**Screen Shots of Fly Away (An airline booking portal)-Simplilearn-Phase2 Project**

**--By Dr. Jyothi N.M.**

**Git Link-**

[**https://github.com/jyothiarunkr/phase2-project2-flyaway-jyothinm.git**](https://github.com/jyothiarunkr/phase2-project2-flyaway-jyothinm.git)

**Project objective:**

As a Full Stack Developer, design and develop an airline booking portal named as FlyAway. Use the GitHub repository to manage the project artifacts.

**Background of the problem statement:**

FlyAway is a ticket-booking portal that lets people book flights on their website.

**The website needs to have the following features:**

● A search form in the homepage to allow entry of travel details, like the date of travel, source, destination, and the number of persons.  
● Based on the travel details entered, it will show the available flights with their ticket prices.  
● Once a person selects a flight to book, they will be taken to a register page where they must fill in their personal details. In the next page, they are shown the flight details of the flight that they are booking, and the payment is done via a dummy payment gateway. On completion of the payment, they are shown a confirmation page with the details of the booking.

For the above features to work, there will be an admin backend with the following features:

● An admin login page where the admin can change the password after login, if he wishes  
● A master list of places for source and destination  
● A master list of airlines  
● A list of flights where each flight has a source, destination, airline, and ticket price  
  
The goal of the company is to deliver a high-end quality product as early as possible. 

**The flow and features of the application:**

● Plan more than two sprints to complete the application  
● Document the flow of the application and prepare a flow chart   
● List the core concepts and algorithms being used to complete this application  
● Implement the appropriate concepts, such as exceptions, collections, and sorting techniques for source code optimization and increased performance

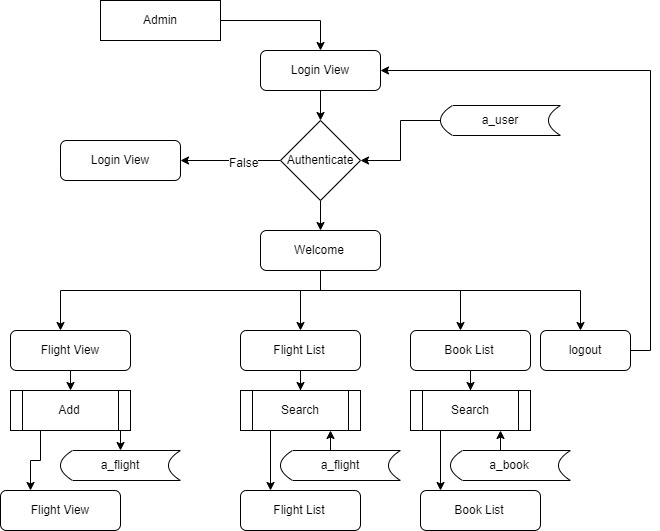
**Software Requirements**

● Eclipse/IntelliJ: An IDE to code for the application   
● Java: A programming language to develop the web pages, databases, and others  
● SQL: To create tables for admin, airlines, and other specifics  
● Maven: To create a web-enabled Maven project  
● Git: To connect and push files from the local system to GitHub   
● GitHub: To store the application code and track its versions   
● Scrum: An efficient agile framework to deliver the product incrementally   
● Search and Sort techniques: Data structures used for the project   
● Specification document: Any open-source document or Google Docs

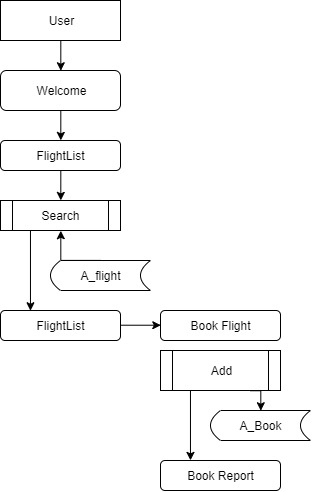
**Functional Requirements**

● An admin login page where the admin can change the password after login, if he wishes  
● A master list of places for source and destination  
● A master list of airlines  
● A list of flights where each flight has a source, destination, airline, and ticket price

