Exp No : 1

CREATE GITHUB REPOSITORY AND EXPLORE ITS FEATURES

Date :

AIM:

To create github repository and explore its features

TOOLS REQUIRED:

GitHub

PROCEDURE:

1. Log in to GitHub → Click “New” → Enter repository name → Set visibility (Public/Private) → Click “Create repository”.

2. Use git clone to copy the repo locally, add your project files, then use the commands

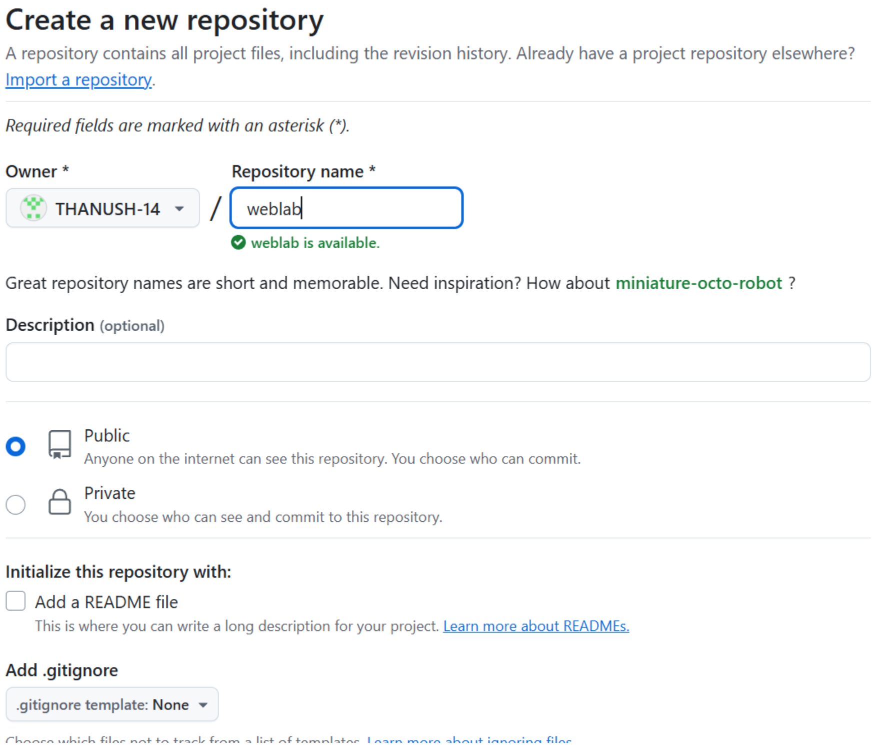
   - git add .
   - git commit -m "message"
   - git push -u origin main

3. Create branches for new features using git branch and git checkout, then merge them into the main branch via pull requests

4. Use “Issues” to report bugs or suggest features, and “Projects” (Kanban-style boards) to manage tasks and workflows.

5. Add a README.md file to describe your project. Use Wiki and GitHub Pages for extended documentation and hosting.

OUTPUT:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * THANUSH-14 / **Repository name *** weblab

✓ weblab is available.

Great repository names are short and memorable. Need inspiration? How about **miniature-octo-robot** ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

RESULT:

Repository had been created in github and its features had been explored

Exp No : 2 **DESIGN A STATIC WEBSITE USING HTML TAGS & HOST**
Date : **IT IN GITHUB REPOSITORY**

AIM:

To design a static website using html tags & host it in github repository

TOOLS REQUIRED:

GitHub,HTML

PROCEDURE:

1. Create HTML website → Write HTML/CSS code → Include index.html → Test locally → Save files

2. Log in to GitHub → Click "New" → Enter repository name → Set to Public → Click "Create repository"

3. Use git clone to copy repo → Add website files → Run git add . → Commit with git commit -m "message" → Push with git push

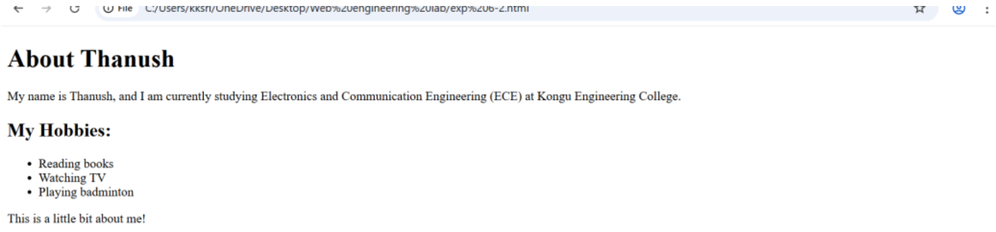
4. Go to repository Settings → Find "Pages" section → Select main branch → Choose root folder → Enable GitHub Pages

