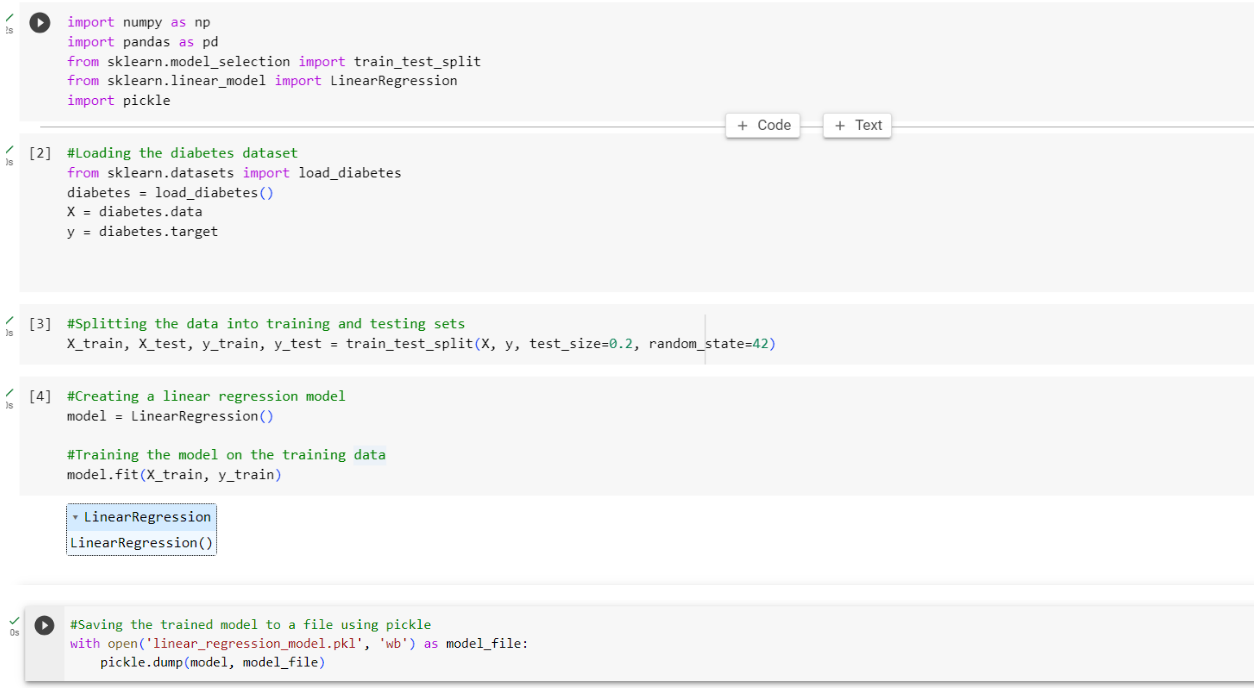
Name: Anusha Asim
Batch Code: LISUM24
Submission Date: 27th August, 2023
Submitted To: GitHub

Steps of Deployment

1. Building a model and downloading it as ‘model.pkl’



```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import pickle
```

```
[2] #Loading the diabetes dataset
    from sklearn.datasets import load_diabetes
    diabetes = load_diabetes()
    X = diabetes.data
    y = diabetes.target
```

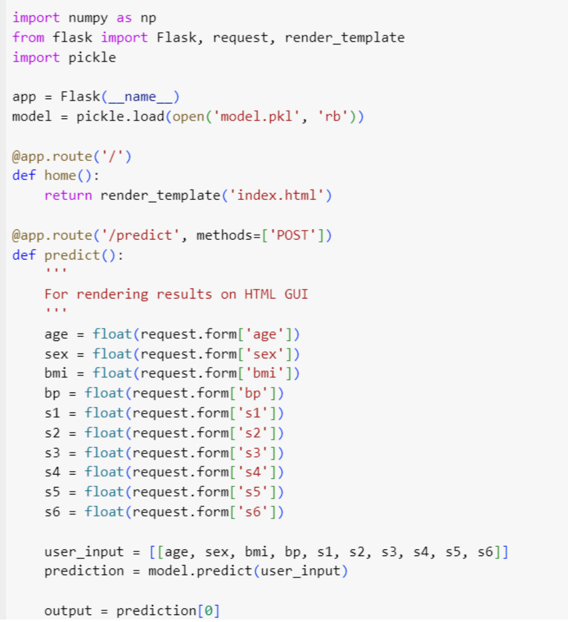
```
[3] #Splitting the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[4] #Creating a linear regression model
    model = LinearRegression()

    #Training the model on the training data
    model.fit(X_train, y_train)
```

