İdil BAYAR 32267
Dilara ALKANALKA 30758

# CS 306- PROJECT PHASE TWO REPORT

## Introduction:

In this report, the newly added components to the SQL system and backhand codes will be explained. They are inspected by their functionality, how they affected the system and importance for the process. Furthermore, each component will be explained by their changes on the database system. The changes will be supported and visualized by the screenshots taken before and after of the executions. Lastly the css, php and html codes will be shown as well as trigger and procedure sql files.

## Trigger Implementation and Testing:

In this phase of the project we have added two trigger models; check_seat_availabilty and check_flight_time_overlap.

The aim of the "check_seat_availabilty" trigger is allowing or preventing one from operating one already booked seat. The process follows the steps:

1- Creates a boolean to store whether the seat is already in Belongs_to table
2- Traverse the table to detect any matching between the input variables and the table.
3- If there is a match, the boolean turns to true and an error is raised to alert the person that this seat is already taken.

```
CREATE TRIGGER check_seat_availability
BEFORE INSERT ON Belongs_to
FOR EACH ROW
BEGIN
    DECLARE seat_already_booked BOOLEAN;

    SELECT
        CASE
            WHEN EXISTS (SELECT 1 FROM Seats WHERE seat_number = NEW.seat_number AND f_id = NEW.f_id AND is_booked = TRUE) THEN TRUE
            ELSE FALSE
        END
    INTO seat_already_booked
    FROM dual;

    IF seat_already_booked THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Seat is already booked.';
    END IF;
END //
```

```
INSERT INTO Belongs_to (seat_number, f_id) VALUES ('12A', 2);
```

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| 1 | 01:55:01 | INSERT INTO Belongs_to (seat_number, f_id) VALUES ('12A', 2) | Error Code: 1644. Seat is already booked. | 0.093 sec |

It is observed that the example command resulted in an error since 12B is entered to Belongs_to table already.

The trigger “check_flight_time_overlap” ensures that, when a new flight is added, the plane does not have another overlapping flight given the certain time interval of the plane’s flight duration. This trigger secures the integrity of the flight schedule, and automates conflict detection while scheduling.

The trigger calculates the end time by using flight time and duration, then checks for conflicts from the Flights table by matching the p_id and raises an error if the conflict is found.

Below are the SQL scripts and the implementation of the check_flight_time_overlap trigger:

SQL File 1 | flight_reservation

Limit to 1000 rows

```
DELIMITER //

CREATE TRIGGER check_flight_time_overlap
BEFORE INSERT ON Flights
FOR EACH ROW
BEGIN
    DECLARE conflict_count INT;
    DECLARE new_end_time DATETIME;
    -- Calculate the end time of the new flight
    SET new_end_time = NEW.flight_date + INTERVAL TIME_TO_SEC(NEW.duration) SECOND;
    -- Check for overlapping flights with the same plane
    SELECT COUNT(*)
    INTO conflict_count
    FROM Flights
    WHERE p_id = NEW.p_id
      AND (
          (NEW.flight_date < (flight_date + INTERVAL TIME_TO_SEC(duration) SECOND))
          AND
          (flight_date < new_end_time)
      );

    IF conflict_count > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Flight schedule conflict detected for the plane.';
    END IF;
END //

DELIMITER ;
```

Flights table before and after the trigger implementation:

```
SELECT * FROM `cs306-flight_reservation`.flights;
```

