CLASSIFICATION OF ARRHYTHMIA BY USING DEEP LEARNING WITH 2-D SPECTRAL IMAGE REPRESENTATION

IBM NALAIYA THIRAN

PROJECT REPORT

TEAM ID: PNT2022TMID02236

Submitted by

Team head:

Rifhath Aslam J (190701162)

Team members:

Shahul Hameed PTS (190701196)

Yogesh Kumar R (190701270)

Rohit Gangadhar P (190701165)

TABLE OF CONTENTS

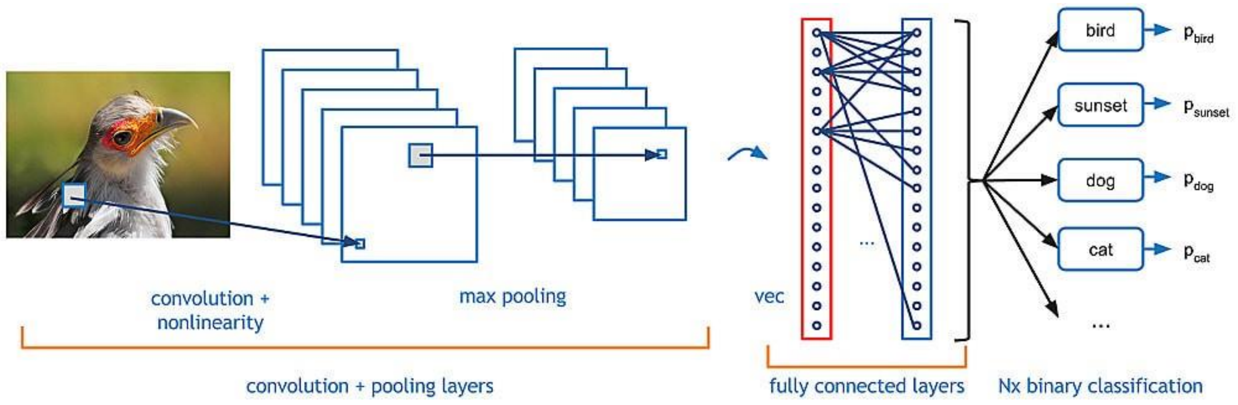
1. Introduction:

a. Overview:

Over 17 million people are estimated to die each year from cardiovascular diseases (CVDs), making them the largest cause of mortality in humans. Three-fourths of all CVD fatalities, according to the World Heart Federation, occur in the low- and middle-income groups of society. By offering prompt treatment, a classification model to detect CVDs in their early stages might significantly lower death rates. Cardiac arrhythmia, in which heartbeats are known to vary from their regular beating pattern, is one of the primary causes of CVDs. With age, body size, exercise, and emotions, a normal heartbeat fluctuates. Palpitations are a condition when the heartbeat seems abnormally rapid or sluggish. Although an arrhythmia may indicate that the heart is beating excessively quickly or slowly, It suggests that the heart's normal rhythm is irregular. It might indicate tachycardia (heart rate greater than 100 beats per minute (bpm)), bradycardia (heart rate less than 60 bpm), a missed pulse, or in severe circumstances, cardiac arrest. Atrial fibrillation, atrial flutter, and ventricular fibrillation are a few further typical varieties of irregular cardiac rhythms. These deviations indicate diverse heart arrhythmia types and can be divided into several subclasses. Patients with cardiac disease may benefit from a precise classification of these categories to aid in diagnosis and therapy. Arrhythmia can refer to irregular heartbeats, whether they are rapid or slow, or to patterns that cannot be explained by a regular heartbeat. In clinical practise, an automated detection of these patterns is extremely important. The recognised features of cardiac arrhythmia include, where the detection requires expert clinical knowledge

b. Purpose:

In the past few decades,Deep Learning has proved to be a compelling toolbecause of its ability to handle large amounts of data. The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is AI Convolution Neural Networks.



In deep learning, a convolution al neural network(CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modifiedby the other.

2. Literature Survey:

2.1 Existing Problem:

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 worldwhich 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 refersto any irregular change from the normalheart rhythms.

There are severaltypes of arrhythmia including **atrial fibrillation, prematurecontraction, ventricular fibrillation, and tachycardia**.

2.2 References:

1.Mc Namara, K.; Alzubaidi, H.; Jackson, J.K. Cardiovascular disease as a leading cause of death: How are
pharmacists getting involved? Integr. Pharm. Res. Pract. 2019,8, 1. [CrossRef] [PubMed]

2.Lackland, D.T.; Weber, S.M.A. Global burden of cardiovascular disease and stroke:
hypertension at the core.
Can. J. Cardiol. 2015,31, 569–571. [CrossRef] [PubMed]

3.Mustaqeem, A.; Anwar, S.M.; Majid, M. A modular cluster based collaborative
recommender system for
cardiac patients. Artif. Intell. Med. 2020,102, 101761. [CrossRef] [PubMed]

4.Irmakci, I.; Anwar, S.M.; Torigian, D.A.; Bagci, U. Deep Learning for Musculoskeletal Image
Analysis. arXiv
2020, arXiv:2003.00541.

5.Anwar, S.M.; Majid, M.; Qayyum, A.; Awais, M.; Alnowami, M.; Khan, M.K. Medical image
analysis using
convolutional neural networks: A review. J. Med. Syst. 2018,42, 226. [CrossRef]

6.Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai,
J.; et al.
Recent advances in convolutional neural networks. Pattern Recognit. 2018,77, 354–377.
[CrossRef]

7.Wu, Y.; Yang, F.; Liu, Y.; Zha, X.; Yuan, S. A comparison of 1-D and 2-D deep convolutional
neural networks
in ECG classification. arXiv 2018, arXiv:1810.07088.

8.Zhao, J.; Mao, X.; Chen, L. Speech emotion recognition using deep 1D & 2-D CNN LSTM
networks.
Biomed. Signal Process. Control 2019,47, 312–323.

9.Ortega, S.; Fabelo, H.; Iakovidis, D.K.; Koulaouzidis, A.; Callico, G.M. Use of
hyperspectral/multispectral
imaging in gastroenterology. Shedding some–different–light into the dark. J. Clin. Med.
2019,8, 36.[CrossRef]

10.Feng, Y.-Z.; Sun, D.-W. Application of Hyperspectral Imaging in Food Safety Inspection and
Control:
A Review. Crit. Rev. Food Sci. Nutr. 2012,52, 1039–1058. [CrossRef]

11.Lorente, D.; Aleixos, N.; Gómez-Sanchis, J.; Cubero, S.; García-Navarrete, O.L.; Blasco, J.
Recent
Advances and Applications of Hyperspectral Imaging for Fruit and Vegetable Quality
Assessment.
Food Bioprocess Technol. 2011,5, 1121–1142. [CrossRef]

12.Tatzer, P.; Wolf, M.; Panner, T. Industrial application for inline material sorting using
hyperspectral imaging
in the NIR range. Real-Time Imaging 2005,11, 99–107. [CrossRef]

13.Kubik, M. Chapter 5 Hyperspectral Imaging: A New Technique for the Non-Invasive Study
of Artworks.
Phys. Tech. Study Art Archaeol. Cult. Herit. 2007,2, 199–259.

14.Hassan, H.; Bashir, A.K.; Abbasi, R.; Ahmad, W.; Luo, B. Single image defocus estimation
by modified
gaussian function. Trans. Emerg. Telecommun. Technol. 2019,30, 3611. [CrossRef]

15.
Ahmad, M.; Bashir, A.K.; Khan, A.M. Metric similarity regularizer to enhance pixel similarity
performance
for hyperspectral unmixing. Optik 2017,140, 86–95. [CrossRef]

16.Salem, M.; Taheri, S.; Yuan, J.S. ECG arrhythmia classification using transfer learning from
2-dimensional
deep CNN features. In Proceedings of the 2018 IEEE Biomedical Circuits and Systems
Conference (BioCAS),
Cleveland, OH, USA, 17–19 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.

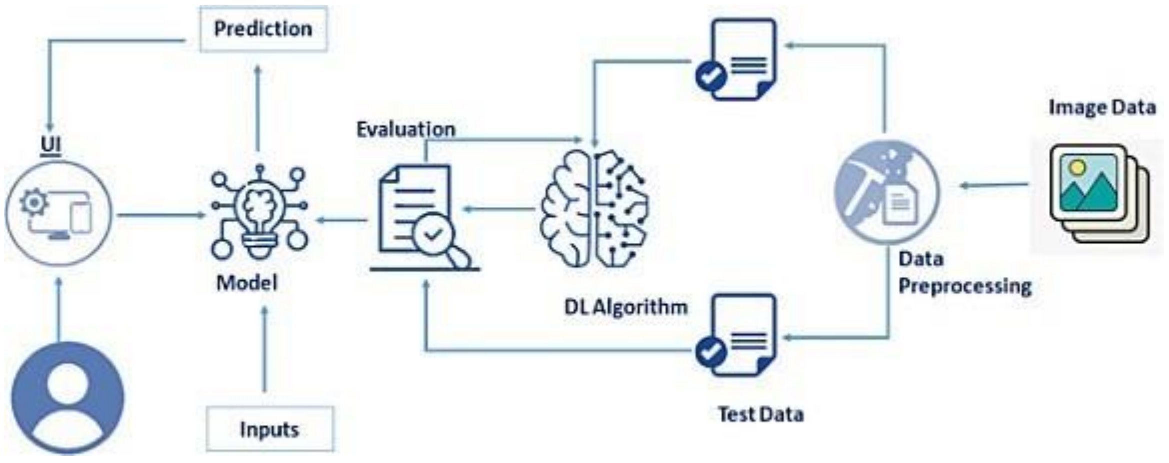
17.Mustaqeem, A.; Anwar, S.M.; Khan, A.R.; Majid, M. A statistical analysis based
recommender model for
heart disease patients. Int. J. Med. Inform. 2017,108, 134–145. [CrossRef]

2.3 **ProposedSolution:**

An "ambulatory electrocardiogram" or an ECG) about the size of a postcardor digital camera that the patient will be using for 1 to 2 days, or up to 2 weeks. The test measures the movement of electrical signals or waves through the heart. These signals tell the heart to contract (squeeze) and pump blood. The patient will have electrodes taped to your skin. It's painless, although some people have mild skin irritation from the tape usedto attach the electrodes to the chest. They can do everything but shower or bathe while wearing the electrodes. After the test period, patient will go back to see your doctor.They will be downloading the information.

Theoretical Experience:

Block Diagram:



We will preparethe project by following the below steps:

i. We will be working with Sequential type of modeling

ii. We will be working with Keras capabilities

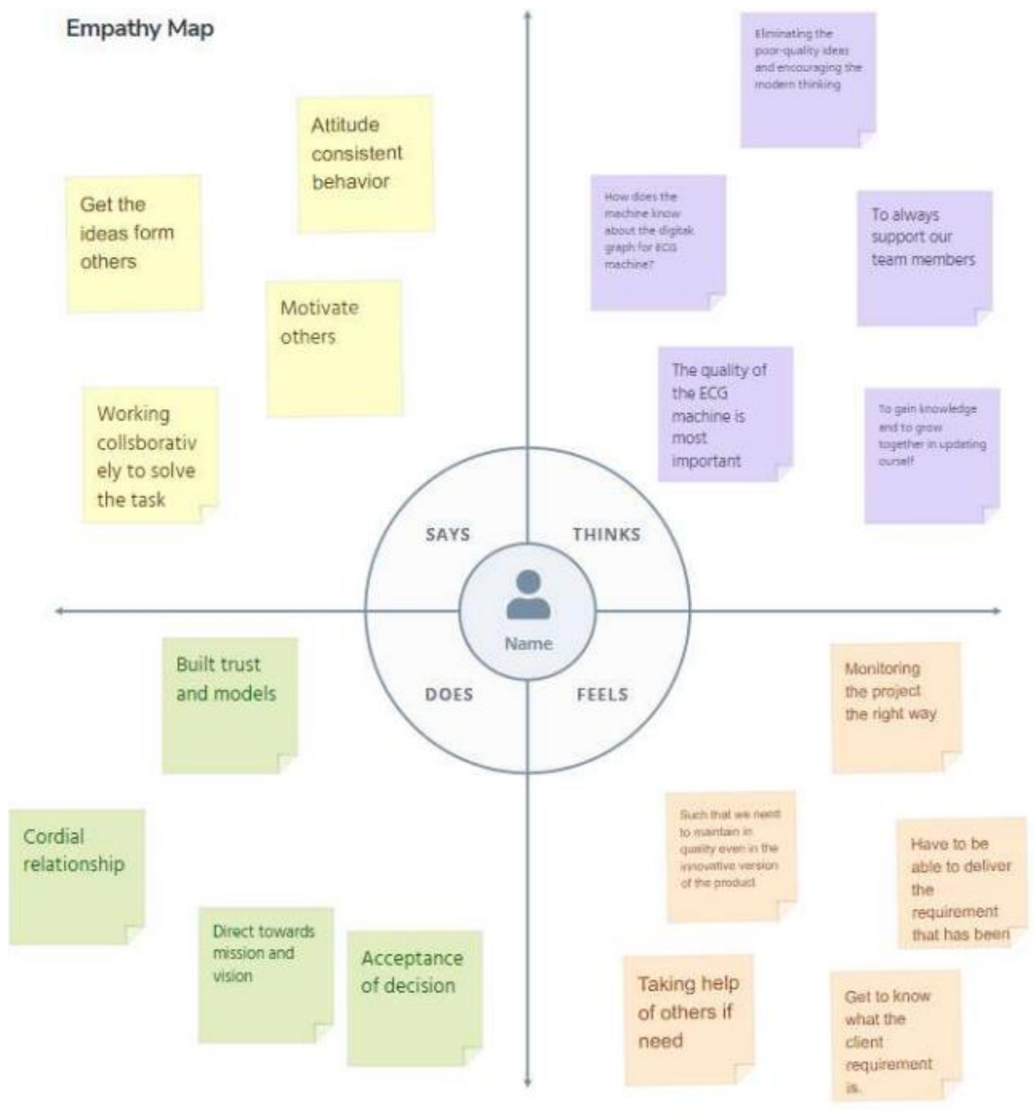
iii. We will be working with image processing techniques

iv. Wewill build a web application using the Flaskframework.

v. Afterwards we will be training our dataset in the IBM cloud andbuilding anothermodel from IBM and we willalso test it.

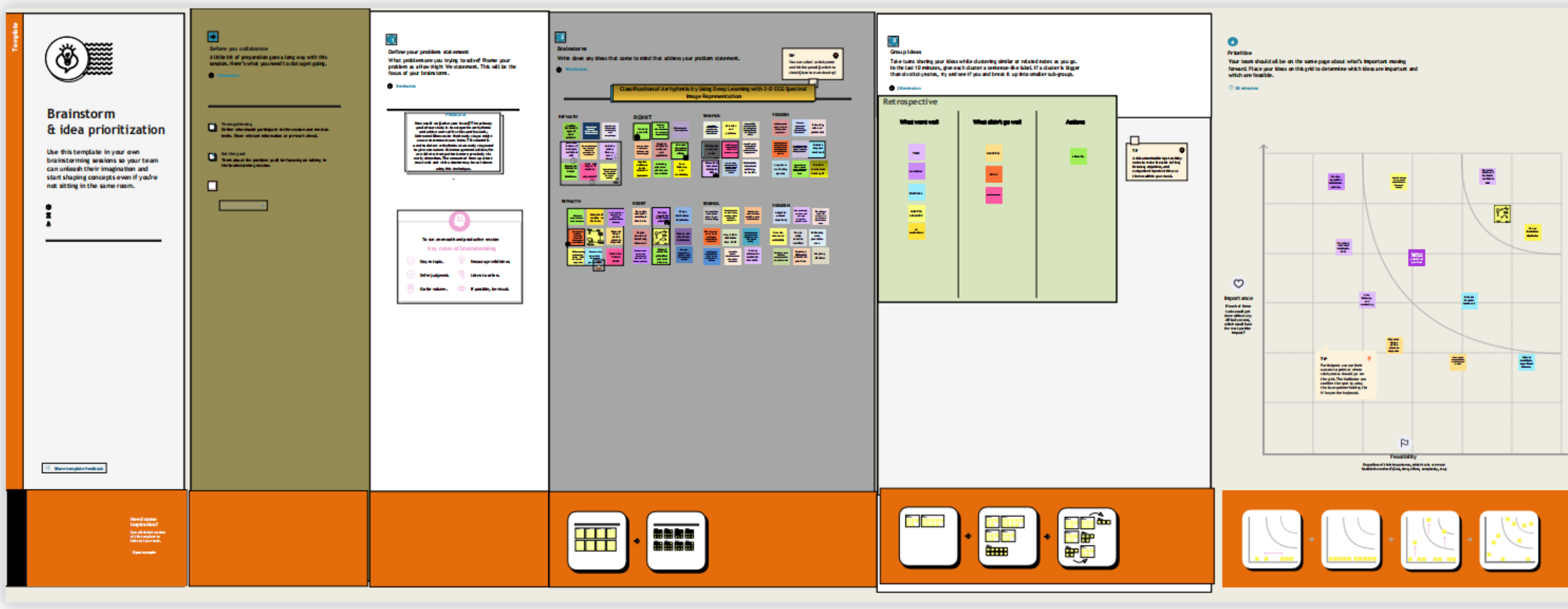
3 IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP:



Empathy Map
Attitude consistent behavior
Get the ideas form others
Motivate others
Working collsboratively to solve the task
Eliminating the poor-quality ideas and encouraging the modern thinking
How does the machine know about the digitak graph for ECG machine?
To always support our team members
The quality of the ECG machine is most important
To gain knowledge and to grow together in updating ourself
SAYS
THINKS
Name
DOES
FEELS
Built trust and models
Cordial relationship
Direct towards mission and vision
Acceptance of decision
Monitoring the project the right way
Such that we need to maintain in quality even in the innovative version of the product
Have to be able to deliver the requirement that has been
Taking help of others if need
Get to know what the client requirement is.

