‘Shark Ticket’

Chosen Problem Statement: Develop a secure Ticketing and Aircraft Tracking system

"Like a shark gliding effortlessly through the ocean's expanse, aviation cuts through the sky with the same determined grace, embodying the spirit of exploration and unfaltering progress."

Introduction & Motivation:

We chose this topic because different websites use different layouts to reserve tickets for flights, this made us realize that some websites have better layouts than others which leads us to use that website more. We aim to create a website that is easy to navigate and secure so that the users can easily book websites. The website will be made using the best practices we observe from each of the websites. Incorporating the best from each website will help in optimization of the website and accessing the application will be easier for the customers.

Meet The Team:

1. Arun U S (22N208)
2. Lohith S (22N228)
3. Mehul Dinesh (22N232)
4. Richard Samuel D (22N242)
5. Saminathan (22N247)

Where do we stand out ?

The airline tracking system employs cutting-edge tracking technologies, harnessing real-time data analytics and predictive modeling to enhance operational efficiency and safety. Features like dynamic seat selection have also been implemented.

The airline tracking system significantly elevates operational efficiency for airlines by providing real-time insights into aircraft positions, weather conditions, and air traffic. This empowers airlines to make informed decisions that minimize delays and optimize routes, translating to punctuality, fuel savings, and ultimately, an improved travel experience for passengers.this project addresses crucial pain points in both customer-facing and operational aspects of air travel, making it an invaluable asset for the aviation industry.

Why did we choose the technologies we used?

Why Python?

1. **Existence of Strong Security Libraries:** Python offers robust security libraries and frameworks like cryptography, enabling seamless implementation of encryption and secure data handling protocols. This helps safeguard sensitive user information and payment data in the ticketing system.

2. **Code Readability:** Python's clean and readable syntax promotes writing secure code by reducing the chances of human error, making it easier to avoid vulnerabilities like SQL injection and cross-site scripting (XSS).

3. **Community-Driven Security:** Python has an active community that continuously works on identifying and addressing security issues, providing timely updates and patches to enhance project security.

4. **Integration Capabilities:** Python seamlessly interfaces with other languages and technologies, making it flexible to connect with secure payment gateways, external APIs, and authentication systems.

5. **Rapid Development:** Python's simplicity accelerates development cycles, crucial when implementing new security measures or making changes to accommodate evolving security requirements.

6. **Scalability:** Python offers both scripting and object-oriented paradigms, enabling developers to choose the best approach for different parts of the project, ensuring flexibility for future expansion.

Why did we choose mySQL for the backend?

- **Reliability:** MySQL is known for its stable performance in managing application data.

- **Performance:** With optimized read and write operations, MySQL ensures efficient handling of data even in high-traffic applications.

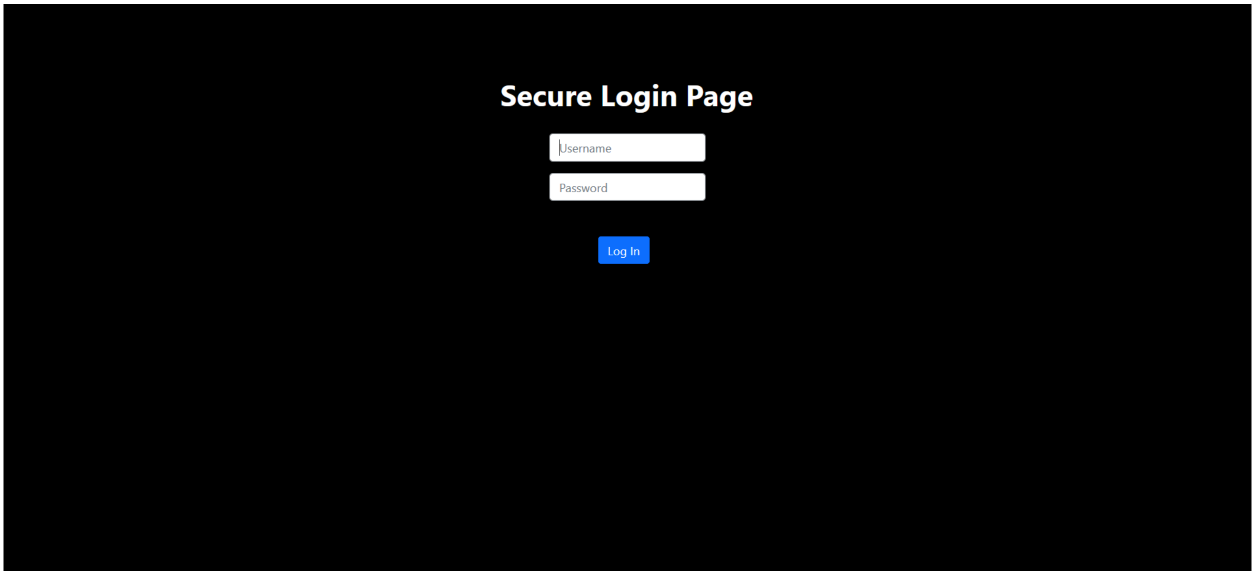
- **Scalability:** MySQL supports scaling strategies like replication, clustering, for accommodating growth.

- **Community and Support:** A large and active community provides resources and expertise for troubleshooting and best practices due to being Open Sourced.

Refer to the Appendix 1.3 for the mySQL initialization and implementation.