**Admin Flow Chart Diagram**

****

**User Flow Chart Diagram**

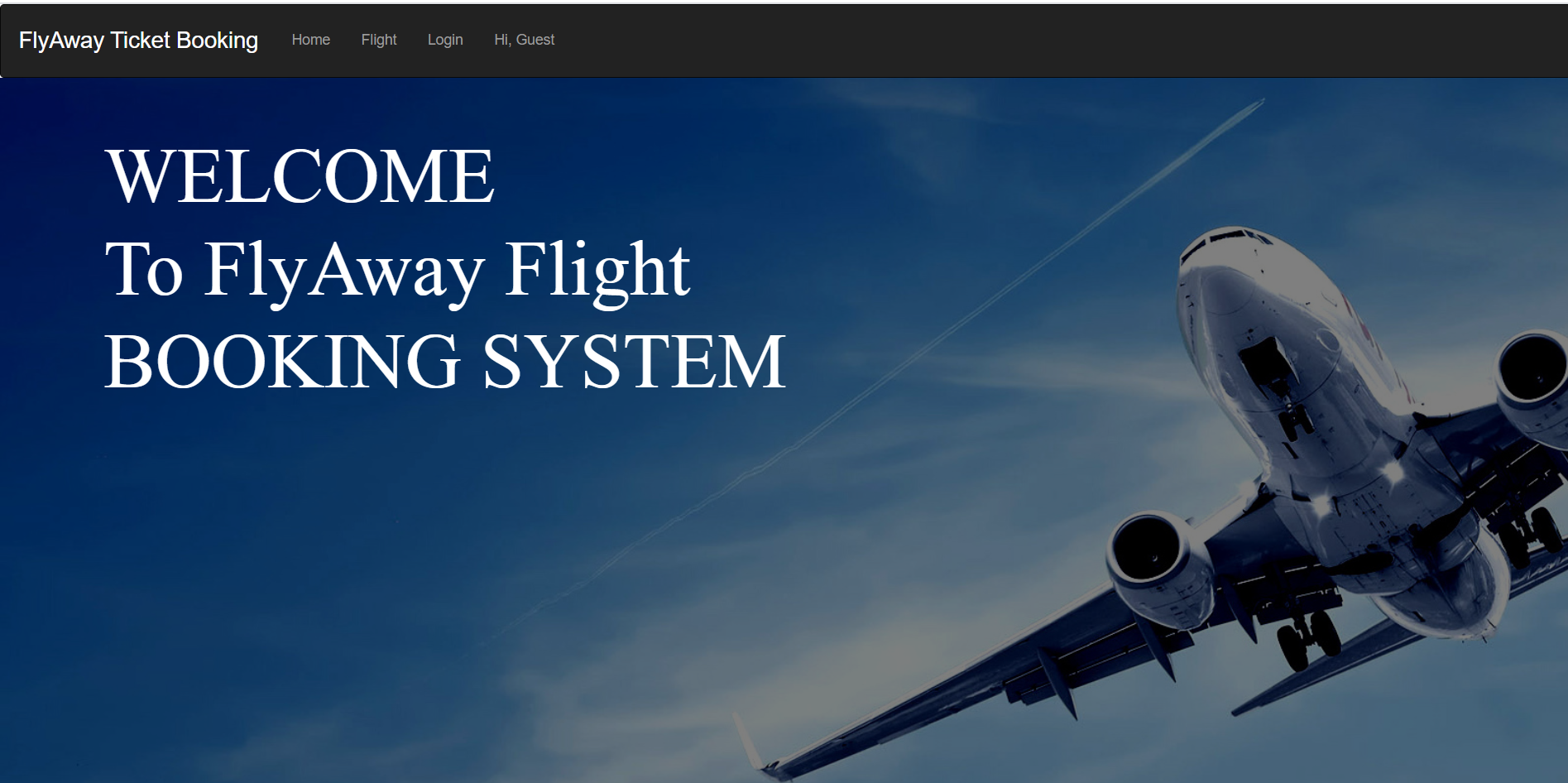


**Output Screens**

**Project Workflow- shown in step by step**

**Welcome Screen**

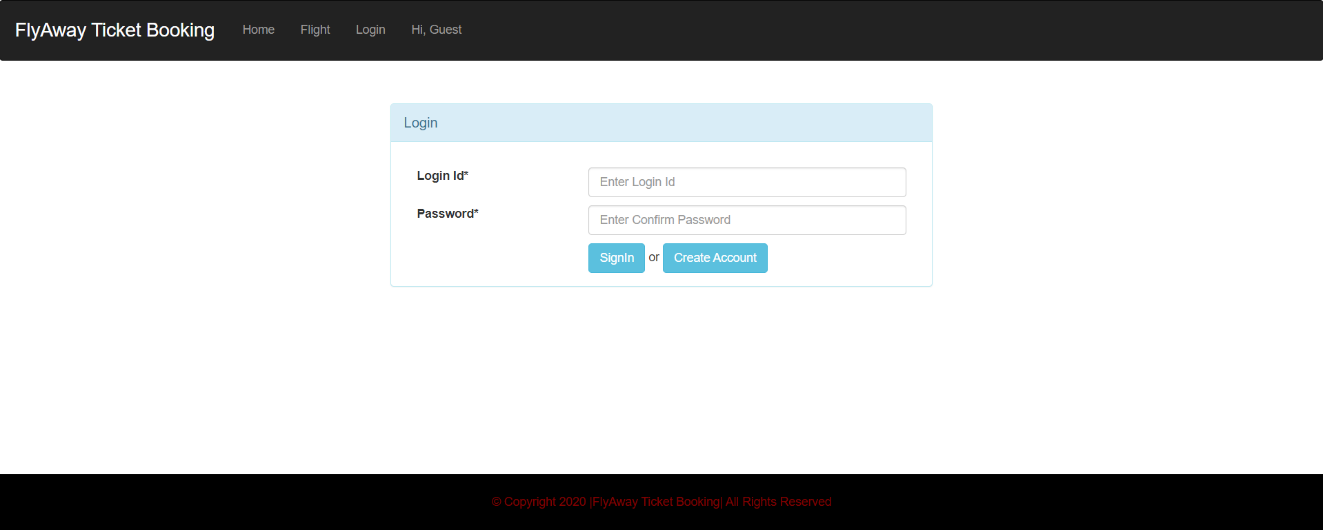
This screen has the Company Name along with few options for the user to Navigate to different pages as per their requirements.



**Login Page**

In this Login Page the guest has to two options:

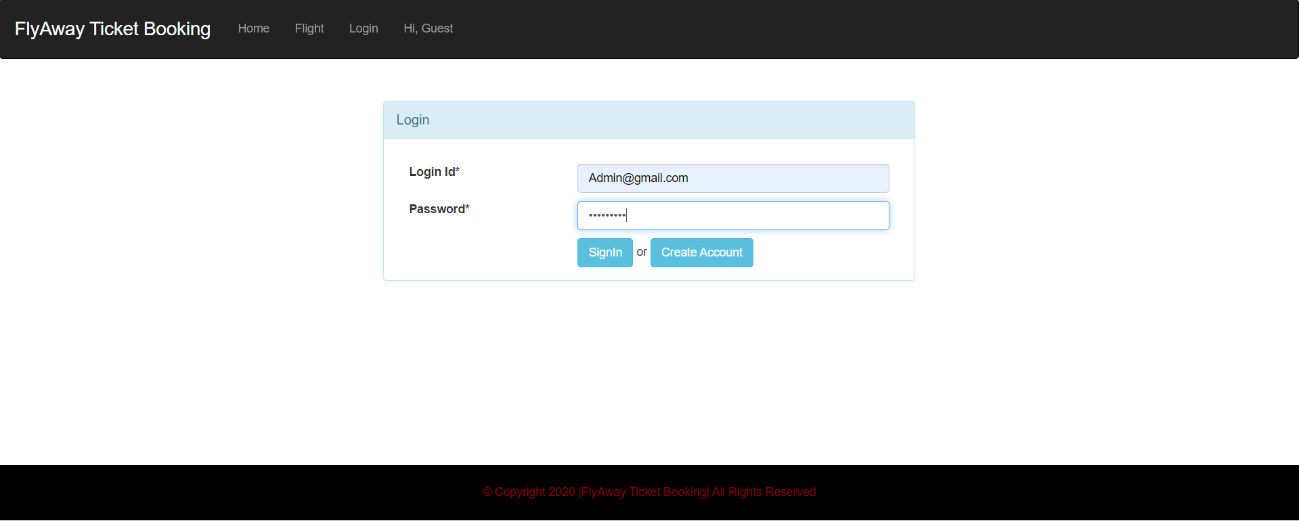
* Guest can login using their credentials
* Guest can newly register themselves to the web page by creating a new login credentials.
* To get the Admin Privileges Admin can login using the same login Page but with the Admin Credentials

****

**Admin Workflow**

The Admin Credentials are:

* Login ID: [Admin@gmail.com](mailto:Admin@gmail.com)
* Password: Admin@123

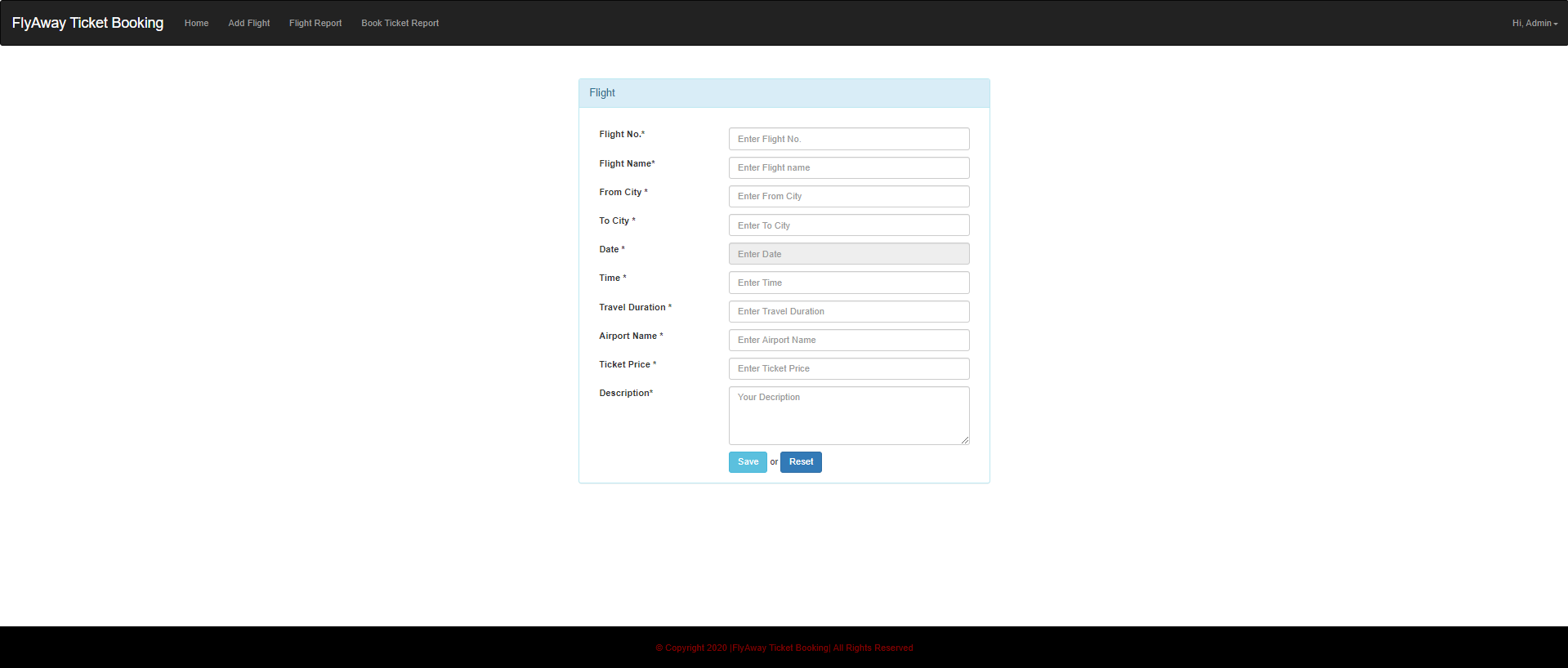
****

**Admin Welcome Screen**

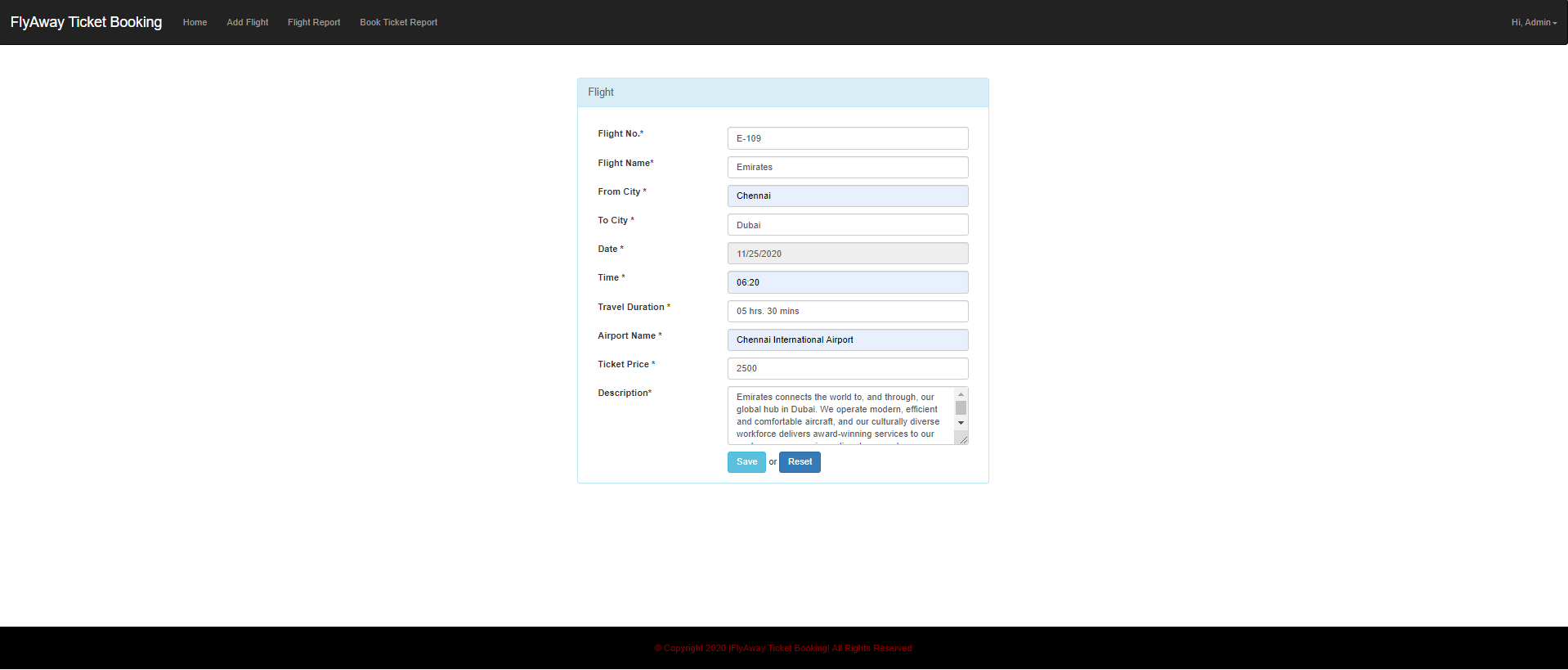
As soon as the Admin logged in, Admin has the access to do the following operations mentioned below:

* Add Flight – Admin can Navigate to Add flight page where he can fill in flight details and add it