3.2 PROBLEM SOLUTION FIT:



Template

Brainstorm
& idea prioritization

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.

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

Key rules of brainstorming
Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Brainstorm
Write down any ideas that come to mind that address your problem statement.

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

Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

Retrospective

What went well	What didn't go well	Actions

Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

4. PREREQUIREMENTS ANALYSIS:

Hardware Components used:

Since we are using the IBM cloud as a platform to execute this project we don’t need any hardwarecomponents other than our system.

Software Components Used:

We will be using Visual Studio which is installed in our system and Watsonstudio from the IBM cloud to complete the project.

Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux andmacOS. Features include support for debugging, syntax highlighting, intelligentcode completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions thatadd additional functionality

WATSON STUDIO:

Watson Studio is one of thecore services in Cloud Pak for Data as a Service.

Watson Studio provides you with the environment and tools to solve your business problems by collaboratively workingwith data. You can choosethe tools you need toanalyze and visualize data, to cleanse and shape data, or to build machine learning

models.

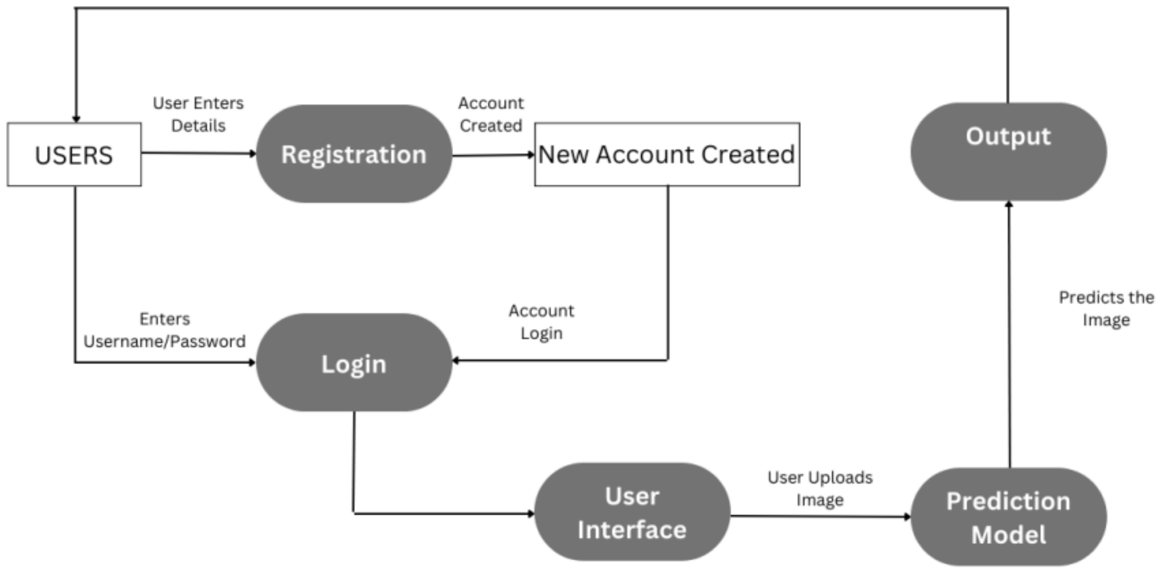
This illustration shows how the architecture of Watson Studio is centeredaround the

project. A project is a workspace where you organize your resources and workwith data.

Watson Studioprojects fully integrate with the catalogs and deployment spaces:

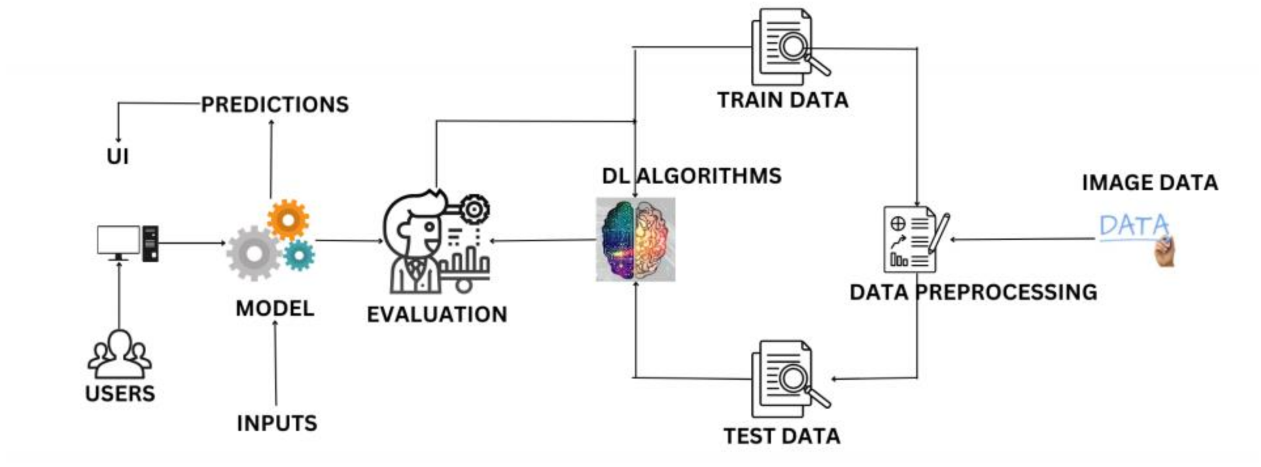
1. Deployment spaces are provided by the Watson Machine Learning serviceYou can easily move assets between projects and deploymentspaces.

5. DATAFLOW DIAGRAM:



6.

SOLUTION ARCHITECTURE:



7. PROJECT PLANNING:

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	5 Days	24 Oct 2022	28 Oct 2022	20	28 Oct 2022
Sprint-2	20	5 Days	30 Oct 2022	04 Nov 2022	20	04 Nov 2022
Sprint-3	20	5 Days	06 Nov 2022	11 Nov 2022	20	11 Nov 2022
Sprint-4	20	5 Days	13 Nov 2022	18 Nov 2022	20	18 Nov 2022

8. CODING AND SOLUTION:

Experimental Investigations:

In this project, we have deployed our training model using CNN on IBM Watson studioand in our local machine.We are deploying 4 typesof CNN layers in a sequential manner, starting from :

- **Convolutional layer 2D:**A 2-D convolutional layer applies slidingconvolutional filtersto 2-D input. The layer convolves theinput bymoving the filters along the input vertically and horizontally and computingthe dot product of the weights and the input, and then adding a bias term.
- **Pooling Layer :**Pooling layers are used to reduce the dimensions of the featuremaps. Thus, it reduces the number of parameters to learnand the amount of computation performed in the network. Thepooling layer summarises the features present in a region of the feature map generatedby a convolution layer.

- **Fully-Connected layer :**After extracting features from multiple convolution layers and pooling layers, the fully-connected layer is used to expand the connection of all features. Finally, the SoftMax layer makes a logisticregression classification. Fully-connected layertransfers the weighted sum of the output of the previous layerto the activation function.

- **Dropout Layer :**There is usually a dropout layer before the fully-connected layer. The dropout layer will temporarily disconnect someneurons from the network according to the certain probability duringthe trainingof the convolution neural network,which reduces the joint adaptability between neuron nodes, reduces overfitting, and enhances the generalization abilityof the network.

Flow Chart & Results with Screenshots:

a. Flow Chart & Resultsby training model in localmachine:

i. Dataset Collection:

The dataset containssix classes:

1. Left BundleBranch Block
2. Normal
3. Premature AtrialContraction
4. Premature Ventricular Contractions
5. Right BundleBranch Block
6. Ventricular Fibrillation

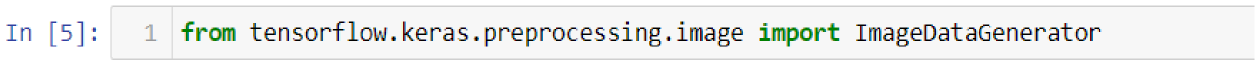
ii. Image Preprocessing:

Image Pre-processing includesthe following main tasks

1. **Import ImageDataGenerator Library:**

Image data augmentation is a techniquethat can be used to artificially expand thesizeof a training dataset by creating modified versions of images in the dataset.

The Keras deep learning neuralnetwork library providesthe capability to fitmodelsusing image data augmentation via the ImageDataGenerator class.



```
In [5]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

1. **Configure ImageDataGenerator Class:**

There are five main types of data augmentation techniques for image data; specifically:

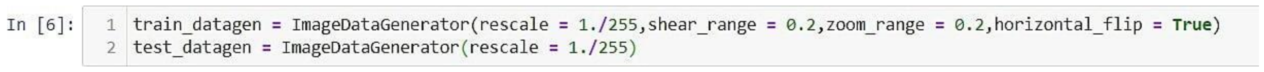
1. Image shifts via the width_shift_range and height_shift_range arguments.
2. Image flips via the horizontal_flip and vertical_flip arguments.

3. Image rotates via the rotation_range argument

4. Image brightness via the brightness_range argument.

5. Image zooms via the zoom_range argument.

An instance of the ImageDataGenerator class can be constructed for train and test.



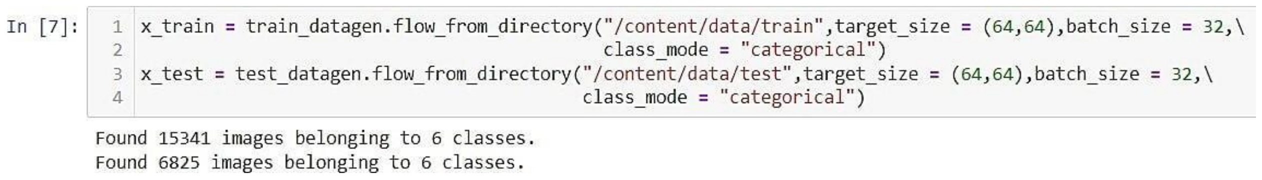
```
In [6]: train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = True)
        test_datagen = ImageDataGenerator(rescale = 1./255)
```

1. **Applying ImageDataGenerator functionality to the trainset and test set:**

We will apply ImageDataGenerator functionality to Trainset and Testset by using the following code

This function will return batches of images from the subdirectories Left Bundle Branch Block, Normal, Premature Atrial Contraction, Premature Ventricular Contractions, RightBundle Branch Block and Ventricular Fibrillation, together with labels 0 to 5{'Left BundleBranch Block': 0, 'Normal': 1, 'Premature Atrial Contraction': 2, 'Premature Ventricular

Contractions': 3, 'Right Bundle Branch Block': 4, 'Ventricular Fibrillation': 5}



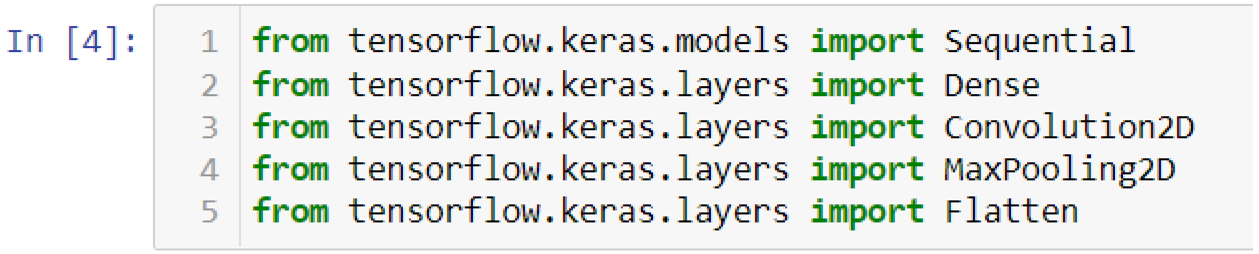
```
In [7]: x_train = train_datagen.flow_from_directory("/content/data/train",target_size = (64,64),batch_size = 32,\
                                                   class_mode = "categorical")
        x_test = test_datagen.flow_from_directory("/content/data/test",target_size = (64,64),batch_size = 32,\
                                                 class_mode = "categorical")

        Found 15341 images belonging to 6 classes.
        Found 6825 images belonging to 6 classes.
```

We can see that for training there are 15341 images belonging to 6 classes and fortesting there are 6825 images belonging to 6 classes.

1. **Model Building**

We are ready with the augmented and pre-processed image data,we will begin our build our model by following the below steps:

1. **Import the model building Libraries:**



```
In [4]:
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
```

- **Initializing the model:**

Keras has 2 ways to define a neural network:

1. Sequential
2. Function API

The Sequential class is used to define linear initializations of network layers which then,collectively, constitute a model. In our examplebelow, we will use the Sequential constructor to create a model, which will then have layers added to it using the add ()method.

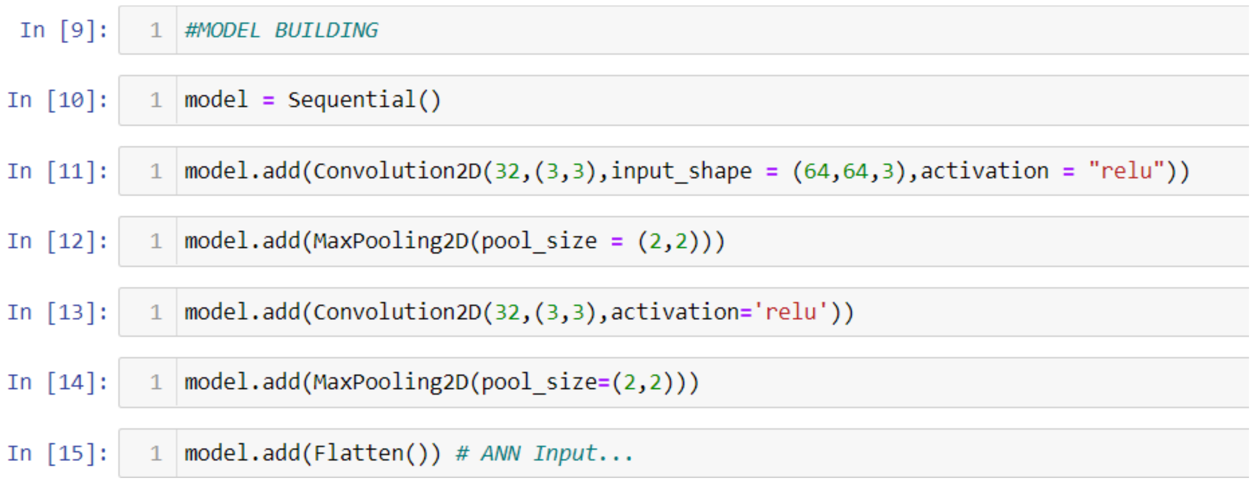
Now, will initialize our model.

1. **Adding CNN Layers:**

We are adding a convolution layer with an activation functionas “relu” and with asmallfiltersize (3,3) and a number of filters as (32) followed by a max-pooling layer.

The Max pool layer is used to downsample the

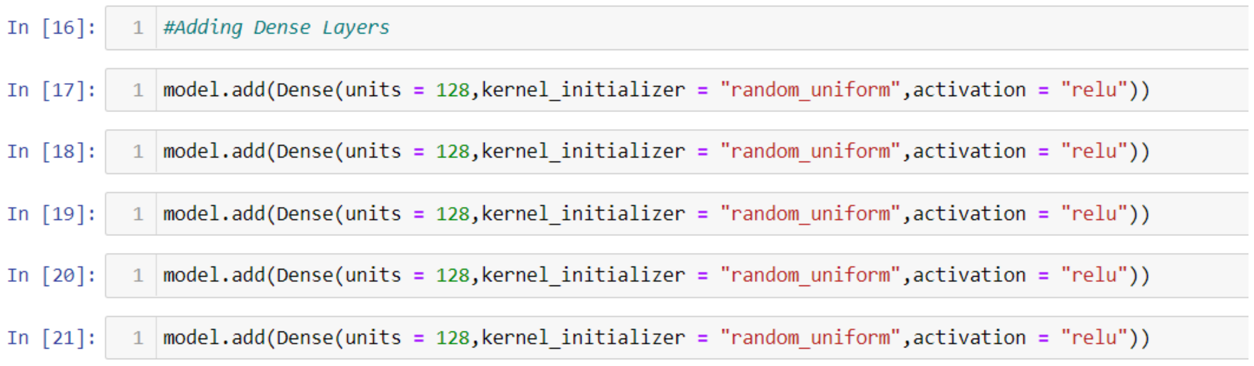
input.The flatten layer flattens the input.