Plan/Design:

Login:

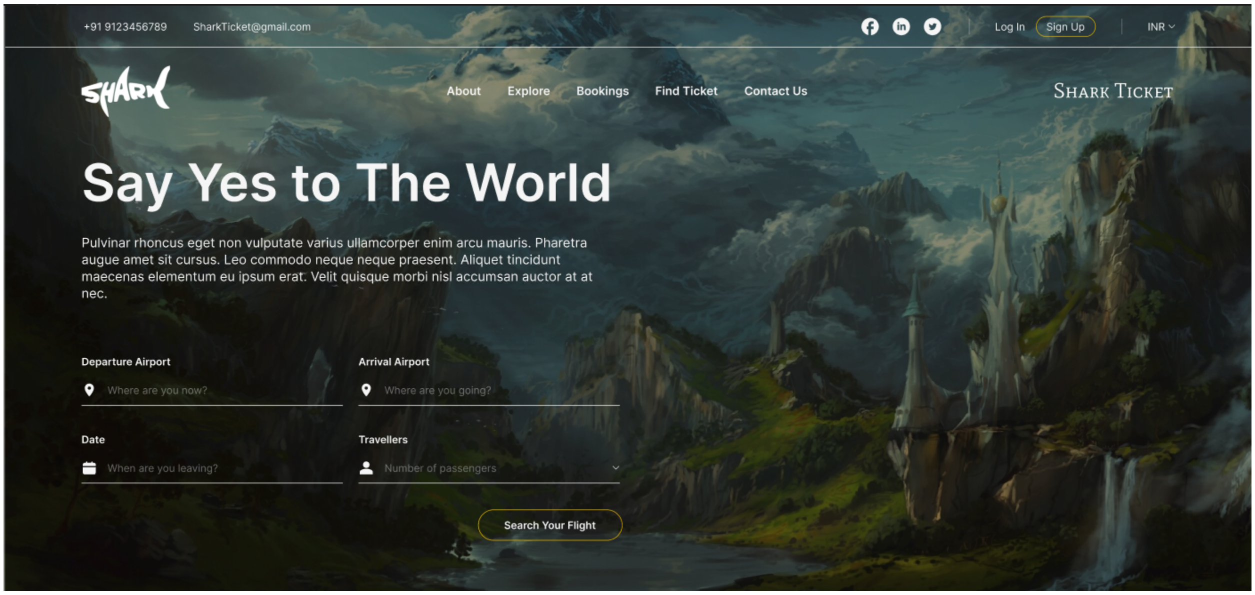


Secure Login Page

A minimalistic Login page to ensure no distractions during login and sign up, since these parts can be boring and could lead to the user switching to a different site mid process.

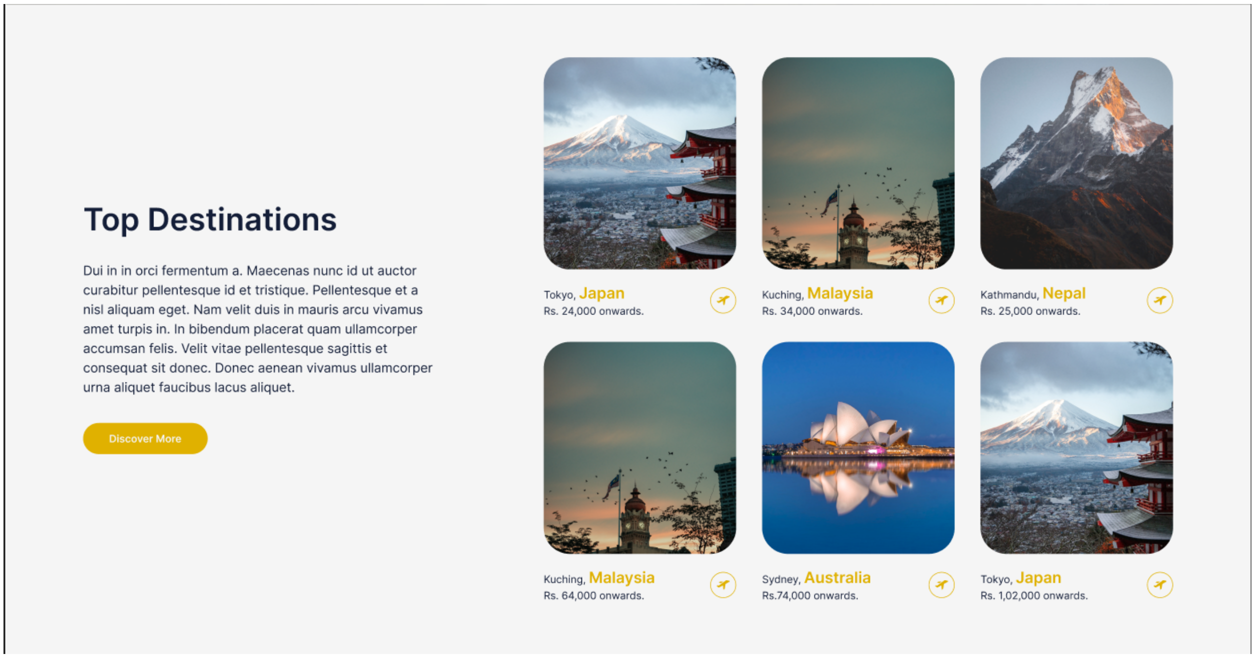
Refer to Appendix 1.2 for the source code of the Login Page

Homepage:



Say Yes to The World

Pulvinar rhoncus eget non vulputate varius ullamcorper enim arcu mauris. Pharetra augue amet sit cursus. Leo commodo neque neque praesent. Aliquet tincidunt maecenas elementum eu ipsum erat. Velit quisque morbi nisl accumsan auctor at at nec.



