Codebase Documentation

Files and Folder Structure

→ hdad

- → __pycache__ (Contains Compiled Python File)
- → static
  - → css
    - →dashboard.css
    - →docid.css
    - →index.css
    - →login.css
    - →main.css
    - → patient_details.css
    - → signin.css
  - → images
    - →favicon.svg
  - → template.zip
  - → templateDiagnostics.zip
- → templates
    - →access_denied.html
    - →account_deleted.html
    - →dashboard.html
    - →docid.html
    - →error.html
    - →index.html
    - →login.html
    - →patient_details.html
    - →profile.html

→register.html

→ __init__.py

→ CHF_Detection_Model.pkl

→ create_graphs.py

→ Diagnostics-Counter.txt

→ forms.py

→ Heart_disease_CatBoost_Model.pkl

→ HF_Classification_Model.pkl

→ models.py

→ routes.py

→ .gitignore

→ app.py

→ Procfile

→ pyenv.cfg

→ requirements

Rest other folders are for python virtual environment, python configurations,etc.

How to Run the app

Open this folder in terminal and type the following command in the terminal

Windows : -

➔ Scripts/activate

Ubuntu : -

➔ Scripts\activate

Then install the requirements if the requirements are not already installed using the command below :-

➔ pip install -r requirements.txt

Now you can start the app using

➔ python app.py

Files and their Functions

- .gitinore
  Contains all the names of files and folders that are to be ignored by git.
- Procfile
  A necessary file for deploying the app on Heroku. It contains the information about the type of app.
- Requirements.txt
  It contains list of all python packages required for the app to run along with their versions.
- Diagnostics-Counter.txt
  It contains a integer number that tell that how many diagnostics has been performed on the app.
- CHF_Detection_Model.pkl
  Pickle file for the Congestive Heart Failure Detection model.
- Heart_disease_CatBoost_Model.pkl
  Pickle file for the Heart disease prediction model.
- HF_Classification_Model.pkl
  Pickle file for the Heart failure classification model.
- app.py
  This python files starts the app and import app and db from the hdad package.
- __init__.py
  This python file import flask and some other packages required for the application to work.
  The packages are initialized in this file.
  It also contains environment variables and passwords.
- models.py
  It contains structure of all the tables in the database.
- forms.py
  It contains all the form that are on the website the forms are created by using flask_wtf from Flaskforms. It contains all the conditions for the forms. Input Fields are being validated by wtforms.validators. All the inputs are first checked and then passed to the app.
- create_graph.py
  This file creates the html code for the graph that is displays the statistics in the patient details page. It takes patient_id, feature_name , feature_data , date_data , and two optional variables that are single feature which is true by default if the second feature is something other than date , and the last variable feature_name2, if the second feature is something other than date then the data for second features are passed instead of date_data.

- routes.py
  It contains all the routes(paths of websites) and the functions of those routes
  - Route : /
    Methods : GET , POST

Return : if current user is authenticated then it redirects to dashboard else it opens index.html

- Route : /login
  Methods : GET , POST
  Return : if current user is authenticated then it redirects to dashboard else if the form(login form) is validated then it match values of username and hashed password with the values in database if the values match then it redirects to dashboard otherwise it prompts “Try Again” else it opens login.html.

- Route : /register
  Methods : GET , POST
  Return : if current user is authenticated then it redirects to dashboard else if request Method is POST then , if the form (register form) is validated it creates new user and redirects to the docid.html else it prompts error message. If there is any problem accessing the db or sending email then it prompts a error message.
  If the request method is GET then it opens register.html
- Route : /profile
  Methods : GET ,POST
  Return : It can only accessed by an authenticated user.
  If the request Method is Post then It validates the inout and matches with “Delete” the account of user is deleted and redirected to account_deleted.html else it prompts an error message.
  If the requesrt method is GET then it opens profile.html
- Route : /dashboard
  Method : GET , POST
  Return : It can only accessed by an authenticated user.
  If the request method is Get then it opens dashboard.html
  Else if request method is POST, then it checks what form is validated and performs according to the that function and shows the result in dashboard page, else if any error happens then it prompts error message
- Route:/logout
  Method : GET , POST
  Return : It can only accessed by an authenticated user.
  If the request method logs out the user and redirects to Homepage.
- Route:/<pid>
  Method : GET , POST
  Return : It can only accessed by an authenticated user.
  If the request method is Get then it opens patient_details.html
  You can send mails using this route.
  Else if the request method is POST then it checks what form is validated and performs according to the that function and shows the result in Patient details page, else if any error happens then it prompts error message.