```
In [9]:  #MODEL BUILDING

In [10]: model = Sequential()

In [11]: model.add(Convolution2D(32,(3,3),input_shape = (64,64,3),activation = "relu"))

In [12]: model.add(MaxPooling2D(pool_size = (2,2)))

In [13]: model.add(Convolution2D(32,(3,3),activation='relu'))

In [14]: model.add(MaxPooling2D(pool_size=(2,2)))

In [15]: model.add(Flatten()) # ANN Input...
```

- ## Adding Hidden Layers:

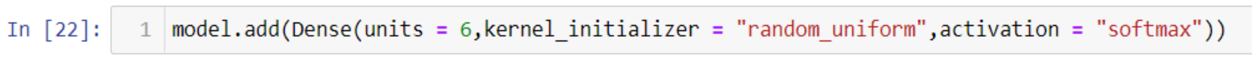
Dense layer is deeply connected neuralnetwork layer. It is most common and frequently used layer



```
In [16]: #Adding Dense Layers

In [17]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation = "relu"))

In [18]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation = "relu"))

In [19]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation = "relu"))

In [20]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation = "relu"))

In [21]: model.add(Dense(units = 128,kernel_initializer = "random_uniform",activation = "relu"))
```

- ## Adding Output Layer:

Understanding the model is very important phase to properly use it for trainingand prediction purposes.Keras provides a simple method, summary to get the full information about the modeland its layers.



```
In [22]: model.add(Dense(units = 6,kernel_initializer = "random_uniform",activation = "softmax"))
In [23]: model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248      
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0         
 2D)                                                             
                                                                 
 flatten (Flatten)           (None, 6272)              0         
                                                                 
 dense (Dense)               (None, 128)               802944    
                                                                 
 dense_1 (Dense)             (None, 128)               16512     
                                                                 
 dense_2 (Dense)             (None, 128)               16512     
                                                                 
 dense_3 (Dense)             (None, 128)               16512     
                                                                 
 dense_4 (Dense)             (None, 128)               16512     
                                                                 
 dense_5 (Dense)             (None, 6)                 774       
                                                                 
=================================================================
Total params: 879,910
Trainable params: 879,910
Non-trainable params: 0
_________________________________________________________________
```

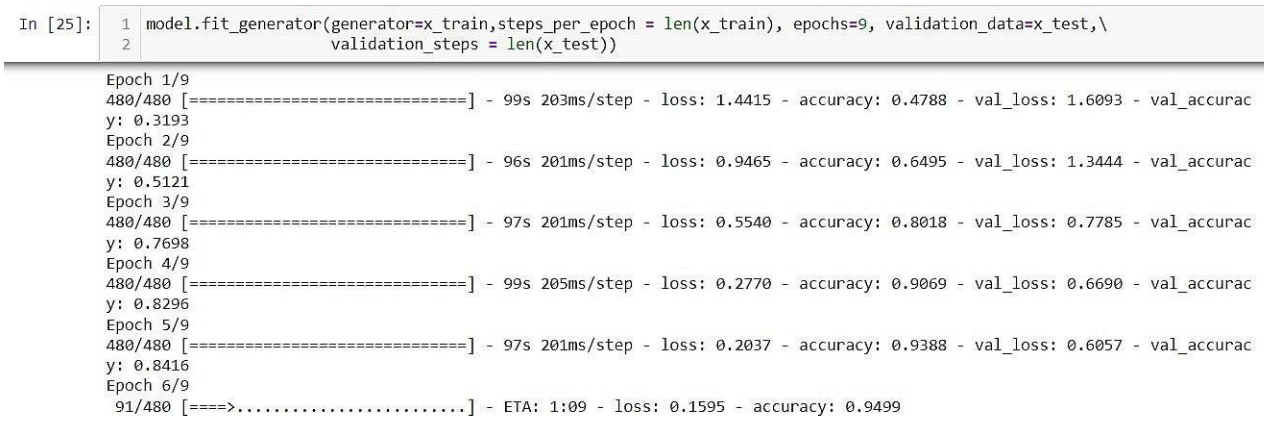
1. **Configure the Learning Process:**

1. The compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase. The loss function is used to find error or deviation in the learning process. Keras requires loss function during themodel compilation process.
2. Optimization is an importantprocess that optimizesthe input weightsby comparing the prediction and the loss function. Here we are using adam optimizer
3. Metrics is used to evaluate the performance of your model. It is similar to loss function, but not used in the training process.

```
In [24]: model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

- ## Training the model:

We will train our model with our image dataset. fit_generator functions used to train adeep learning neural network.



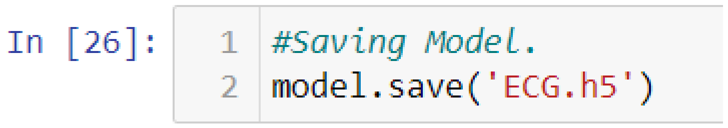
```
In [25]:  model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=9, validation_data=x_test,\
                              validation_steps = len(x_test))
```

```
Epoch 1/9
480/480 [==============================] - 99s 203ms/step - loss: 1.4415 - accuracy: 0.4788 - val_loss: 1.6093 - val_accurac
y: 0.3193
Epoch 2/9
480/480 [==============================] - 96s 201ms/step - loss: 0.9465 - accuracy: 0.6495 - val_loss: 1.3444 - val_accurac
y: 0.5121
Epoch 3/9
480/480 [==============================] - 97s 201ms/step - loss: 0.5540 - accuracy: 0.8018 - val_loss: 0.7785 - val_accurac
y: 0.7698
Epoch 4/9
480/480 [==============================] - 99s 205ms/step - loss: 0.2770 - accuracy: 0.9069 - val_loss: 0.6690 - val_accurac
y: 0.8296
Epoch 5/9
480/480 [==============================] - 97s 201ms/step - loss: 0.2037 - accuracy: 0.9388 - val_loss: 0.6057 - val_accurac
y: 0.8416
Epoch 6/9
 91/480 [====>.........................] - ETA: 1:09 - loss: 0.1595 - accuracy: 0.9499
```

- ## Saving the model:

The model is saved with .h5 extension as follows

An H5 file is a data file saved in the Hierarchical Data Format (HDF).It contains multidimensional arrays of scientific data.



```
In [26]:  #Saving Model.
           model.save('ECG.h5')
```

1. ## Testing the model:

Load necessary libraries and load the saved model using

load_modelTaking an image as inputand checking the results

Note: The target size should for the image that is should be the same as the target sizethat you have used for training.

```
In [26]: #Saving Model.
         model.save('ECG.h5')

In [28]: from tensorflow.keras.models import load_model
         from tensorflow.keras.preprocessing import image

In [29]: model=load_model('ECG.h5')

In [30]: img=image.load_img("/content/Unknown_image.png",target_size=(64,64))

In [31]: x=image.img_to_array(img)

In [32]: import numpy as np

In [33]: x=np.expand_dims(x,axis=0)

In [34]: pred = model.predict(x)
         y_pred=np.argmax(pred)
         y_pred

Out[34]: 1

In [35]: index=['left Bundle Branch block',
                'Normal',
                'Premature Atrial Contraction',
                'Premature Ventricular Contraction',
                'Right Bundle Branch Block',
                'Ventricular Fibrillation']
         result = str(index[y_pred])
         result

Out[35]: 'Normal'
```

The unknown image uploaded is:

Here the output for the uploaded result is normal.

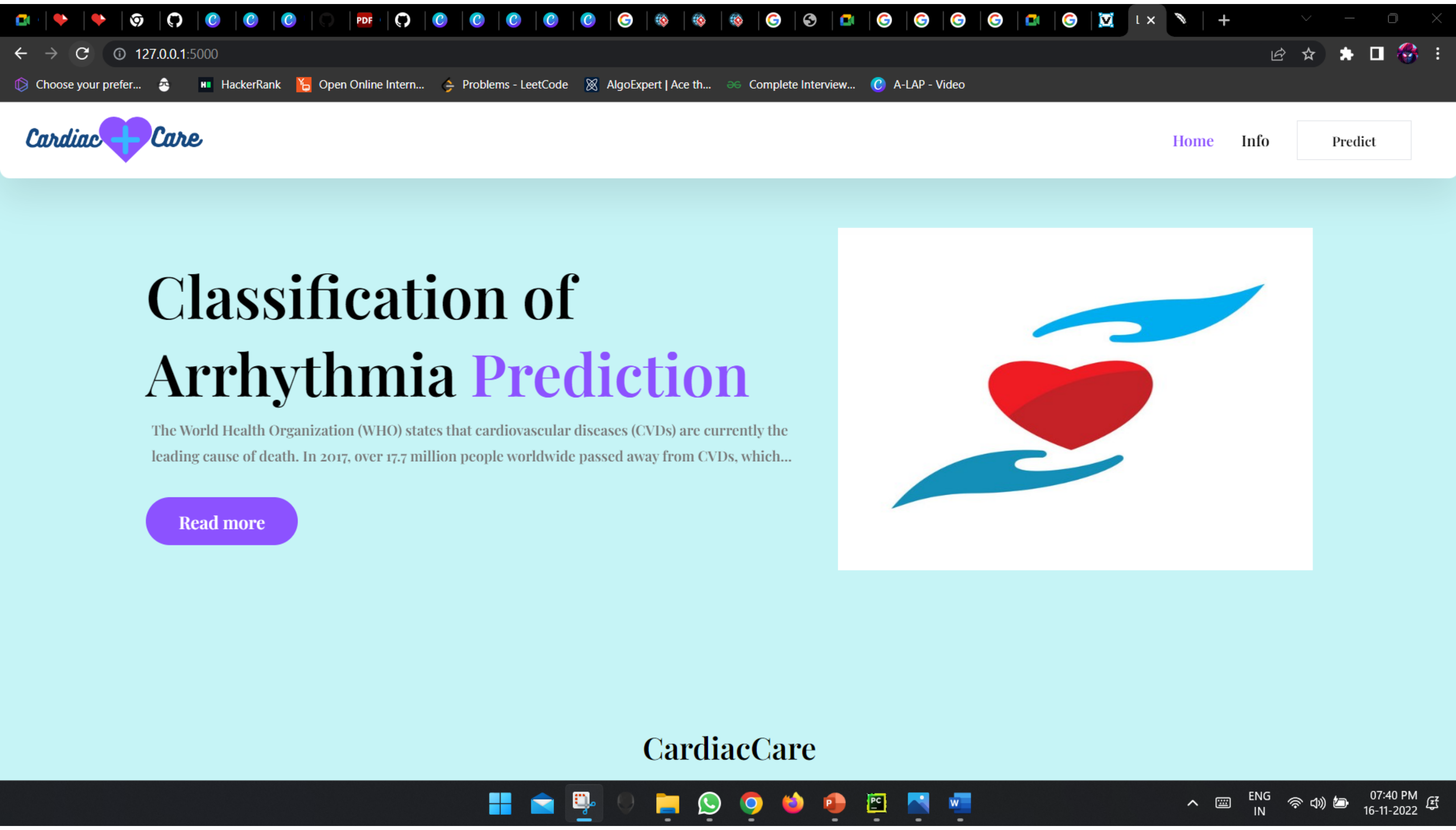
1. Application Building:

In this section, we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has uploaded an image. The uploaded image is given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- **Building HTML Pages:**
- We use HTML to create the front end part of the web page.
- Here, we created 4 html pages- home.html, predict_base.html, predict.html, information.html.

home.html displays the home page.



Home Info Predict

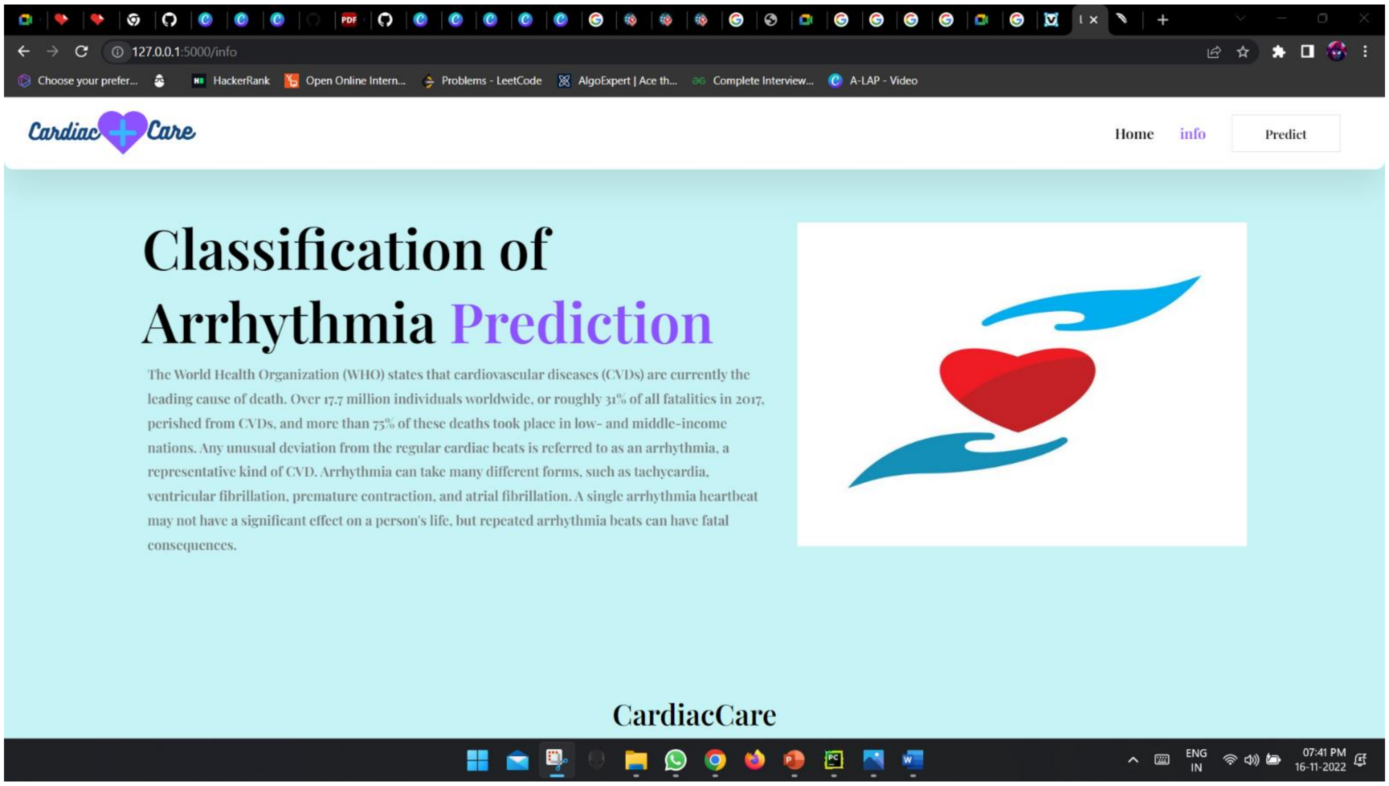
Classification of Arrhythmia Prediction

The World Health Organization (WHO) states that cardiovascular diseases (CVDs) are currently the leading cause of death. In 2017, over 17.7 million people worldwide passed away from CVDs, which...