| f_id | flight_date | gate_no | departure_airport | arrival_airport | duration | p_id |
|---|---|---|---|---|---|---|
| 1 | 2024-10-17 | A1 | CUR | MIA | 03:10:00 | AAL708 |
| 2 | 2024-10-17 | D6 | BAQ | MIA | 02:50:00 | AAL1124 |
| 3 | 2024-10-16 | E7 | MIA | PHL | 02:50:00 | AAL661 |
| 4 | 2024-10-17 | D22 | MIA | SFO | 06:15:00 | AAL2933 |
| 5 | 2024-10-16 | A2 | HOG | MIA | 01:45:00 | AAL2726 |
| 6 | 2024-10-17 | 43 | LAX | MIA | 05:10:00 | AAL1815 |
| 7 | 2024-10-16 | 11 | SJO | MIA | 03:05:00 | AAL1600 |
| 8 | 2024-10-16 | D43 | MIA | GND | 03:40:00 | AAL1546 |
| 9 | 2024-10-17 | A18 | DFW | MIA | 02:45:00 | AAL1405 |
| 10 | 2024-10-18 | D19 | MIA | KIN | 01:50:00 | AAL1400 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

```
SELECT * FROM `cs306-flight_reservation`.flights;
INSERT INTO Flights (f_id, flight_date, gate_no, departure_airport, arrival_airport, duration, p_id)
VALUES (11, '2024-10-17 02:30:00', 'D5', 'JFK', 'MIA', '02:00:00', 'AAL708');
```

| f_id | flight_date | gate_no | departure_airport | arrival_airport | duration | p_id |
|---|---|---|---|---|---|---|
| 1 | 2024-10-17 | A1 | CUR | MIA | 03:10:00 | AAL708 |
| 2 | 2024-10-17 | D6 | BAQ | MIA | 02:50:00 | AAL1124 |
| 3 | 2024-10-16 | E7 | MIA | PHL | 02:50:00 | AAL661 |
| 4 | 2024-10-17 | D22 | MIA | SFO | 06:15:00 | AAL2933 |
| 5 | 2024-10-16 | A2 | HOG | MIA | 01:45:00 | AAL2726 |
| 6 | 2024-10-17 | 43 | LAX | MIA | 05:10:00 | AAL1815 |
| 7 | 2024-10-16 | 11 | SJO | MIA | 03:05:00 | AAL1600 |
| 8 | 2024-10-16 | D43 | MIA | GND | 03:40:00 | AAL1546 |
| 9 | 2024-10-17 | A18 | DFW | MIA | 02:45:00 | AAL1405 |
| 10 | 2024-10-18 | D19 | MIA | KIN | 01:50:00 | AAL1400 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

| # | Time | Action | Message |
|---|---|---|---|
| 1 | 23:18:37 | SELECT * FROM `cs306-flight_reservation`.flights LIMIT 0, 1000 | 10 row(s) returned |
| 2 | 23:18:37 | INSERT INTO Flights (f_id, flight_date, gate_no, departure_airport, arrival_airport, duration, p_id) VALUES (11, '202... | Error Code: 1644. Flight schedule conflict detected for the plane. |

The example command resulted in an error since there is another flight for the plane with p_id=11 for the inserted duration of flying time.