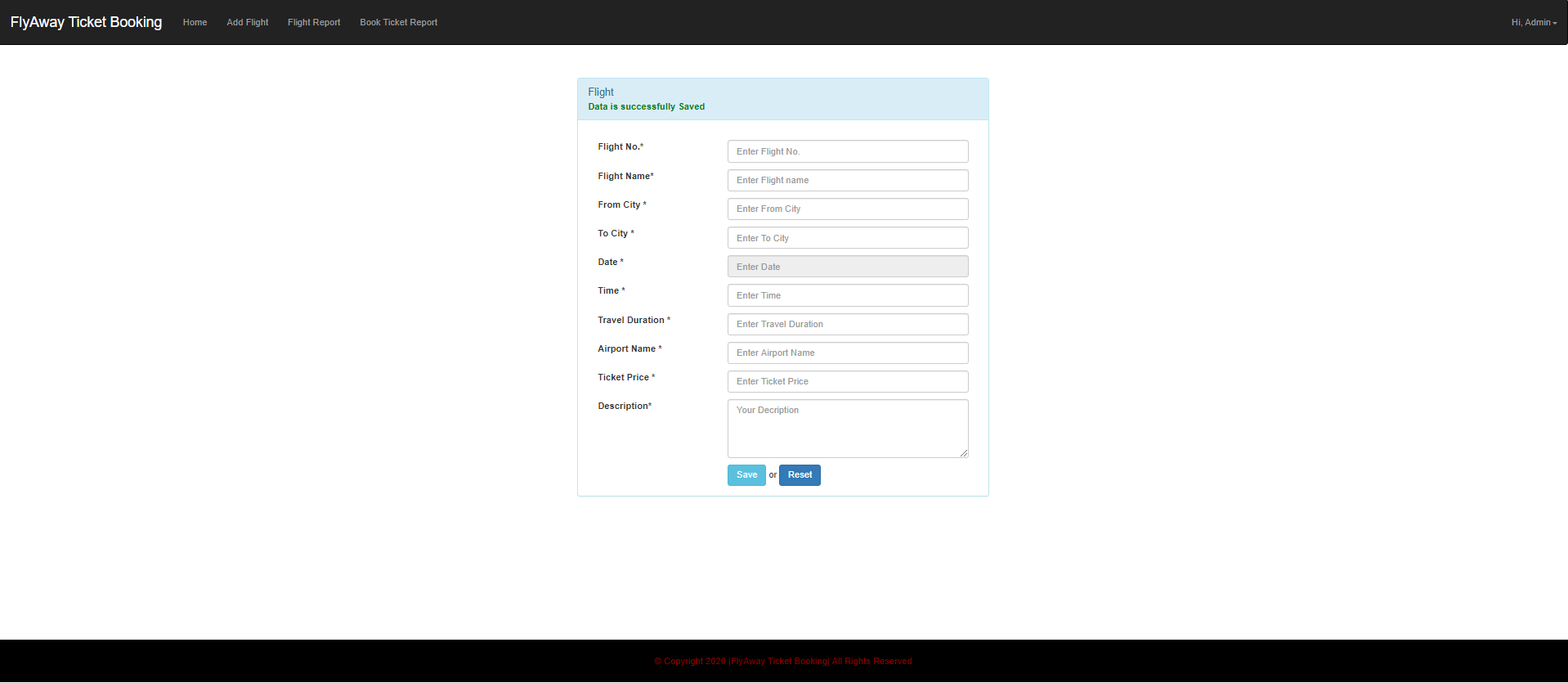
As soon as the Admin is Navigated to Add Flight page, then he needs to fill all the details of the flight



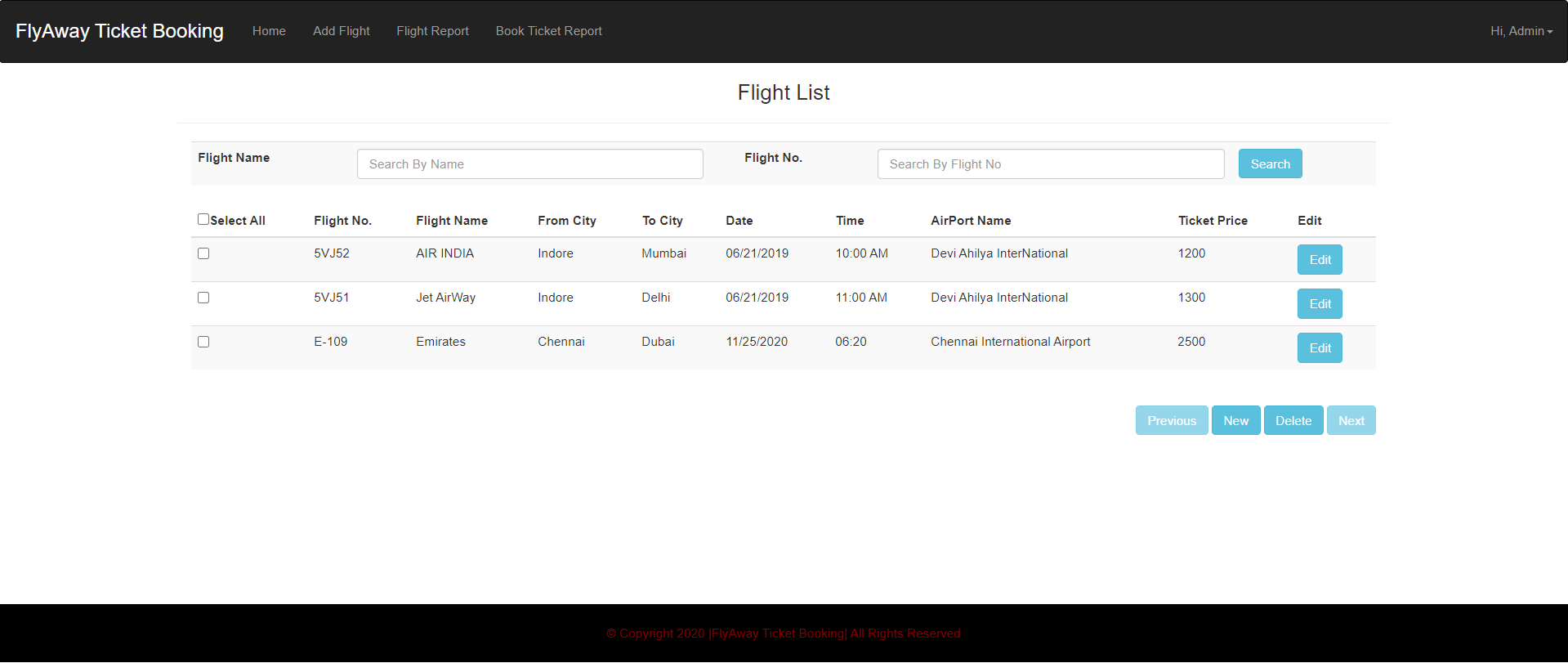
As soon as this page appear then the details of the new flight is entered and Save button is clicked

****

* Flight Report – In this Page the admin can see all the flights currently is available for the booking

