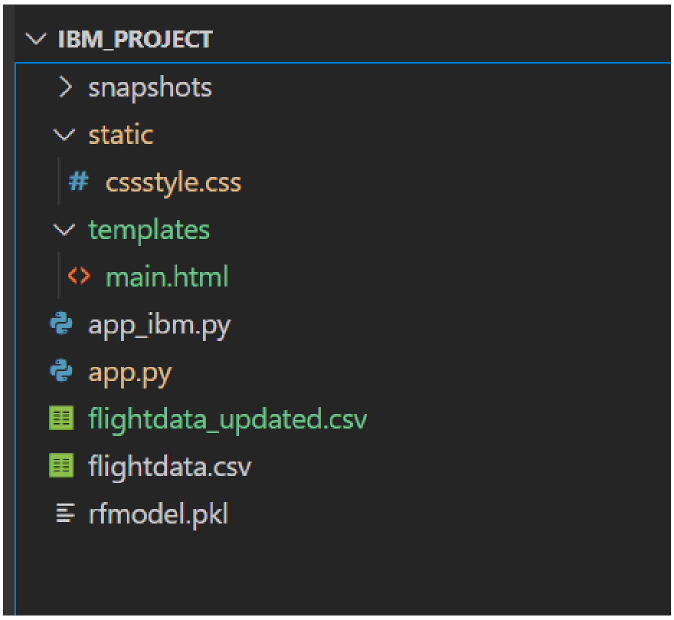
PROJECT DEVELOPMENT PHASE

SPRINT 3 – CODE

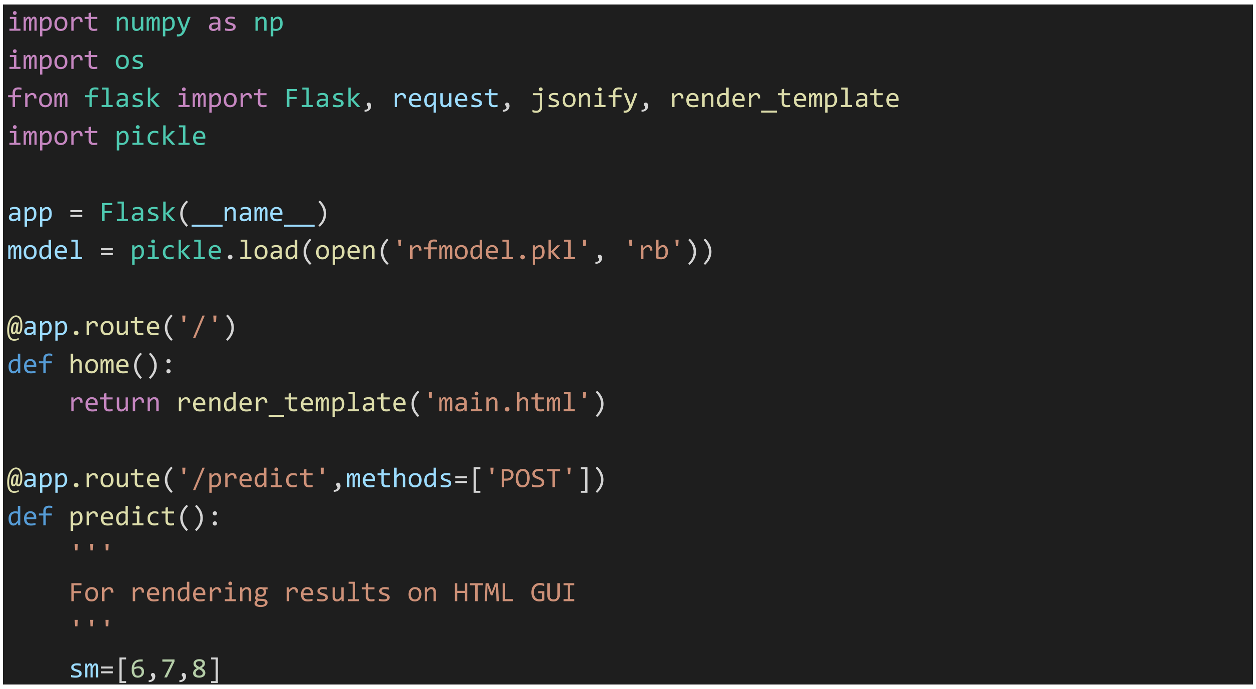
Team ID	PNT2022TMID53187
Project	Flight delay prediction using Machine learning
Marks	8 Marks