5. Visit username.github.io/repository → Verify website loads → Check all links → Fix any issues → Share URL

CODE : INDEX.HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

    <title>Thanush</title>

</head>
<body>
    <div class="container">
        <h1>About Thanush</h1>
        <p>My name is Thanush, and I am currently studying Electronics and Communication Engineering (ECE) at Kongu Engineering College.</p>
        <h2>My Hobbies:</h2>
        <ul>
            <li>Reading books</li>
            <li>Watching TV</li>
            <li>Playing badminton</li>
        </ul>
        <p>This is a little bit about me!</p>
    </div>
</body>
</html>
```

OUTPUT:



C:/Users/kksri/OneDrive/Desktop/Web%20engineering%20lab/exp%206-2.html

About Thanush

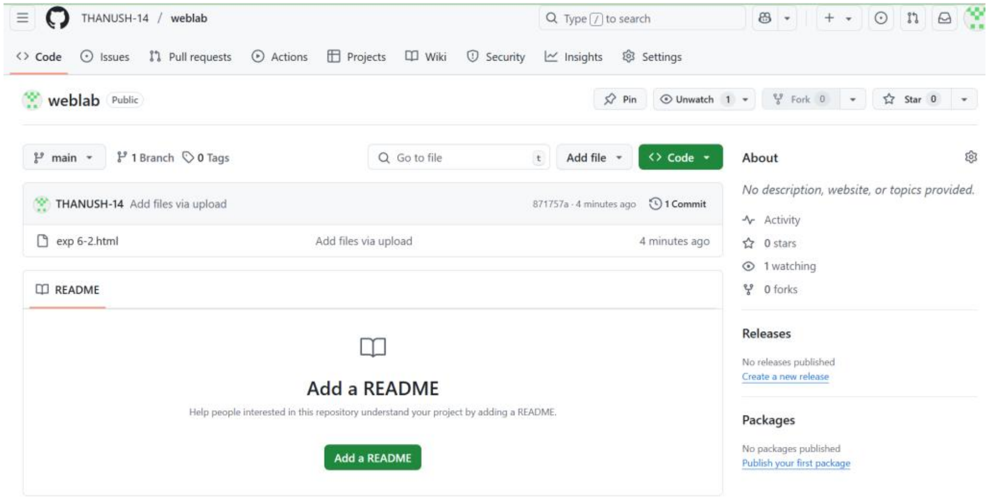
My name is Thanush, and I am currently studying Electronics and Communication Engineering (ECE) at Kongu Engineering College.

My Hobbies:

- Reading books
- Watching TV
- Playing badminton

This is a little bit about me!

OUTPUT:



THANUSH-14 Add files via upload 871757a · 4 minutes ago 1 Commit

exp 6-2.html Add files via upload 4 minutes ago

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

RESULT:

A Static page had been designed using html tags and pushed into the repository and successfully deployed.

Exp no: 3
Date :

APPLY CSS3 TEXT, BACKGROUND AND BORDER PROPERTIES TO DESIGN AN ATTRACTIVE WEB PAGE

AIM:

To apply css3 text, background and border properties to design an attractive web page

TOOLS REQUIRED:

HTML, CSS3

PROCEDURE:

1. Create HTML structure → Define header, main, footer → Add content sections → Include CSS file → Test basic layout

2. Apply CSS3 text properties → Set font-family, size → Use text-shadow, letter-spacing → Add color gradients → Style headings, paragraphs

3. Implement background properties → Use linear-gradient → Add background images → Set background-size, position → Apply opacity → Ensure contrast

4. Design borders with CSS3 → Use border-radius → Apply box-shadow → Set border-style, width → Add hover effects → Style buttons, cards

5. Enhance and test page → Add transitions, animations → Ensure responsiveness → Test cross-browser → Adjust spacing, alignment → Deploy to GitHub Pages

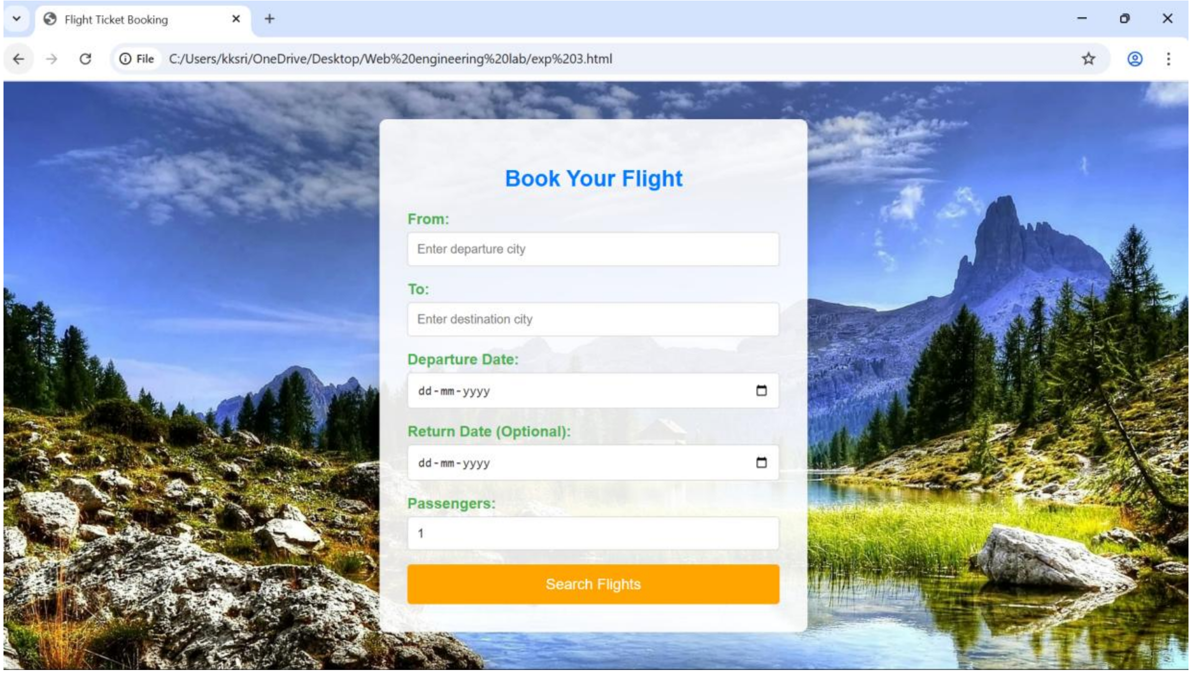
CSS.STYLE

```
<style>
body {
font-family: sans-serif;
background-color: #f4f4f4;
display: flex;
justify-content: center;
align-items: center;
min-height: 100vh;
margin: 0;
background-image:url('https://images.pexels.com/photos/371633/pexels-photo-371633.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=2');
background-size: cover;
background-repeat: no-repeat;
}
.booking-container {
background-color: rgba(255, 255, 255, 0.9);
padding: 30px;
border-radius: 8px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
width: 400px;
}
h2 {
text-align: center;
margin-bottom: 20px;
```

```
color: rgb(0, 123, 255);
}
.form-group {
margin-bottom: 15px;
}
label {
display: block;
margin-bottom: 5px;
color: rgb(76, 175, 80);
font-weight: bold;
}
input[type="text"],
input[type="date"],
select {
width: 100%;
padding: 10px;
border: 1px solid rgb(204, 204, 204);
border-radius: 4px;
box-sizing: border-box;

color: rgb(51, 51, 51);
}
select {
appearance: none;
background-image: url('data:image/svg+xml;utf8,<svg fill="currentColor" viewBox="0 0 24 24"><path d="M7 10l5 5 5-5z"/></svg>');
```

```
}
background-repeat: no-repeat;
background-position-x: 95%;
background-position-y: center;
background-size: 1em;
}
button {
background-color: rgb(255, 165, 0);
color: rgb(255, 255, 255);
padding: 12px 20px;
border: none;
border-radius: 4px;
cursor: pointer;
font-size: 1em;
width: 100%;
}
button:hover {
background-color: rgb(255, 140, 0);
}
.note {
font-size: 0.8em;
color: rgb(119, 119, 119);
margin-top: 10px;
text-align: center;
}
</style>
```

OUTPUT:


Flight Ticket Booking
File
C:/Users/kksri/OneDrive/Desktop/Web%20engineering%20lab/exp%203.html

Book Your Flight

From:

Enter departure city

To:

Enter destination city

Departure Date:

dd-mm-yyyy

Return Date (Optional):

dd-mm-yyyy

Passengers:

1

Search Flights

RESULT:

CSS3 text, background and border properties to design an attractive web page had been done.

Exp No : 4

Date :

DESIGN A WEBPAGE WITH NAVIGATION BAR USING APPROPRIATE CSS3 PROPERTIES

AIM:

To design a webpage with navigation bar using appropriate css3 properties

SOFTWARE / TOOLS USED:

HTML,CSS

PROCEDURE:

1. Create HTML structure → Define header, nav, main → Add navigation links → Include CSS file Test basic layout

2. Style navigation bar → Use flexbox for layout → Apply background gradient → Set text properties → Add hover effects

3. Enhance nav with CSS3 → Use border-radius, shadows → Add transitions → Implement active states → Ensure accessibility

4. Design page content → Apply text styling → Use background properties → Add card-like sections → Set responsive margins

5. Test and deploy → Check responsiveness → Verify link functionality → Host on GitHub Pages → Adjust for cross-browser → Share URL

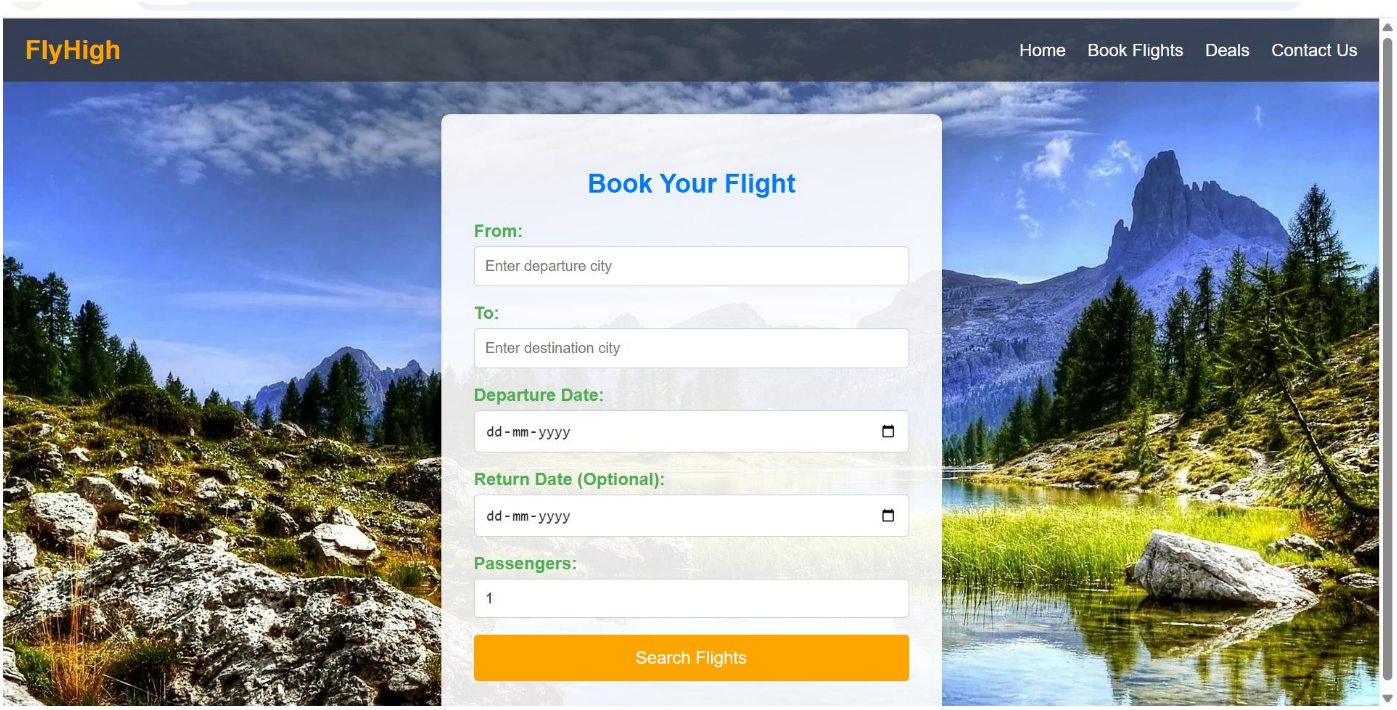
CSS STYLE:

```
nav {
    background-color: rgba(52, 58, 64, 0.8); /* Dark semi-transparent background */
    color: white;
    padding: 15px 20px;
    width: 100%;
    box-sizing: border-box;
    display: flex;
    justify-content: space-between;
    align-items: center;
 }

 nav .logo {
    font-size: 1.5em;
    font-weight: bold;
    color: rgb(255, 165, 0); /* Orange logo */
 }

 nav ul {
    list-style: none;
    padding: 0;
    margin: 0;
    display: flex;
 }

 nav ul li {
    margin-left: 20px;
 }

 nav ul li:first-child {
    margin-left: 0;
 }

 nav ul li a {
    color: white;
    text-decoration: none;
    transition: color 0.3s ease;
 }

 nav ul li a:hover {
    color: rgb(0, 123, 255); /* Blue on hover */
 }
```

ADD INSIDE HEADER TAG:

```
<div class="logo">FlyHigh</div>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Book Flights</a></li>
            <li><a href="#">Deals</a></li>
            <li><a href="#">Contact Us</a></li>
        </ul>
```

OUTPUT:



Book Your Flight

From: Enter departure city

To: Enter destination city

Departure Date: dd-mm-yyyy

Return Date (Optional): dd-mm-yyyy

Passengers: 1

RESULT:

A webpage with navigation bas using appropriate css3 properties had been designed

Exp No : 5 **DESIGN AN ONLINE REGISTRATION FORM FOR ANY**
Date :

REAL TIME APPLICATION

AIM:

To design an online registration form for any real time application

TOOLS USED:

HTML,CSS

PROCEDURE:

1. Build HTML structure → Create login/registration forms → Add email, password, name inputs → Include submit buttons → Link CSS file or use In-Line Styling

2. Style layout with CSS3 → Use flexbox to align forms → Apply background gradient → Set margins, padding → Center forms on page

3. Design inputs, labels → Add border-radius, shadows → Style placeholders, focus effects → Use clear text styling → Ensure label visibility

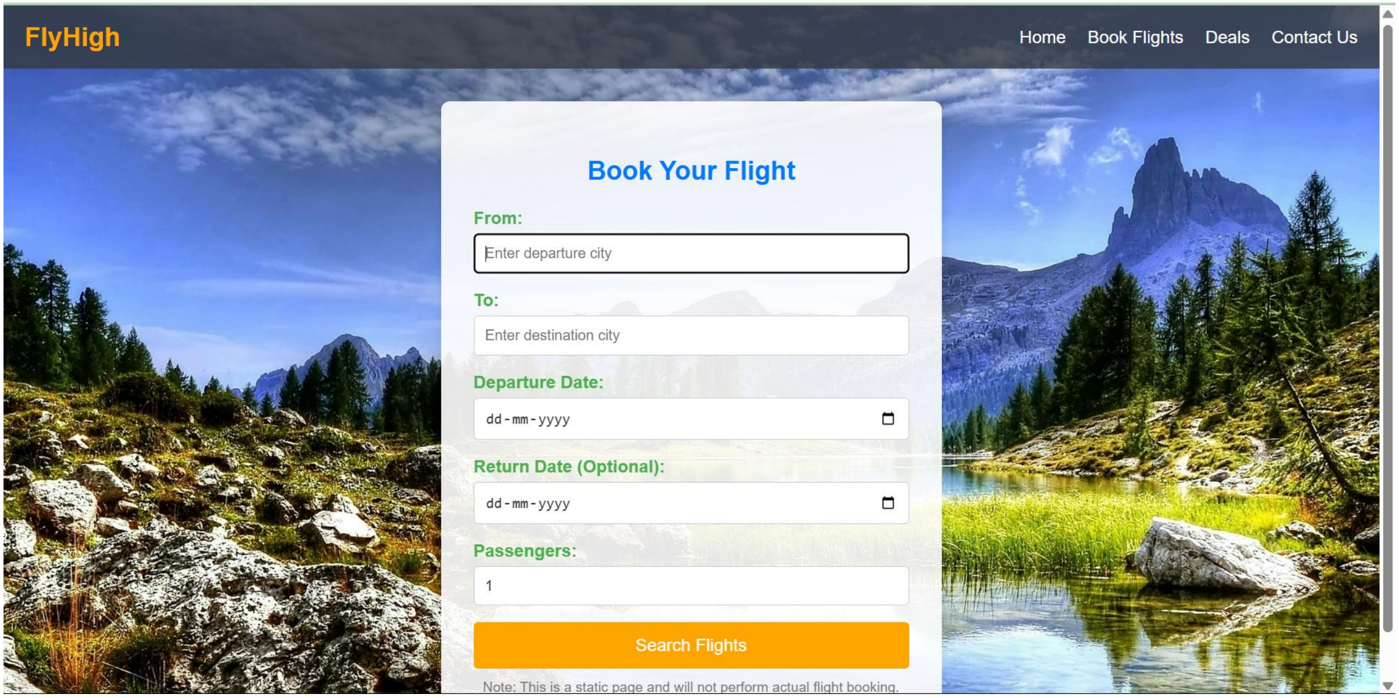
4. Style buttons with CSS3 → Apply gradient backgrounds → Add hover, click transitions → Set button text shadow → Ensure button contrast

5. Finalize design → Test responsiveness on devices → Adjust spacing, alignment → Verify form aesthetics → Save and review → Prepare for hosting

CODE:

```
<div class="booking-container">
    <h2>Book Your Flight</h2>
    <form action="#" method="get"> <div class="form-group">
        <label for="from">From:</label>
        <input type="text" id="from" name="from" placeholder="Enter departure city" required>
      </div>
      <div class="form-group">
        <label for="to">To:</label>
        <input type="text" id="to" name="to" placeholder="Enter destination city" required>
      </div>
      <div class="form-group">
        <label for="departure_date">Departure Date:</label>
        <input type="date" id="departure_date" name="departure_date" required>
      </div>
      <div class="form-group">
        <label for="return_date">Return Date (Optional):</label>
        <input type="date" id="return_date" name="return_date">
      </div>
      <div class="form-group">
        <label for="passengers">Passengers:</label>
        <select id="passengers" name="passengers">
          <option value="1">1</option>
          <option value="2">2</option>
          <option value="3">3</option>
          <option value="4">4</option>
          <option value="5">5+</option>
        </select>
      </div>
      <button type="submit">Search Flights</button>
    </form>
```

OUTPUT:



Book Your Flight

From:

Enter departure city

To:

Enter destination city

Departure Date:

dd-mm-yyyy

Return Date (Optional):

dd-mm-yyyy

Passengers:

1

Note: This is a static page and will not perform actual flight booking.

RESULT:

An online registration form for any real time application had been designed

Exp No : 6

Date :

VALIDATE THE VALUES OF VARIOUS FIELDS IN A REGISTRATION FORM USING JAVA SCRIPT

AIM:

To validate the values of various fields in a registration form using java script

LANGUAGE / TOOLS USED:

HTML,CSS, JavaScript

PROCEDURE:

1. Create HTML form → Add name, email, password, phone fields → Include submit button → Link CSS, JavaScript → Set form structure

2. Style form with CSS3 → Use flexbox for layout → Apply gradient background → Style inputs, labels

   → Add error message styling

3. Add JavaScript validation → Validate required fields → Check email format → Ensure password strength → Verify phone number

4. Display validation errors → Show error messages → Highlight invalid inputs → Clear errors on re-entry → Style valid inputs

5. Test form validation → Submit with empty fields → Test invalid inputs → Verify error messages → Ensure valid submission

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flight Ticket Booking</title>
    <style>
        body {
            font-family: sans-serif;
            background-color: #f4f4f4;
            display: flex;
```

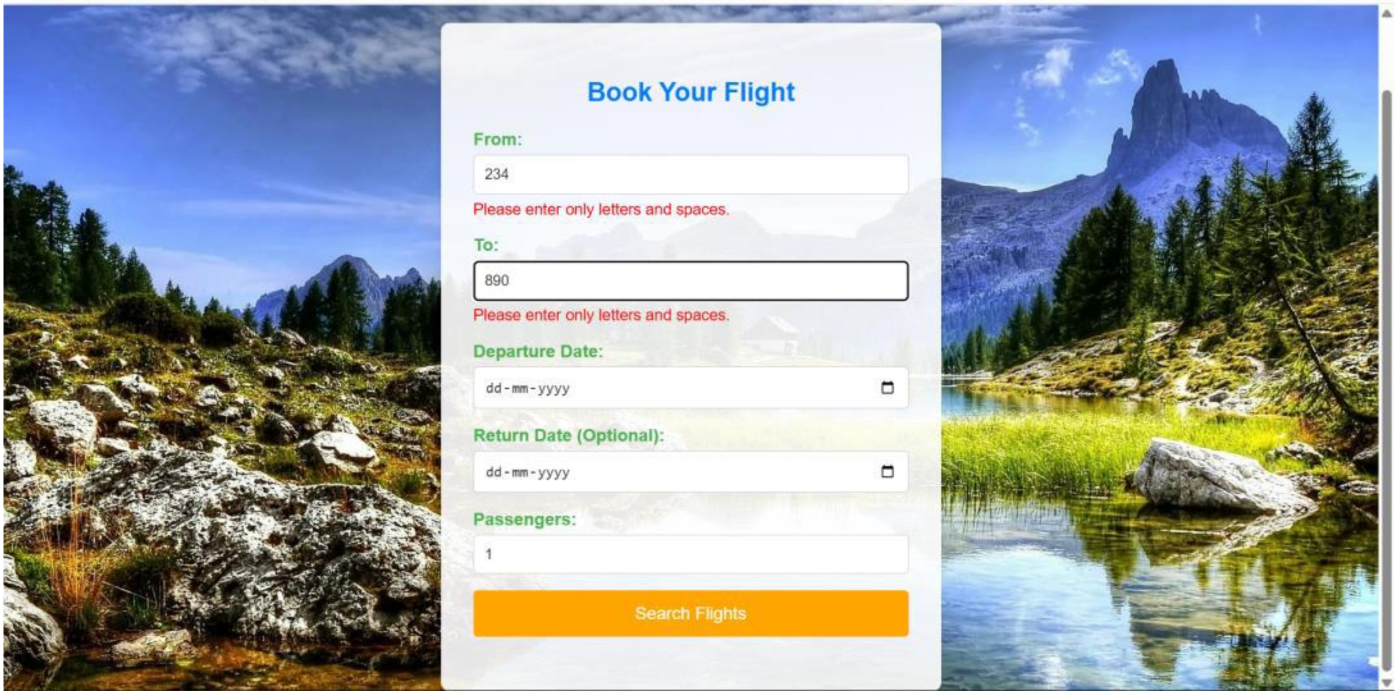
```
      flex-direction: column; /* Stack navigation and content */
      align-items: center;
      min-height: 100vh;
      margin: 0;
      /* 3D Flight Image Background */
      background-image: url('https://images.pexels.com/photos/371633/pexels-photo-371633.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=2');
      background-size: cover;
      background-repeat: no-repeat;
    }

    /* Navigation Bar Styles */
    nav {
      background-color: rgba(52, 58, 64, 0.8); /* Dark semi-transparent background */
      color: white;
      padding: 15px 20px;
      width: 100%;
      box-sizing: border-box;
      display: flex;
      justify-content: space-between;
      align-items: center;
    }

    nav .logo {
      font-size: 1.5em;
      font-weight: bold;
      color: rgb(255, 165, 0); /* Orange logo */
    }

    nav ul {
      list-style: none;
      padding: 0;
      margin: 0;
      display: flex;
    }

    nav ul li {
      margin-left: 20px;
    }

    nav ul li:first-child {
      margin-left: 0;
    }

    nav ul li a {
      color: white;
      text-decoration: none;
      transition: color 0.3s ease;
    }

    nav ul li a:hover {
      color: rgb(0, 123, 255); /* Blue on hover */
    }
```

```
     .booking-container {
        background-color: rgba(255, 255, 255, 0.9); /* Slightly transparent white */
        padding: 30px;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        width: 400px;
        margin-top: 30px; /* Add some space below the navbar */
    }

    h2 {
        text-align: center;
        margin-bottom: 20px;
        color: rgb(0, 123, 255); /* Blue */
    }

    .form-group {
        margin-bottom: 15px;
    }

    label {
        display: block;
        margin-bottom: 5px;
        color: rgb(76, 175, 80); /* Green */
        font-weight: bold;
    }

    input[type="text"],
    input[type="date"],
    select {
        width: 100%;
        padding: 10px;
        border: 1px solid rgb(204, 204, 204); /* Light Grey */
        border-radius: 4px;
        box-sizing: border-box;
        color: rgb(51, 51, 51); /* Dark Grey */
    }

    select {
        appearance: none;
        background-image: url('data:image/svg+xml;utf8,<svg fill="currentColor" viewBox="0 0 24
24"><pathd="M7 10l5 5 5-5z"/></svg>');
        background-repeat: no-repeat;
        background-position-x: 95%;
        background-position-y: center;
        background-size: 1em;
    }

    button {
        background-color: rgb(255, 165, 0); /* Orange */
        color: rgb(255, 255, 255); /* White */
        padding: 12px 20px;
        border: none;
        border-radius: 4px;
        cursor: pointer;
```

```
            font-size: 1em;
            width: 100%;
        }

        button:hover {
            background-color: rgb(255, 140, 0); /* Darker Orange */
        }

        .note {
            font-size: 0.8em;
            color: rgb(119, 119, 119); /* Grey */
            margin-top: 10px;
            text-align: center;
        }

        .error-message {
            color: red;
            font-size: 0.9em;
            margin-top: 5px;
        }
    </style>
</head>
<body>
    <nav>
        <div class="logo">FlyHigh</div>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Book Flights</a></li>
            <li><a href="#">Deals</a></li>
            <li><a href="#">Contact Us</a></li>
        </ul>
    </nav>

    <div class="booking-container">
        <h2>Book Your Flight</h2>
        <form id="flightBookingForm" action="#" method="get">
            <div class="form-group">
                <label for="from">From:</label>
                <input type="text" id="from" name="from" placeholder="Enter departure city" required
oninput="validateCity('from')">
                <div id="fromError" class="error-message"></div>
            </div>
            <div class="form-group">
                <label for="to">To:</label>
                <input type="text" id="to" name="to" placeholder="Enter destination city" required
oninput="validateCity('to')">
                <div id="toError" class="error-message"></div>
            </div>
            <div class="form-group">
                <label for="departure_date">Departure Date:</label>
                <input type="date" id="departure_date" name="departure_date" required>
            </div>
            <div class="form-group">
                <label for="return_date">Return Date (Optional):</label>
                <input type="date" id="return_date" name="return_date">
            </div>
            <div class="form-group">
                <label for="passengers">Passengers:</label>
                <select id="passengers" name="passengers">
                    <option value="1">1</option>
                    <option value="2">2</option>
                    <option value="3">3</option>
                    <option value="4">4</option>
                    <option value="5">5+</option>
                </select>
            </div>
            <button type="submit" onclick="handleSearch(event)" disabled>Search Flights</button>
        </form>
        <div id="searchResults" class="note" style="margin-top: 20px; text-align: left;">
            </div>
    </div>

    <script>
        const cityRegex = /^[a-zA-Z\s]+$/;
        const searchButton = document.querySelector('button[type="submit"]');
        let isFromValid = false;
        let isToValid = false;

        function validateCity(inputId) {
            const inputElement = document.getElementById(inputId);
            const errorDivId = inputId + 'Error';
            const errorDiv = document.getElementById(errorDivId);
            const inputValue = inputElement.value;

            if (!cityRegex.test(inputValue)) {
                errorDiv.textContent = 'Please enter only letters and spaces.';
                if (inputId === 'from') {
                    isFromValid = false;
                } else if (inputId === 'to') {
                    isToValid = false;
                }
            } else {
                errorDiv.textContent = '';
                if (inputId === 'from') {
                    isFromValid = true;
                } else if (inputId === 'to') {
                    isToValid = true;
                }
            }
            updateSearchButtonState();
        }

        function updateSearchButtonState() {
            searchButton.disabled = !(isFromValid && isToValid);
        }

        function handleSearch(event) {
            event.preventDefault(); // Prevent the default form submission
```

```
            if (!isFromValid || !isToValid) {
                alert('Please correct the city name formats.');
                return;
            }

            const fromCity = document.getElementById('from').value;
            const toCity = document.getElementById('to').value;
            const departureDate = document.getElementById('departure_date').value;
            const returnDate = document.getElementById('return_date').value;
            const passengers = document.getElementById('passengers').value;
            const searchResultsDiv = document.getElementById('searchResults');

            // For this static page, we'll just display the search parameters.
            // In a real application, you would send this data to a server
            // and process the flight search.

            let resultsHTML = "<h3>Flight Search Parameters:</h3><ul>";
            resultsHTML += `<li><strong>From:</strong> ${fromCity}</li>`;
            resultsHTML += `<li><strong>To:</strong> ${toCity}</li>`;
            resultsHTML += `<li><strong>Departure Date:</strong> ${departureDate}</li>`;
            if (returnDate) {
                resultsHTML += `<li><strong>Return Date:</strong> ${returnDate}</li>`;
            } else {
                resultsHTML += `<li><strong>Return Date:</strong> One-way</li>`;
            }
            resultsHTML += `<li><strong>Passengers:</strong> ${passengers}</li>`;
            resultsHTML += "</ul><p>Note: This is a static webpage, so no actual flight search is performed.</p>";

            searchResultsDiv.innerHTML = resultsHTML;
        }

        // Initial state of the search button
        updateSearchButtonState();
    </script>
</body>
</html>
```