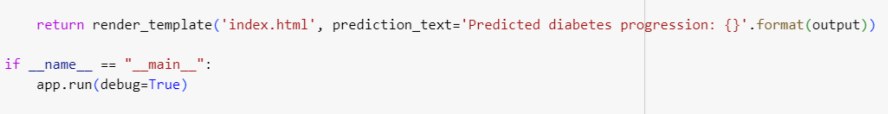
```
▾ LinearRegression
LinearRegression()
```

```
#Saving the trained model to a file using pickle
with open('linear_regression_model.pkl', 'wb') as model_file:
    pickle.dump(model, model_file)
```

2. Creating the app.py file, uploading the model file there and writing the code to integrate with index.html for flask web application

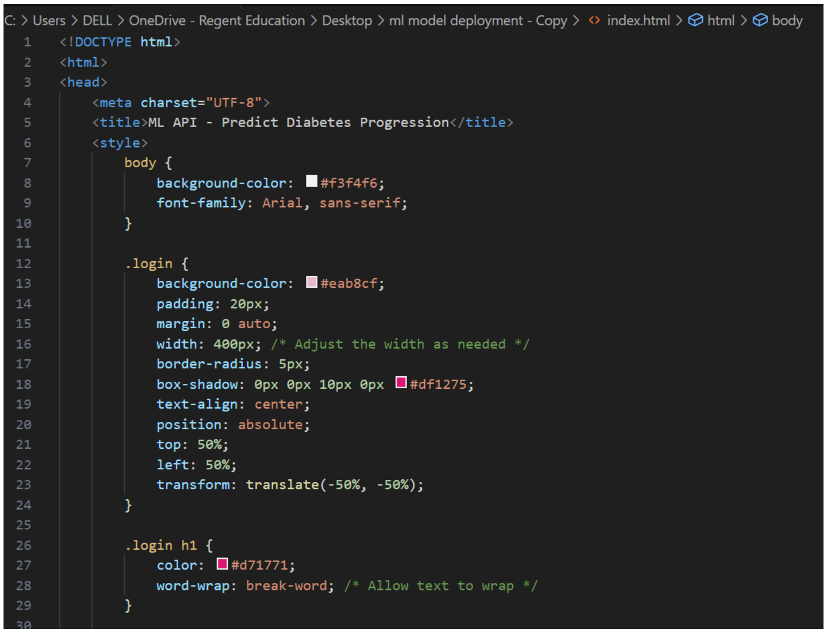


```python
import numpy as np
from flask import Flask, request, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

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

@app.route('/predict', methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    age = float(request.form['age'])
    sex = float(request.form['sex'])
    bmi = float(request.form['bmi'])
    bp = float(request.form['bp'])
    s1 = float(request.form['s1'])
    s2 = float(request.form['s2'])
    s3 = float(request.form['s3'])
    s4 = float(request.form['s4'])
    s5 = float(request.form['s5'])
    s6 = float(request.form['s6'])

    user_input = [[age, sex, bmi, bp, s1, s2, s3, s4, s5, s6]]
    prediction = model.predict(user_input)

    output = prediction[0]
```



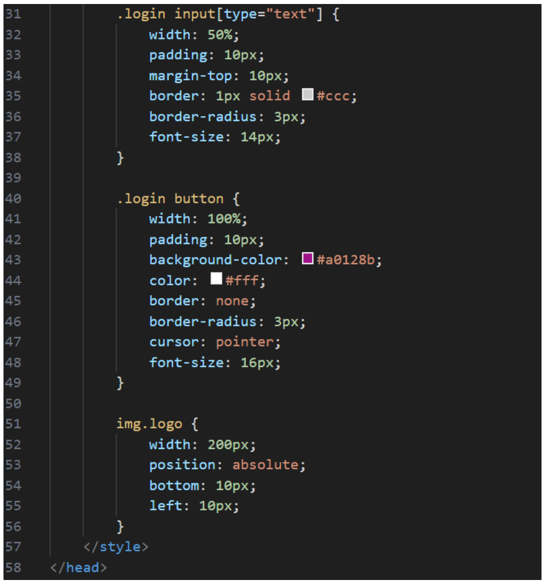
```python
    return render_template('index.html', prediction_text='Predicted diabetes progression: {}'.format(output))

if __name__ == "__main__":
    app.run(debug=True)
```