In this sprint development phase, we used pre-processed dataset to build a model. In order to create our models, we used the Decision Tree Classifier algorithm. Additionally, we have implemented a technique to evaluate the precision of our model and convert it into a Pkl file by importing the Pickle Python Library. The prediction is carried out by Flask App with the aid of the pickle model file.



```
IBM_PROJECT
  > snapshots
  v static
      cssstyle.css
  v templates
      main.html
    app_ibm.py
    app.py
    flightdata_updated.csv
    flightdata.csv
    rfmodel.pkl
```

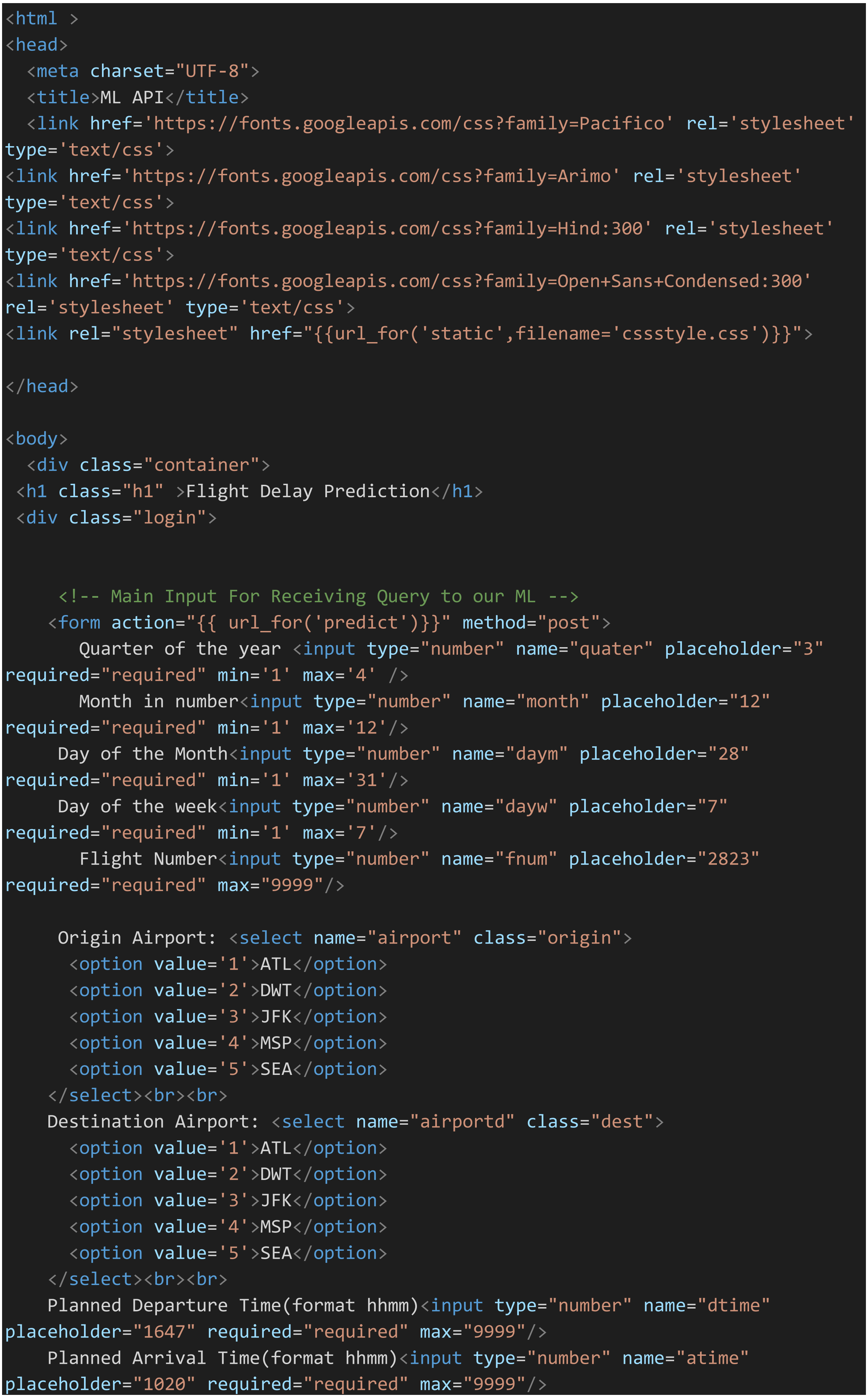
App.py:



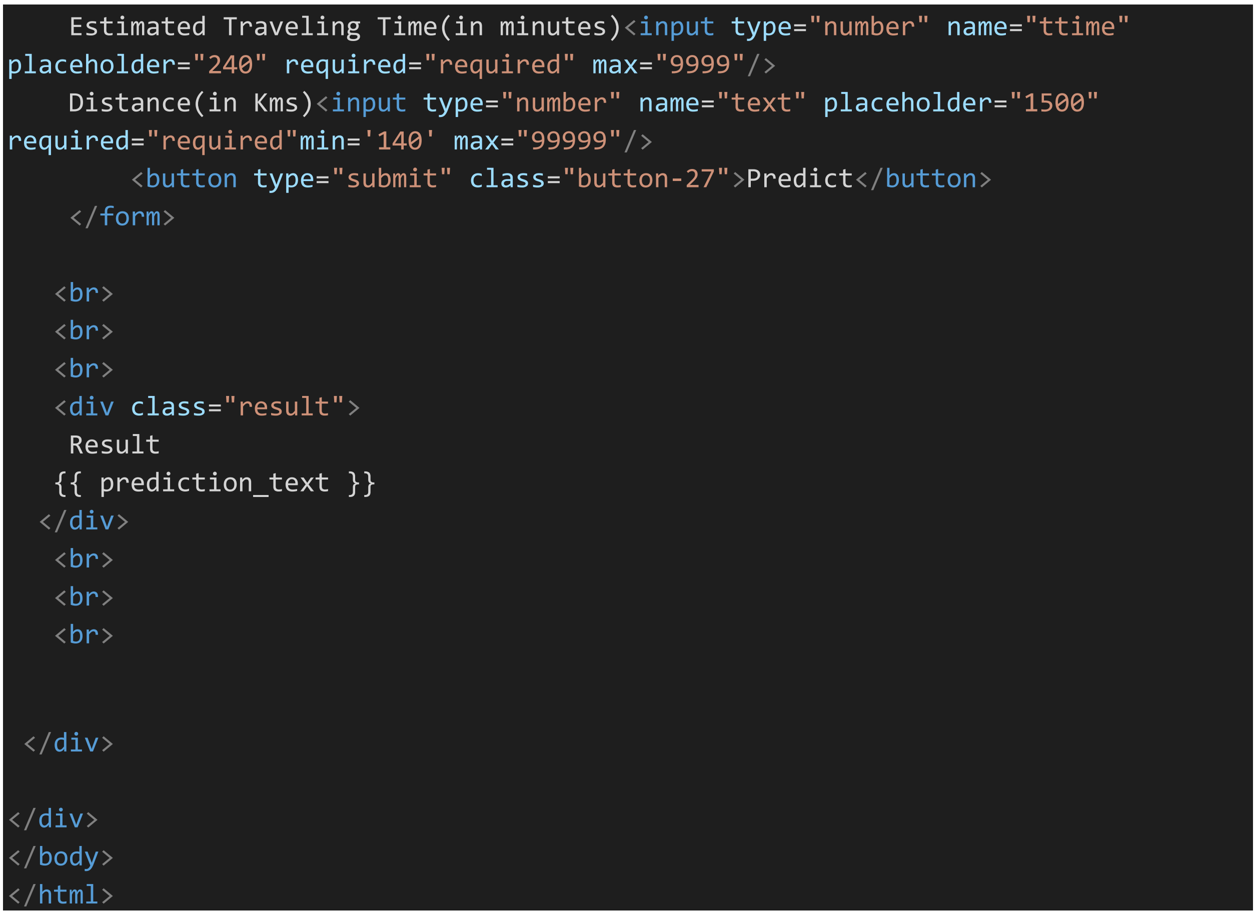
```python
import numpy as np
import os
from flask import Flask, request, jsonify, render_template
import pickle

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

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

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    sm=[6,7,8]
    wt=[9,10,11]
    sp=[12,1,2,3]
    fl=[4,5]
    farr= [int(x) for x in request.form.values()]
    if farr[1] in sm:
        farr.append(0)
    elif farr[1] in wt:
        farr.append(1)
    elif farr[1] in sp:
        farr.append(2)
    else:
        farr.append(3)
    final_features=np.array(farr,dtype='int64')
    print(final_features)
    prediction = model.predict([final_features])


    output = round(prediction[0])

    if output==0:
        return render_template('main.html', prediction_text='No delay will 
happen {}'.format(output))
    elif output==1:
        return render_template('main.html', prediction_text='There is a chance 
to departure delay will happen {}'.format(output))
    elif output==2:
        return render_template('main.html', prediction_text='here is a chance to 
both departure and arrival delay will happen {}'.format(output))
    elif output==3:
        return render_template('main.html', prediction_text='here is a chance to 
flight  will diverted {}'.format(output))
    elif output==4:
        return render_template('main.html', prediction_text='here is a chance to 
cancel the flight {}'.format(output))
    else:
        return render_template('main.html', prediction_text='output 
{}'.format(output))
'''@app.route('/predict_api',methods=['POST'])
def predict_api():

    For direct API calls trought request

    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])
    output = prediction[0]
    return jsonify(output)'''

if __name__ == "__main__":
    os.environ.setdefault('FLASK_ENV', 'development')
    app.run(debug=False)
```