Top Destinations

Dui in in orci fermentum a. Maecenas nunc id ut auctor curabitur pellentesque id et tristique. Pellentesque et a nisl aliquam eget. Nam velit duis in mauris arcu vivamus amet turpis in. In bibendum placerat quam ullamcorper accumsan felis. Velit vitae pellentesque sagittis et consequat sit donec. Donec aenean vivamus ullamcorper urna aliquet faucibus lacus aliquet.

Tokyo, Japan
Rs. 24,000 onwards.

Kuching, Malaysia
Rs. 34,000 onwards.

Kathmandu, Nepal
Rs. 25,000 onwards.

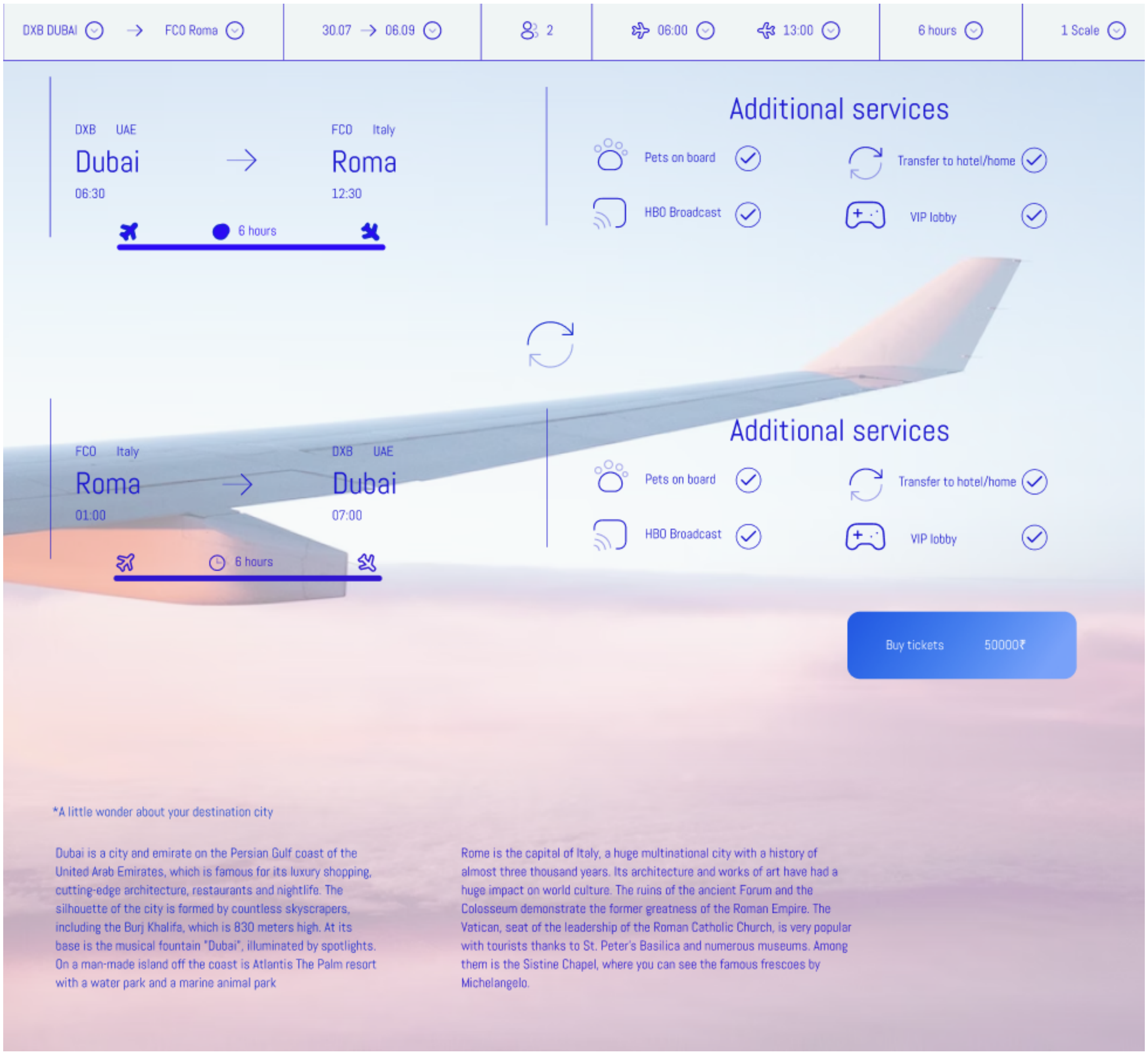
Kuching, Malaysia
Rs. 64,000 onwards.

Sydney, Australia
Rs.74,000 onwards.

Tokyo, Japan
Rs. 1,02,000 onwards.

The homepage creates the first impression of the website, so it was made to please the eyes of the user, using state of the art design, the user retentivity is maintained. Giving the user options along with the pricing for their next travel destination in the homepage along with visually appealing pictures for those destinations helps the user decide on their next destination, which can lead to an increase in the number of bookings on the website.

