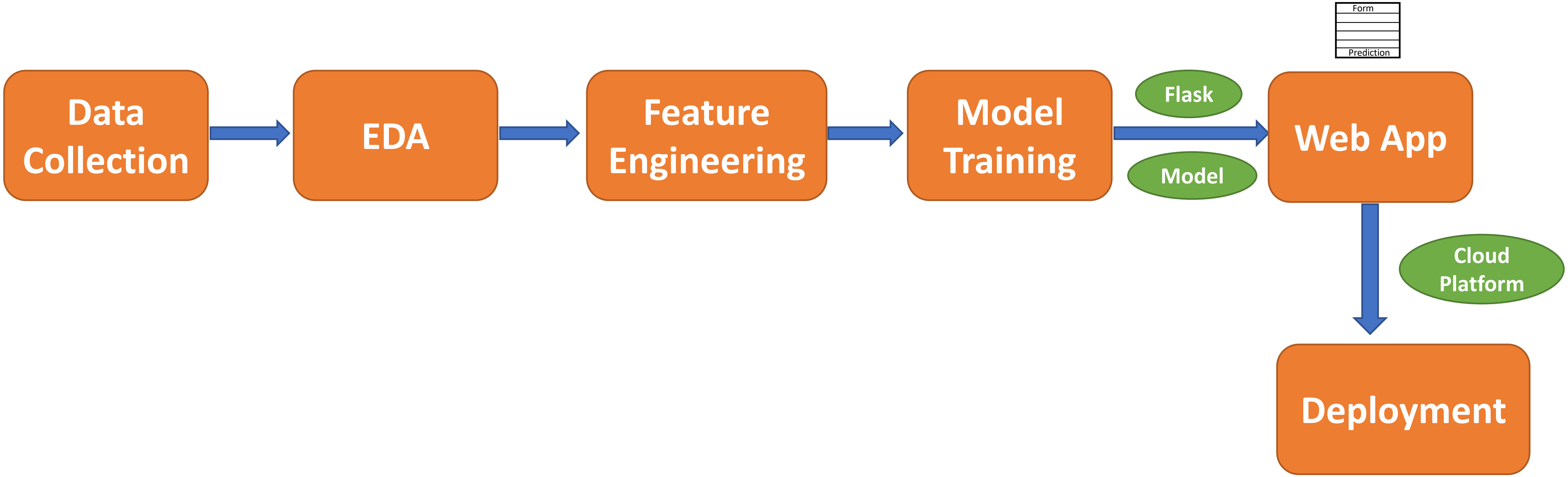
Data Science Masters

Machine Learning End-to-End Project

Logistic Regression Project

Project Life Cycle



Folder and File

- Step 1.

  * Create **dataset** folder for *data*

  * Create **models** folder for *.pkl models*

  * Create **notebooks** folder for *.ipynb files*

- Step 2.