Read more

CardiacCare

information.html displays all important details to be known about ECG.

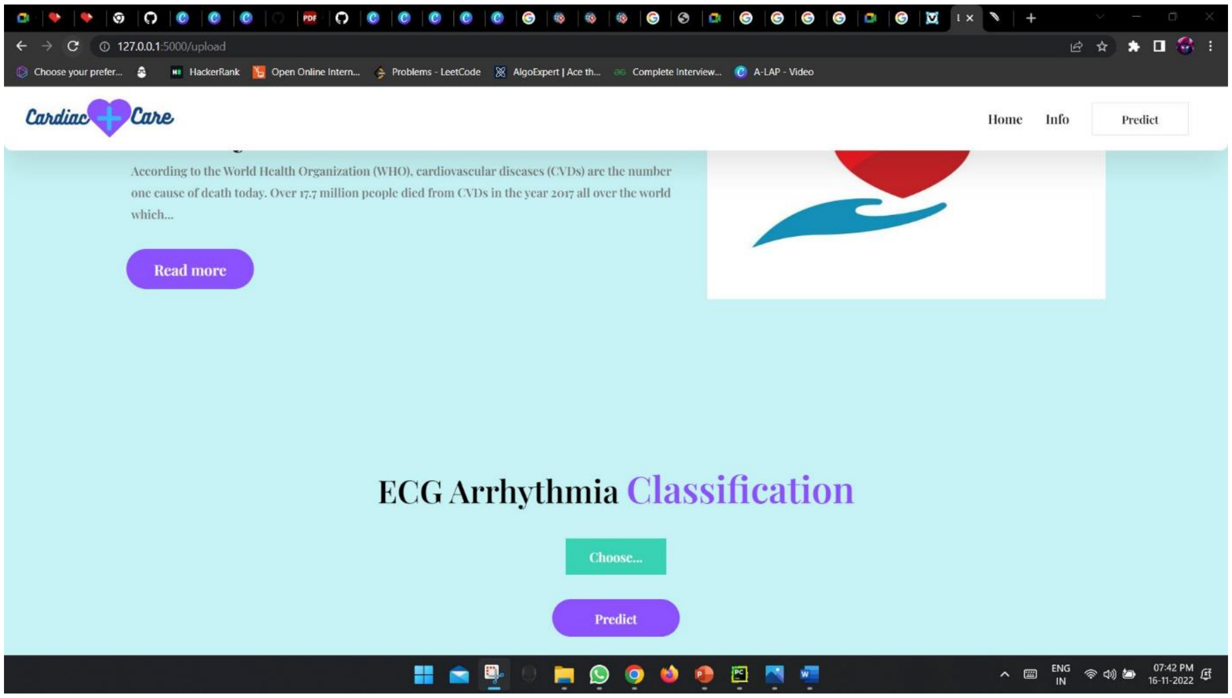


Home info Predict

Classification of Arrhythmia Prediction

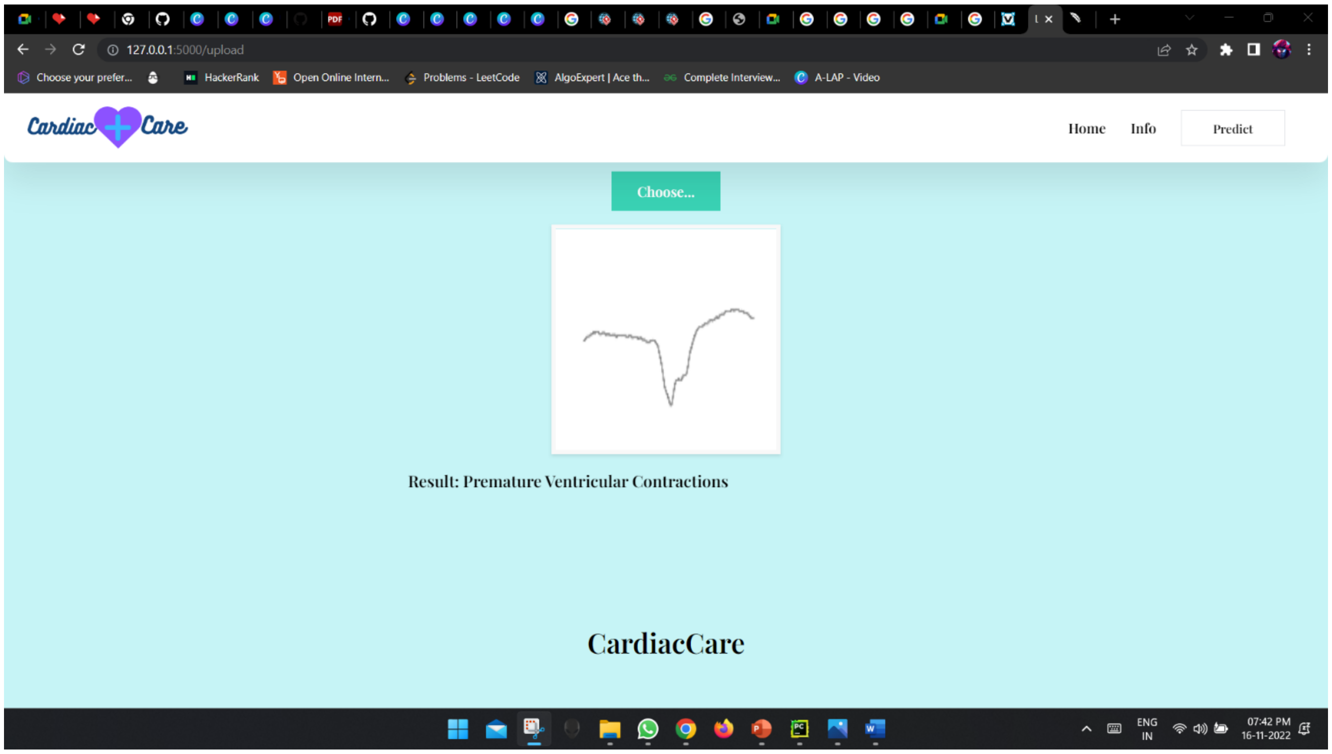
The World Health Organization (WHO) states that cardiovascular diseases (CVDs) are currently the leading cause of death. Over 17.7 million individuals worldwide, or roughly 31% of all fatalities in 2017, perished from CVDs, and more than 75% of these deaths took place in low- and middle-income nations. Any unusual deviation from the regular cardiac beats is referred to as an arrhythmia, a representative kind of CVD. Arrhythmia can take many different forms, such as tachycardia, ventricular fibrillation, premature contraction, and atrial fibrillation. A single arrhythmia heartbeat may not have a significant effect on a person's life, but repeated arrhythmia beats can have fatal consequences.

CardiacCare



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...

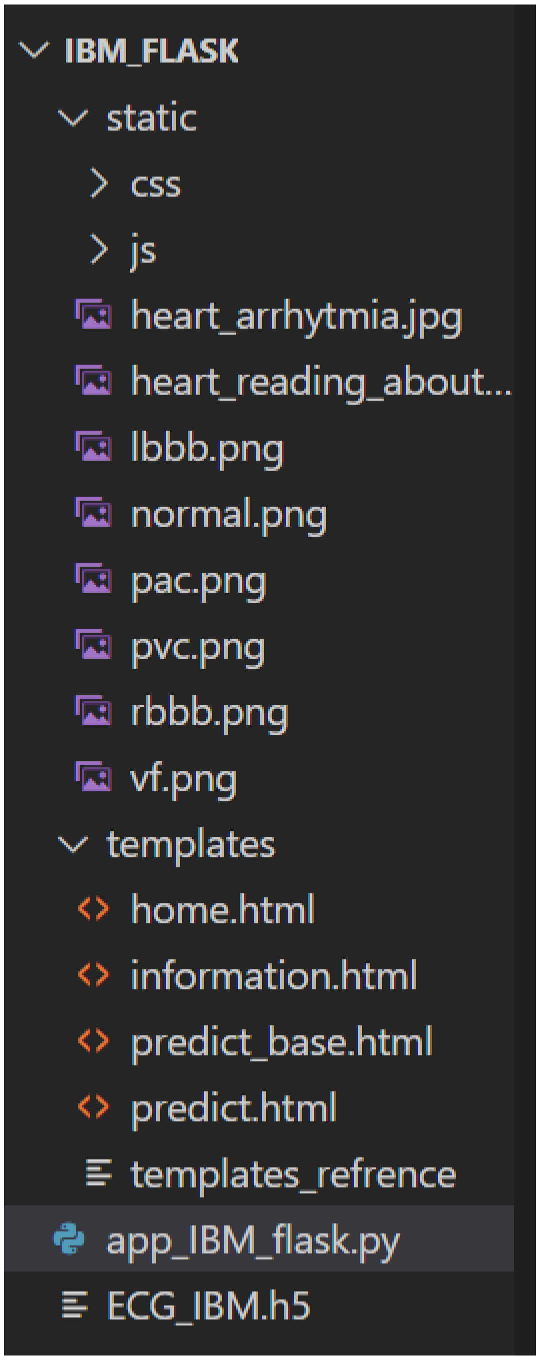
ECG Arrhythmia Classification



Result: Premature Ventricular Contractions

CardiacCare

- predict-base.html and predict.html acceptinput from the user and predicts thevalues.



```
IBM_FLASK
  static
    css
    js
    heart_arrhytmia.jpg
    heart_reading_about...
    lbbb.png
    normal.png
    pac.png
    pvc.png
    rbbb.png
    vf.png
  templates
    home.html
    information.html
    predict_base.html
    predict.html
    templates_refrence
  app_IBM_flask.py
  ECG_IBM.h5
```

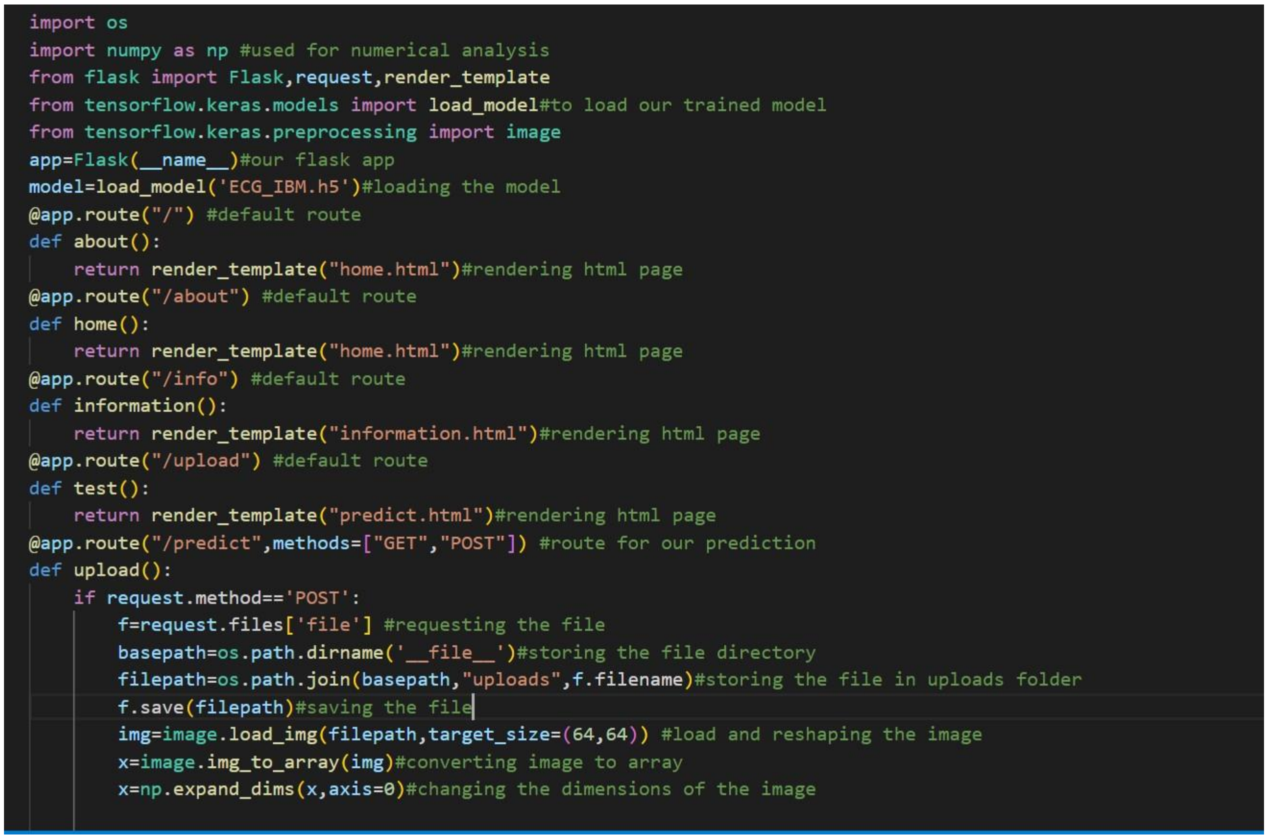
- **Building server-side script:**

We will build the ask le ‘app.py’ which is a web frameworkwritten in pythonfor server-side scripting.

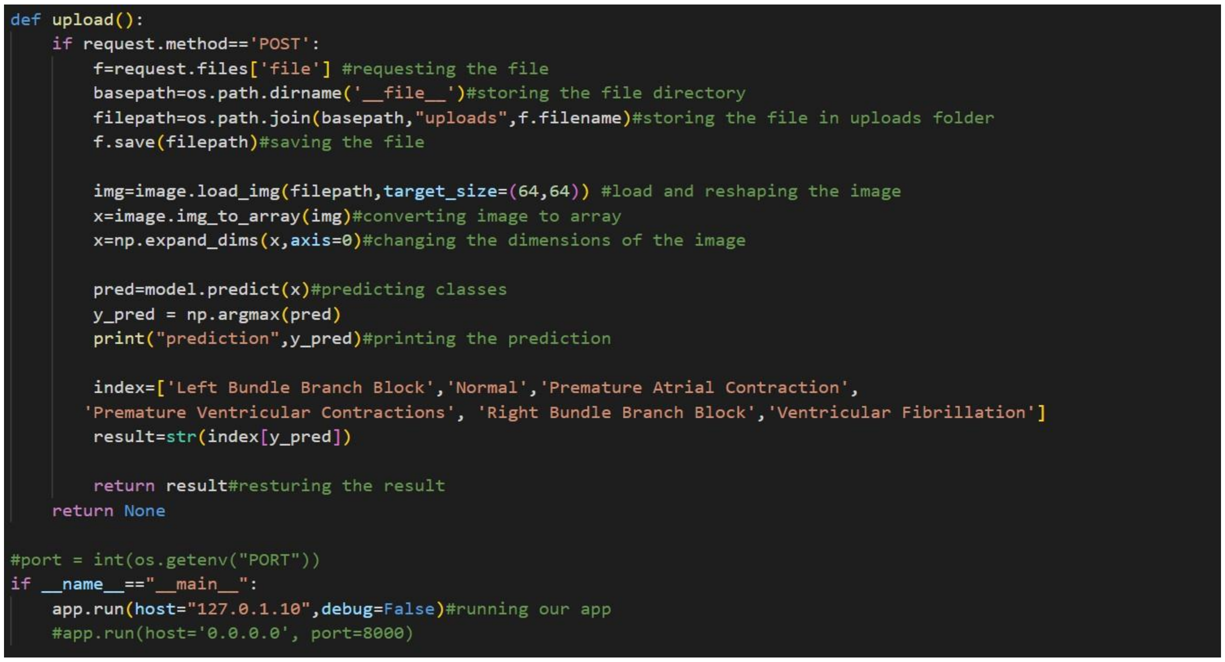
1. The app starts running when the “_name_”constructor is called in main.
2. render_template is used to return HTML le.

3. “GET” method is used to take input from the user.

4. “POST” method is used to display the output to the user.