OUTPUT:



Book Your Flight

From:

234

Please enter only letters and spaces.

To:

890

Please enter only letters and spaces.

Departure Date:

dd-mm-yyyy

Return Date (Optional):

dd-mm-yyyy

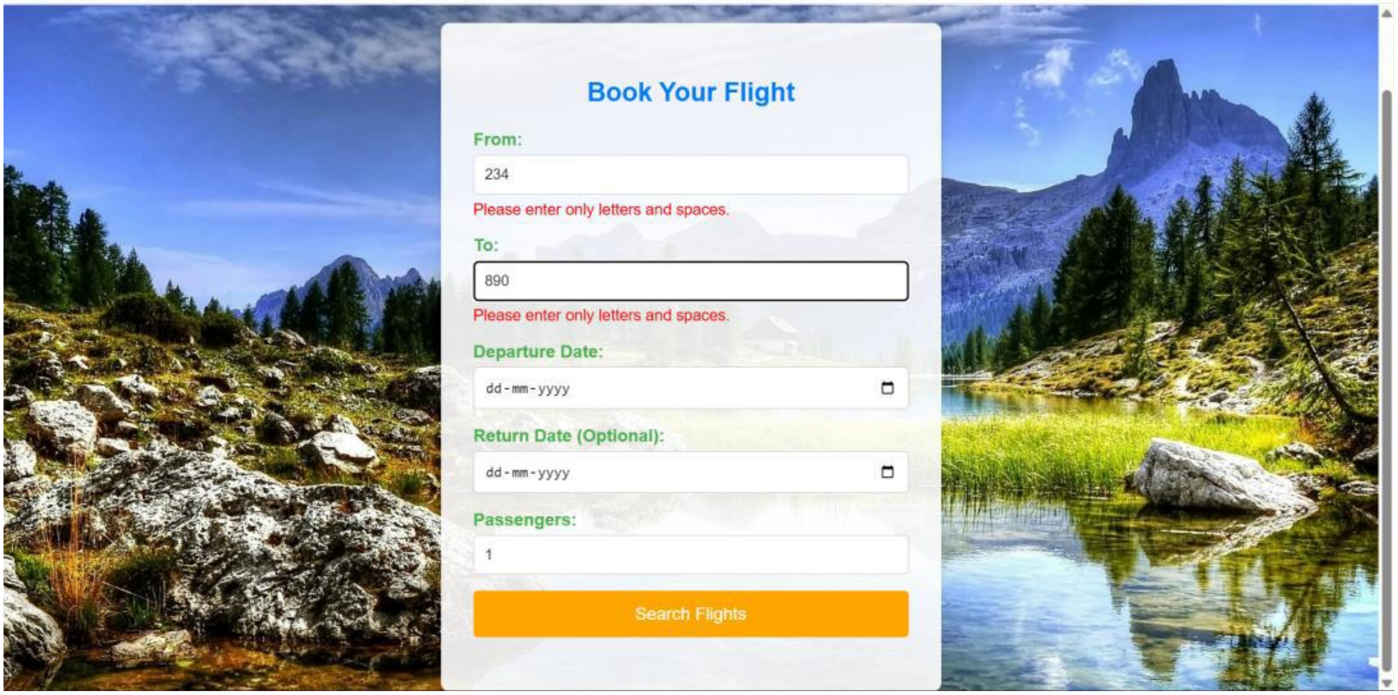
Passengers:

1

RESULT:

Validation of the values of various fields in a registration form using java script has been done.

OUTPUT:



Book Your Flight

From:

234

Please enter only letters and spaces.

To:

890

Please enter only letters and spaces.

Departure Date:

dd-mm-yyyy

Return Date (Optional):

dd-mm-yyyy

Passengers:

1

RESULT:

Validation of the values of various fields in a registration form using java script has been done.

EXP: 7. DEVELOP A SIMPLE DYNAMIC WEB APPLICATION TO RETRIEVE THE USER DETAILS FROM A WEB FORM AND DISPLAY THE SAME USING PHP

DATE:

AIM:

To build a simple dynamic web page that collects and displays user data using PHP.