main.html:

```
<!DOCTYPE html>
```

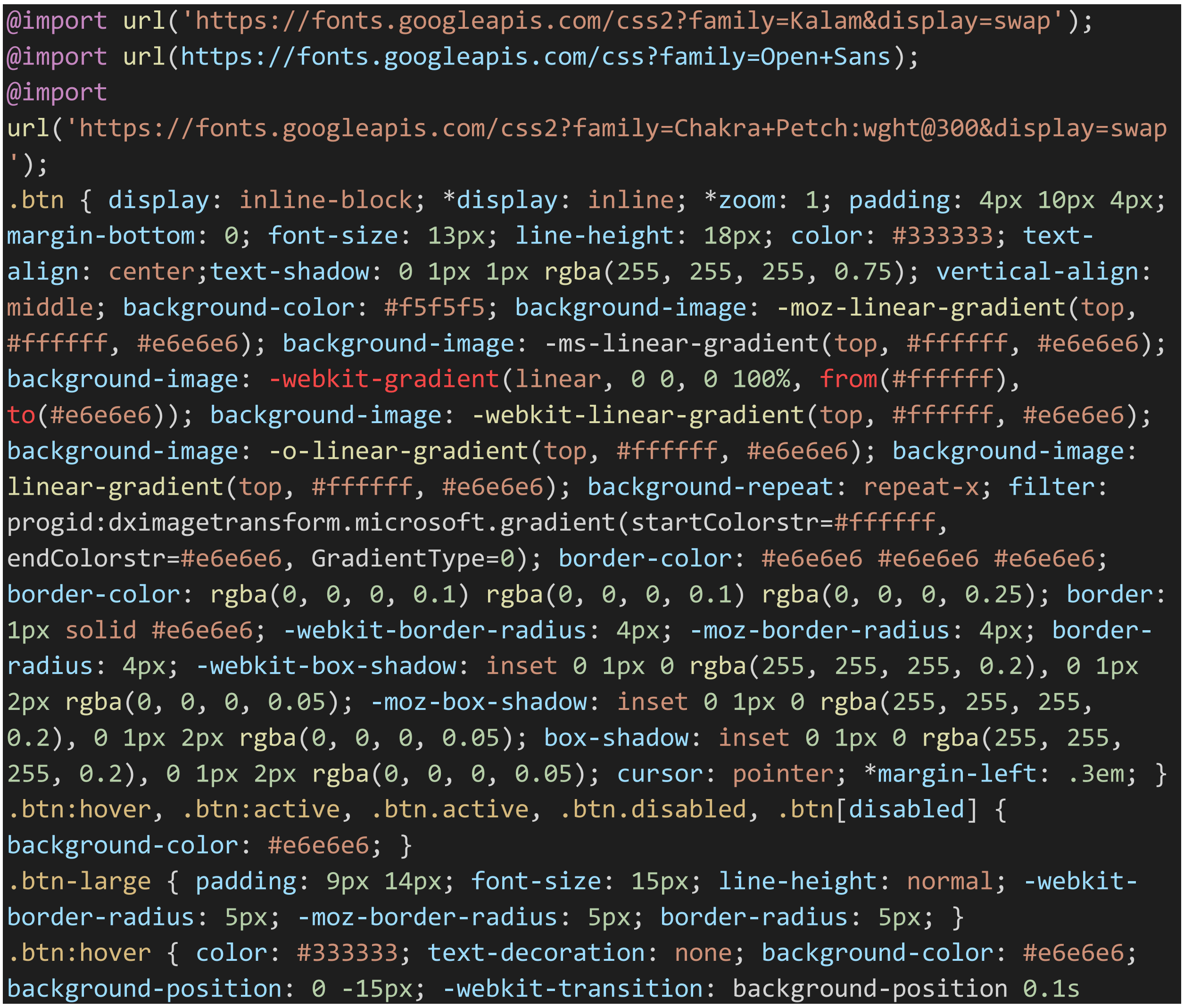


```
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{url_for('static',filename='cssstyle.css')}}">

</head>

<body>
  <div class="container">
 <h1 class="h1" >Flight Delay Prediction</h1>
 <div class="login">


     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}" method="post">
       Quarter of the year <input type="number" name="quater" placeholder="3"
required="required" min='1' max='4' />
       Month in number<input type="number" name="month" placeholder="12"
required="required" min='1' max='12'/>
     Day of the Month<input type="number" name="daym" placeholder="28"
required="required" min='1' max='31'/>
     Day of the week<input type="number" name="dayw" placeholder="7"
required="required" min='1' max='7'/>
       Flight Number<input type="number" name="fnum" placeholder="2823"
required="required" max="9999"/>

     Origin Airport: <select name="airport" class="origin">
      <option value='1'>ATL</option>
      <option value='2'>DWT</option>
      <option value='3'>JFK</option>
      <option value='4'>MSP</option>
      <option value='5'>SEA</option>
    </select><br><br>
    Destination Airport: <select name="airportd" class="dest">
      <option value='1'>ATL</option>
      <option value='2'>DWT</option>
      <option value='3'>JFK</option>
      <option value='4'>MSP</option>
      <option value='5'>SEA</option>
    </select><br><br>
    Planned Departure Time(format hhmm)<input type="number" name="dtime"
placeholder="1647" required="required" max="9999"/>
    Planned Arrival Time(format hhmm)<input type="number" name="atime"
placeholder="1020" required="required" max="9999"/>
```