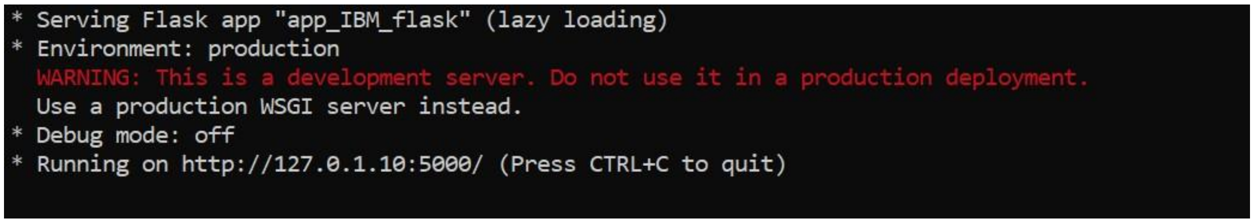
```
import os
import numpy as np #used for numerical analysis
from flask import Flask,request,render_template
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image
app=Flask(__name__)#our flask app
model=load_model('ECG_IBM.h5')#loading the model
@app.route("/") #default route
def about():
    return render_template("home.html")#rendering html page
@app.route("/about") #default route
def home():
    return render_template("home.html")#rendering html page
@app.route("/info") #default route
def information():
    return render_template("information.html")#rendering html page
@app.route("/upload") #default route
def test():
    return render_template("predict.html")#rendering html page
@app.route("/predict",methods=["GET","POST"]) #route for our prediction
def upload():
    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 array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image
```



```
def upload():
    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 array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image

        pred=model.predict(x)#predicting classes
        y_pred = np.argmax(pred)
        print("prediction",y_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[y_pred])

        return result#resturing the result
    return None

#port = int(os.getenv("PORT"))
if __name__=="__main__":
    app.run(host="127.0.1.10",debug=False)#running our app
    #app.run(host='0.0.0.0', port=8000)
```

1. **Running The App:**

Run the file as : python app_IBM_flask.py



```
* Serving Flask app "app_IBM_flask" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.1.10:5000/ (Press CTRL+C to quit)
```

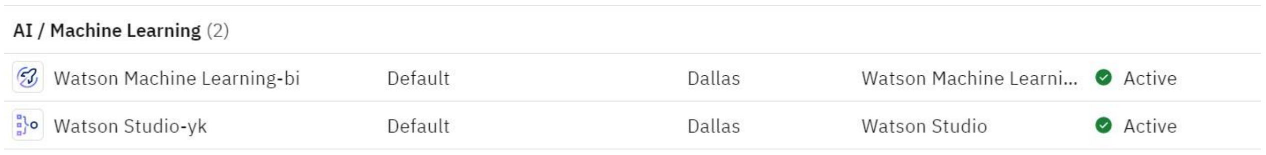
Navigate to the localhost (http://127.0.1.10:5000/)where you can view your web page.

Flow Chart & Results by training model in IBM WATSON STUDIO:

Creating IBM cloud account:

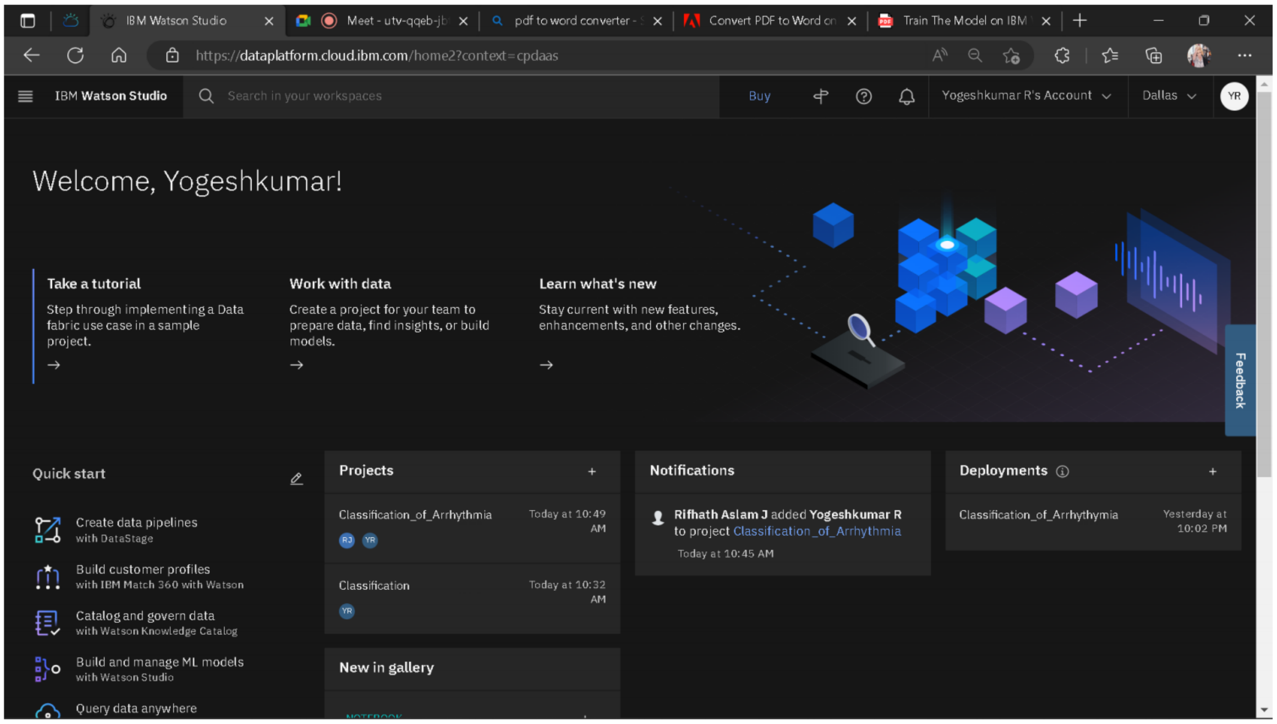
We have to create an IBM Cloud Account and should log in.

Creating Watson StudioService & MachineLearning Service:



AI / Machine Learning (2)

Watson Machine Learning-bi	Default	Dallas	Watson Machine Learni...	✓ Active
Watson Studio-yk	Default	Dallas	Watson Studio	✓ Active



Welcome, Yogeshkumar!

Take a tutorial
Step through implementing a Data fabric use case in a sample project.

Work with data
Create a project for your team to prepare data, find insights, or build models.

Learn what's new
Stay current with new features, enhancements, and other changes.

Quick start
Create data pipelines with DataStage
Build customer profiles with IBM Match 360 with Watson
Catalog and govern data with Watson Knowledge Catalog
Build and manage ML models with Watson Studio
Query data anywhere

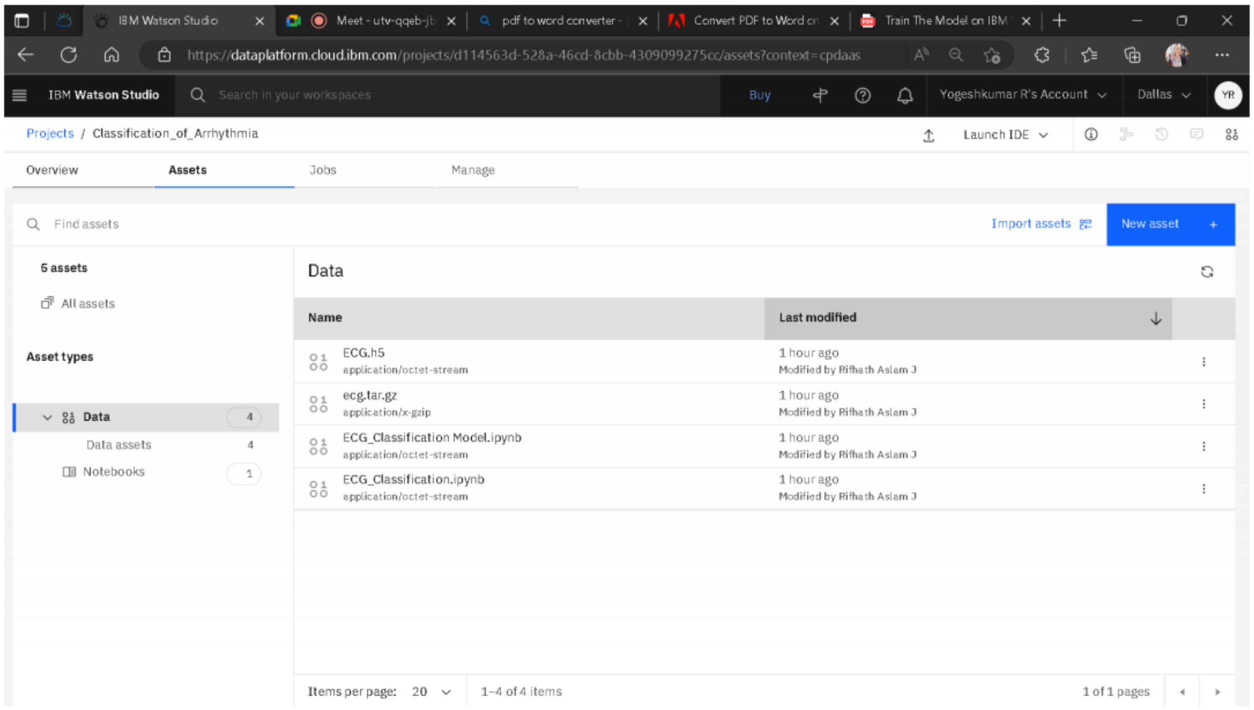
Projects
Classification_of_Arrhythmia — Today at 10:49 AM
Classification — Today at 10:32 AM

New in gallery

Notifications
Rifhath Aslam J added Yogeshkumar R to project Classification_of_Arrhythmia
Today at 10:45 AM

Deployments
Classification_of_Arrhythmia — Yesterday at 10:02 PM

- **Upload The datasetand create a jupytersource □le in the created project:**



https://dataplatform.cloud.ibm.com/projects/d114563d-528a-46cd-8cbb-4309099275cc/assets?context=cpdaas

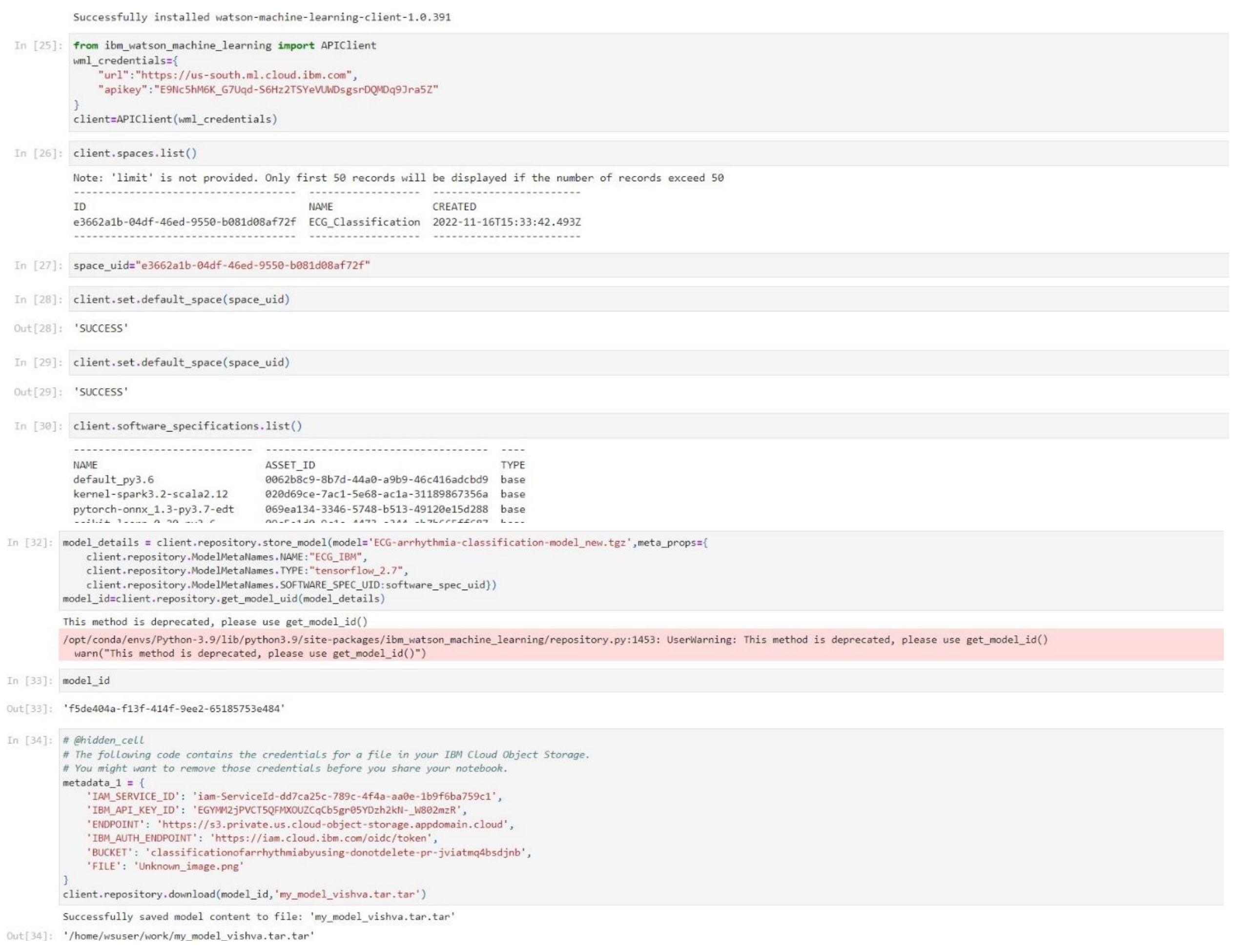
Data

Name	Last modified
ECG.h5 application/octet-stream	1 hour ago Modified by Rifhath Aslam J
ecg.tar.gz application/x-gzip	1 hour ago Modified by Rifhath Aslam J
ECG_Classification Model.ipynb application/octet-stream	1 hour ago Modified by Rifhath Aslam J
ECG_Classification.ipynb application/octet-stream	1 hour ago Modified by Rifhath Aslam J