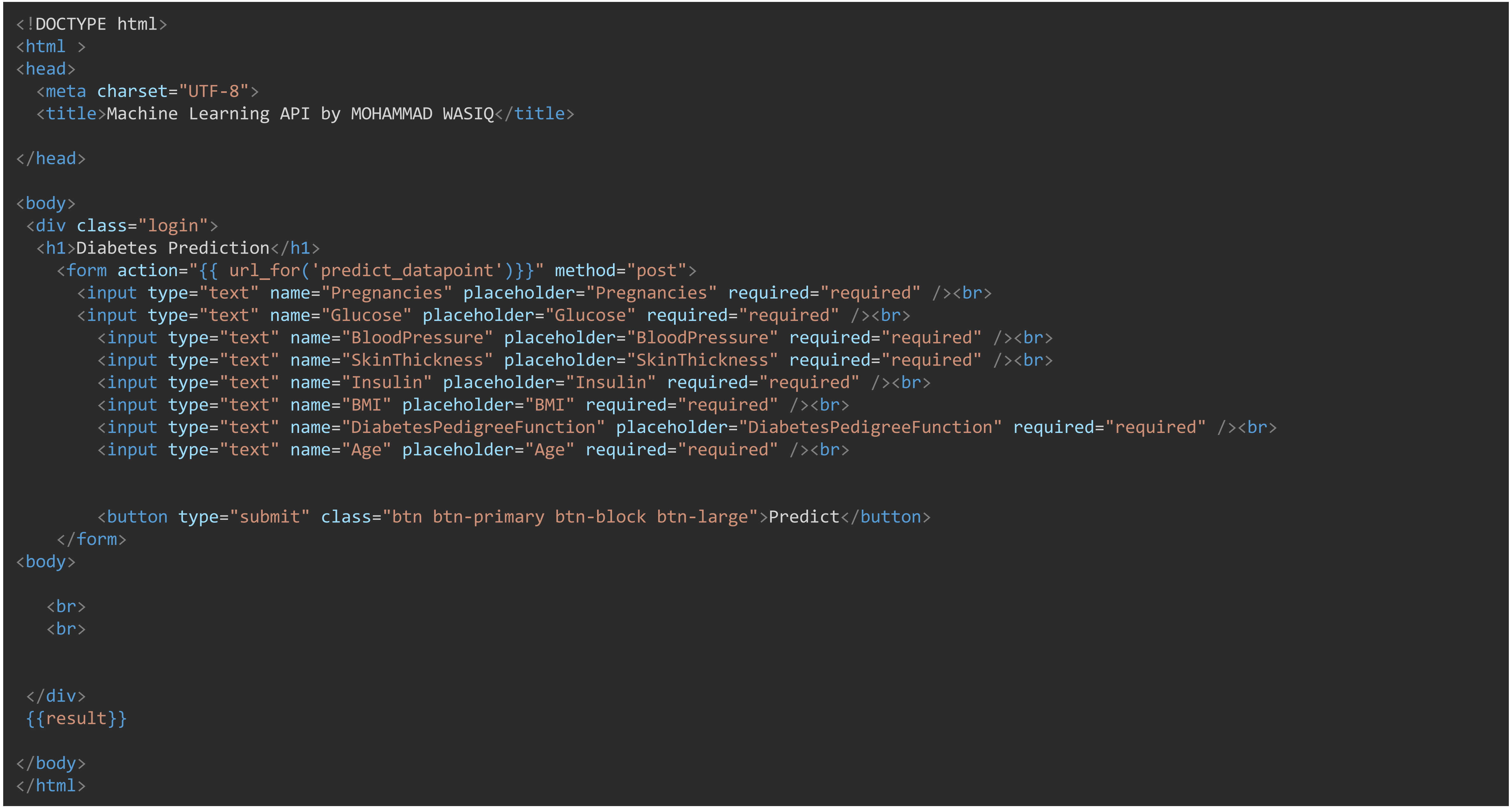
  * Create **application.py** file for *flask app*

  * Create **requirements.txt** file for require libraries

  * Create **templates** folder for *index.html, home.html*

- **home.html**

Coding



```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Machine Learning API by MOHAMMAD WASIQ</title>

</head>

<body>
 <div class="login">
  <h1>Diabetes Prediction</h1>
    <form action="{{ url_for('predict_datapoint')}}" method="post">
      <input type="text" name="Pregnancies" placeholder="Pregnancies" required="required" /><br>
      <input type="text" name="Glucose" placeholder="Glucose" required="required" /><br>
        <input type="text" name="BloodPressure" placeholder="BloodPressure" required="required" /><br>
        <input type="text" name="SkinThickness" placeholder="SkinThickness" required="required" /><br>
        <input type="text" name="Insulin" placeholder="Insulin" required="required" /><br>
        <input type="text" name="BMI" placeholder="BMI" required="required" /><br>
        <input type="text" name="DiabetesPedigreeFunction" placeholder="DiabetesPedigreeFunction" required="required" /><br>
        <input type="text" name="Age" placeholder="Age" required="required" /><br>


        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>
<body>

   <br>
   <br>


 </div>
 {{result}}

</body>
</html>
```