TOOLS REQUIRED:

- XAMPP/WAMP (Apache server and PHP)
- Text editor (VS Code, Sublime Text)

PROCEDURE:

1. Open XAMPP and start the Apache server.
2. Create a folder in htdocs (e.g., user_form).
3. Inside the folder, create an HTML file (index.html) with a form to input name, email, etc.
4. Create a display.php file to retrieve data using $_POST or $_GET.
5. In the HTML form, set action="display.php" and method="post".
6. Use echo statements in display.php to print submitted data.
7. Test the form by accessing it via http://localhost/user_form/index.html.

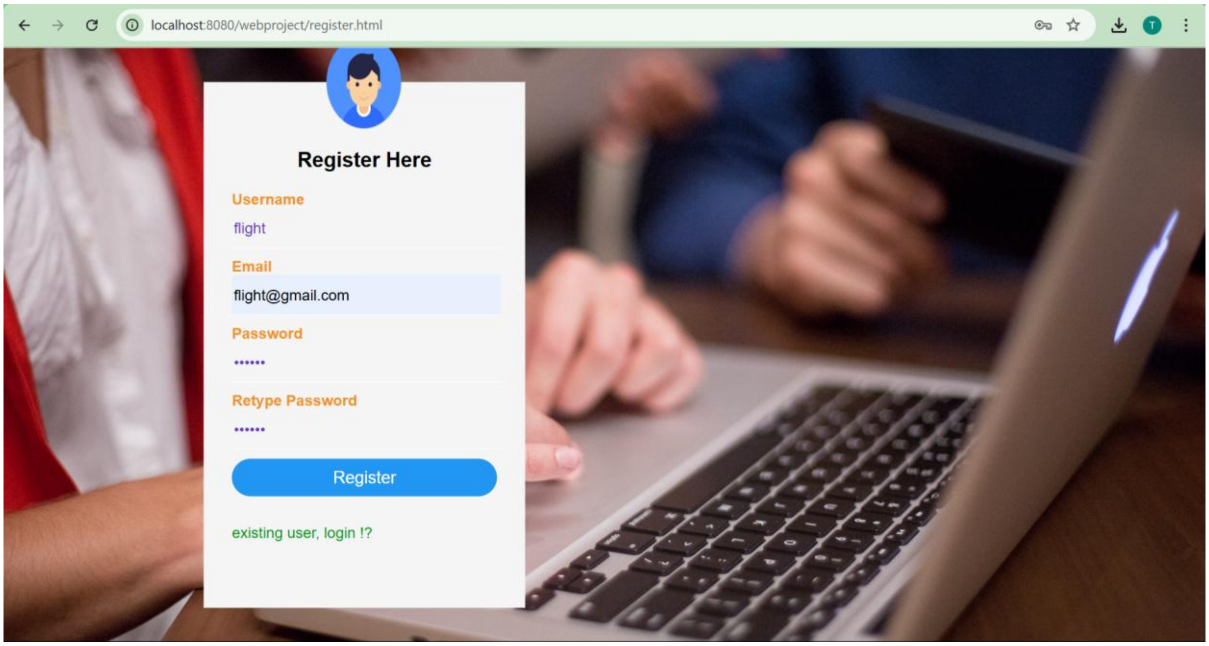
CODE :

register.html:

```
<form name="myform2" action="register.php" method="POST">
```

login.php:

```
<!DOCTYPE html>
<head>
<title>Register Form Design</title>
<link rel="stylesheet" type="text/css" href="style.css">

<body>

<div class="box">
<img src="user.png" class="user">

 <h1>Register Here</h1>

 <form name="myform2" action="register.php" method="POST">

     <p>Username</p>
     <input type="text" name="uname1" placeholder="Enter Username" required="">

     <p>Email</p>
     <input type="Email" name="email" placeholder="Enter email id" required="">

     <p>Password</p>
     <input type="password" name="upswd1" placeholder="Enter Password" required="">

     <p>Retype Password</p>
     <input type="password" name="upswd2" placeholder="Re-Enter Password" required="">

     <input type="submit" name="" value="Register">

     <br><br>
     <a href="index.html">existing user, login !?</a>
 </form>

</div>

</body>
</head>
</html>
```



Register Here

Username

flight

Email

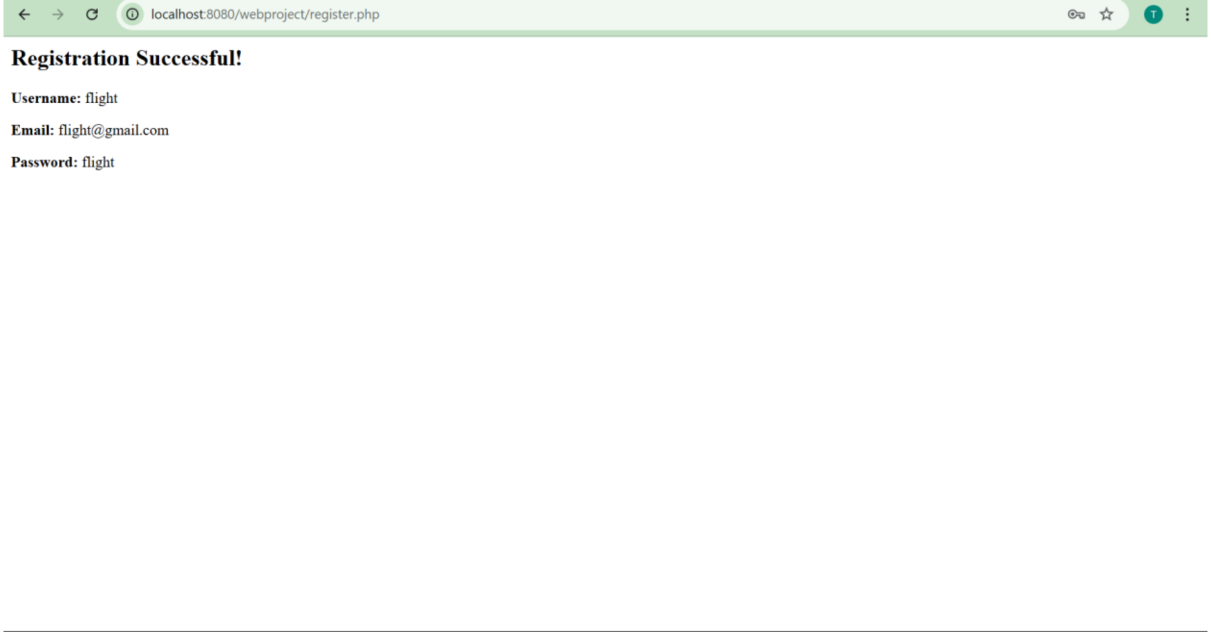
flight@gmail.com

Password

••••••

Retype Password

••••••



Registration Successful!

Username: flight

Email: flight@gmail.com

Password: flight

RESULT:

Thus the Develop a simple dynamic web application to retrieve the user details from a web form and display the same using Php are done successfully.

EXP: 8. CREATE A DATABASE WITH NECESSARY TABLES AND EXECUTE SQL QUERIES USING PHPMYADMIN AND MYSQL

DATE:

AIM:

To create a database and perform basic SQL operations using phpMyAdmin.

TOOLS REQUIRED:

- XAMPP/WAMP
- phpMyAdmin

PROCEDURE:

1. Open XAMPP and start Apache and MySQL.
2. Access phpMyAdmin via http://localhost/phpmyadmin.
3. Click “New” and create a new database (e.g., user_db).
4. Create a table (e.g., users) with fields like id, name, email.
5. Insert sample records manually or using SQL query:
6. INSERT INTO users (name, email) VALUES ('John Doe', 'john@example.com');
7. Execute SELECT, UPDATE, DELETE queries using the SQL tab.
8. View changes in the "Browse" section.

