

## Sprint - 1 objectives:

- Data preprocessing
- Login page and registration page along with database connection

### Data preprocessing:

- Removing unwanted columns
- Data cleaning
- EDA/Data analysis
- Feature Extraction
- Label hot encoding

### To-do:

- Splitting dependent and independent variables
- Encoding Categorical columns to numerical
- Split Train-Test Data
- Model Building

Refer ipynb for data preprocessing operations.

### Login Page and Registration page:

#### login.html:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<link rel="stylesheet" href="{{ url_for('static', filename='des.css') }}">
<body>
<br>
<h2 style="text-align:center;">A Flight Delay Prediction Model</h2>
<form action="{{ url_for('dashh') }}" method="POST">
  <div class="imgcontainer">
    
  </div>

  <div class="container">
    <label for="uname"><b>Username</b></label>
    <input type="text" placeholder="Enter Username" name="uname" required>

    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="psw" required>
```

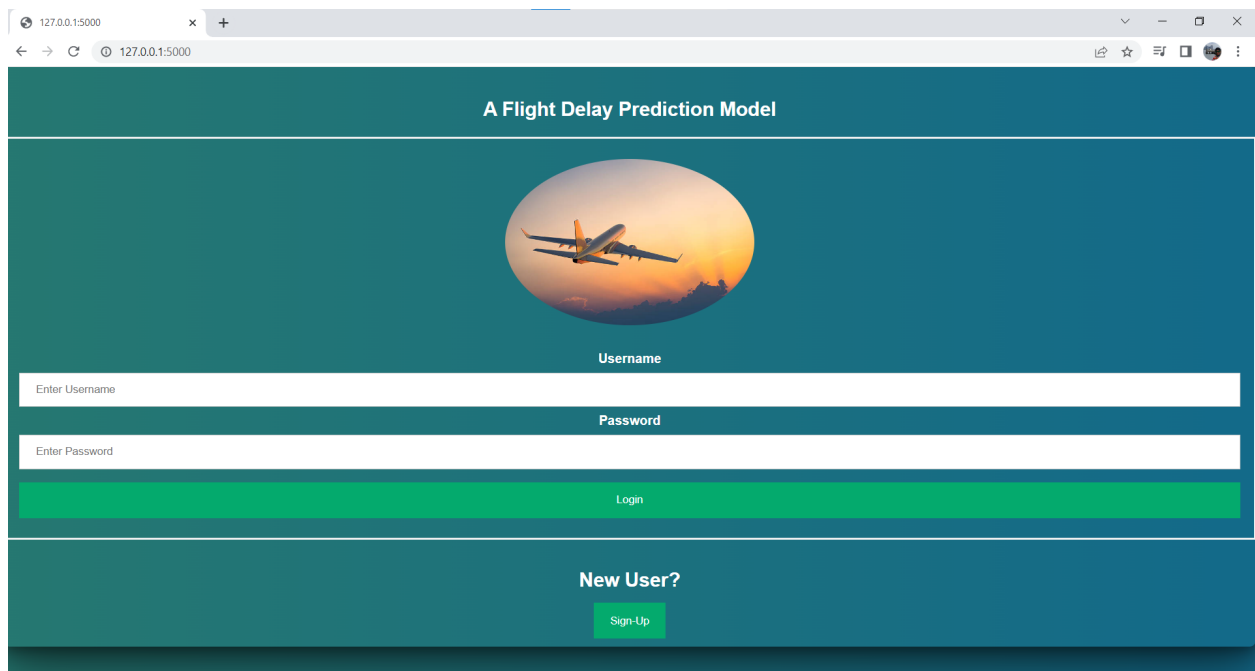
```

    <button type="submit">Login</button>
</form>
</div>

<div class="container">
<h2 style="text-align:center;">New User?</h2>
<div class="container">
<div class="center">

<form style="border:none;" action="{{ url_for('sign') }}">
    <button type="submit">Sign-Up</button>
</form>
</div>
</div>
</div>
</div>
</body>
</html>

```



signup.html:

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="{{ url_for('static', filename='des.css') }}">

```

```

</head>
<body>
<br>
<h2 style="text-align:center;">Create an account</h2>
<form action="{{ url_for('home') }}" method="POST">

<div class="container">
  <label for="em"><b>Enter Email address</b></label>
  <input type="email" placeholder="Enter an email address" name="em" required>

  <label for="unme"><b>Enter a Username</b></label>
  <input type="text" placeholder="Enter a Username" name="unme" required>

  <label for="pswd"><b>Enter a Password</b></label>
  <input type="password" placeholder="Enter a Password" name="pswd" required>

  <button type="submit">Create</button>
</div>
</div>
</div>
</div>
</form>
</body>
</html>

```

The screenshot displays a web browser window with a single tab titled '127.0.0.1:5000/sign?'. The browser's address bar also shows this URL. The main content of the page is a 'Create an account' form. The form has a dark teal header with the title 'Create an account'. Below the header, there are three input fields with labels 'Enter Email address', 'Enter a Username', and 'Enter a Password'. Each input field has a placeholder text matching the label. At the bottom of the form is a green 'Create' button. The background of the page is a solid dark teal color.

## Database connection:

### For signup:

```
if request.method=='POST':
    email=request.form['em']
    uname=request.form['unme']
    password=request.form['pswd']
    login.insert_one({"email":email,"username":uname,"password":password})
    return render_template('login.html')
return render_template('login.html')
```

### For login:

```
if request.method=='POST':
    coll=login.find()
    uname=request.form['uname']
    password=request.form['psw']
    for i in coll:
        if(uname==i['username'] and password==i['password']):
            return render_template('dashboard.html')
```

## Mongodb:

The screenshot shows the MongoDB Compass web interface. At the top, the collection name 'login.users' is displayed. On the right, there are two green boxes with the numbers '2' and '1', labeled 'DOCUMENTS' and 'INDEXES' respectively. Below the collection name, there are tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. The 'Documents' tab is selected. A filter bar shows a filter query: '{ field: 'value' }'. To the right of the filter bar are buttons for 'OPTIONS', 'FIND', 'RESET', and a refresh icon. Below the filter bar, there is a section for 'ADD DATA' and a 'VIEW' section with icons for list, grid, and table views. The main area displays three documents, each with a unique ObjectId, email, username, and password. The status bar at the bottom indicates 'Displaying documents 1 - 3 of 3' and includes a 'REFRESH' button.

Document	ObjectId	email	username	password
1	ObjectId('63769df4456f0aa7aa4c9ce9')	"em@s"	"d"	"pwd"
2	ObjectId('63769df7b456f0aa7aa4c9cee')	"j@j"	"aad"	"1234"
3	ObjectId('63770cef94d9c0b3dddb5870')	"m@m"	"anish"	"1"