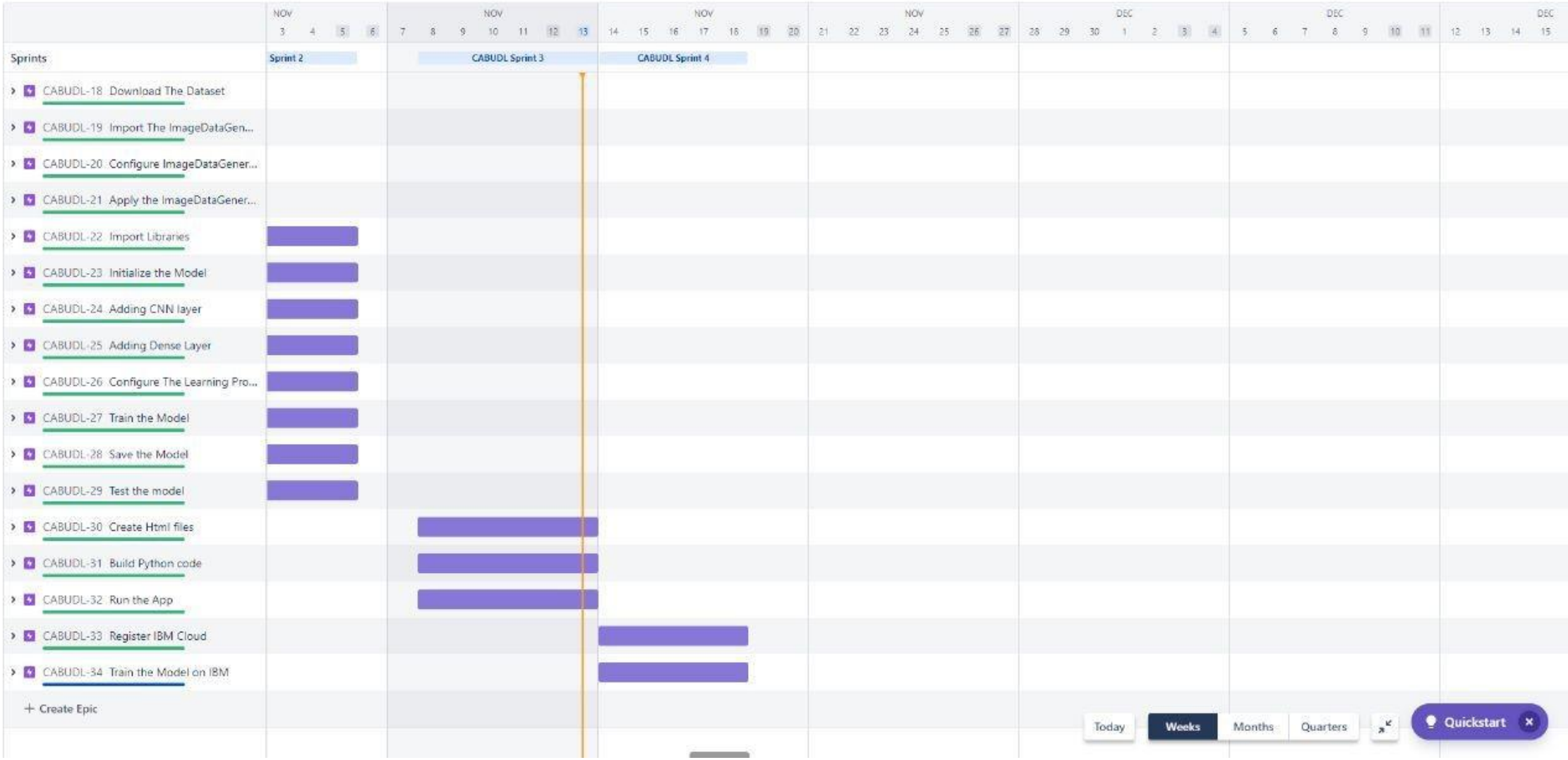

NOV
3 4 5 6
NOV
7 8 9 10 11 12 13
NOV
14 15 16 17 18 19 20
NOV
21 22 23 24 25 26 27
DEC
28 29 30 1 2 3 4
DEC
5 6 7 8 9 10 11
DEC
12 13 14 15
Sprints
Sprint 2
CABUDL Sprint 3
CABUDL Sprint 4
CABUDL-18 Download The Dataset
CABUDL-19 Import The ImageDataGen...
CABUDL-20 Configure ImageDataGener...
CABUDL-21 Apply the ImageDataGener...
CABUDL-22 Import Libraries
CABUDL-23 Initialize the Model
CABUDL-24 Adding CNN layer
CABUDL-25 Adding Dense Layer
CABUDL-26 Configure The Learning Pro...
CABUDL-27 Train the Model
CABUDL-28 Save the Model
CABUDL-29 Test the model
CABUDL-30 Create Html files
CABUDL-31 Build Python code
CABUDL-32 Run the App
CABUDL-33 Register IBM Cloud
CABUDL-34 Train the Model on IBM
+ Create Epic
Today
Weeks
Months
Quarters
Quickstart

o html 9 style

c lus- "mbar" >

Sans To ECG-

64 b

size-"S8" size--15" 5u.s

Applicatim Arrhyttwia

classifiGtim

71 v py+ont to Health

72 of d—th today. 17.' died CVUs in the 2017

11 the *ich is atmt of all deaths, and wer 7SX of deaths in

'Iddle countries _ is a 0+ COD that to

ny change fræ the rhythu. There typæ of arrhytlwia

(ons)