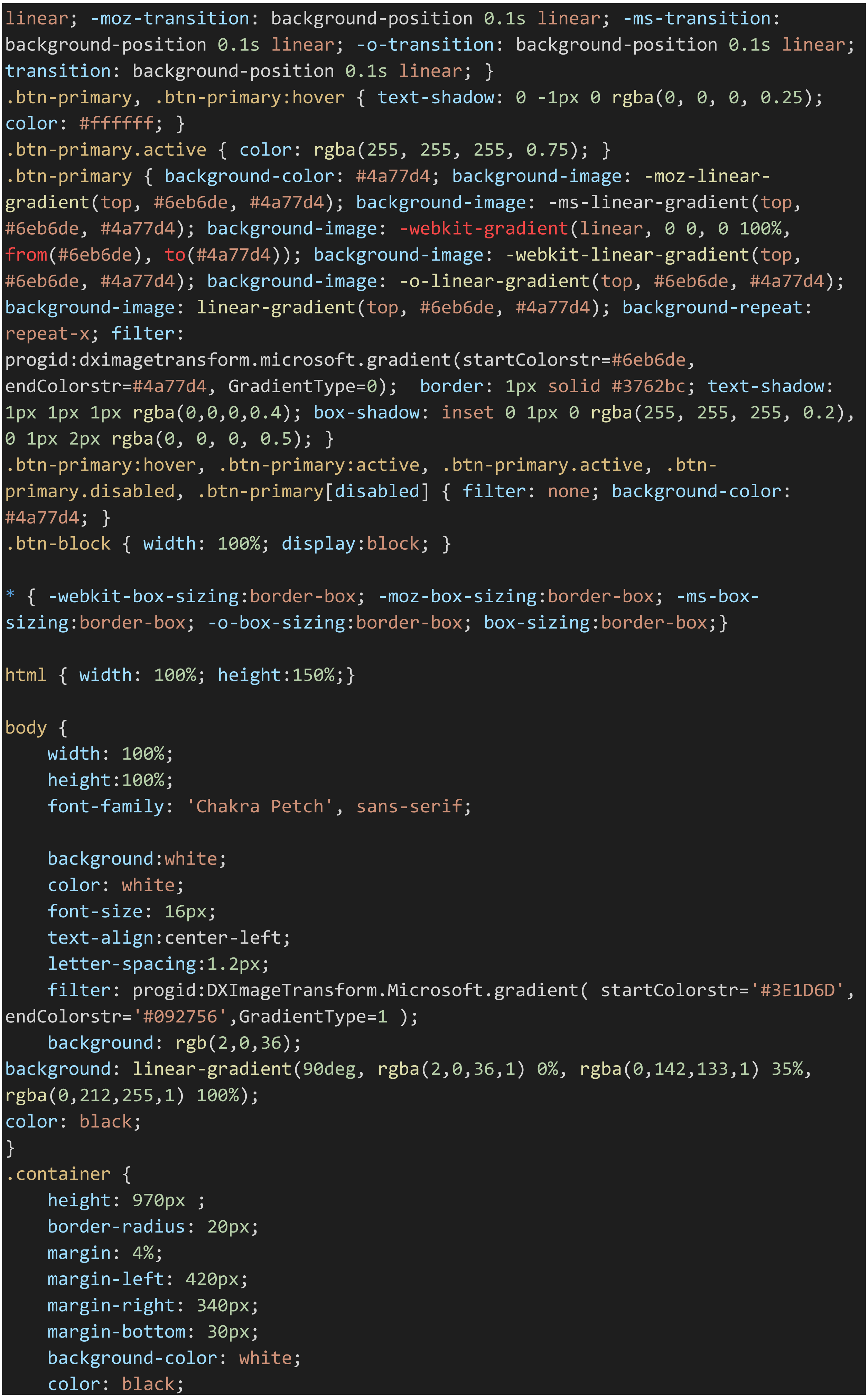


```
    Estimated Traveling Time(in minutes)<input type="number" name="ttime"
placeholder="240" required="required" max="9999"/>
    Distance(in Kms)<input type="number" name="text" placeholder="1500"
required="required"min='140' max="99999"/>
        <button type="submit" class="button-27">Predict</button>
    </form>

   <br>
   <br>
   <br>
   <div class="result">
    Result
   {{ prediction_text }}
  </div>
   <br>
   <br>
   <br>


 </div>

</div>
</body>
</html>
```

Cssstyle.css:



```
@import url('https://fonts.googleapis.com/css2?family=Kalam&display=swap');
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
@import
url('https://fonts.googleapis.com/css2?family=Chakra+Petch:wght@300&display=swap
');
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px;
margin-bottom: 0; font-size: 13px; line-height: 18px; color: #333333; text-
align: center;text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75); vertical-align:
middle; background-color: #f5f5f5; background-image: -moz-linear-gradient(top,
#ffffff, #e6e6e6); background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6);
background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff),
to(#e6e6e6)); background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6);
background-image: -o-linear-gradient(top, #ffffff, #e6e6e6); background-image:
linear-gradient(top, #ffffff, #e6e6e6); background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff,
endColorstr=#e6e6e6, GradientType=0); border-color: #e6e6e6 #e6e6e6 #e6e6e6;
border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border:
1px solid #e6e6e6; -webkit-border-radius: 4px; -moz-border-radius: 4px; border-
radius: 4px; -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px
2px rgba(0, 0, 0, 0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255,
0.2), 0 1px 2px rgba(0, 0, 0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255,
255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] {
background-color: #e6e6e6; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-
border-radius: 5px; -moz-border-radius: 5px; border-radius: 5px; }
.btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6;
background-position: 0 -15px; -webkit-transition: background-position 0.1s
```