SQL File 1 | flight_reservation | stored222 | flights

Limit to 1000 rows

```sql
INSERT INTO Flights (f_id, flight_date, gate_no, departure_airport, arrival_airport, duration, p_id)
VALUES (12, '2024-10-19 06:30:00', 'D7', 'MIA', 'ATL', '01:30:00', 'AAL708');
SELECT * FROM `cs306-flight_reservation`.flights;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| f_id | flight_date | gate_no | departure_airport | arrival_airport | duration | p_id |
|---|---|---|---|---|---|---|
| 2 | 2024-10-17 | D6 | BAQ | MIA | 02:50:00 | AAL1124 |
| 3 | 2024-10-16 | E7 | MIA | PHL | 02:50:00 | AAL661 |
| 4 | 2024-10-17 | D22 | MIA | SFO | 06:15:00 | AAL2933 |
| 5 | 2024-10-16 | A2 | HOG | MIA | 01:45:00 | AAL2726 |
| 6 | 2024-10-17 | 43 | LAX | MIA | 05:10:00 | AAL1815 |
| 7 | 2024-10-16 | 11 | SJO | MIA | 03:05:00 | AAL1600 |
| 8 | 2024-10-16 | D43 | MIA | GND | 03:40:00 | AAL1546 |
| 9 | 2024-10-17 | A18 | DFW | MIA | 02:45:00 | AAL1405 |
| 10 | 2024-10-18 | D19 | MIA | KIN | 01:50:00 | AAL1400 |
| 12 | 2024-10-19 | D7 | MIA | ATL | 01:30:00 | AAL708 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

flights 5

Output

Action Output

| # | Time | Action | Message |
|---|---|---|---|
| 1 | 23:25:57 | INSERT INTO Flights (f_id, flight_date, gate_no, departure_airport, arrival_airport, duration, p_id) VALUES (12, '202... | 1 row(s) affected, 1 warning(s): 1265 Data t |
| 2 | 23:25:57 | SELECT * FROM `cs306-flight_reservation`.flights LIMIT 0, 1000 | 11 row(s) returned |

## Stored Procedure Implementation:

We have implemented two stored procedure implementations; SearchFlights and book_seat.

The aim of the “book_seat” procedure is to successfully book an available seat from the Seats table and update the required table status. The procedure happens as following:

1- The seat existence and availability is checked
2- If the seat is bookable, the status of is_booked converted to true for specified seat in Seats table
3- Then the new booking is added to Belongs_to table
4- The transactions are committed sequentially with the end of procedures.

```sql
CREATE PROCEDURE book_seat(
    IN p_seat_number VARCHAR(10),
    IN p_flight_id INT,
    OUT p_status VARCHAR(100)
)
BEGIN
    -- Declare variables for checking
    DECLARE seat_exists INT;
    DECLARE is_booked BOOLEAN;

    -- Check if seat exists for the flight
    SELECT COUNT(*) INTO seat_exists
    FROM Seats
    WHERE seat_number = p_seat_number AND f_id = p_flight_id;


IF seat_exists = 0 THEN
    SET p_status = 'Error: Seat does not exist for this flight';
ELSE
    -- Check if seat is already booked
    SELECT is_booked INTO is_booked
    FROM Seats
    WHERE seat_number = p_seat_number AND f_id = p_flight_id;

    IF is_booked = TRUE THEN
        SET p_status = 'Error: Seat is already booked';
    ELSE
```

```
        IF is_booked = TRUE THEN
            SET p_status = 'Error: Seat is already booked';
        ELSE
            START TRANSACTION;
            UPDATE Seats
            SET is_booked = TRUE
            WHERE seat_number = p_seat_number AND f_id = p_flight_id;
            INSERT INTO Belongs_to (seat_number, f_id)
            VALUES (p_seat_number, p_flight_id);
            -- If we get here, both operations succeeded
            COMMIT;
            SET p_status = 'Success: Seat booked successfully';
        END IF;
    END IF;
END //
```

```
SET @status = '';
CALL book_seat('A1', 101, @status);
SELECT @status AS booking_status;
```

Result Grid | Filter Rows:

| booking_status |
| --- |
| Error: Seat does not exist for this flight |

As it can be seen from the procedure piece that the flight doesn't own an A1 seat so that the variables raised an error to alert the person.

The "SearchFlights" stored procedure retrieves flights based on the departure airport, arrival airport, and flight date. This allows users to efficiently search for available flights matching their criteria.
Below are the SQL scripts of the stored procedures and their retrieval examples:

```
DELIMITER $$
DROP PROCEDURE IF EXISTS SearchFlights $$
CREATE PROCEDURE SearchFlights(
    IN dep_airport VARCHAR(50),   -- Input: Departure airport
    IN arr_airport VARCHAR(50),   -- Input: Arrival airport
    IN flight_date DATE           -- Input: Flight date
)
BEGIN
    -- Select flights based on the input parameters
    SELECT f_id, flight_date, gate_no, departure_airport, arrival_airport, duration
    FROM Flights
    WHERE departure_airport = dep_airport
    AND arrival_airport = arr_airport
    AND flight_date = flight_date;
END $$