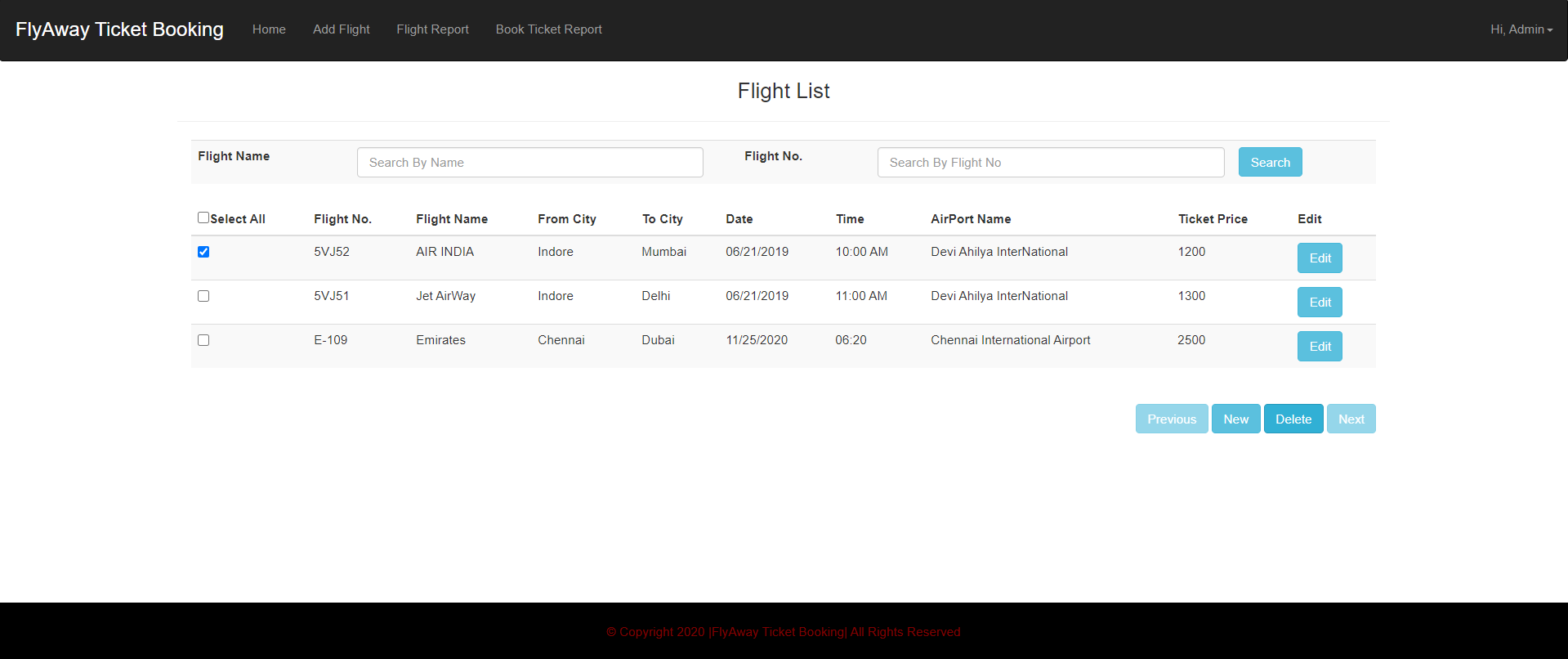
****

* Book Ticket Report – In this page Admin can see the booking details of the tickets

****

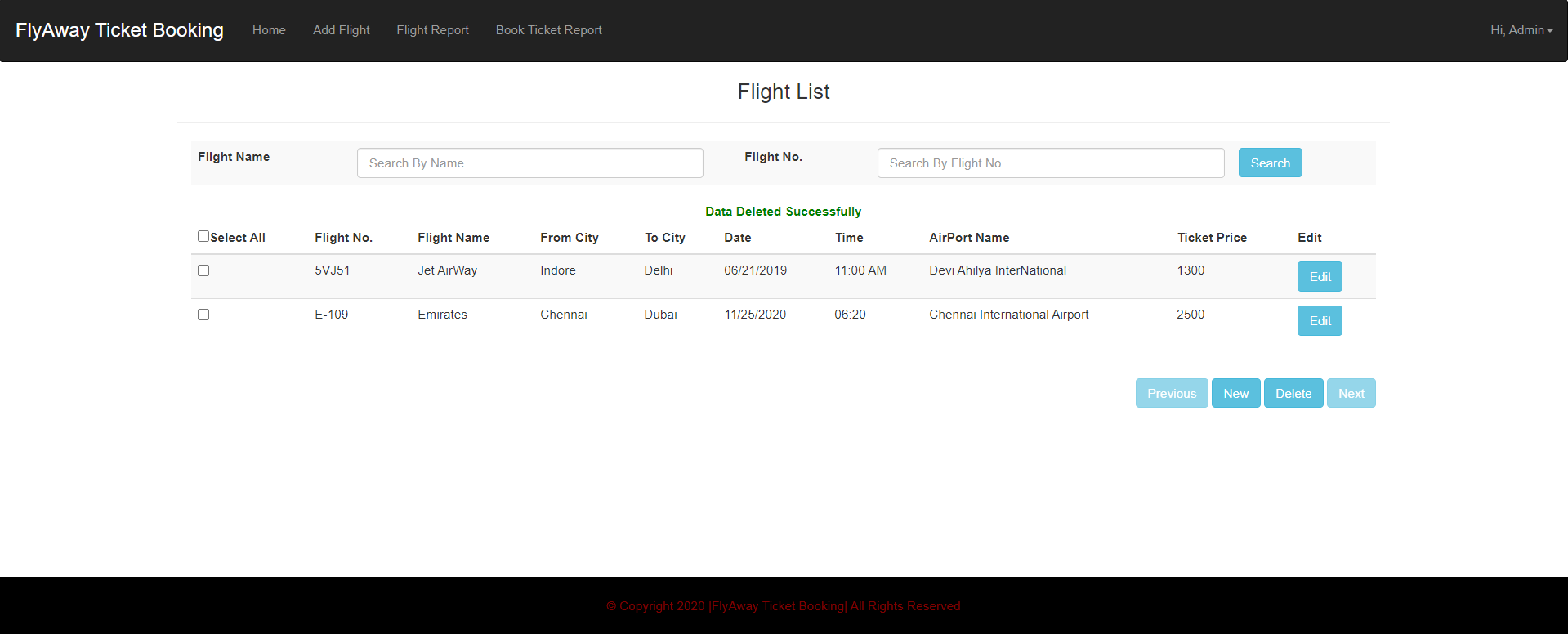
**Delete flight detail**

* The Admin has an option to delete the flight detail by selecting the flight details by checking the check box and clicking the delete button to complete the action