Code:

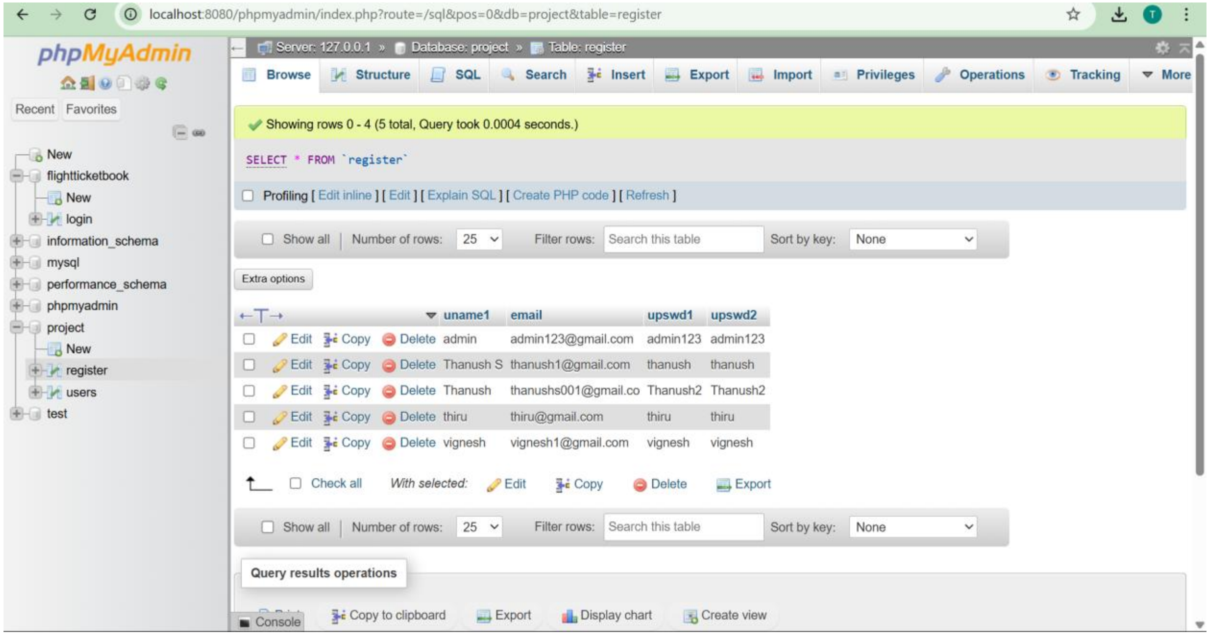
```php
<?php
$uname = $_POST['uname'];
$upswd = $_POST['upswd'];

$conn = new mysqli("localhost", "root", "", "project");
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT upswd1 FROM register WHERE uname1 = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("s", $uname);
$stmt->execute();
$stmt->store_result();

if ($stmt->num_rows > 0) {
    $stmt->bind_result($stored_password);
    $stmt->fetch();
    if ($upswd === $stored_password) {
        include("flyhigh.html");
    } else {
        echo "<h2>Invalid username or password.</h2>";
    }
} else {
    echo "<h2>Invalid username or password.</h2>";
}
```

```
$stmt->close();

$conn->close();

?>
```

Sql commands:

```
CREATE DATABASE IF NOT EXISTS project;

USE project;


CREATE TABLE IF NOT EXISTS register (
id INT AUTO_INCREMENT PRIMARY KEY,
uname1 VARCHAR(50) NOT NULL,
email VARCHAR(100) NOT NULL UNIQUE,
upswd1 VARCHAR(255) NOT NULL,
upswd2 VARCHAR(255) NOT NULL
);
```

OUTPUT:



localhost:8080/phpmyadmin/index.php?route=/sql&pos=0&db=project&table=register

Server: 127.0.0.1 » Database: project » Table: register

✓ Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

```
SELECT * FROM `register`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

				uname1	email	upswd1	upswd2
□	Edit	Copy	Delete	admin	admin123@gmail.com	admin123	admin123
□	Edit	Copy	Delete	Thanush S	thanush1@gmail.com	thanush	thanush
□	Edit	Copy	Delete	Thanush	thanushs001@gmail.co	Thanush2	Thanush2
□	Edit	Copy	Delete	thiru	thiru@gmail.com	thiru	thiru
□	Edit	Copy	Delete	vignesh	vignesh1@gmail.com	vignesh	vignesh

□ Check all *With selected:* Edit Copy Delete Export

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

RESULT:

Thus the Creation of database with necessary tables and execute SQL queries using phpMyAdmin and MySQL are done successfully.

EXP: 9. DEVELOP ANY REAL-TIME WEB APPLICATION USING PHP AND MYSQL

DATE:

AIM:

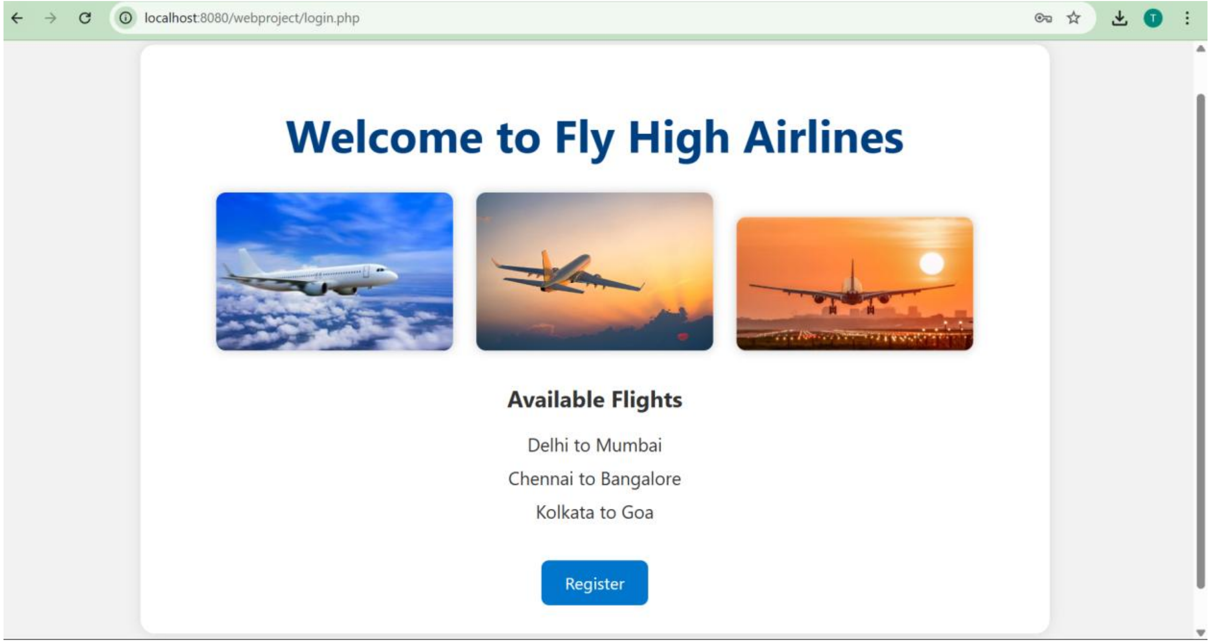
To build a basic real-time web application (e.g., feedback form) using PHP and MySQL.

TOOLS REQUIRED:

- XAMPP/WAMP
- phpMyAdmin
- Text Editor **PROCEDURE:**

1. Create a database (feedback_db) and table (feedback) in phpMyAdmin.
2. Design a web form (index.html) to collect feedback (name, message).
3. Create a submit.php to handle form submission and insert into MySQL using mysqli_connect() and INSERT query.
4. Create another page view.php to display all records using SELECT query.
5. Link pages and test functionality via http://localhost/project-folder.

Outputs:



localhost:8080/webproject/login.php

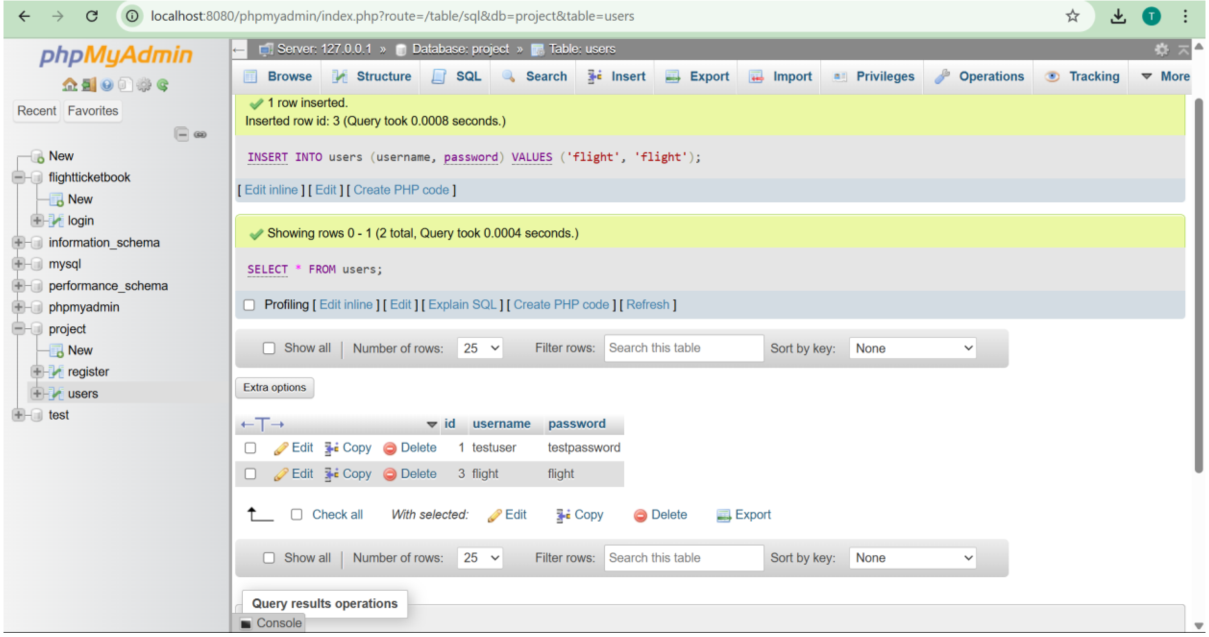
Welcome to Fly High Airlines

Available Flights

Delhi to Mumbai

Chennai to Bangalore

Kolkata to Goa



localhost:8080/phpmyadmin/index.php?route=/table/sql&db=project&table=users

✓ 1 row inserted.
Inserted row id: 3 (Query took 0.0008 seconds.)