3. Creating and formatting the index.html file

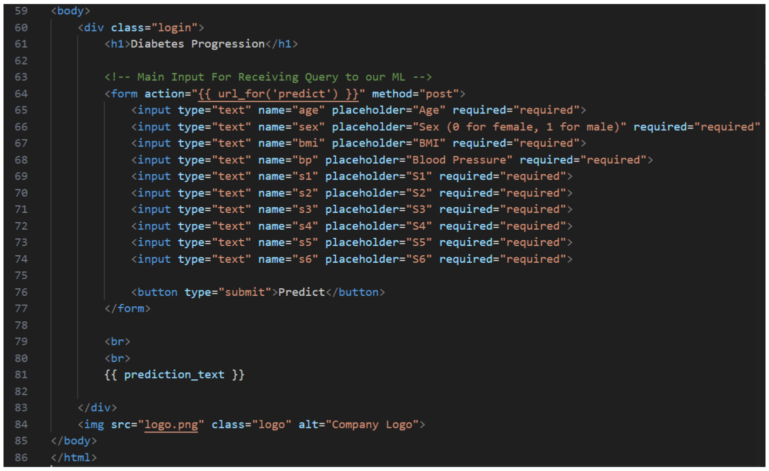


C: > Users > DELL > OneDrive - Regent Education > Desktop > ml model deployment - Copy > index.html > html > body

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>ML API - Predict Diabetes Progression</title>
    <style>
        body {
            background-color: #f3f4f6;
            font-family: Arial, sans-serif;
        }

        .login {
            background-color: #eab8cf;
            padding: 20px;
            margin: 0 auto;
            width: 400px; /* Adjust the width as needed */
            border-radius: 5px;
            box-shadow: 0px 0px 10px 0px #df1275;
            text-align: center;
            position: absolute;
            top: 50%;
            left: 50%;
            transform: translate(-50%, -50%);
        }

        .login h1 {
            color: #d71771;
            word-wrap: break-word; /* Allow text to wrap */
        }
```

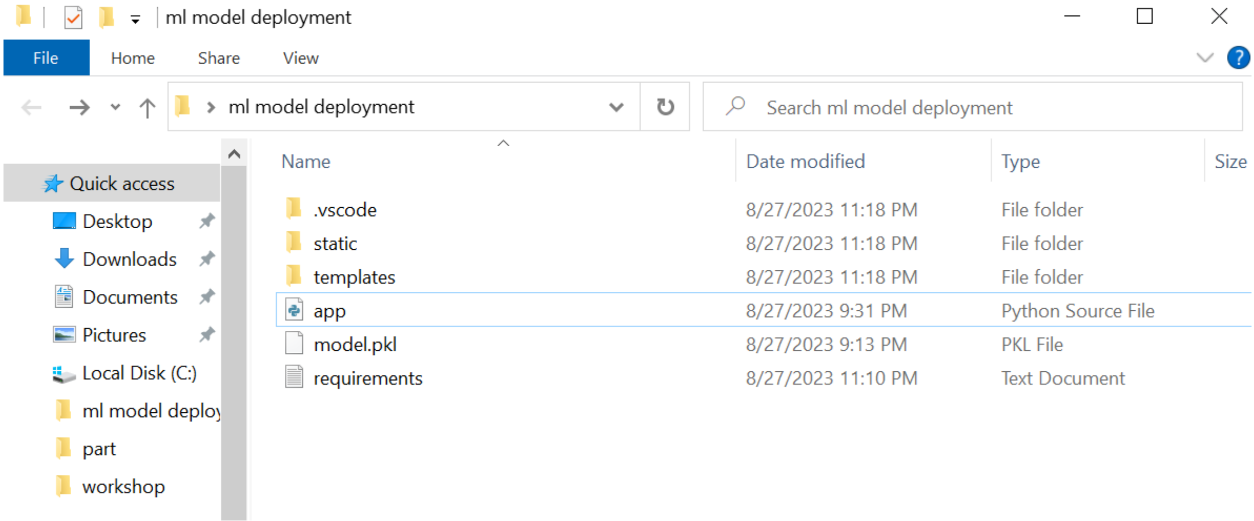


```
        .login input[type="text"] {
            width: 50%;
            padding: 10px;
            margin-top: 10px;
            border: 1px solid ■#ccc;
            border-radius: 3px;
            font-size: 14px;
        }