Items per page: 20 1–4 of 4 items 1 of 1 pages

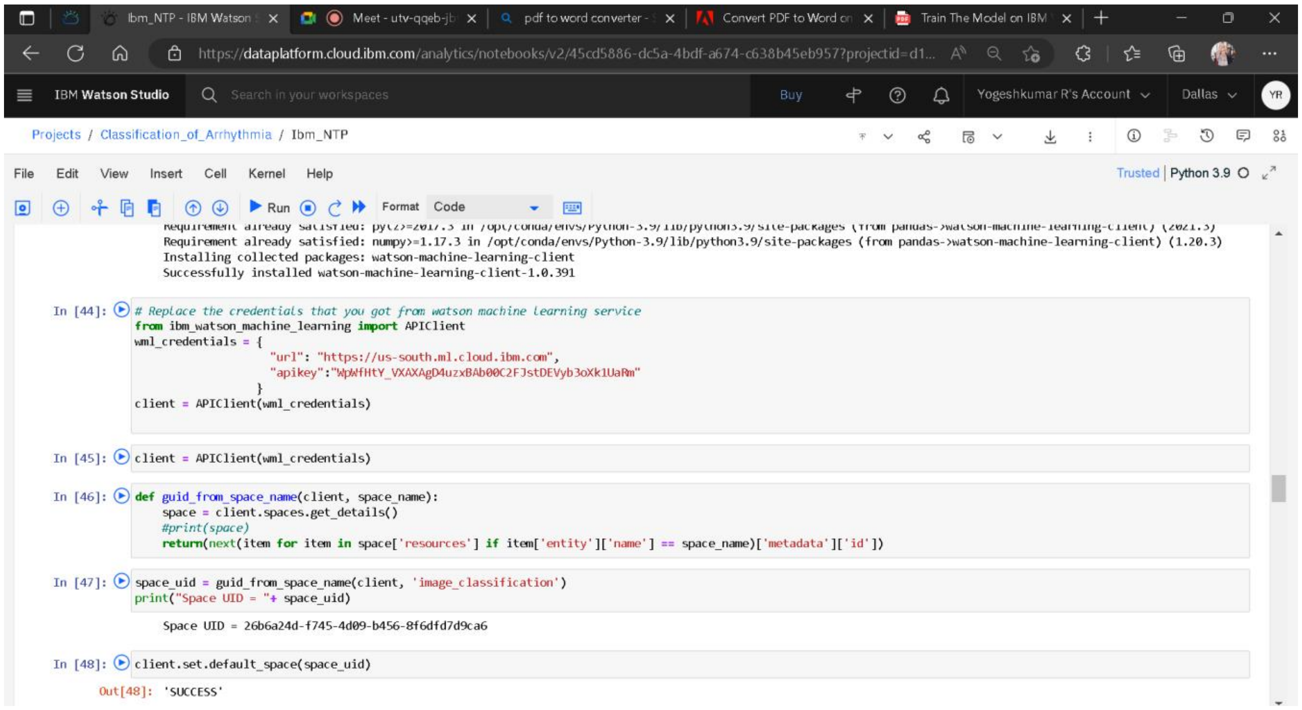
<u>Apply CNN algorithmand save the model and deploy it using API key generated:</u>

```
In [77]:  model.save('ECG_IBM.h5')

In [109]: !tar -zcvf ECG-arrhythmia-classification-model_new.tgz ECG_IBM.h5

          ECG_IBM.h5

In [110]: ls -1

          data/
          ECG-arrhythmia-classification-model_new.tgz
          ECG-classification.tgz
          ECG_IBM.h5
```



```
        Successfully installed watson-machine-learning-client-1.0.391

In [25]: from ibm_watson_machine_learning import APIClient
         wml_credentials={
             "url":"https://us-south.ml.cloud.ibm.com",
             "apikey":"E9Nc5hM6K_G7Uqd-S6Hz2TSYeVUWDsgsrDQMDq9Jra5Z"
         }
         client=APIClient(wml_credentials)

In [26]: client.spaces.list()

         Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
         ------------------------------------  ------------------  ------------------------
         ID                                    NAME                CREATED
         e3662a1b-04df-46ed-9550-b081d08af72f  ECG_Classification  2022-11-16T15:33:42.493Z
         ------------------------------------  ------------------  ------------------------

In [27]: space_uid="e3662a1b-04df-46ed-9550-b081d08af72f"

In [28]: client.set.default_space(space_uid)

Out[28]: 'SUCCESS'

In [29]: client.set.default_space(space_uid)

Out[29]: 'SUCCESS'

In [30]: client.software_specifications.list()

         -----------------------------  ------------------------------------  ----
         NAME                           ASSET_ID                              TYPE
         default_py3.6                  0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
         kernel-spark3.2-scala2.12      020d69ce-7ac1-5e68-ac1a-31189867356a  base
         pytorch-onnx_1.3-py3.7-edt     069ea134-3346-5748-b513-49120e15d288  base

In [32]: model_details = client.repository.store_model(model='ECG-arrhythmia-classification-model_new.tgz',meta_props={
             client.repository.ModelMetaNames.NAME:"ECG_IBM",
             client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
             client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid})
         model_id=client.repository.get_model_uid(model_details)

         This method is deprecated, please use get_model_id()
         /opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ibm_watson_machine_learning/repository.py:1453: UserWarning: This method is deprecated, please use get_model_id()
           warn("This method is deprecated, please use get_model_id()")

In [33]: model_id

Out[33]: 'f5de404a-f13f-414f-9ee2-65185753e484'

In [34]: # @hidden_cell
         # The following code contains the credentials for a file in your IBM Cloud Object Storage.
         # You might want to remove those credentials before you share your notebook.
         metadata_1 = {
             'IAM_SERVICE_ID': 'iam-ServiceId-dd7ca25c-789c-4f4a-aa0e-1b9f6ba759c1',
             'IBM_API_KEY_ID': 'EGYMM2jPVCT5QFMXOUZCqCb5gr05YDzh2kN-_W802mzR',
             'ENDPOINT': 'https://s3.private.us.cloud-object-storage.appdomain.cloud',
             'IBM_AUTH_ENDPOINT': 'https://iam.cloud.ibm.com/oidc/token',
             'BUCKET': 'classificationofarrhythmiabyusing-donotdelete-pr-jviatmq4bsdjnb',
             'FILE': 'Unknown_image.png'
         }
         client.repository.download(model_id,'my_model_vishva.tar.tar')

         Successfully saved model content to file: 'my_model_vishva.tar.tar'
Out[34]: '/home/wsuser/work/my_model_vishva.tar.tar'
```

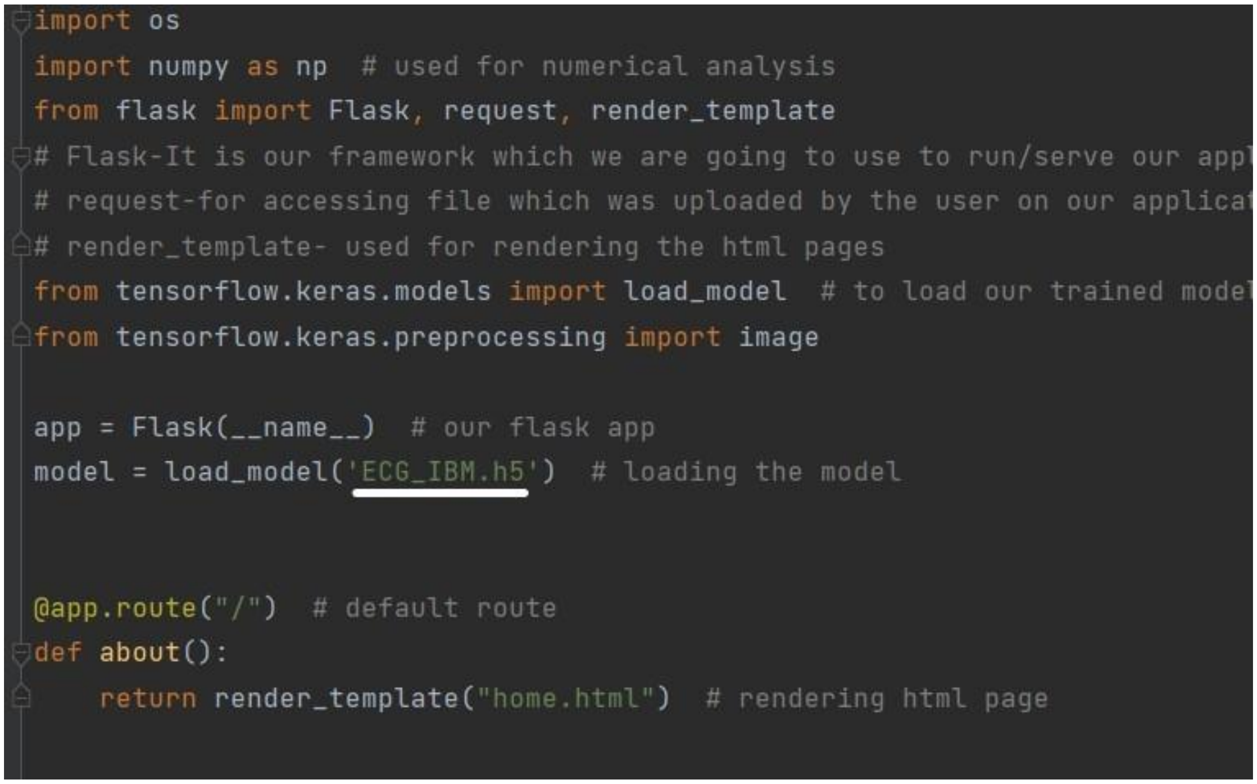
<u>For downloading the model we have to run the last part of the above codein the local jupyternotebook:</u>



```
Projects / Classification_of_Arrhythmia / Ibm_NTP

        Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)
        Installing collected packages: watson-machine-learning-client
        Successfully installed watson-machine-learning-client-1.0.391

In [44]: # Replace the credentials that you got from watson machine learning service
         from ibm_watson_machine_learning import APIClient
         wml_credentials = {
                           "url": "https://us-south.ml.cloud.ibm.com",
                           "apikey":"WpWfHtY_VXAXAgD4uzxBAb00C2FJstDEVyb3oXk1UaRm"
                          }
         client = APIClient(wml_credentials)

In [45]: client = APIClient(wml_credentials)

In [46]: def guid_from_space_name(client, space_name):
             space = client.spaces.get_details()
             #print(space)
             return(next(item for item in space['resources'] if item['entity']['name'] == space_name)['metadata']['id'])

In [47]: space_uid = guid_from_space_name(client, 'image_classification')
         print("Space UID = "+ space_uid)

             Space UID = 26b6a24d-f745-4d09-b456-8f6dfd7d9ca6

In [48]: client.set.default_space(space_uid)

   Out[48]: 'SUCCESS'
```

Now we will extractthe .h5 modelle and will do the app deploymentusing

ask as done in the previoustraining:



```
import os
import numpy as np  # used for numerical analysis
from flask import Flask, request, render_template
# Flask-It is our framework which we are going to use to run/serve our appl
# request-for accessing file which was uploaded by the user on our applicat
# render_template- used for rendering the html pages
from tensorflow.keras.models import load_model  # to load our trained model
from tensorflow.keras.preprocessing import image

app = Flask(__name__)  # our flask app
model = load_model('ECG_IBM.h5')  # loading the model


@app.route("/")  # default route
def about():
    return render_template("home.html")  # rendering html page
```

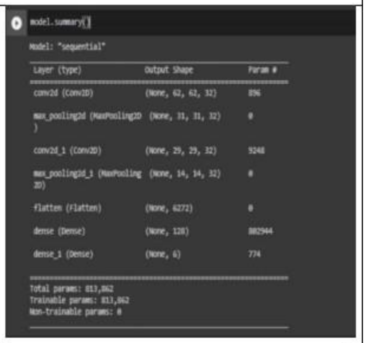
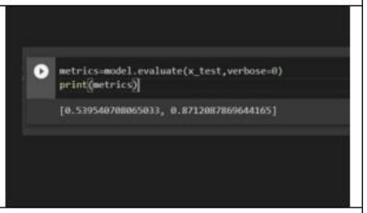
Hence we trained the model using IBM Watson.

9. TESTING:

a. PERFORMANCE TESTING:

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	
2.	Accuracy	Training Accuracy – 0.539540708065 Validation Accuracy -0.871208786964	
3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	-

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 6)                 774

=================================================================
Total params: 813,862
Trainable params: 813,862
Non-trainable params: 0
_________________________________________________________________
```

```
metrics=model.evaluate(x_test,verbose=0)
print(metrics)

[0.539540708065033, 0.8712087869644165]
```

b. USER ACCEPTANCE TESTING

This report shows the count of the bugs at each severity level, and how they were fixed.

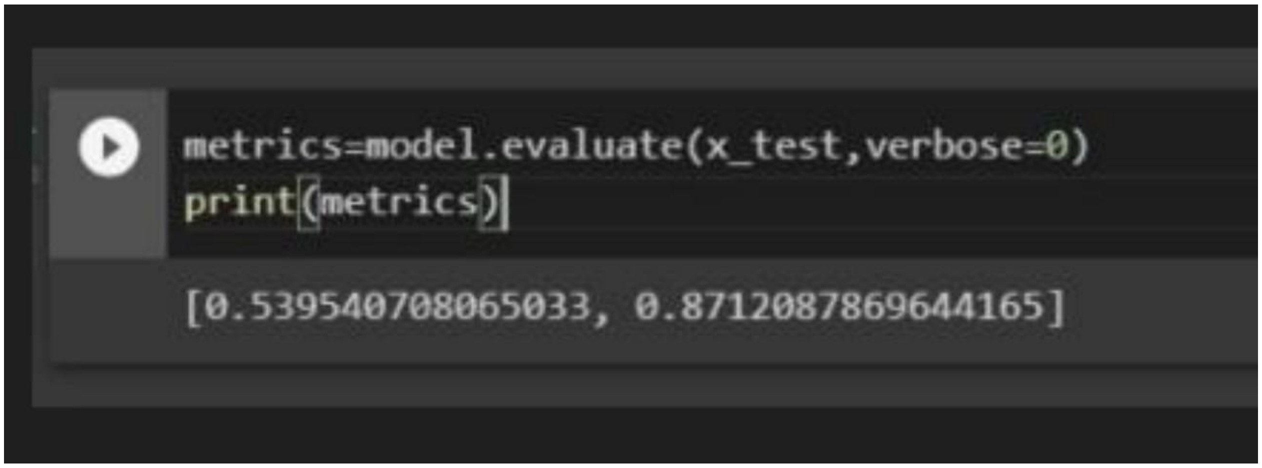
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	4	2	3	14
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	15	30
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	17	14	13	21	65

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4

10. **RESULT**

a. **PERFORMANCE METRICS:**



```
metrics=model.evaluate(x_test,verbose=0)
print(metrics)

[0.539540708065033, 0.8712087869644165]
```

11. Advantages & Disadvantages:

a. **Advantages:**

i. The proposed modelpredicts Arrhythmia in images with a highaccuracy rate of nearly 96%

1. The early detection of Arrhythmia gives better understanding of disease causes, initiates therapeutic interventions and enablesdeveloping appropriate treatments.

b. **Disadvantages:**

i. Notuseful for identifying the different stagesof Arrhythmia disease.

ii. Notuseful in monitoring motor symptoms

Applications :

iii. Itis useful for identifying the arrhythmia diseaseat an earlystage.

iv. It is usefulin detecting cardiovascular disorders

.

12. Conclusion:

1. Cardiovascular disease is a major health problem in today's world. The early diagnosis of cardiac arrhythmia highly relies onthe ECG.
2. Unfortunately, the expert level of medicalresources is rare, visually identify the ECG signalis challenging and time-consuming.
3. The advantages of the proposed CNN network have been put toevidence.
4. It is endowed with an ability to effectively process the non-filtereddataset with its potential anti-noise features. Besides that, ten- foldcross-validation is implemented in this work to furtherdemonstratethe robustness of the network.

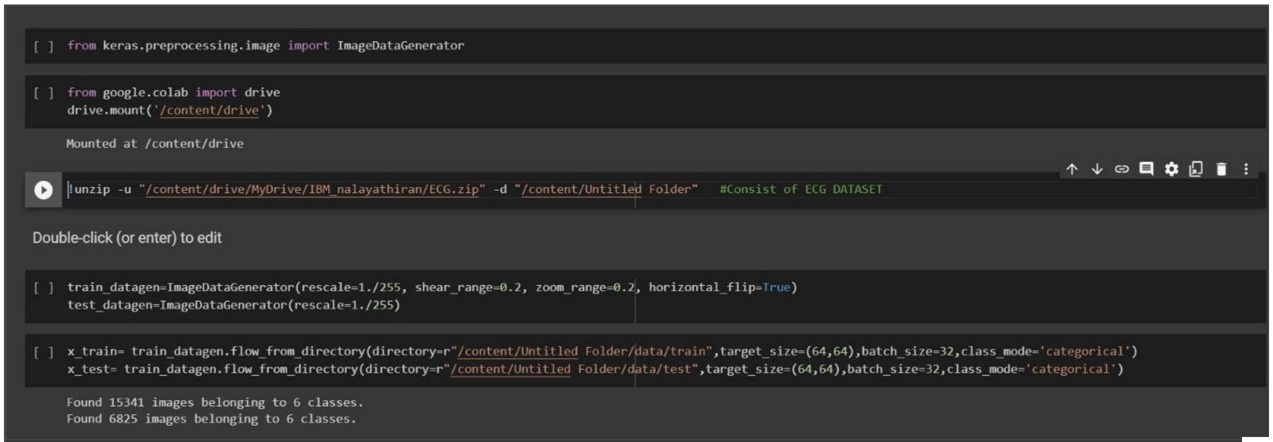
13. Future Scope:

For future work, it would be interesting to explore the use of optimization techniques to find a feasibledesign and solution. The limitation of our studyis thatwe have yet to apply any optimization techniques to optimize the model parameters and we believe that with the implementation of the optimization, it will be able to further elevate the performance of the proposedsolution to the next level.

14. APPENDIX:

SOURCE CODE:

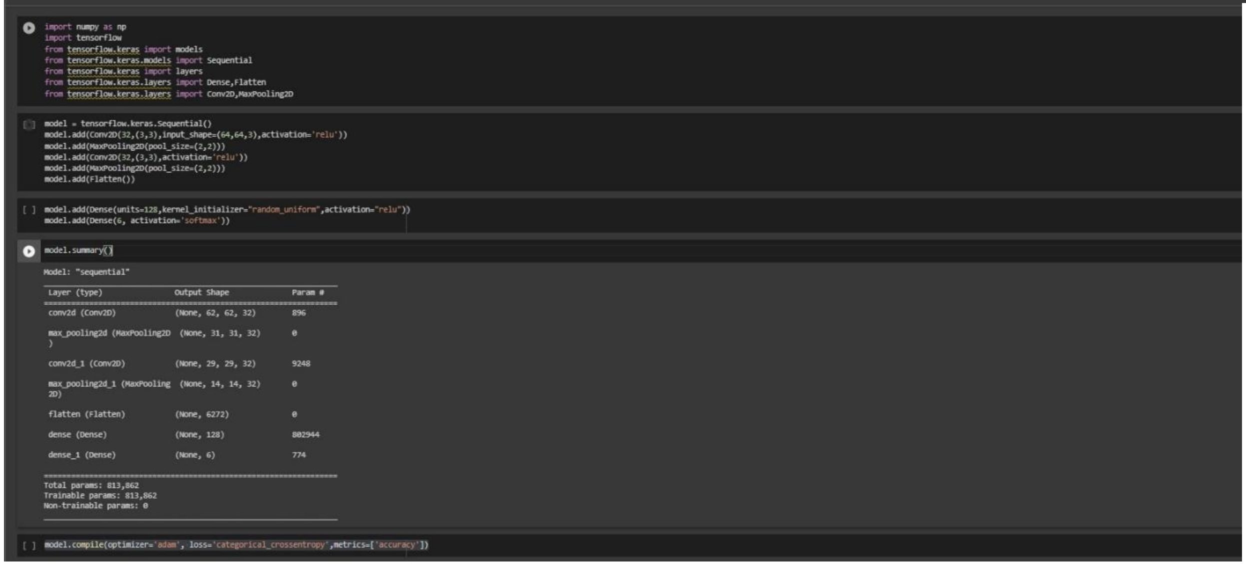
Model_generator.py



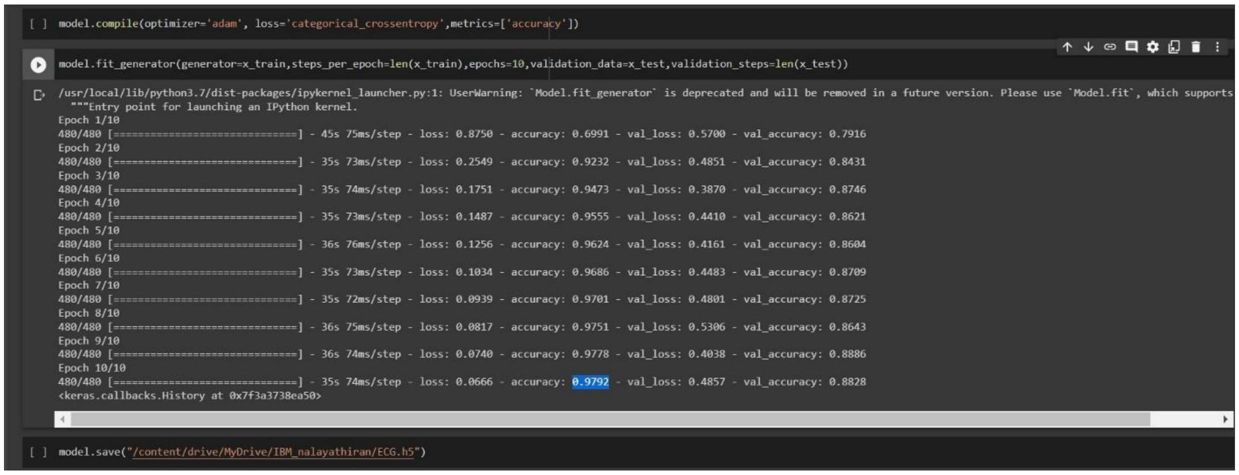
```
[ ] from keras.preprocessing.image import ImageDataGenerator

[ ] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

!unzip -u "/content/drive/MyDrive/IBM_nalayathiran/ECG.zip" -d "/content/Untitled Folder"   #Consist of ECG DATASET
```

Double-click (or enter) to edit

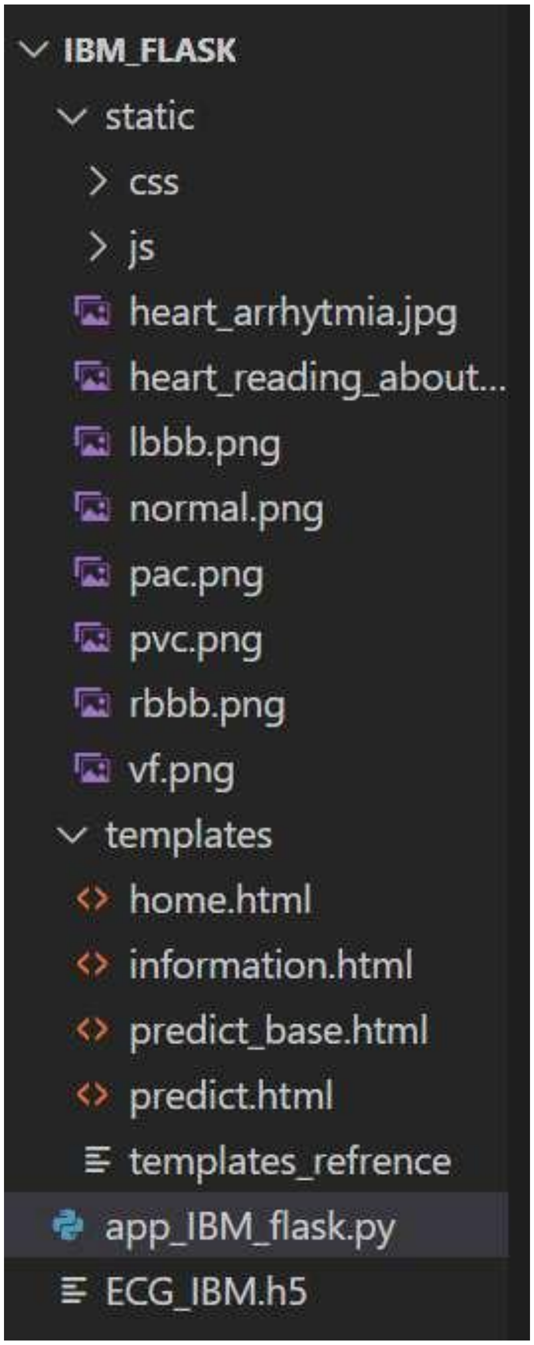
```
[ ] train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
    test_datagen=ImageDataGenerator(rescale=1./255)

[ ] x_train= train_datagen.flow_from_directory(directory=r"/content/Untitled Folder/data/train",target_size=(64,64),batch_size=32,class_mode='categorical')
    x_test= train_datagen.flow_from_directory(directory=r"/content/Untitled Folder/data/test",target_size=(64,64),batch_size=32,class_mode='categorical')

    Found 15341 images belonging to 6 classes.
    Found 6825 images belonging to 6 classes.
```



```
import numpy as np
import tensorflow
from tensorflow.keras import models
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D,MaxPooling2D

model = tensorflow.keras.Sequential()
model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

[ ] model.add(Dense(units=128,kernel_initializer="random_uniform",activation="relu"))
    model.add(Dense(6, activation='softmax'))

model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 6)                 774

=================================================================
Total params: 813,862
Trainable params: 813,862
Non-trainable params: 0
_________________________________________________________________

[ ] model.compile(optimizer='adam', loss='categorical_crossentropy',metrics=['accuracy'])
```



```
[ ] model.compile(optimizer='adam', loss='categorical_crossentropy',metrics=['accuracy'])

model.fit_generator(generator=x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports
  """Entry point for launching an IPython kernel.
Epoch 1/10
480/480 [==============================] - 45s 75ms/step - loss: 0.8750 - accuracy: 0.6991 - val_loss: 0.5700 - val_accuracy: 0.7916
Epoch 2/10
480/480 [==============================] - 35s 73ms/step - loss: 0.2549 - accuracy: 0.9232 - val_loss: 0.4851 - val_accuracy: 0.8431
Epoch 3/10
480/480 [==============================] - 35s 74ms/step - loss: 0.1751 - accuracy: 0.9473 - val_loss: 0.3870 - val_accuracy: 0.8746
Epoch 4/10
480/480 [==============================] - 35s 73ms/step - loss: 0.1487 - accuracy: 0.9555 - val_loss: 0.4410 - val_accuracy: 0.8621
Epoch 5/10
480/480 [==============================] - 36s 76ms/step - loss: 0.1256 - accuracy: 0.9624 - val_loss: 0.4161 - val_accuracy: 0.8604
Epoch 6/10
480/480 [==============================] - 35s 73ms/step - loss: 0.1034 - accuracy: 0.9686 - val_loss: 0.4483 - val_accuracy: 0.8709
Epoch 7/10
480/480 [==============================] - 35s 72ms/step - loss: 0.0939 - accuracy: 0.9701 - val_loss: 0.4801 - val_accuracy: 0.8725
Epoch 8/10
480/480 [==============================] - 36s 75ms/step - loss: 0.0817 - accuracy: 0.9751 - val_loss: 0.5306 - val_accuracy: 0.8643
Epoch 9/10
480/480 [==============================] - 36s 74ms/step - loss: 0.0740 - accuracy: 0.9778 - val_loss: 0.4038 - val_accuracy: 0.8886
Epoch 10/10
480/480 [==============================] - 35s 74ms/step - loss: 0.0666 - accuracy: 0.9792 - val_loss: 0.4857 - val_accuracy: 0.8828
<keras.callbacks.History at 0x7f3a3738ea50>

[ ] model.save("/content/drive/MyDrive/IBM_nalayathiran/ECG.h5")
```



```
IBM_FLASK
  static
    css
    js
    heart_arrhytmia.jpg
    heart_reading_about...
    lbbb.png
    normal.png
    pac.png
    pvc.png
    rbbb.png
    vf.png
  templates
    home.html
    information.html
    predict_base.html
    predict.html
    templates_refrence
  app_IBM_flask.py
  ECG_IBM.h5
```

app_IBM_flask.py

```
import os
import numpy as np  # used for numerical analysis
from flask import Flask, request, render_template
# Flask-It is our framework which we are going to use to run/serve our application.
# request-for accessing file which was uploaded by the user on our application.
# render_template- used for rendering the html pages
from tensorflow.keras.models import load_model  # to load our trained model
from tensorflow.keras.preprocessing import image

app = Flask(__name__)  # our flask app
model = load_model('ECG.h5')  # loading the model

@app.route("/") #default route
@app.route("/home") #Home page set to default page
def default():
   return render_template('index.html') #rendering index.html

@app.route("/info") #route to info page
def information():
   return render_template("info.html") #rendering info.html

@app.route("/about") #route to about us page
def about_us():
   return render_template('about.html')  #rendering about.html
```

```
@app.route("/contact") #route to contact us page
def contact_us():
    return render_template('contact.html')  #rendering contact.html


@app.route("/upload") #default route
def test():
    return render_template("predict.html")  #rendering contact.html


@app.route("/predict",methods=["GET","POST"]) #route for our prediction
def upload():
    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 array
        x = np.expand_dims(x, axis=0)  # changing the dimensions of the image


        preds = model.predict(x)  # predicting classes
        pred = np.argmax(preds, axis=1)  # predicting classes
        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]])
        return result  # resturing the result
    return None



# port = int(os.getenv("PORT"))
if __name__ == "__main__":
    app.run(debug=False)  # running our app
    # app.run(host='0.0.0.0', port=8000)
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Life Care - Heart Prediction Online</title>
<link
rel="shortcut
icon"
href="{{url_for('static',
filename='images/fevicon.png' )}}" type="image/x-icon">
<link
rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/aos/2.3.1/aos.css" />
<link
href="https://fonts.googleapis.com/css2?family=Playfair+Display:wght@600&dis
play=swap" rel="stylesheet" /><link rel="stylesheet" href="{{url_for('static',
filename='css/style.css' )}}" />
<script
src="https://kit.fontawesome.com/64d58efce2.js"
crossorigin="anonymous">
</script>
</head>
<body>
```

```
<div class="wrapper">
<!--Navigation Bar-->
<div class="nav">
<div class="logo">
<a href="/">
<img src="static\images\logo.png" style="width:190px" />
</a>
</div>
<div class="links">
<a href="/home" class="mainLink">Home</a>
<a href="/info">Info</a>
<a href="/about">About Us</a>
<a href="/contact">Contact Us</a>
<a href="/upload" class="btn1">Predict</a>
</div>
</div>
<!--Landing Page-->
<div class="landing">
<div class="landingText" data-aos="fade-up" data-aous-duration="1000">
<h1>Classification of Arrhythmia
<span style="color: #e0501b; font-size: 4vw">Prediction</span>
</h1>
<h3>
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...
```

```
</h3>
<div class="btn2"><a href="/info">Read more</a>
</div>
</div>
<div
class="landingImage"
data-aos="fade-down"
data-aous
duration="2000">
<img src="static/images/banner_img.jpg" alt="bannerImg" style="width:
500px; height:360px" />
</div>
</div>
<!--Service Section-->
<div class="about">
<div class="aboutText" data-aos="fade-up" data-aous-duration="1000">
<h1 style="margin: 20px;">
Our Patients Are at Centre
<span style="color: #2f8be0; font-size: 3vw">of Every We Do</span>
</h1><div class="image-container">
<img src="/static/images/connsultPationt.png" alt="consultPationt"
style="width:400px; margin:100px 0px 0px 90px;"></img>
</div>
</div>
<div class="aboutList" data-aos="fade-left" data-aous-duration="1000">
<ol>
<li>
```

```
<span>01</span>
<p>99.8% accurate result.</p>
</li>
<li>
<span>02</span>
<p>No need to go hospital.</p>
</li>
<li>
<span>03</span>
<p>No need to login</p>
</li>
<li>
<span>04</span>
<p>24/7 Support.</p>
</li>
</ol>
</div>
</div><!--Info Section-->
<div class="infoSection">
<div class="infoHeader" data-aos="fade-up" data-aous-duration="1000">
<h1>
We Analyse Youe Health states <br /><span style="color: #e0501b">In
Order to Top Service.</span>
</h1>
</div>
<div class="infoCards">
<div class="card one" data-aos="fade-up" data-aous-duration="1000">
```

```
<img src="static/images/banner_1.svg" class="cardoneImg" alt="" data

aos="fade-up"

data-aous-duration="1100" />

<div class="cardbgone"></div>

<div class="cardContent">

<h2>Health State</h2>

<p>

Easy to know Health state

</p>

<a href="/">

<div class="cardBtn">

<img src="static/images/next.png" alt="" class="cardIcon" />

</div>

</a>

</div>

</div>

<div class="card two" data-aos="fade-up" data-aous-duration="1300"><img
src="static/images/banner_1.svg" class="cardtwoImg" alt="" data

aos="fade-up"