Coding

- **index.html**

```
<h1> MOHAMMAD WASIQ Welcome to the Home Page </h1>
```

- **requirements.txt**

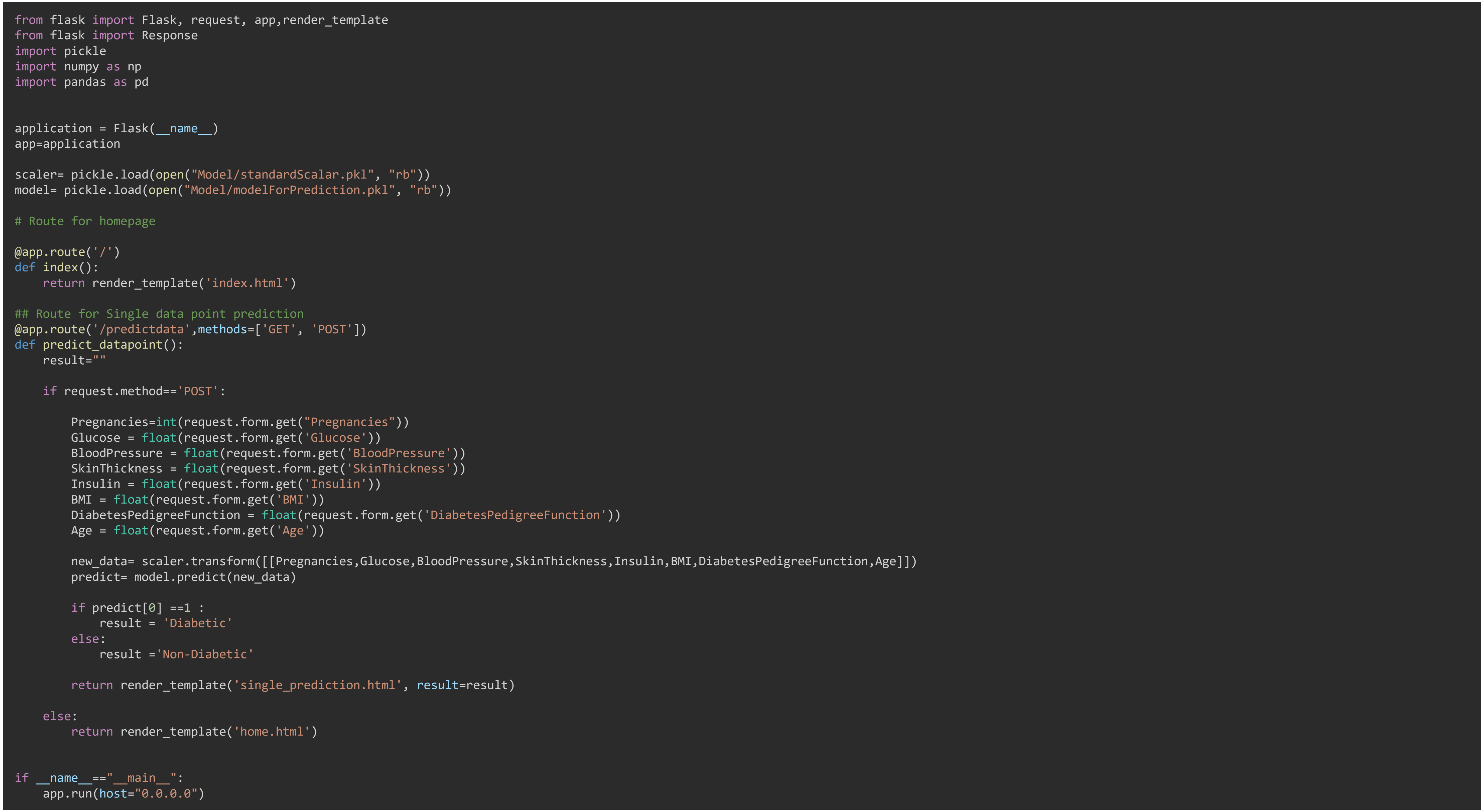
```
pandas
numpy
scikit-learn
seaborn
Flask
```

```
pip install –r requirements.txt
```