atrial fibrillation,

77 single arrhytlmia have tm1 that displays and stetu

beats in fatal (ECG) is a f of irregular t.•t rhyttm signals 1

igni+tcant task In field 0+

(uh-RITH-æ-uh) is en rhyttm errhytYias) the that the s beats

eut Qy like a or may signs ssmptm_

it •s fw a to hm fast or rate. the tmr•t rate Hith dwing

lifestyle help

dm•t faulty signal

ump CALF vrML

arrhythmie præd#es. implanted deviæs to control eliminate fut. SIN heartbuts. A

height—sap

dim' is t 0+ all deaths, and middle countries. is a

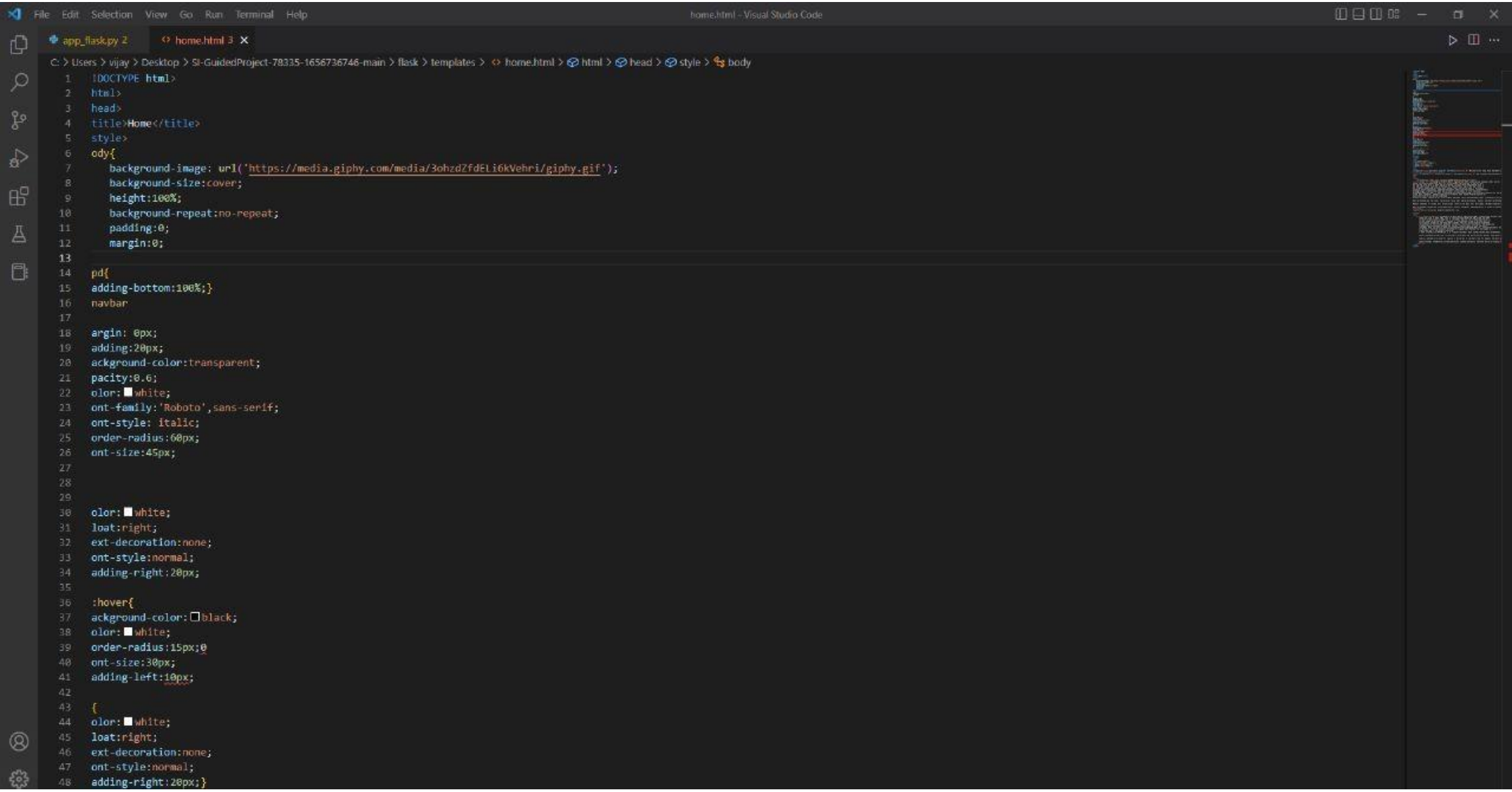
o cam of death t«lay. Over 17. J million r change rhythm. are atrial single may not a