        .login button {
            width: 100%;
            padding: 10px;
            background-color: ■#a0128b;
            color: □#fff;
            border: none;
            border-radius: 3px;
            cursor: pointer;
            font-size: 16px;
        }

        img.logo {
            width: 200px;
            position: absolute;
            bottom: 10px;
            left: 10px;
        }
    </style>
</head>
```



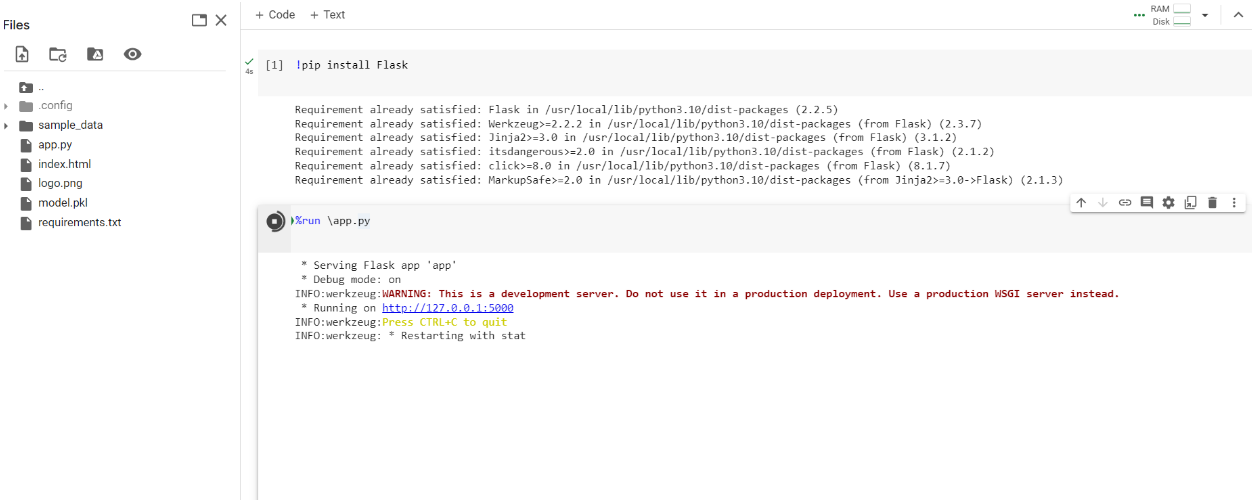
```
<body>
    <div class="login">
        <h1>Diabetes Progression</h1>

        <!-- Main Input For Receiving Query to our ML -->
        <form action="{{ url_for('predict') }}" method="post">
            <input type="text" name="age" placeholder="Age" required="required">
            <input type="text" name="sex" placeholder="Sex (0 for female, 1 for male)" required="required"
            <input type="text" name="bmi" placeholder="BMI" required="required">
            <input type="text" name="bp" placeholder="Blood Pressure" required="required">
            <input type="text" name="s1" placeholder="S1" required="required">
            <input type="text" name="s2" placeholder="S2" required="required">
            <input type="text" name="s3" placeholder="S3" required="required">
            <input type="text" name="s4" placeholder="S4" required="required">
            <input type="text" name="s5" placeholder="S5" required="required">
            <input type="text" name="s6" placeholder="S6" required="required">