****

**Data deletion successful**

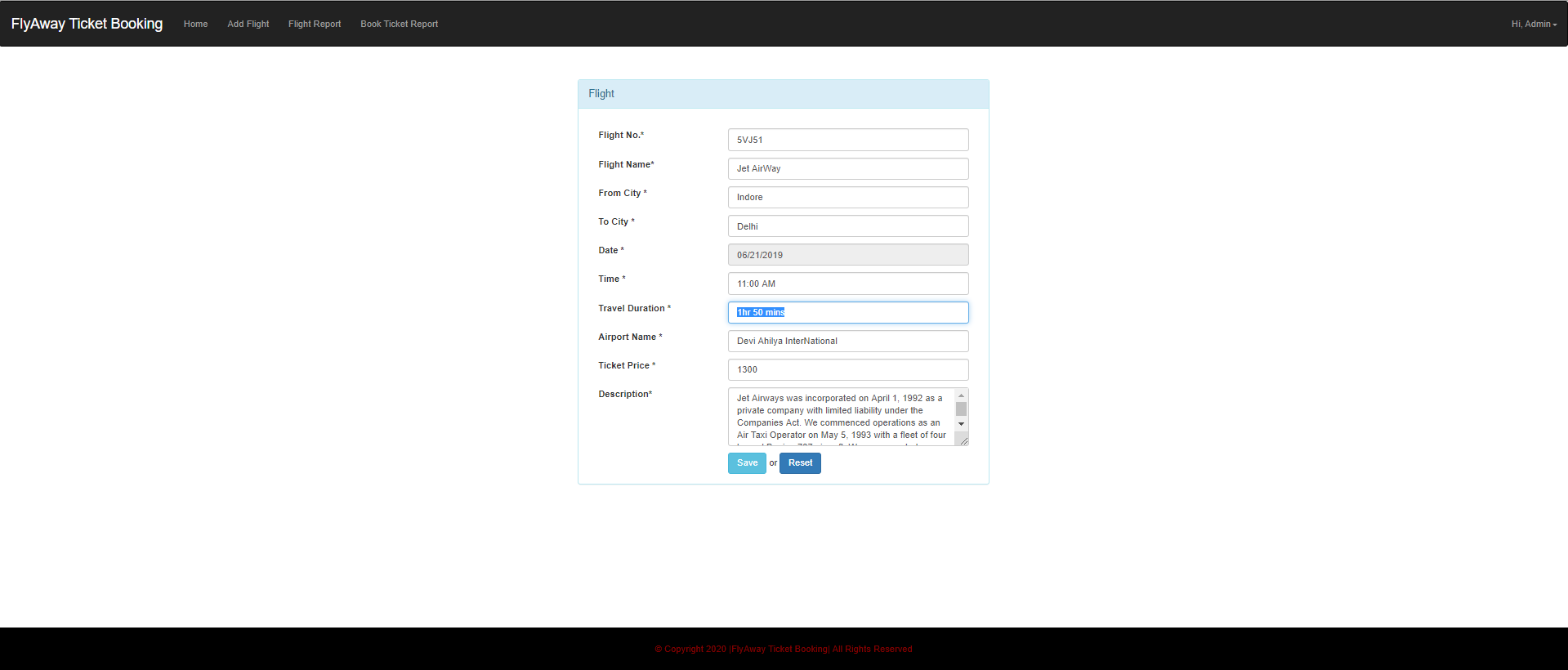
* Once the selected flight ticket is deleted then there is a message to make the user know that the data is deleted successfully as shown in the below screenshot