Ticket Booking UI:

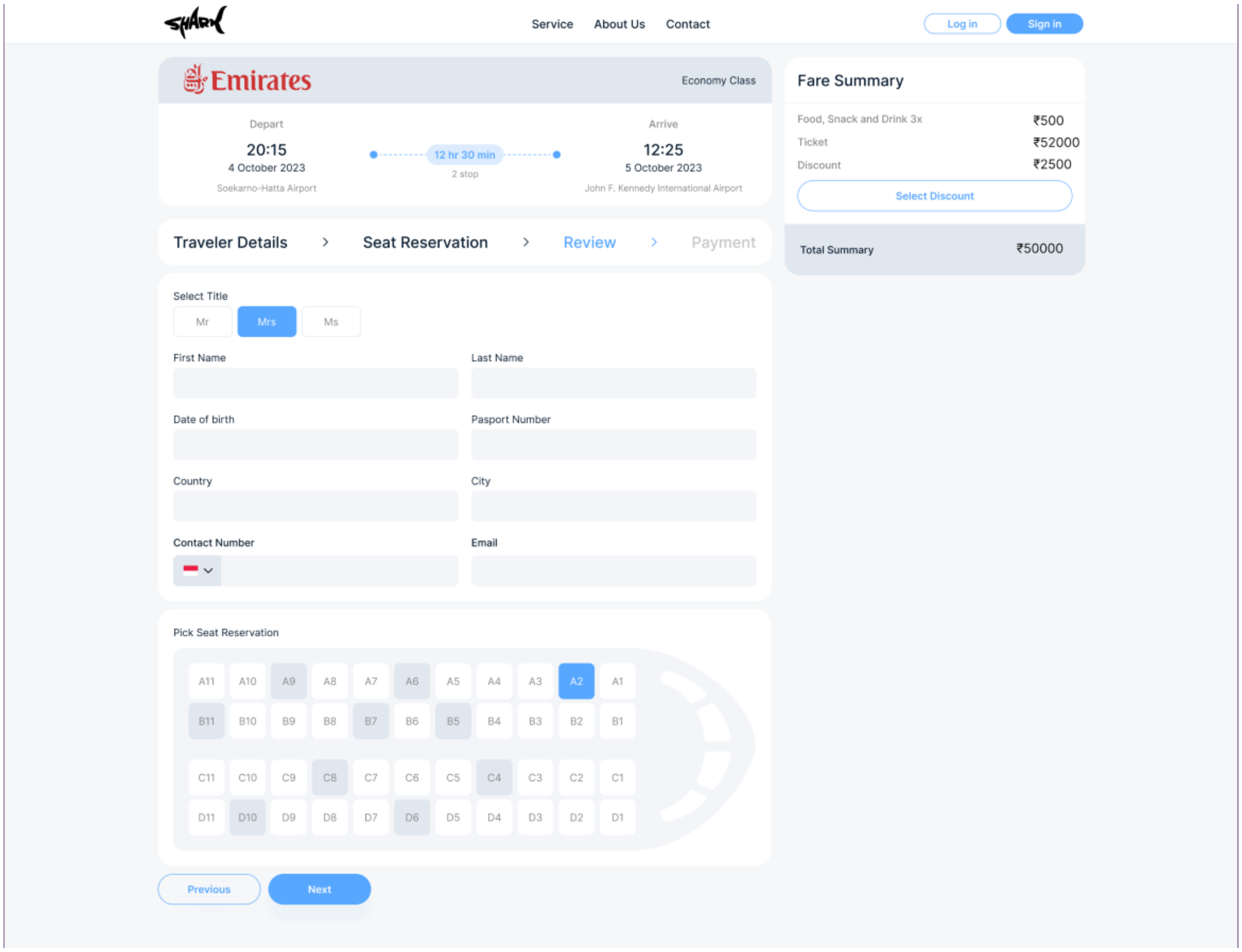


*A little wonder about your destination city

Dubai is a city and emirate on the Persian Gulf coast of the United Arab Emirates, which is famous for its luxury shopping, cutting-edge architecture, restaurants and nightlife. The silhouette of the city is formed by countless skyscrapers, including the Burj Khalifa, which is 830 meters high. At its base is the musical fountain "Dubai", illuminated by spotlights. On a man-made island off the coast is Atlantis The Palm resort with a water park and a marine animal park

Rome is the capital of Italy, a huge multinational city with a history of almost three thousand years. Its architecture and works of art have had a huge impact on world culture. The ruins of the ancient Forum and the Colosseum demonstrate the former greatness of the Roman Empire. The Vatican, seat of the leadership of the Roman Catholic Church, is very popular with tourists thanks to St. Peter's Basilica and numerous museums. Among them is the Sistine Chapel, where you can see the famous frescoes by Michelangelo.

There is also a major emphasis on the design of the flight listings, the user spends a lot of time on choosing the right flight to go to their destination, ensuring that the user has a blissful experience, the design choices were made. Furthermore, quick facts about the city being traveled to are located in the bottom of the page, this creates a personalized experience for the user and gives positives about a location, which can help the user make their choice on the city they are traveling to.