DELIMITER ;
```

```
CALL SearchFlights('CUR', 'MIA', '2024-10-17');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| f_id | flight_date | gate_no | departure_airport | arrival_airport | duration |
|---|---|---|---|---|---|
| 1 | 2024-10-17 | A1 | CUR | MIA | 03:10:00 |

## Web Access Module:

There are two web access modules for users to interact with the database, one for checking whether a flight is available by entering departure and arrival airports with the flight date.Users then may check whether there are available unbooked seats on the flight that they have found.

For booking:

This html file is creating the web interference for the php file to be seen with the combination of css file. It creates two interactive input blocks for the user to enter their desired seat and flight id. This information is available under the available seats table. When the booking is successful,

the specified seat is erased from the table. Depending on the choice, system can raise the following messages:

1- Successful booking
2- Not available seat
3- There is no such seat for that flight

Before selecting the seat:

**Book a Flight Seat**

Success: Seat booked successfully!
Seat Number:
1A

Flight ID:
1

Book Seat

**Available Seats**

| Seat Number | Flight ID | Price | Flight Date | Gate | Departure | Arrival |
|---|---|---|---|---|---|---|
| 1A | 1 | $500.00 | 2024-10-17 | A1 | CUR | MIA |
| 12B | 2 | $220.00 | 2024-10-17 | D6 | BAQ | MIA |
| 12C | 2 | $220.00 | 2024-10-17 | D6 | BAQ | MIA |

After booking:

**Book a Flight Seat**

Success: Seat booked successfully!
Seat Number:

Flight ID:

Book Seat

**Available Seats**

| Seat Number | Flight ID | Price | Flight Date | Gate | Departure | Arrival |
|---|---|---|---|---|---|---|
| 12B | 2 | $220.00 | 2024-10-17 | D6 | BAQ | MIA |
| 12C | 2 | $220.00 | 2024-10-17 | D6 | BAQ | MIA |

For seat table:

Before booking:

| Seat Number | Flight ID | Is Booked | Price | Near Emergency Exit |
|---|---|---|---|---|
| 12A | 2 | Yes | $220.00 | No |
| 12B | 2 | No | $220.00 | No |
| 12C | 2 | No | $220.00 | No |
| 13A | 3 | Yes | $180.00 | Yes |
| 13B | 3 | Yes | $180.00 | Yes |
| 13C | 3 | Yes | $180.00 | Yes |
| 14A | 4 | Yes | $100.00 | No |
| 15A | 5 | Yes | $300.00 | Yes |
| 1A | 1 | No | $500.00 | Yes |
| 1B | 1 | Yes | $500.00 | Yes |

After booking:

| Seat Number | Flight ID | Is Booked | Price | Near Emergency Exit |
|---|---|---|---|---|
| 12A | 2 | Yes | $220.00 | No |
| 12B | 2 | No | $220.00 | No |
| 12C | 2 | No | $220.00 | No |
| 13A | 3 | Yes | $180.00 | Yes |
| 13B | 3 | Yes | $180.00 | Yes |
| 13C | 3 | Yes | $180.00 | Yes |
| 14A | 4 | Yes | $100.00 | No |
| 15A | 5 | Yes | $300.00 | Yes |
| 1A | 1 | Yes | $500.00 | Yes |
| 1B | 1 | Yes | $500.00 | Yes |

For flight checking:

The web interface allows users to search for flights by entering departure and arrival airports with the flight date.The results based on the form submission are gathered from the database and displayed on the result page.
Below are the screenshots of the input form, insertion confirmation and data view.

localhost/flight_search/flight_search.html

YouTube Gmail SUCoursehttp://suc... mySU bann

## Search Flights

Departure Airport:

Arrival Airport:

Flight Date (YYYY-MM-DD): gg.aa.yyyy

Search Flights

localhost/flight_search/flight_search.html

YouTube Gmail SUCoursehttp://suc... mySU

## Search Flights

Departure Airport: CUR

Arrival Airport: MIA

Flight Date (YYYY-MM-DD): 19.10.2024

Search Flights

localhost/flight_search/search_flights.php

YouTube Gmail SUCoursehttp://suc... mySU

Input Date: 2024-10-19

Departure Airport: CUR

Arrival Airport: MIA

No flights found for the given parameters.

localhost/flight_search/search_flights.php

YouTube Gmail SUCoursehttp://suc... mySU bannerweb.sabanci...

Input Date: 2024-10-17

Departure Airport: CUR

Arrival Airport: MIA

**Flights Found:**

| Flight ID | Departure Airport | Arrival Airport | Flight Date | Duration | Gate Number |
|---|---|---|---|---|---|
| 1 | CUR | MIA | 2024-10-17 | 03:10:00 | A1 |

Back-end HTML and PHP codes:

```html
<!DOCTYPE html>
<html>
<head>
    <title>Flight Seat Booking System</title>
    <link rel="stylesheet" href="seat.css">
</head>
<body>
    <div class="container">
        <h2>Book a Flight Seat</h2>

        <?php if ($message) echo $message; ?>

        <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
            <div class="form-group">
                <label for="seat_number">Seat Number:</label>
                <input type="text" id="seat_number" name="seat_number" required>
            </div>

            <div class="form-group">
                <label for="flight_id">Flight ID:</label>
                <input type="number" id="flight_id" name="flight_id" required>
            </div>

            <input type="submit" value="Book Seat">
        </form>

        <h3>Available Seats</h3>
        <table>
```

```
<h3>Available Seats</h3>
<table>
    <tr>
        <th>Seat Number</th>
        <th>Flight ID</th>
        <th>Price</th>
        <th>Flight Date</th>
        <th>Gate</th>
        <th>Departure</th>
        <th>Arrival</th>
    </tr>
    <?php
    if ($available_seats) {
        while($row = $
    ?>
    <tr class="clickab
        <td><?php echo
        <td><?php echo htmlspecialchars($row['f_id']); ?></td>
        <td>$<?php echo htmlspecialchars($row['price']); ?></td>
        <td><?php echo htmlspecialchars($row['flight_date']); ?></td>
        <td><?php echo htmlspecialchars($row['gate_no']); ?></td>
        <td><?php echo htmlspecialchars($row['departure_airport']); ?></td>
        <td><?php echo htmlspecialchars($row['arrival_airport']); ?></td>
    </tr>
    <?php
        endwhile;
    }
```

Convert special characters to HTML entities

```
htmlspecialchars( string $string [, int $flags = ENT_COMPAT | ENT_HTML401 [, string $encoding =
ini_get("default_charset") [, bool $double_encode ]]]): string
```

```
            <td><?php echo htmlspecialchars($row['arrival_airport']); ?></td>
        </tr>
        <?php
            endwhile;
        }
        ?>
    </table>
</div>

<script>
    function fillForm(seatNumber, flightId) {
        document.getElementById('seat_number').value = seatNumber;
        document.getElementById('flight_id').value = flightId;
    }
</script>
</body>
</html>
```

For seat table:

This html file shows the seat table variables in a table format.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flight Seats Information</title>
    <link rel="stylesheet" href="table.css">
</head>
<body>
    <div class="container">
        <h1>Flight Seats Information</h1>

        <div class="table-container">
            <h2>Seats Table</h2>
            <table>
                <thead>
                    <tr>
                        <th>Seat Number</th>
                        <th>Flight ID</th>
                        <th>Is Booked</th>
                        <th>Price</th>
                        <th>Near Emergency Exit</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    if ($seats_result) {
                        while($row = mysqli_fetch_assoc($seats_result)) {
```