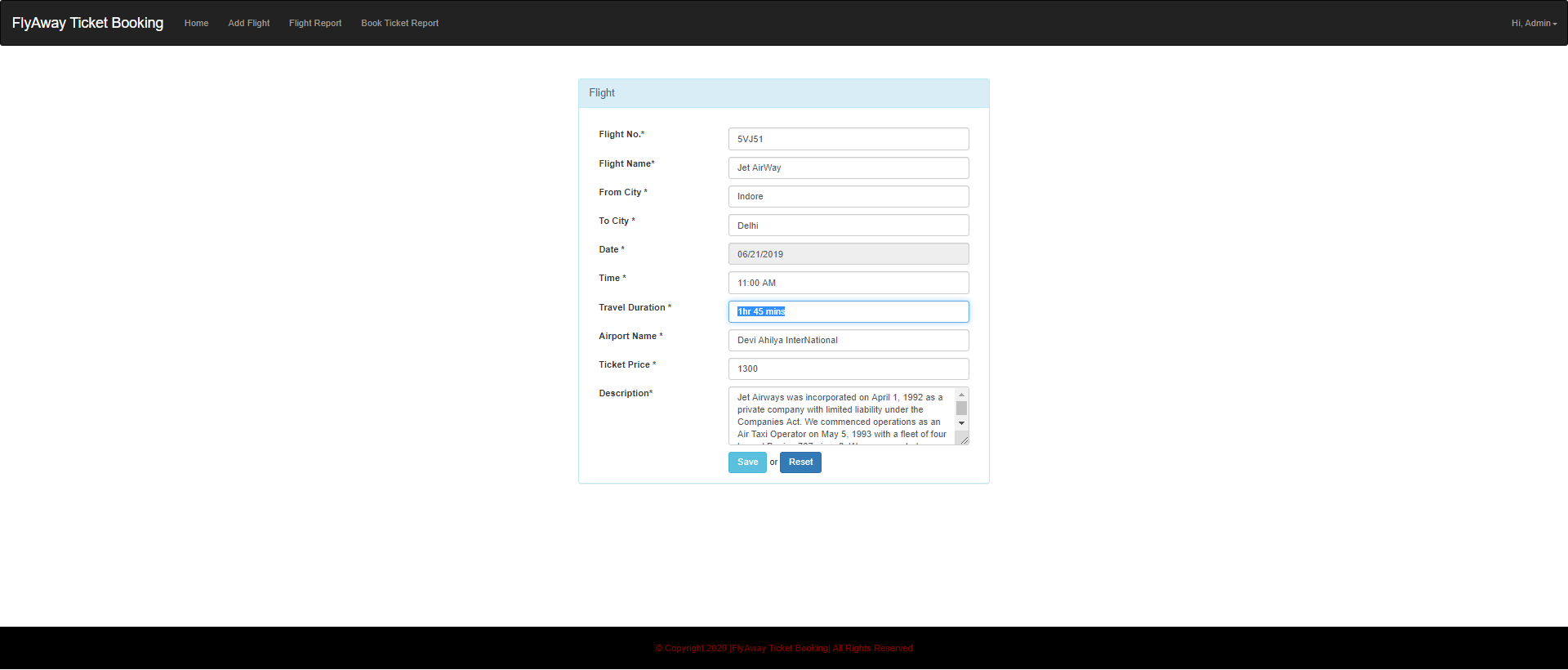
****

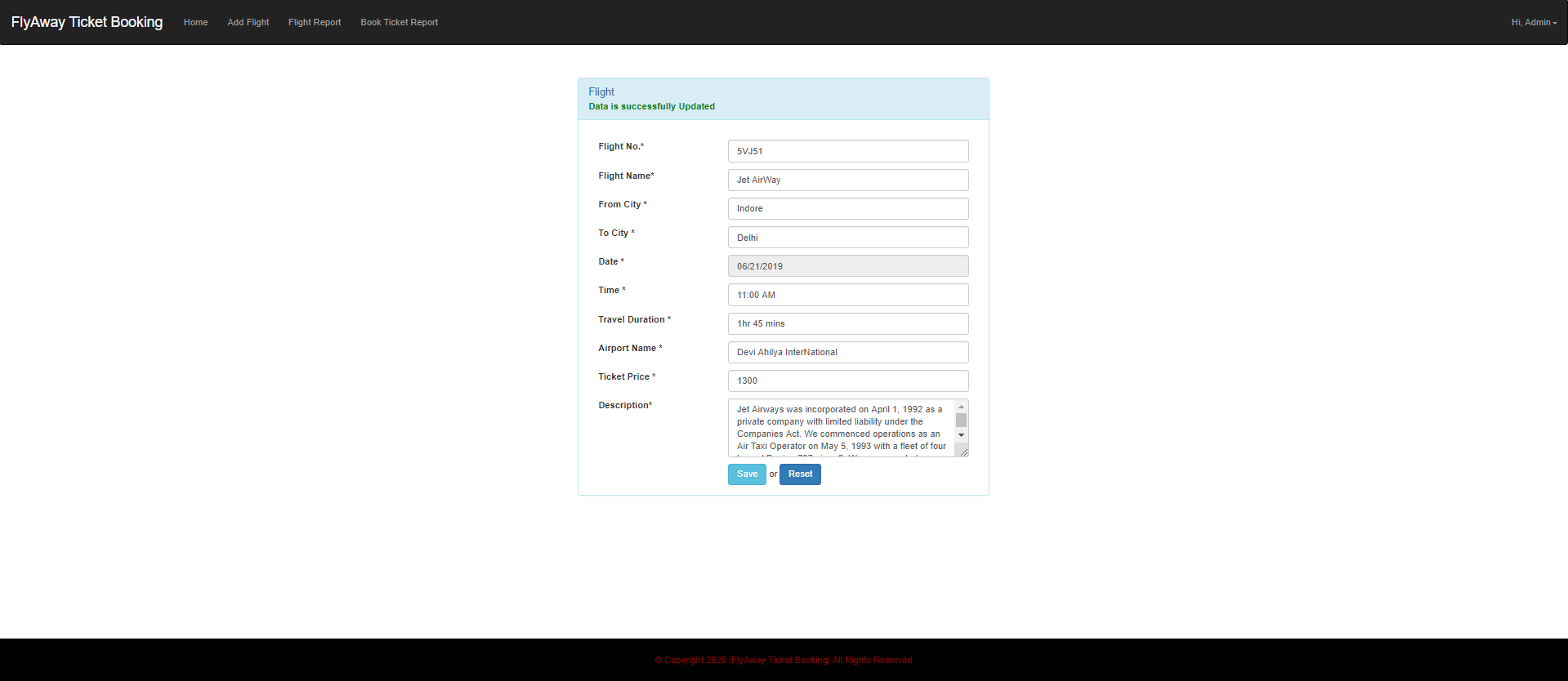
**Edit flight details**

To Edit the flight details of a particular flight the Admin has to click the Edit button available on each item of the list and click it to display all the details of the particular selection where Admin can Update the Details of the Flight

**Time duration was edited from 1hr 50mins to 1hr 45mins:**

****

****

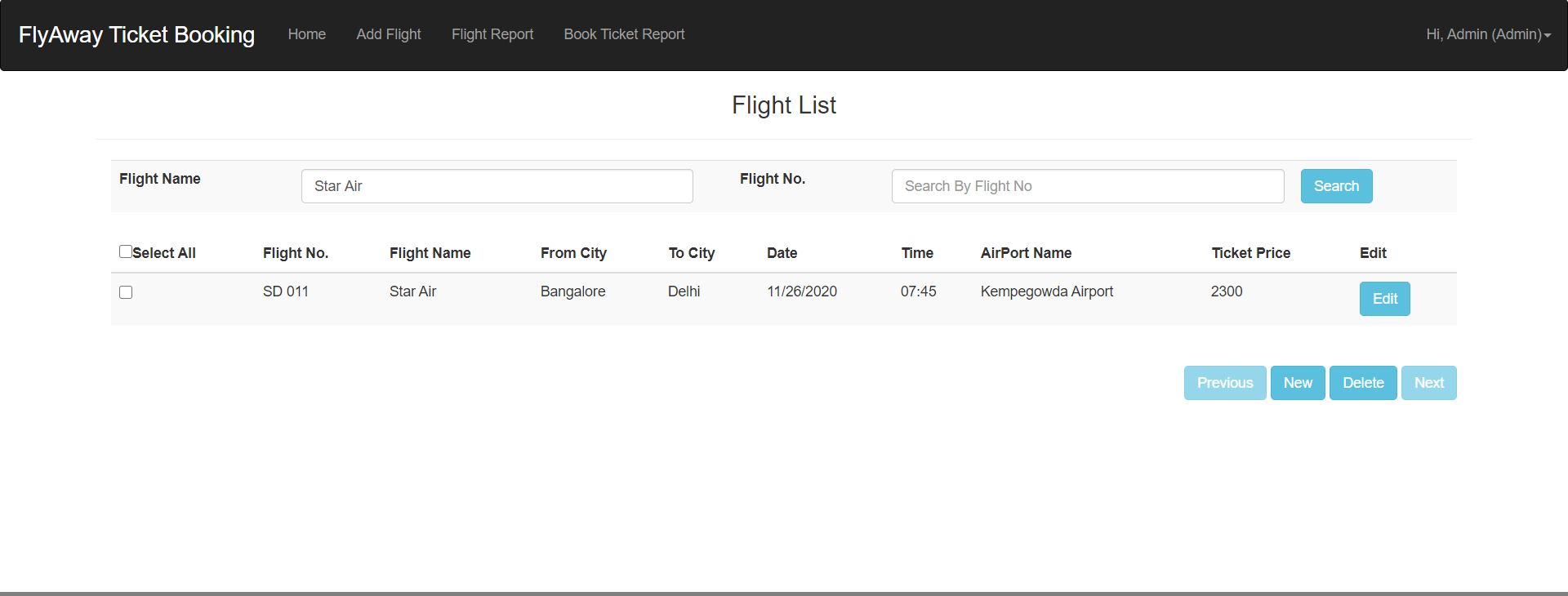
**Edited detail has been updated successfully:**