arrhytt.ia of of significant in field of ECG ts a tm1 that displays rhytt. statw

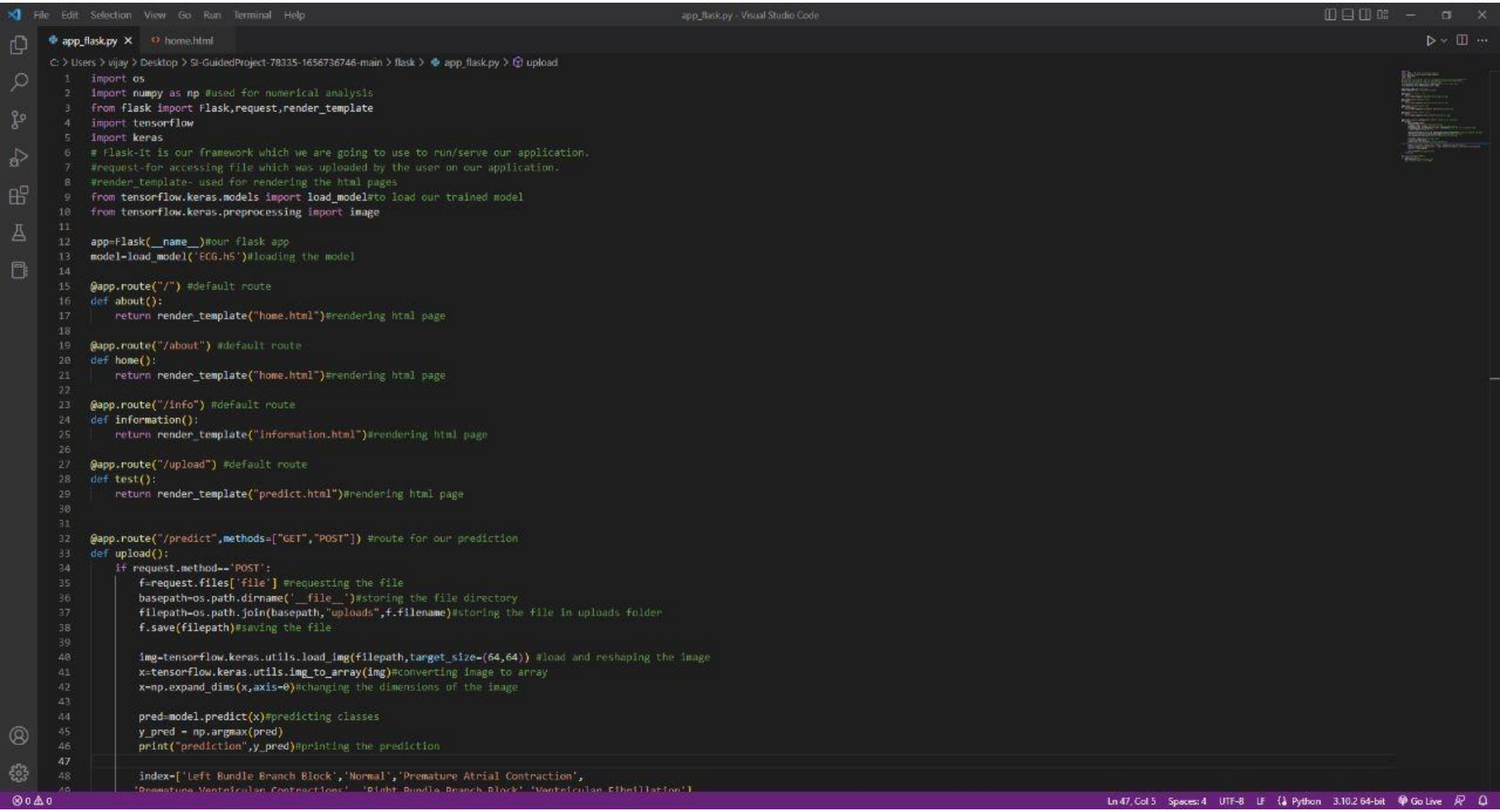
cardiology. {9 A h&rt is an rhrttw

CVOs in the 2817 0+ in of that to ty;æs of u•rhythmia +1br111atton, on life. a non cardiovascular dis (cws) are the 1 t signals

p


File Edit Selection View Go Run Terminal Help
home.html - Visual Studio Code
app_flask.py 2
home.html 3
C: > Users > vijay > Desktop > SI-GuidedProject-78335-1656736746-main > flask > templates > home.html > html > head > style > body
!DOCTYPE html>
html>
head>
title>Home</title>
style>
ody{
background-image: url('https://media.giphy.com/media/3ohzdZfdELi6kVehri/giphy.gif');
background-size:cover;
height:100%;
background-repeat:no-repeat;
padding:0;
margin:0;
pd{
adding-bottom:100%;}
navbar
argin: 0px;
adding:20px;
ackground-color:transparent;
pacity:0.6;
olor: white;
ont-family:'Roboto',sans-serif;
ont-style: italic;
order-radius:60px;
ont-size:45px;
olor: white;
loat:right;
ext-decoration:none;
ont-style:normal;
adding-right:20px;
:hover{
ackground-color: black;
olor: white;
order-radius:15px;0
ont-size:30px;
adding-left:10px;
{
olor: white;
loat:right;
ext-decoration:none;
ont-style:normal;
adding-right:20px;}

p


File Edit Selection View Go Run Terminal Help
app_flask.py - Visual Studio Code
app_flask.py
home.html
C: > Users > vijay > Desktop > SI-GuidedProject-78335-1656736746-main > flask > app_flask.py > upload
import os
import numpy as np #used for numerical analysis
from flask import Flask,request,render_template
import tensorflow
import keras
# 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
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=tensorflow.keras.utils.load_img(filepath,target_size=(64,64)) #load and reshaping the image
x=tensorflow.keras.utils.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',
Ln 47, Col 5 Spaces: 4 UTF-8 LF Python 3.10.2 64-bit Go Live