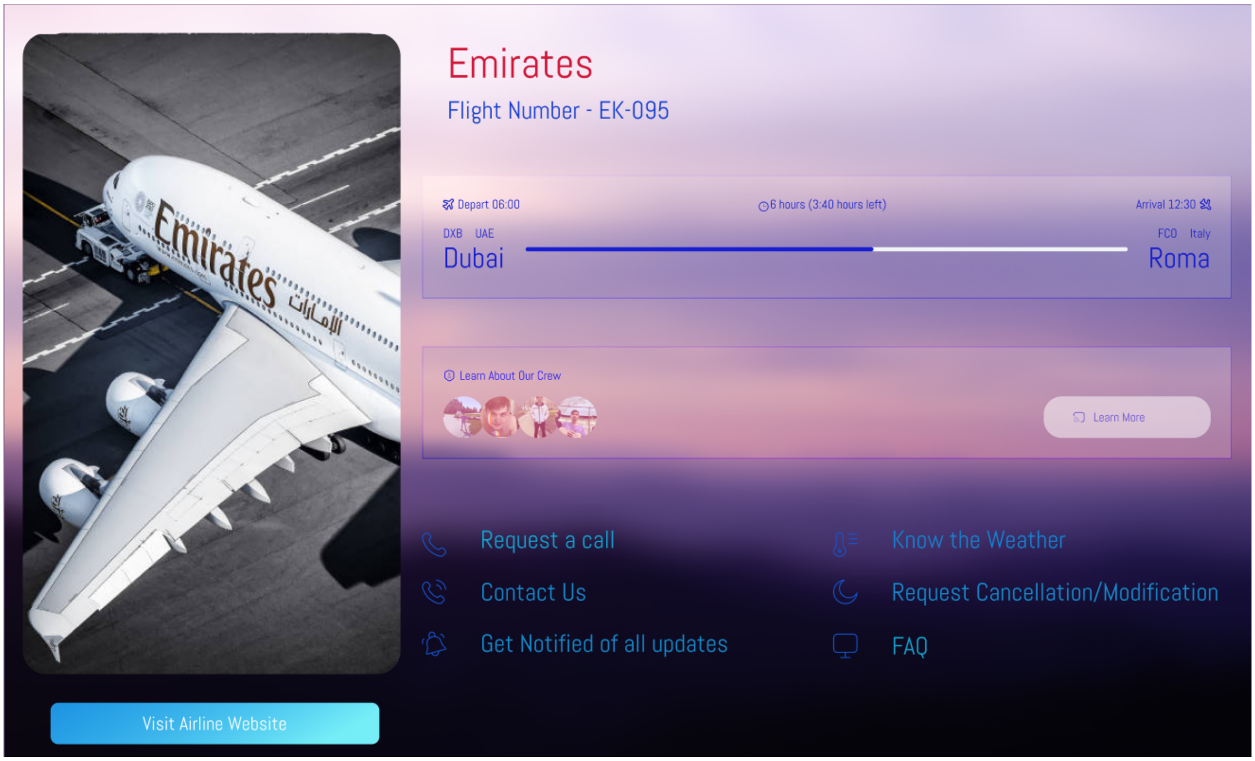
Food, Snack and Drink 3x	₹500
Ticket	₹52000
Discount	₹2500
Total Summary	₹50000

A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1
C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1

A simple yet effective UI for the booking page ensures quickness in checking out, as well as ease in booking. This is crucial to inviting users to book from our website the next time as well.

A fare summary is given alongside when the ticket is booked so that the user is aware of the cost throughout the process. This gives the user the confidence that no hidden charges are being added. Complete transparency gains the users trust on our software.

Flight Tracking UI:


Emirates

Flight Number - EK-095

Depart 06:00 6 hours (3:40 hours left) Arrival 12:30

DXB UAE Dubai FCO Italy Roma

A beautifully designed flight tracker is what the traveler’s loved ones would wait on during the travel, when the site is visually appealing it creates a sense of trust with the users, furthermore, waiting on this website for tracking would encourage them to browse through and book tickets for their own journey around the world.

More elements such as more facts about popular destinations on the tracking page could increase user interaction and retentivity while they are waiting on the flight. A small box allowing them to know more about the crew assigned will create a deeper connection between the airline and the user.

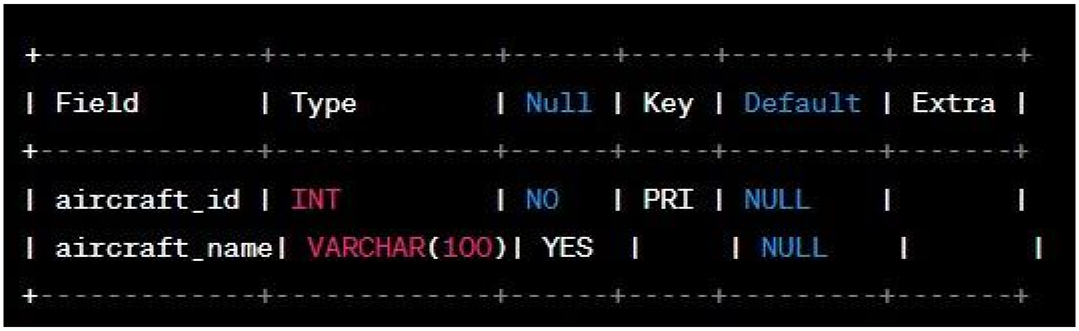
The overall design aspects of the website emphasizes user experience and state of the art UI design. This design ensures the user is able to navigate through all parts of the website seamlessly. The aesthetically pleasing design of the flight list, homepage, and the flight tracking page ensures user retentivity.

Working:

The General idea of the working is to store data in a MySQL, the database design will contain 4 general tables,

1. **Aircraft Details**:

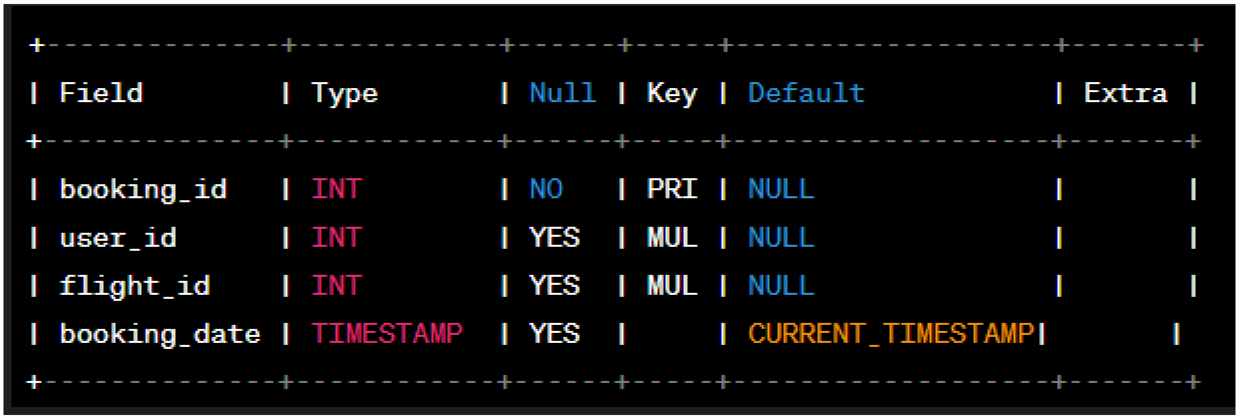
    This table contains information about the airline operating and the aircraft's ID.



```
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| aircraft_id | INT          | NO   | PRI | NULL    |       |
| aircraft_name| VARCHAR(100)| YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
```

2. **Booking details:**

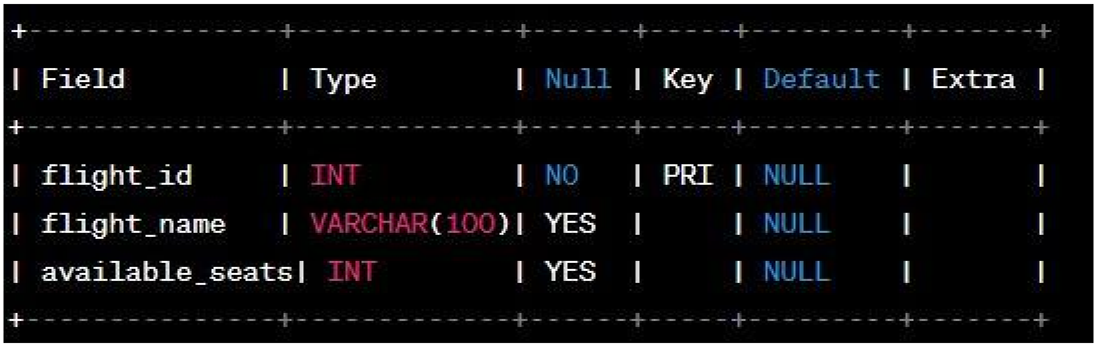
    This table contains important information on flight bookings and enables tracking of the bookings, it holds the Booking ID, The ID of the user making the booking, The flight's ID, and the date of booking.



```
+--------------+------------+------+-----+-------------------+-------+
| Field        | Type       | Null | Key | Default           | Extra |
+--------------+------------+------+-----+-------------------+-------+
| booking_id   | INT        | NO   | PRI | NULL              |       |
| user_id      | INT        | YES  | MUL | NULL              |       |
| flight_id    | INT        | YES  | MUL | NULL              |       |
| booking_date | TIMESTAMP  | YES  |     | CURRENT_TIMESTAMP|       |
+--------------+------------+------+-----+-------------------+-------+
```

3. **Availability**:

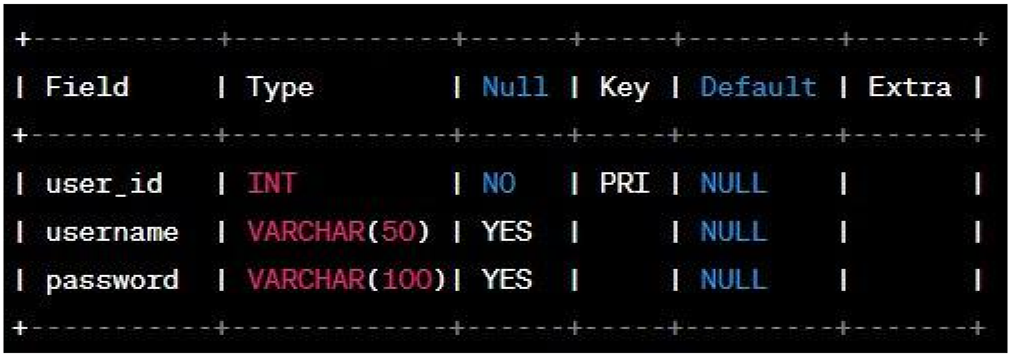
For each plane, an availability table is created, where the flight's ID, name and the number of seats available to book is stored.



```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| flight_id     | INT         | NO   | PRI | NULL    |       |
| flight_name   | VARCHAR(100)| YES  |     | NULL    |       |
| available_seats| INT        | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

4. **User Account Table:**

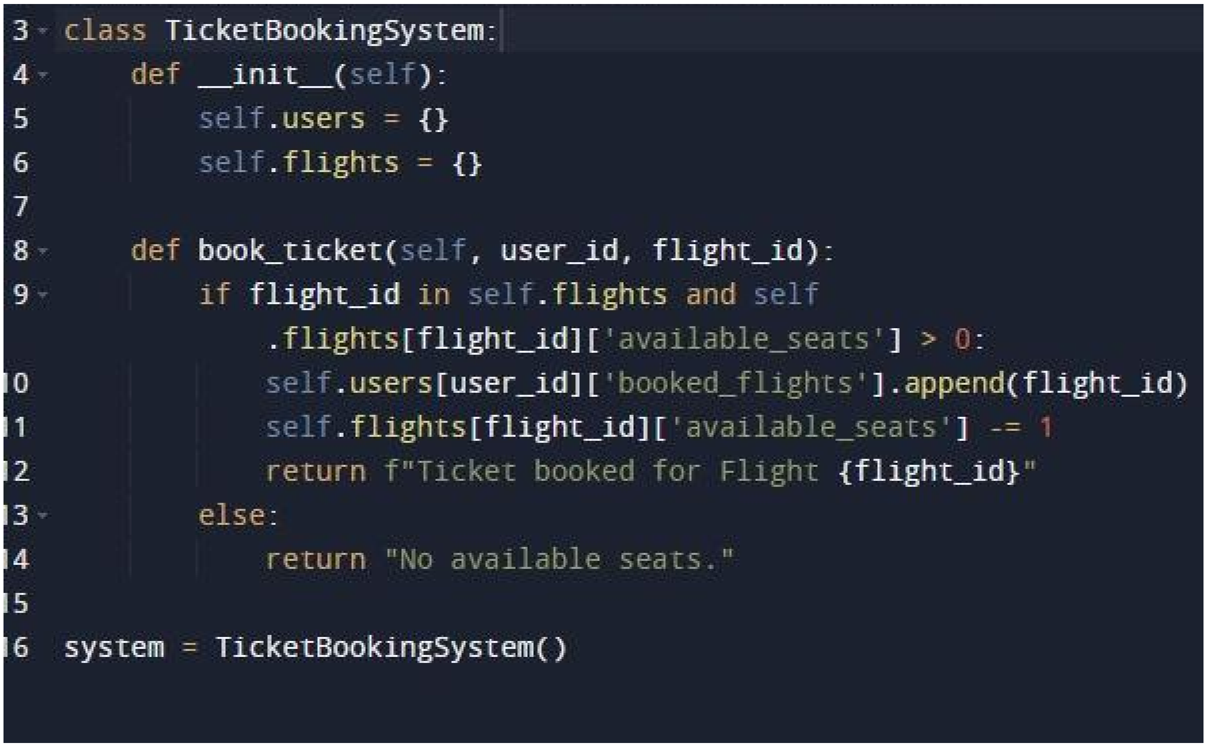
This table stores details about the user, the username, password and the user_id - generated by the website to keep track of the user.