```
    </thead>
    <tbody>
        <?php
        if ($seats_result) {
            while($row = mysqli_fetch_assoc($seats_result)) {
                echo "<tr>";
                echo "<td>" . $row["seat_number"] . "</td>";
                echo "<td>" . $row["f_id"] . "</td>";
                echo "<td class='" . ($row["is_booked"] ? "true" : "false") . "'>" . ($row["is_booked"] ?
                echo "<td>$" . number_format($row["price"], 2) . "</td>";
                echo "<td class='" . ($row["near_emergency_exit"] ? "true" : "false") . "'>" . ($row["nea
                echo "</tr>";
            }
        } else {
            echo "<tr><td colspan='5'>Error retrieving seats data: " . mysqli_error($db) . "</td></tr>";
        }
        ?>
    </tbody>
</table>
iv>
```

The php files give the dynamic procedure to the database systems. There are two php files for the seat checking and booking process which obtain data from the system and put it on the web server. One for seat availability check and one for the current conditions of the seat table itself.

There is another php file for the flight checking which checks whether there exists a flight based on given conditions.

For displaying seat table:

```php
<?php
// Include the configuration file.
require_once 'config.php';


// Query to get seats data
$seats_query = "SELECT * FROM Seats";
$seats_result = mysqli_query($db, $seats_query);

?>
```

For evaluating desired seat from available seat table:

```php
<?php
include 'config.php';

$message = '';
$status = '';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $seat_number = $_POST['seat_number'];
    $flight_id = $_POST['flight_id'];

    // First check if the seat exists and is available
    $check_query = "SELECT is_booked FROM Seats WHERE seat_number = ? AND f_id = ?";
    $check_stmt = $db->prepare($check_query);
    $check_stmt->bind_param("si", $seat_number, $flight_id);
    $check_stmt->execute();
    $result = $check_stmt->get_result();

    if ($result->num_rows === 0) {
        $message = '<div style="color: red;">Error: Seat does not exist for this flight.</div>';
    } else {
        $seat_data = $result->fetch_assoc();
        if ($seat_data['is_booked']) {
            $message = '<div style="color: red;">Error: This seat is already booked.</div>';
        } else {
            // Start transaction
            $db->begin_transaction();
```

```php
        try {
            // Update the seat status
            $update_query = "UPDATE Seats SET is_booked = TRUE WHERE seat_number = ? AND f_id = ?";
            $update_stmt = $db->prepare($update_query);
            $update_stmt->bind_param("si", $seat_number, $flight_id);
            $update_stmt->execute();

            // Add to Belongs_to table
            $belongs_query = "INSERT INTO Belongs_to (seat_number, f_id) VALUES (?, ?)";
            $belongs_stmt = $db->prepare($belongs_query);
            $belongs_stmt->bind_param("si", $seat_number, $flight_id);
            $belongs_stmt->execute();

            // If we got here, commit the transaction
            $db->commit();
            $message = '<div style="color: green;">Success: Seat booked successfully!</div>';
        } catch (Exception $e) {
            // If there was an error, roll back the transaction
            $db->rollback();
            $message = '<div style="color: red;">Error: ' . $e->getMessage() . '</div>';
        }

        if (isset($update_stmt)) $update_stmt->close();
        if (isset($belongs_stmt)) $belongs_stmt->close();
    }
```

```php
    $check_stmt->close();
}

// Query to get all available seats with additional information
$available_seats_query = "
    SELECT
        s.seat_number,
        s.f_id,
        s.price,
        s.is_booked,
        f.departure_airport,
        f.arrival_airport,
        f.flight_date,
        f.gate_no
    FROM Seats s
    JOIN Flights f ON s.f_id = f.f_id
    WHERE s.is_booked = FALSE
    ORDER BY f.flight_date, s.f_id, s.seat_number";

