

<b>TEAM ID</b>	PNT2022TMID48226
<b>PROJECT NAME</b>	Statistical Machine Learning Approaches to Liver Disease Prediction

**Team Leader:** P.M.Priyadharshini

**Team Member:** N.N.Deepika

**Team Member:** T.G.Yamini

**Team Member:** S.S.Megha

## Run the App

### Run the application from anaconda prompt

- Open new anaconda prompt from the start menu . Navigate to the folder where your python script is.
- Now type “python app.py” command
- It will show the local host where your app is running on **http://127.0.0.1.8000/**
- Copy that local host URL and open that URL in the browser. It does navigate me to where you can view your web page.
- Enter the values, click on the predict button and see the result/prediction on the web page.

```

app=Flask(__name__) # our flask app

@app.route('/') # rendering the html template
def home():
    return render_template('home.html')
@app.route('/predict') # rendering the html template
def index() :
    return render_template("index.html")

@app.route('/data_predict', methods=['POST']) # route for our prediction
def predict():
    age = request.form['age'] # requesting for age data
    gender = request.form['gender'] # requesting for gender data
    tb = request.form['tb'] # requesting for Total_Bilirubin data
    db = request.form['db'] # requesting for Direct_Bilirubin data
    ap = request.form['ap'] # requesting for Alkaline_Phosphotase data
    aa1 = request.form['aa1'] # requesting for ALanine_Aminotransferase data
    aa2 = request.form['aa2'] # requesting for Aspartate_Aminotransferase data
    tp = request.form['tp'] # requesting for Total_Protiens data
    a = request.form['a'] # requesting for Albumin data
    agr = request.form['agr'] # requesting for Albumin_and_Globulin_Ratio data

    # covertng data into float format
    data = [[float(age), float(gender), float(tb), float(db), float(ap), float(aa1), float(aa2), float(tp),

    # Loading model which we saved
    model = pickle.load(open('liver_analysis.pkl', 'rb'))

    prediction= model.predict(data)[0]
    if (prediction == 1):
        return render_template('noChance.html', prediction='You have a liver desease problem, You must and :
    else:
        return render_template('chance.html', prediction='You dont have a liver desease problem')

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

```

---

```

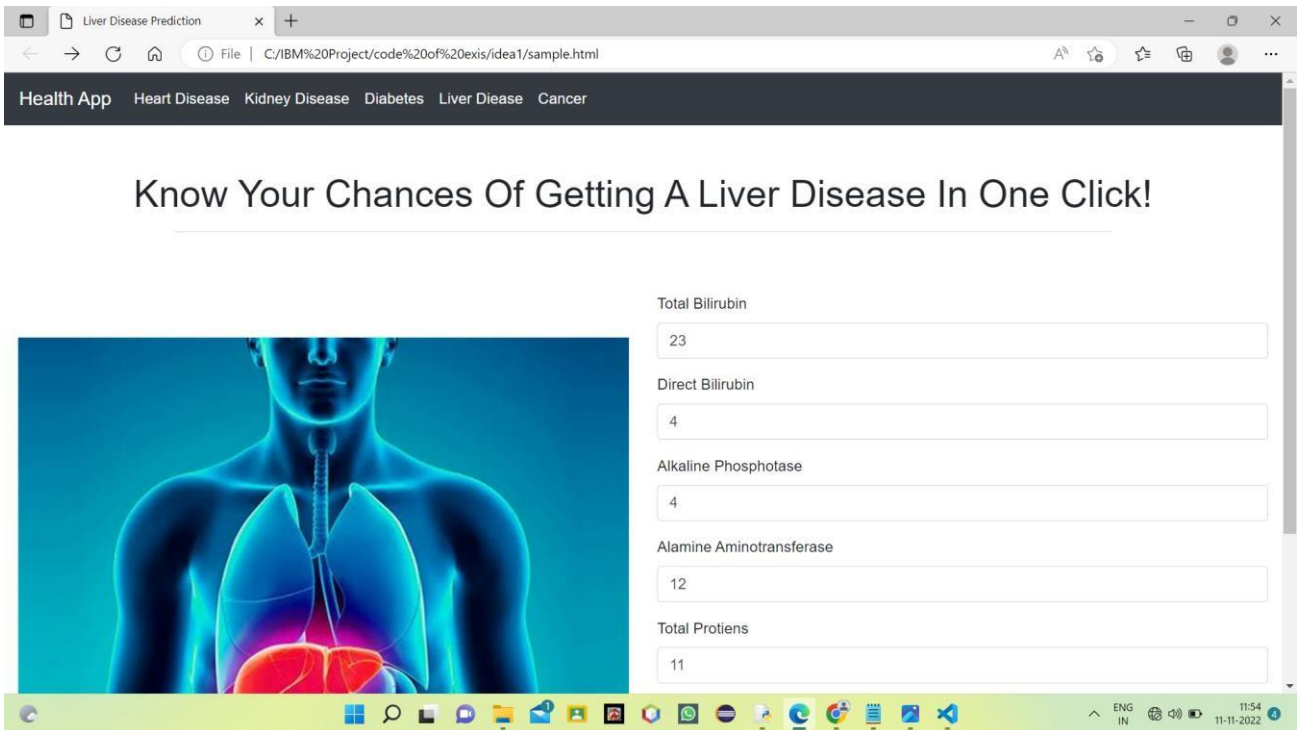
* Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off

* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```


## Showcasing the output on UI

Home page is displayed when home button is clicked. Predict page is displayed when predict button is clicked. In predict page, enter input values to predict the liver disease or not. Finally, the prediction for the given input features is shown.



Health App Heart Disease Kidney Disease Diabetes Liver Disease Cancer

### Know Your Chances Of Getting A Liver Disease In One Click!



Total Bilirubin  
23

Direct Bilirubin  
4


Alkaline Phosphatase  
4

Alamine Aminotransferase  
12

Total Protiens  
11

Liver Disease Prediction

File | C:/IBM%20Project/code%20of%20exis/idea1/sample.html



Total Bilirubin

Direct Bilirubin

Alkaline Phosphatase

Alamine Aminotransferase

Total Protiens

Albumin

Albumin and Globulin Ratio

Predict

Windows taskbar: 11:54 11-11-2022

Output:

Build Python Code2.pdf | Create an HTML File2.pdf | create am HTML file 3.pdf | localhost:8087/predict

localhost:8087/predict

Gmail | YouTube | Maps

**Sorry, you have chances of getting the disease. Please consult the doctor immediately**

[home](#)

Windows taskbar: 70°F Raining now | 12:43 11-11-2022