data-aous-duration="1200" />

<div class="cardbgtwo"></div>

<div class="cardContent">

<h2>User Friendly</h2>

<p>

Easy for people to use, prediction

</p>

<a href="/">
```

```
<div class="cardBtn">
<img src="static/images/next.png" alt="" class="cardIcon" />
</div>
</a>
</div>
</div>
<div class="card three" data-aos="fade-up" data-aous-duration="1600">
<img src="static/images/banner_1.svg" class="cardthreeImg" alt=""
data-aos="fade-up"
data-aous-duration="1000" />
<div class="cardbgthree"></div>
<div class="cardContent">
<h2>Classification of Arrhythmia</h2>
<p>
Prediction Classification of Arrhythmia
</p>
<a href="/upload"><div class="cardBtn">
<img src="static/images/next.png" alt="" class="cardIcon" />
</div>
</a>
</div>
</div>
</div>
</div>
<!--Banner And Footer-->
<div class="banner">
<div
```

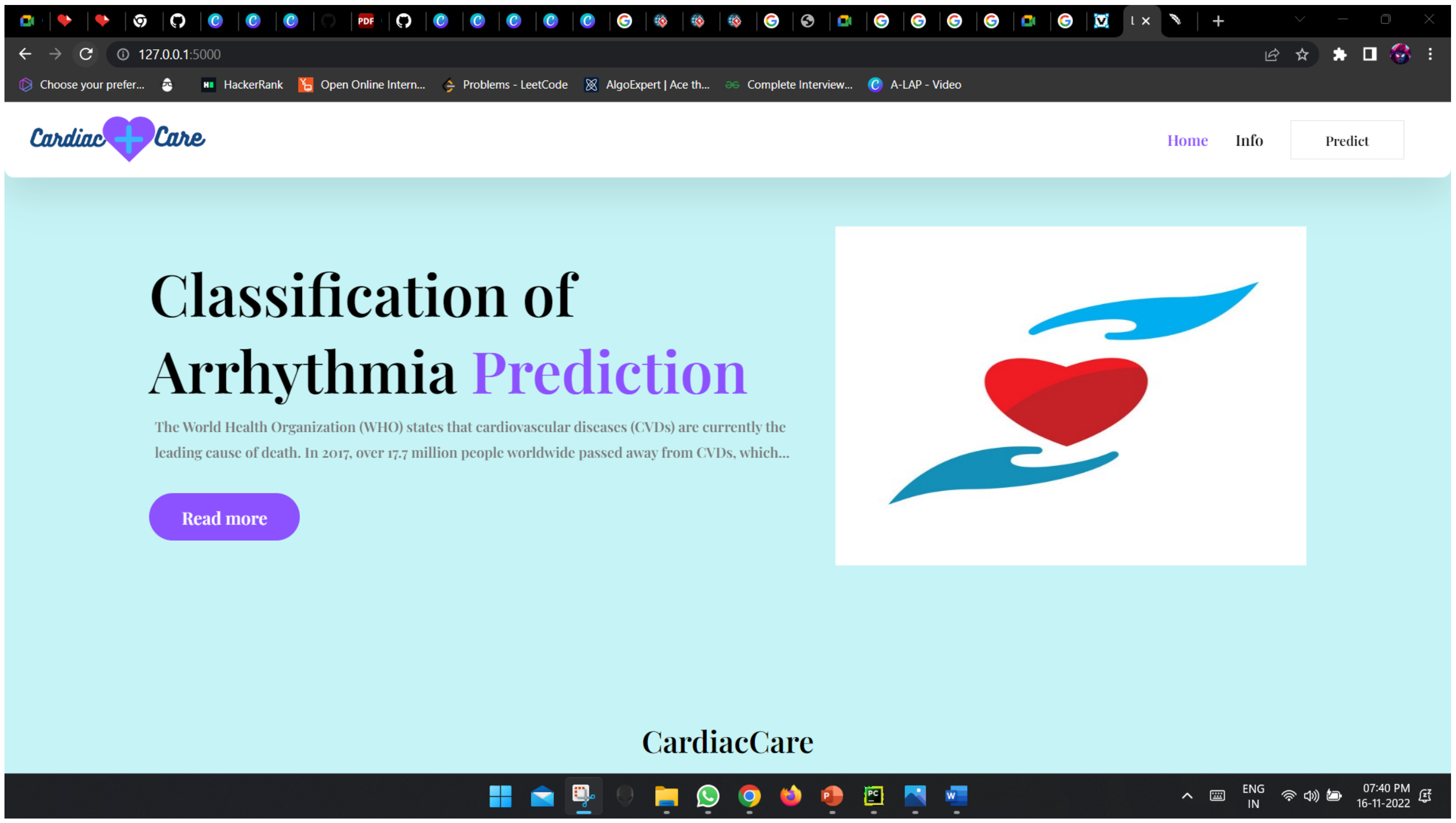
```
class="bannerText"
data-aos="fade-right"
data-aous
duration="1000">
<h1>
Download the LifeCare App Today <br /><span style="font-size: 1.6vw;
font-weight: normal"
class="bannerInnerText">Stay Updated and get all your medical needs
taken care of!</span>
</h1>
<a href="/"><img src="static/images/AndroidPNG.png" alt="" /></a>
<a href="/"><img src="static/images/iosPNG.png" alt="" /></a>
</div>
<div class="bannerImg" data-aos="fade-up" data-aous-duration="1000">
<img src="static/images/app.png" alt="" />
</div>
</div><div class="footer">
<h1>LifeCare</h1>
<div class="footerlinks">
<a href="/home" class="mainLink">Home</a>
<a href="/info">Info</a>
<a href="/about">About Us</a>
<a href="/contact">Contact Us</a>
</div>
</div>
</div>
<scriptsrc="https://cdnjs.cloudflare.com/ajax/libs/aos/2.3.1/aos.js"></script>
```

<script>

& Project Demo Link :-Git Link :-

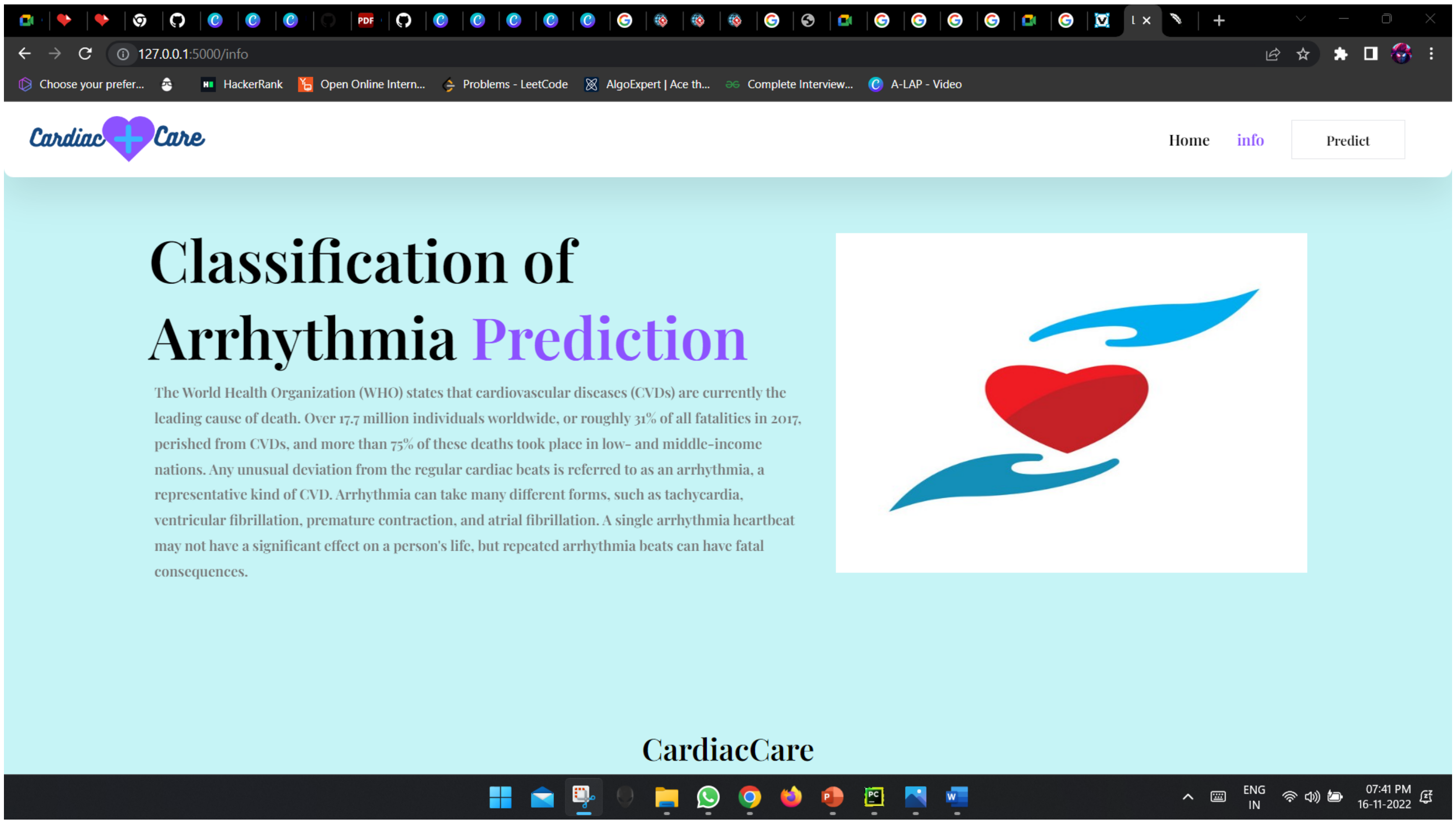
https://github.com/IBM-EPBL/IBM-Project-17429-1659670230

Project Demo :-



Classification of Arrhythmia Prediction

The World Health Organization (WHO) states that cardiovascular diseases (CVDs) are currently the leading cause of death. In 2017, over 17.7 million people worldwide passed away from CVDs, which...



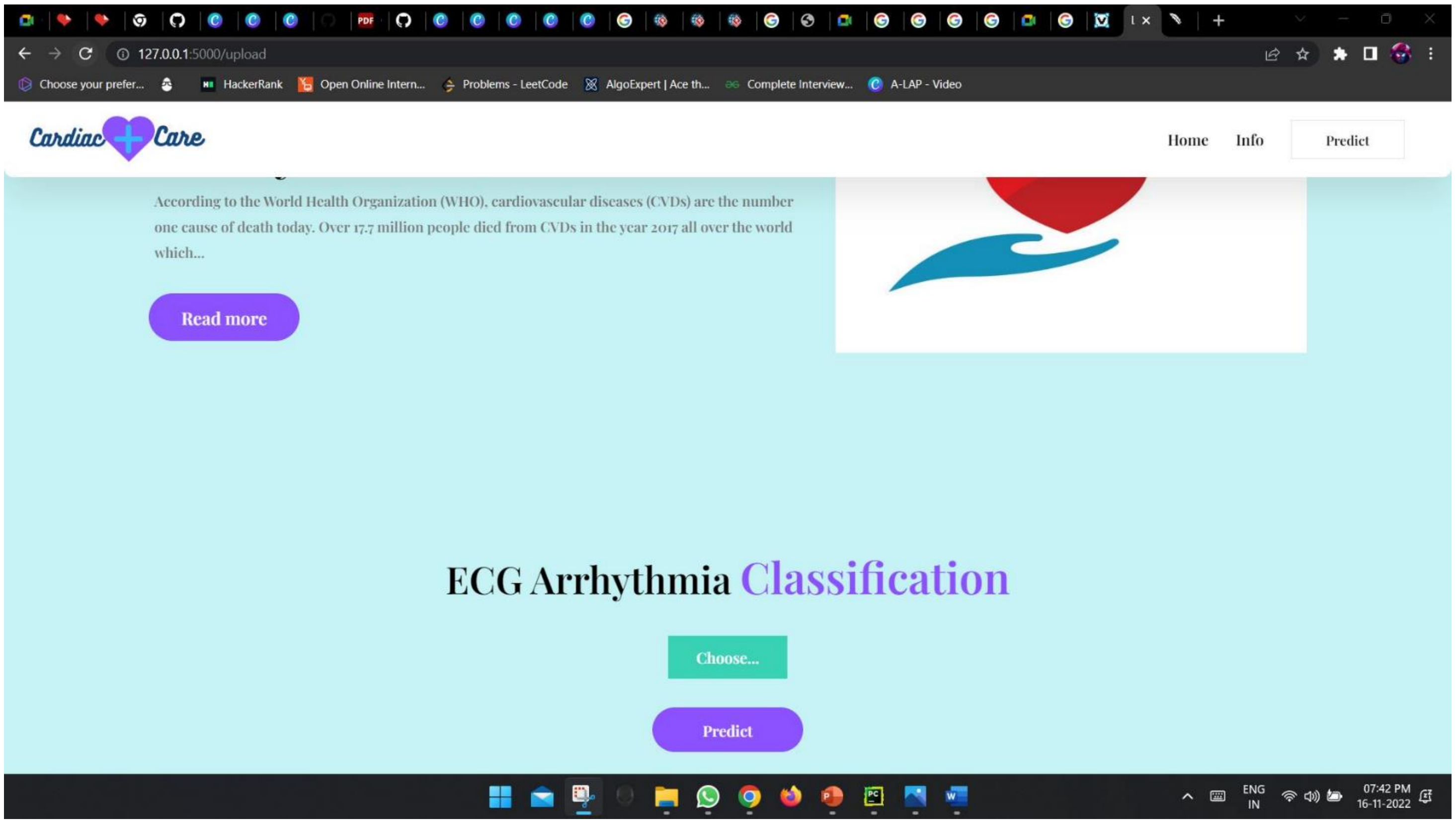
127.0.0.1:5000/info

Home info Predict

Classification of Arrhythmia Prediction

The World Health Organization (WHO) states that cardiovascular diseases (CVDs) are currently the leading cause of death. Over 17.7 million individuals worldwide, or roughly 31% of all fatalities in 2017, perished from CVDs, and more than 75% of these deaths took place in low- and middle-income nations. Any unusual deviation from the regular cardiac beats is referred to as an arrhythmia, a representative kind of CVD. Arrhythmia can take many different forms, such as tachycardia, ventricular fibrillation, premature contraction, and atrial fibrillation. A single arrhythmia heartbeat may not have a significant effect on a person's life, but repeated arrhythmia beats can have fatal consequences.

CardiacCare



127.0.0.1:5000/upload

Home Info Predict

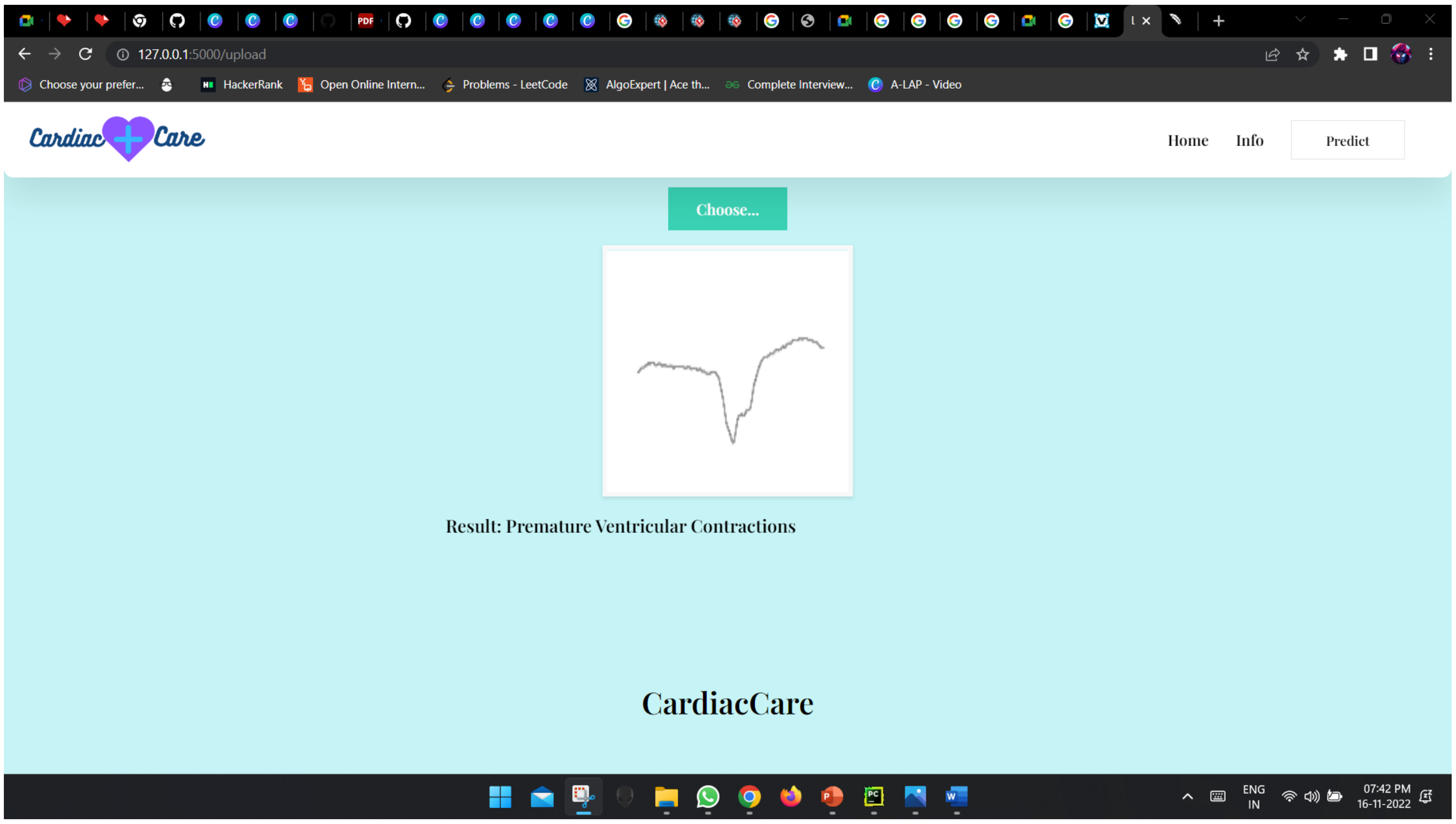
According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the number one cause of death today. Over 17.7 million people died from CVDs in the year 2017 all over the world which...

Read more

ECG Arrhythmia Classification

Choose...

Predict



Result: Premature Ventricular Contractions

CardiacCare

<u>THE END</u>