            <button type="submit">Predict</button>
        </form>

        <br>
        <br>
        {{ prediction_text }}

    </div>
    <img src="logo.png" class="logo" alt="Company Logo">
</body>
</html>
```

4. Saving all the files in the project folder



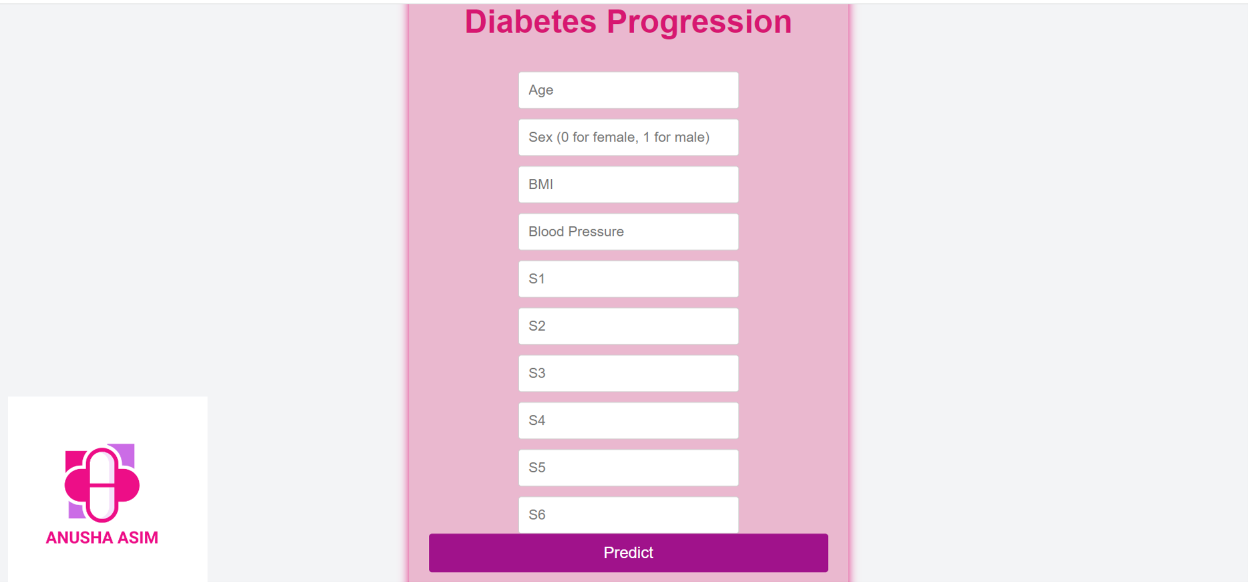
Name	Date modified	Type	Size
.vscode	8/27/2023 11:18 PM	File folder	
static	8/27/2023 11:18 PM	File folder	
templates	8/27/2023 11:18 PM	File folder	
app	8/27/2023 9:31 PM	Python Source File	
model.pkl	8/27/2023 9:13 PM	PKL File	
requirements	8/27/2023 11:10 PM	Text Document	

5. Uploading the files on a new Google Colab Notebook and Running

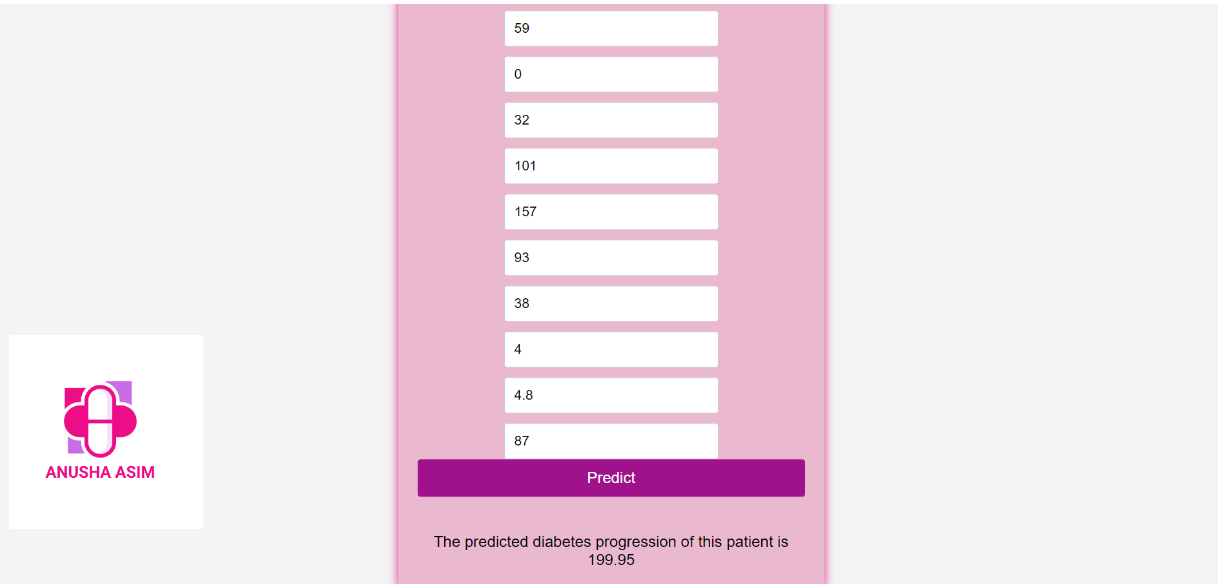


```
[1]  !pip install Flask

     Requirement already satisfied: Flask in /usr/local/lib/python3.10/dist-packages (2.2.5)
     Requirement already satisfied: Werkzeug>=2.2.2 in /usr/local/lib/python3.10/dist-packages (from Flask) (2.3.7)
     Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.10/dist-packages (from Flask) (3.1.2)
     Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.10/dist-packages (from Flask) (2.1.2)
     Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-packages (from Flask) (8.1.7)
     Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=3.0->Flask) (2.1.3)

%run \app.py

      * Serving Flask app 'app'
      * Debug mode: on
     INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
      * Running on http://127.0.0.1:5000
     INFO:werkzeug:Press CTRL+C to quit
     INFO:werkzeug: * Restarting with stat
```

6. Opening the link



7. Entering the data to see prediction



59

0

32

101

157

93

38

4

4.8

87

The predicted diabetes progression of this patient is 199.95