$available_seats = $db->query($available_seats_query);

if (!$available_seats) {
    $message = '<div style="color: red;">Error retrieving available seats: ' . $db->error . '</div>';
}
?>
```

For flight checking process:

flight_search.html × search_flights.php config.php

C: > xampp > htdocs > flight_search > flight_search.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flight Search</title>
</head>
<body>
    <h2>Search Flights</h2>
    <form method="POST" action="search_flights.php">
        <label for="departure_airport">Departure Airport:</label>
        <input type="text" id="departure_airport" name="departure_airport" required>
        <br><br>

        <label for="arrival_airport">Arrival Airport:</label>
        <input type="text" id="arrival_airport" name="arrival_airport" required>
        <br><br>

        <label for="flight_date">Flight Date (YYYY-MM-DD):</label>
        <input type="date" id="flight_date" name="flight_date" required>

        <br><br>

        <button type="submit">Search Flights</button>
    </form>
</body>
</html>
```

flight_search.html | search_flights.php | config.php

C: > xampp > htdocs > flight_search > search_flights.php

```php
<?php
include "config.php";
$form_submitted = false;

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $departure_airport = $_POST['departure_airport']; // departure airport
    $arrival_airport = $_POST['arrival_airport']; // arrival airport
    $input_date = $_POST['flight_date']; // flight date in YYYY-MM-DD format

    $form_submitted = true;

    echo "<p>Input Date: " . $input_date . "</p>";
    echo "<p>Departure Airport: " . $departure_airport . "</p>";
    echo "<p>Arrival Airport: " . $arrival_airport . "</p>";

    $sql = "SELECT * FROM Flights WHERE flight_date = '$input_date'
            AND departure_airport = '$departure_airport'
            AND arrival_airport = '$arrival_airport'";

    $result = mysqli_query($db, $sql);

    if (!$result) {
        die("Query failed: " . mysqli_error($db));
    }

    if (mysqli_num_rows($result) > 0) {
        echo "<h3>Flights Found:</h3>";
        echo "<table border='1'>
                <tr>
                    <th>Flight ID</th>
                    <th>Departure Airport</th>
                    <th>Arrival Airport</th>
                    <th>Flight Date</th>
                    <th>Duration</th>
                    <th>Gate Number</th>
```

C: > xampp > htdocs > flight_search > config.php

```php
<?php
$host = "localhost";
$user = "root"; // Default username for XAMPP
$password = ""; // Default password for XAMPP is empty
$dbname = "cs306-flight_reservation";

$db = mysqli_connect($host, $user, $password, $dbname);

if (!$db) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

```php
        echo "<table border='1'>
                <tr>
                    <th>Flight Date</th>
                    <th>Duration</th>
                    <th>Gate Number</th>
                </tr>";


        while ($row = mysqli_fetch_assoc($result)) {
            echo "<tr>
                    <td>" . $row['f_id'] . "</td>
                    <td>" . $row['departure_airport'] . "</td>
                    <td>" . $row['arrival_airport'] . "</td>
                    <td>" . $row['flight_date'] . "</td>
                    <td>" . $row['duration'] . "</td>
                    <td>" . $row['gate_no'] . "</td>
                </tr>";
        }

        echo "</table>";
    } else {
        echo "<p>No flights found for the given parameters.</p>";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flight Search</title>
</head>
```

C: > xampp > htdocs > flight_search > search_flights.php

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flight Search</title>
</head>
<body>


    <?php if (!$form_submitted) { ?>
        <form method="POST" action="">
            <label for="departure_airport">Departure Airport:</label>
            <input type="text" id="departure_airport" name="departure_airport" required>
            <br><br>

            <label for="arrival_airport">Arrival Airport:</label>
            <input type="text" id="arrival_airport" name="arrival_airport" required>
            <br><br>

            <label for="flight_date">Flight Date (YYYY-MM-DD):</label>
            <input type="date" id="flight_date" name="flight_date" required>

            <br><br>

            <button type="submit">Search Flights</button>
        </form>
    <?php } ?>
</body>
</html>
```