- **single_prediction.html**

```
<h1>You are: {{result}}</h1>
```

- **application.py**



```python
from flask import Flask, request, app,render_template
from flask import Response
import pickle
import numpy as np
import pandas as pd


application = Flask(__name__)
app=application

scaler= pickle.load(open("Model/standardScalar.pkl", "rb"))
model= pickle.load(open("Model/modelForPrediction.pkl", "rb"))

# Route for homepage

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

## Route for Single data point prediction
@app.route('/predictdata',methods=['GET', 'POST'])
def predict_datapoint():
    result=""

    if request.method=='POST':

        Pregnancies=int(request.form.get("Pregnancies"))
        Glucose = float(request.form.get('Glucose'))
        BloodPressure = float(request.form.get('BloodPressure'))
        SkinThickness = float(request.form.get('SkinThickness'))
        Insulin = float(request.form.get('Insulin'))
        BMI = float(request.form.get('BMI'))
        DiabetesPedigreeFunction = float(request.form.get('DiabetesPedigreeFunction'))
        Age = float(request.form.get('Age'))

        new_data= scaler.transform([[Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age]])
        predict= model.predict(new_data)

        if predict[0] ==1 :
            result = 'Diabetic'
        else:
            result ='Non-Diabetic'

        return render_template('single_prediction.html', result=result)

    else:
        return render_template('home.html')


if __name__=="__main__":
    app.run(host="0.0.0.0")
```

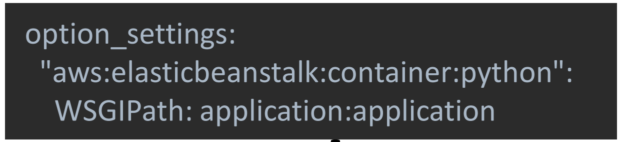
Git Commands

```
➢ ls –a
➢ git remote –v
➢ git remote rm origin
➢ git init
➢ git add README.md
➢ git add .
➢ git status
➢ git commit –m “first commit”
➢ git config –global user.email “mohammadwasiq0786@gmail.com”
➢ git config –global user.name “mohamamdwasiq0”
➢ git commit –m “first commit”
➢ git branch –M main
➢ git branch
➢ git remote add origin < GitHub Repo Link >
➢ git push –u origin main
```

- **Now it requires to permission to sign in the GitHub.**

AWS Deployment

- Create **.ebextensions** folder
- In this folder create **python.config** file and write the following code



```
option_settings:
  "aws:elasticbeanstalk:container:python":
    WSGIPath: application:application
```

Go to Console Home

- Click on Elastic Beanstalk
- On Left Top Click on **Applications**
- After Clicking On Right Side Click on Create Application
- In **Application name** give any *name* < predictiondiabities >
- On **Platform** choose *Platform* as **Python**
- In **Sample code** select *Sample application*
- Click on Create Application . It takes some times to get ready.

Code Pipeline

Search **CodePipeline**

- Click on CodePipeline
- On Left Top Click on **CodePipeline** Create Pipeline
- In **Pipeline name** give any *name* < predictiondiabities >
- Set everything default
- On Right Bottom Click on Next
- After that in **Source Provider** choose **GitHub** (Version 1)
- Click on Connect to GitHub
- Confirm the **Processing OAuth request**
- In **Repository** choose < Repository Name >
- In **Branch** choose **main** and Click on Next

Code Pipeline

- After clicking Next
- On Add build stage Click on Skip
- After Skipping there is an another window names **Add deploy stage** select **deploy provider** as **AWS Elastic Beanstalk**
- In **Region** select **Asia Pacific (Mumbai)**
- In **Application** name select **< predictiondiabities >**
- In **Environment** name select **< predictiondiabities >**
- Click on Next
- After that there is an another window named **Review**
- Go to **Elastic Beanstalk** and click on **predictiondiabities-env** and check **health** when it is Ok then.
- Go to again **Review** and on Right bottom Click on Create Pipeline