```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| user_id  | INT         | NO   | PRI | NULL    |       |
| username | VARCHAR(50) | YES  |     | NULL    |       |
| password | VARCHAR(100)| YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

Refer to Appendix 1.1 for code used to create the tables.

Mechanism for Ticket Booking:



```python
class TicketBookingSystem:
    def __init__(self):
        self.users = {}
        self.flights = {}

    def book_ticket(self, user_id, flight_id):
        if flight_id in self.flights and self
            .flights[flight_id]['available_seats'] > 0:
            self.users[user_id]['booked_flights'].append(flight_id)
            self.flights[flight_id]['available_seats'] -= 1
            return f"Ticket booked for Flight {flight_id}"
        else:
            return "No available seats."

system = TicketBookingSystem()
```

After user authentication, the availability of the seats is checked. If available, then the seat is booked under the book_ticket function which takes in the parameters user_id and flight_id. Once the booking is done, the users dashboard is updated and the number of seats is also updated in the mainframe.

Security:

In order to make the ticketing software more secure the following procedures are followed:

- Multi-factor authentication (MFA) systems are employed so that only authorized users can access the system.
- Sensitive data, including user details, payment information, and booking records, both in transit and at rest are encrypted using industry-standard encryption protocols.
- Role-based access control (RBAC) are employed to ensure that users can only access the features and data that they are authorized to.
- Payments, comply with Payment Card Industry Data Security Standard (PCI DSS) requirements and use secure payment gateways.
- All input from users is sanitized to prevent SQL injection, cross-site scripting (XSS), and other common vulnerabilities.
- Regular Security Audits and Penetration Testing is conducted to proactively identify and address vulnerabilities before they are exploited.

For employing a secure aircraft tracking system:

- Communication between aircraft and the tracking system is encrypted and uses secure protocols to prevent unauthorized access.
- Intrusion detection and prevention systems to monitor for any suspicious activity or unauthorized access attempts.
- Physical access to the tracking system's infrastructure is made secure to prevent unauthorized individuals from gaining direct access to sensitive equipment.
- Real-time Anomaly Detection algorithms to identify unusual flight behavior and potential security threats promptly and inform the respected authorities.

Conclusion:

In conclusion, the development and implementation of this software represents a significant leap forward in enhancing the safety, efficiency, and overall experience of modern air travel. The overall design aspects of the website also play a crucial role, as they emphasize user experience and state-of-the-art UI design. By seamlessly integrating advanced security measures with real-time tracking capabilities, this project addresses crucial aspects of aviation management and passenger well-being.