```
INSERT INTO users (username, password) VALUES ('flight', 'flight');
```

[Edit inline] [Edit] [Create PHP code]

✓ Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
SELECT * FROM users;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	username	password
□ Edit Copy Delete	1	testuser	testpassword
□ Edit Copy Delete	3	flight	flight

□ Check all *With selected:* Edit Copy Delete Export

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

RESULT:

Thus The Development Any Real-Time Web Application Using Php and Mysql was done

EXP: 10. APPLY SESSION TRACKING IN PHP TO MANAGE USERS' SESSION IN A WEBSITE

DATE:

AIM:

To implement session tracking in PHP to retain user data across pages.

TOOLS REQUIRED:

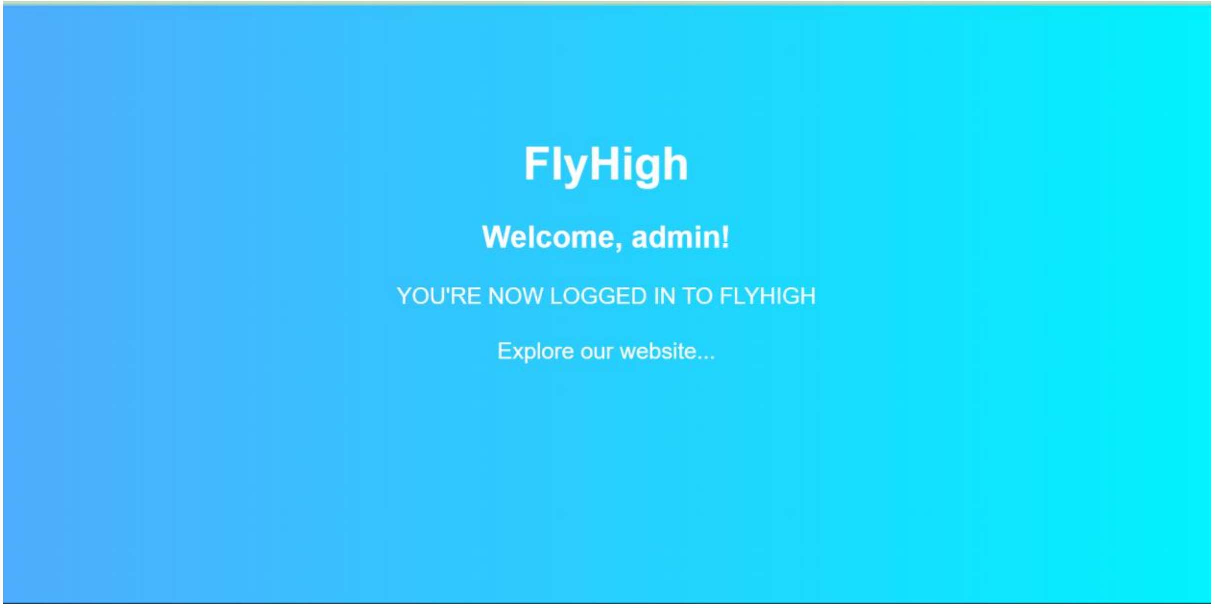
•XAMPP/WAMP

•Text Editor **PROCEDURE:**

1. Create a folder (e.g., session_demo) in htdocs.
2. In login.php, create a login form and use session_start() to initiate a session.
3. Upon successful login, store user data in session variables:
4. session_start();
5. $_SESSION['username'] = $_POST['username'];
6. In dashboard.php, retrieve session variables using $_SESSION['username'].
7. Add logout.php to destroy session using session_destroy().
8. Use session checks to prevent unauthorized access: 9. if (!isset($_SESSION['username'])) { header("Location: login.php"); }

CODE:

```
<?php
session_start();
$host = "localhost";
$dbname = "flyhigh";
$username = "root";
$password = "";

$conn = new mysqli($host, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $fullname = $conn->real_escape_string($_POST['fullname']);
    $email = strtolower(trim($conn->real_escape_string($_POST['email'])));
    $password_raw = $_POST['password'];
    $phone = $conn->real_escape_string($_POST['phone']);
    $address = $conn->real_escape_string($_POST['address']);
    $hashed_password = password_hash($password_raw, PASSWORD_DEFAULT);
    $check = $conn->prepare("SELECT id FROM users WHERE email = ?");
    $check->bind_param("s", $email);
    $check->execute();
    $check->store_result();

    if ($check->num_rows > 0) {
        echo "<h2>Email already registered!</h2>";
        echo "<p><a href='register.html'>← Go back</a></p>";
    } else {
        $stmt = $conn->prepare("INSERT INTO users (fullname, email, password, phone, address)
VALUES (?, ?, ?, ?, ?)");
        $stmt->bind_param("sssss", $fullname, $email, $hashed_password, $phone, $address);

        if ($stmt->execute()) {
            $_SESSION['user_id'] = $stmt->insert_id;
            $_SESSION['fullname'] = $fullname;
            $_SESSION['email'] = $email;
        echo "<h2>Registration successful!</h2>";
        echo "<p><a href='index.html'>Go to Home</a></p>";
      } else {
        echo "Error: Could not register user.";
      }
 $stmt->close();
   }
 $check->close();
}
$conn->close();
?>
```



FlyHigh

Welcome, admin!

YOU'RE NOW LOGGED IN TO FLYHIGH

Explore our website...

RESULT:

Thus the Application Session Tracking In Php To Manage Users’ Session In A Website are done.