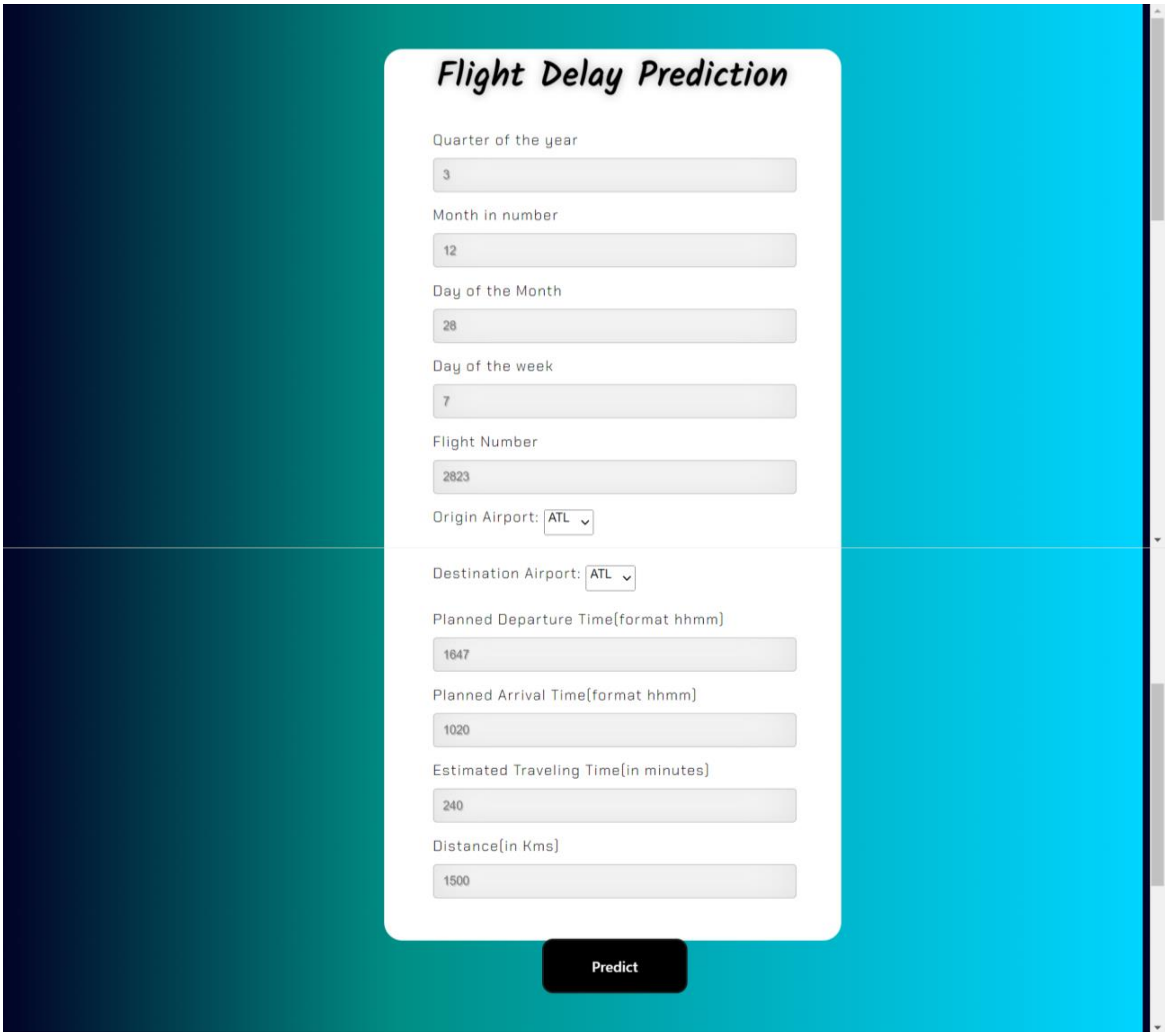


```
linear; -moz-transition: background-position 0.1s linear; -ms-transition:
background-position 0.1s linear; -o-transition: background-position 0.1s linear;
transition: background-position 0.1s linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);
color: #ffffff; }
.btn-primary.active { color: rgba(255, 255, 255, 0.75); }
.btn-primary { background-color: #4a77d4; background-image: -moz-linear-
gradient(top, #6eb6de, #4a77d4); background-image: -ms-linear-gradient(top,
#6eb6de, #4a77d4); background-image: -webkit-gradient(linear, 0 0, 0 100%,
from(#6eb6de), to(#4a77d4)); background-image: -webkit-linear-gradient(top,
#6eb6de, #4a77d4); background-image: -o-linear-gradient(top, #6eb6de, #4a77d4);
background-image: linear-gradient(top, #6eb6de, #4a77d4); background-repeat:
repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0);  border: 1px solid #3762bc; text-shadow:
1px 1px 1px rgba(0,0,0,0.4); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2),
0 1px 2px rgba(0, 0, 0, 0.5); }
.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-
primary.disabled, .btn-primary[disabled] { filter: none; background-color:
#4a77d4; }
.btn-block { width: 100%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-
sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box;}

html { width: 100%; height:150%;}

body {
    width: 100%;
    height:100%;
    font-family: 'Chakra Petch', sans-serif;

    background:white;
    color: white;
    font-size: 16px;
    text-align:center-left;
    letter-spacing:1.2px;
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D',
endColorstr='#092756',GradientType=1 );
    background: rgb(2,0,36);
background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(0,142,133,1) 35%,
rgba(0,212,255,1) 100%);
color: black;
}
.container {
    height: 970px ;
    border-radius: 20px;
    margin: 4%;
    margin-left: 420px;
    margin-right: 340px;
    margin-bottom: 30px;
    background-color: white;
    color: black;
```

```css
}
.result{
    background-color: white;
    height: 120px;
    width: 380px;
    border-radius: 5px;