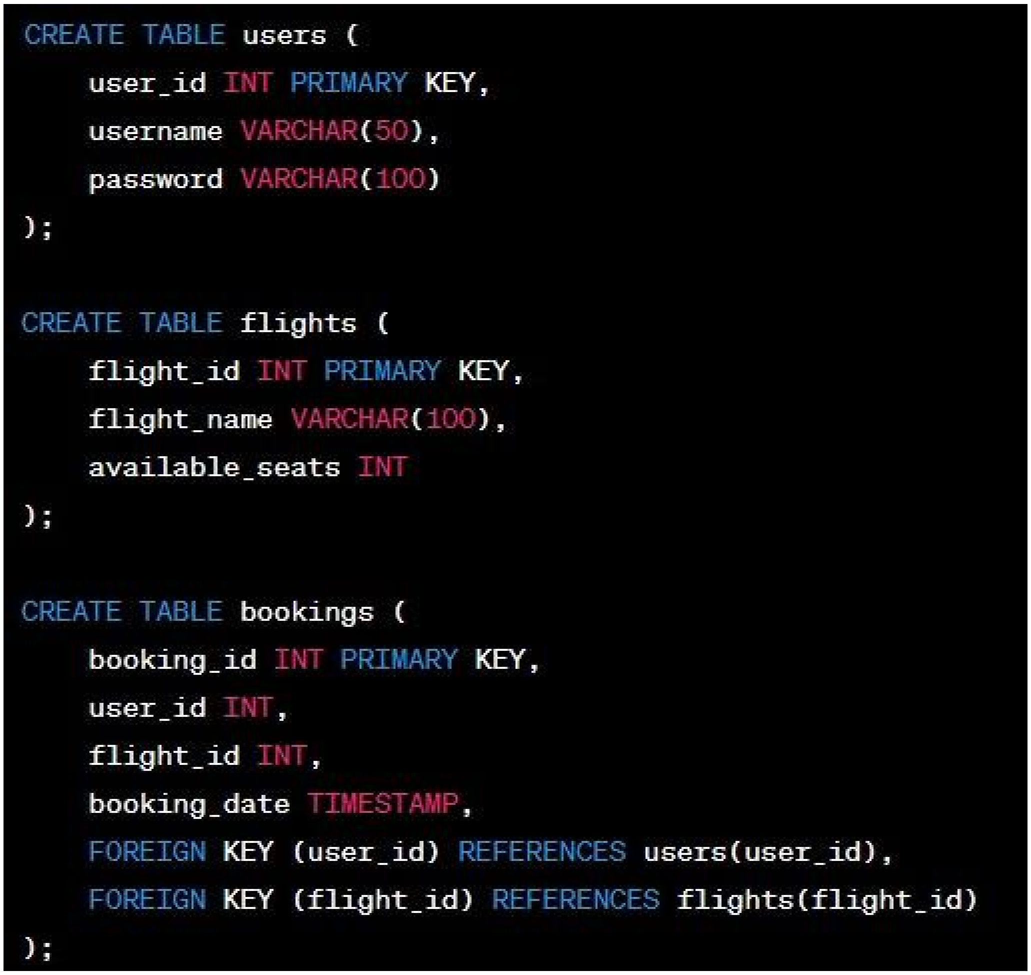
The secure ticketing feature not only streamlines the ticketing process but also ensures the authenticity of travelers, mitigating the risks associated with fraudulent activities and unauthorized access. This, combined with the aesthetically pleasing design of the flight list and homepage, creates a seamless and trustworthy user experience. The utilization of robust encryption techniques and biometric authentication further fortifies the system against potential breaches, assuring both airlines and passengers of the highest level of data protection.

Simultaneously, the aircraft tracking system brings unprecedented accuracy and visibility into the movements of aircraft. Just like the aesthetically pleasing design of the flight tracking page, the system leverages cutting-edge technologies such as GPS, radar, and communication systems to offer a comprehensive understanding of flight paths, potential diversions, and emergency situations. This real-time monitoring significantly enhances situational awareness, reduces response times, and bolsters overall aviation safety.

As the aviation industry continues to expand, the Secure Ticketing and Aircraft Tracking System, with its commitment to user-friendly design, stands as a testament to the innovative capabilities that technology offers. It sets a new standard for secure and efficient air travel, fostering trust among passengers and stakeholders alike. With an unwavering commitment to data security, operational efficiency, and passenger satisfaction, this system undoubtedly contributes to shaping the future landscape of global aviation.

Appendix:

1.1 Table Creation in MySQL:



```
CREATE TABLE users (
    user_id INT PRIMARY KEY,
    username VARCHAR(50),
    password VARCHAR(100)
);

CREATE TABLE flights (
    flight_id INT PRIMARY KEY,
    flight_name VARCHAR(100),
    available_seats INT
);

CREATE TABLE bookings (
    booking_id INT PRIMARY KEY,
    user_id INT,
    flight_id INT,
    booking_date TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (flight_id) REFERENCES flights(flight_id)
);
```

Appendix 1.2:

HTML code for login:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, width=device-width">
    <link crossorigin="anonymous"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jI
W3" rel="stylesheet">
    <script crossorigin="anonymous"
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTo
v4+1p"></script>
    <title>Shark Flights</title>

  </head>

  <body style="background-color: #000000">
    <div class="container">
      <form action="/" method="post" ><br><br><br><br><b style="font-size: 40px;
color:white; margin-left: 475px;">Secure Login Page</b> <br><br>
        <div class="mb-3">
          <input autocomplete="off" autofocus class="form-control mx-auto w-auto"
id="username" name="username" placeholder="Username" type="text">
        </div>
        <div class="mb-3">
          <input class="form-control mx-auto w-auto" id="password" name="password"
placeholder="Password" type="password">
        </div><br>
        <div class="mx-auto p-2">
          <button class="btn btn-primary" type="submit" style="margin-left: 600px;">Log
In</button>
        </div>
      </form>
    </div>
  </body>
</html>
```

Appendix 1.3:
mySQL initialization

```
import mysql.connector

db_connection = mysql.connector.connect(
    host="your_host",
    user="your_user",
    password="your_password",
    database="your_database"
)

cursor = db_connection.cursor()

insert_user_query = "INSERT INTO users (user_id, username, password) VALUES (%s, %s, %s)"
user_data = (1, "user123", "pass123")
cursor.execute(insert_user_query, user_data)
db_connection.commit()

select_users_query = "SELECT * FROM users"
cursor.execute(select_users_query)
users = cursor.fetchall()

for user in users:
    print(user)

cursor.close()
db_connection.close()
```