>

<div class="BG">
        <nav>
          <input type="checkbox" id="check">
          <label for="check" class="checkbtn">
            <i class="fas fa-bars"></i>
          </label>
          <label class="logo">ECG </label>

          <ul>
            <li><a  href="home.html">Home</a></li>
          <li><a  href="about-us.html">Info</a></li>
            <li><a  href="classify.html">Predict</a></li>

          </ul>
        </nav>

      </div>




<video autoplay muted loop id="myVideo">
  <source src="/static/video/ecg.mp4" type="video/mp4">

</video>

<div class="content1">
  <h11><u>Result</u></h11>
  <h2>The classified Arrhythmiya type is : <br>{{result}}</h2>

</div>
```

```
</body>
</html>
```

style.css

```
body{
position:relative;
font-size:18px;
font-weight:bold;
letter-spacing:.7px;
background-image:url("/static/images/bg.jpg");

}

.BG{
  background:  no-repeat;
  background-size: cover;
  background-position: center;
  width: 100%;
  height: 90px;
  align-items: top;
}
nav{
  position: fixed;
   background: linear-gradient(135deg,#01cbfd 0%, #505fcd 100%);
  height: 80px;
  width: 100%;
  border-radius: 25px;

}
label.logo{
  font-family: 'Poppins';
  color: white;
  float: left;
  font-size: 35px;
  line-height: 80px;
  padding: 0 100px;
  font-weight: 500;
}
label.logo:hover{
  color: #070707;
}
nav ul{
  float: right;
  margin-right: 20px;
}
nav ul li {
  display: inline-block;
  line-height: 80px;
  margin: 0 5px;
}
```

```css
nav ul li a{
  font-family: 'Poppins';
  font-size: 17px;
  font-weight: 500;
  padding: 7px 13px;
  border-radius: 3px;
  text-transform: uppercase;
  color: white;
}
a.active,a:hover{
  color: #5eb9ff;
  transition: .5s;
}
.checkbtn{
  font-size: 30px;
  color: white;
  float: right;
  line-height: 80px;
  margin-right: 40px;
  cursor: pointer;
  display: none;
}
#check{
  display: none ;
}




.container,#predict{
display:flex;
flex-wrap:wrap;
justify-content:space-around;
border-radius:10px;
width:80%;
padding:20px;
margin:auto;
margin-top:50px;
}
#home{
margin:auto;
   width:65%;
   font-size:25px;
}

.container:nth-child(n){
margin-bottom:50px;
}
.content{
height:300px;
width:40%;`
}
.content img{
height:80%;
width:100%;
background:#fff;
padding:10px;
}
#predict{
display:block;
```

```
}
h1{margin:auto;
margin-bottom:20px;
text-align:center;
width:70%;

color:#0320f8;

}

h11{margin:auto;
margin-bottom:20px;
text-align:center;
width:70%;
font-size:100px;
color:white;


}

h2{margin:auto;
margin-bottom:10px;
text-align:center;
font-size:40px;
width:70%;
border-radius:10px;
background-color:Aqua;


}

.btn1{
margin:auto;
margin-top:-50px;
display:block;
padding:10px;
text-align:center;
font-size:30px;
font-weight:bold;
background:#fff;
outline:none;
border:none;
border-radius: 25px;
background: linear-gradient(135deg,#08c3e9 0%,#0320f8 100%);
}
.btn2{
margin:auto;
display:block;
padding:10px;
text-align:center;
font-size:30px;
font-weight:bold;

outline:none;
cursor: pointer;

border-radius: 25px;
```

```
background: linear-gradient(135deg,#77f07b 0%,#1f8e00 100%);
}









.img{
margin:auto;
margin-left:18%;
margin-top:20px;
margin-bottom:20px;
width:60%;
height:300px;
background:#fff;
padding:10px;
}



.content1{
  position: relative;
  margin: 130px auto;
  text-align: center;
  padding: 0 20px;
}
.content1 .text{
  font-size: 2.5rem;
  font-weight: 600;
  color: #202020;
}
.content1 .p{
  font-size: 2.1875rem;
  font-weight: 600;
  color: #ffffff;
}
footer{

  bottom: 0px;
  width: 100%;
  align-items:bottom;
  background: linear-gradient(135deg,#01c0f5 0%,#01024b 100%);
}

.bottom center{
  padding: 20px;
  font-size: 0.9375rem;
  background: #151515;
}
.bottom center span{
  color: #656565;
}
.bottom center a{
  color: #f12020;
  text-decoration: none;
}
```

```css
.bottom center a:hover{
  text-decoration: underline;
}



* {
  box-sizing: border-box;
}



#myVideo {
margin-top:80px;
  position: fixed;
  right: 0;
  bottom: 0;
  min-width: 100%;
  min-height: 100%;
}




#myBtn {
  width: 200px;
  font-size: 18px;
  padding: 10px;
  border: none;
  background: #000;
  color: #fff;
  cursor: pointer;
}

#myBtn:hover {
  background: #ddd;
  color: black;
}



@media screen and (max-width: 900px) {
  footer{
    position: relative;
    bottom: 0px;
  }
  .main-content{
    flex-wrap: wrap;
    flex-direction: column;
  }
  .main-content .box{
    margin: 5px 0;
  }
}
```

app.py:

```
import requests
from flask import Flask, render_template, request
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import os


app = Flask(__name__)
model=load_model('ECG.h5')


@app.route('/')
@app.route('/home.html')
def home():
    return render_template('home.html')


@app.route('/about-us.html')
def about():
    return render_template('about-us.html')


@app.route('/classify.html', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':

        f = request.files['file']  # requesting the file
        basepath = os.path.dirname('__file__')  # storing the file
directory
        filepath = os.path.join(basepath, "uploads", f.filename)  # storing
the file in uploads folder
        f.save(filepath)  # saving the file

        img = image.load_img(filepath, target_size=(64, 64))  # load and
reshaping the image
        x = image.img_to_array(img)  # converting image to an array
        x = np.expand_dims(x, axis=0)  # changing the dimensions of the
image

        pred = np.argmax(model.predict(x), axis=1)
        print("prediction", pred)  # printing the prediction
        index=['Left Bundle Branch Block', 'Normal', 'Premature Atrial
Contraction', 'Premature Ventricular Contractions', 'Right Bundle Branch
Block', 'Ventricular Fibrillation']


        result = str(index[pred[0]])

        x = result
        print(x)

        return render_template('result.html', result=x)
```

```
    else:
        return render_template('classify.html')



if __name__ == '__main__':
    app.run()
```