    margin:100px auto;
}
.login {
    position: absolute;
    top: 40%;
    left: 50%;
    margin: -100px 0 0 -150px;
    width:400px;
    height:400px;
}



/* CSS */
.button-27 {

  appearance: none;
  background-color: #000000;
  border: 2px solid #1A1A1A;
  border-radius: 12px;
  box-sizing: border-box;
  color: #FFFFFF;
  cursor: pointer;
  display: inline-block;
  font-family: Roobert,-apple-system,BlinkMacSystemFont,"Segoe UI",Helvetica,Arial,sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol";
  font-size: 16px;
  font-weight: 600;
  line-height: normal;
  margin: 0;
  min-height: 60px;
  min-width: 0;
  outline: none;
  padding: 16px 24px;
  text-align: center;
  text-decoration: none;
  transition: all 300ms cubic-bezier(.23, 1, 0.32, 1);
  user-select: none;
  -webkit-user-select: none;
  touch-action: manipulation;
  width: 40%;
  will-change: transform;
  margin-top: 30px;
  margin-left: 120px;
}
```

```
.button-27:disabled {
  pointer-events: none;
}

.button-27:hover {
  box-shadow: rgba(0, 0, 0, 0.25) 0 8px 15px;
  transform: translateY(-2px);
}

.button-27:active {
  box-shadow: none;
  transform: translateY(0);
}
.h1 {font-size: 38px;color: black;font-family: 'Kalam', cursive; text-shadow: 0
0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center;margin-bottom:
20px;}
body .origin{padding-bottom:10px}
body .dest{padding-bottom:10px}
body .btn{padding:20px}
input {
    width: 100%;
    margin-bottom: 15px;
    margin-top: 10px;
    background: rgb(255, 255, 255);
    border: none;
    outline: none;
    padding: 10px;
    font-size: 13px;
    color: black;
    text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
    border: 1px solid rgba(0,0,0,0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }
```

output screenshots:



Flight Delay Prediction

Quarter of the year: 3

Month in number: 12

Day of the Month: 28

Day of the week: 7

Flight Number: 2823

Origin Airport: ATL

Destination Airport: ATL

Planned Departure Time(format hhmm): 1647

Planned Arrival Time(format hhmm): 1020

Estimated Traveling Time(in minutes): 240

Distance(in Kms): 1500