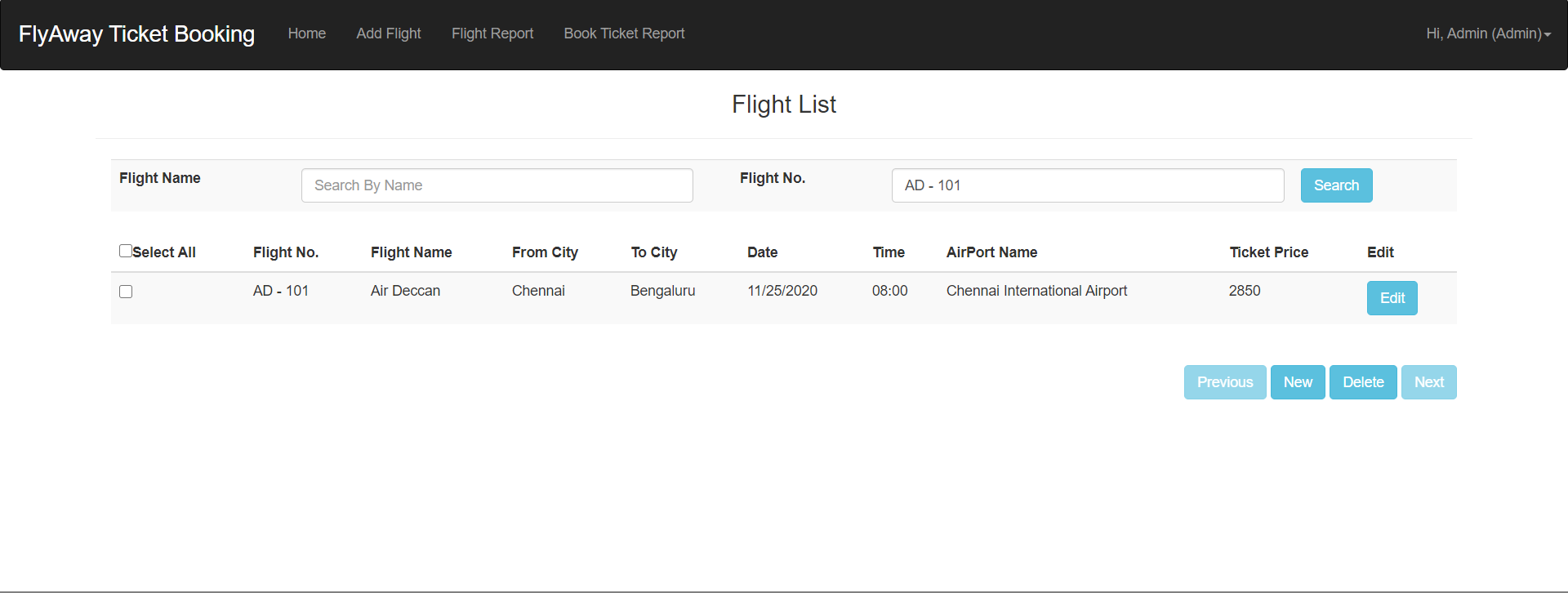
**Filter option**

There are few Filter options which are implemented for searching the appropriate flight for the Admin to make update to the details of the flight or even deleting it etc.

Below are the few Filter options namely for the Admin Credential

* Filter by Flight Name
* Filter by Flight Number

**Filter by flight name:** ****

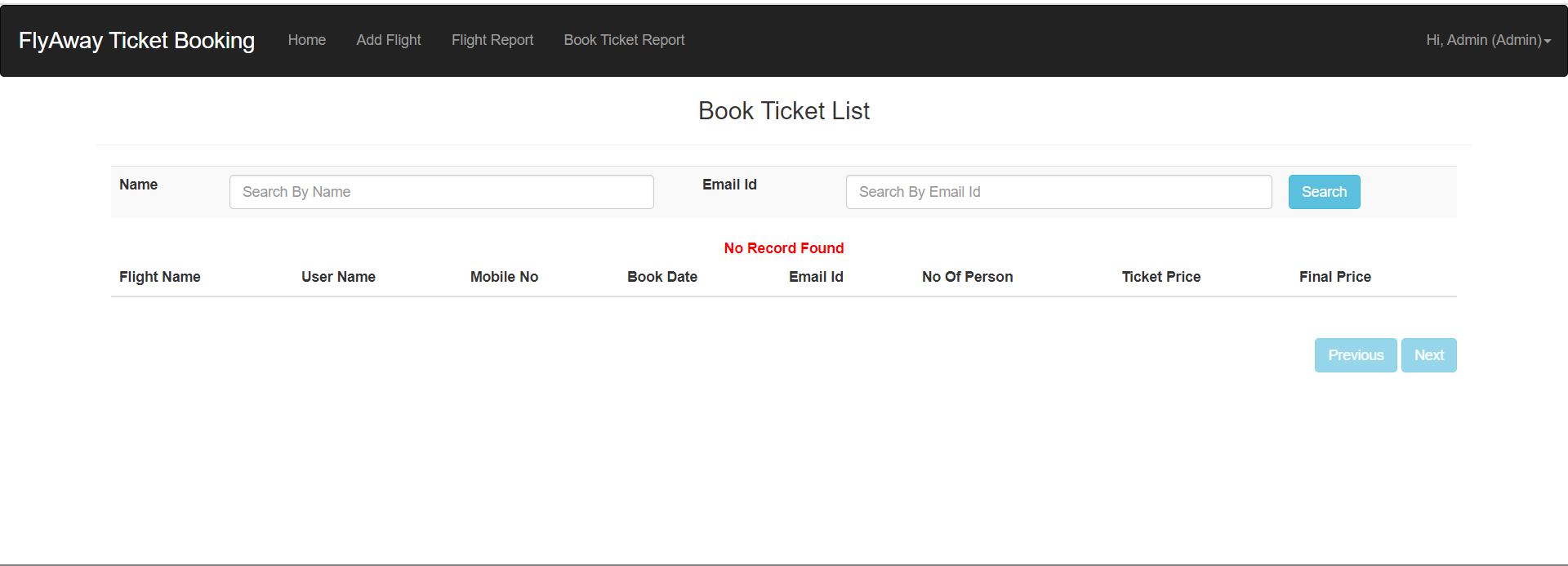
**Filter by flight number:**

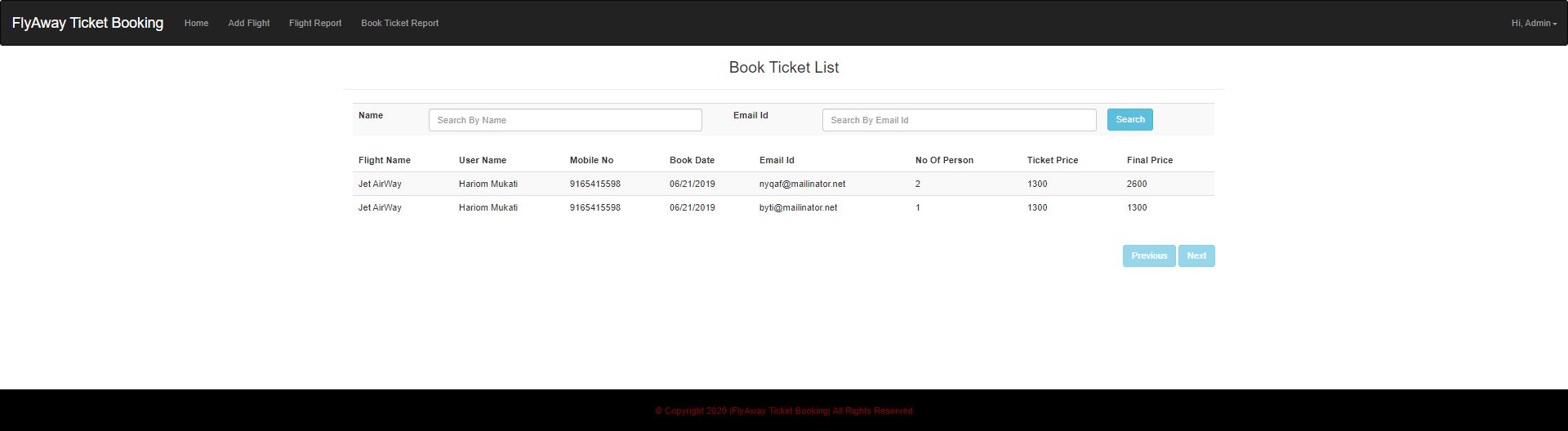
**Book Ticket Report**

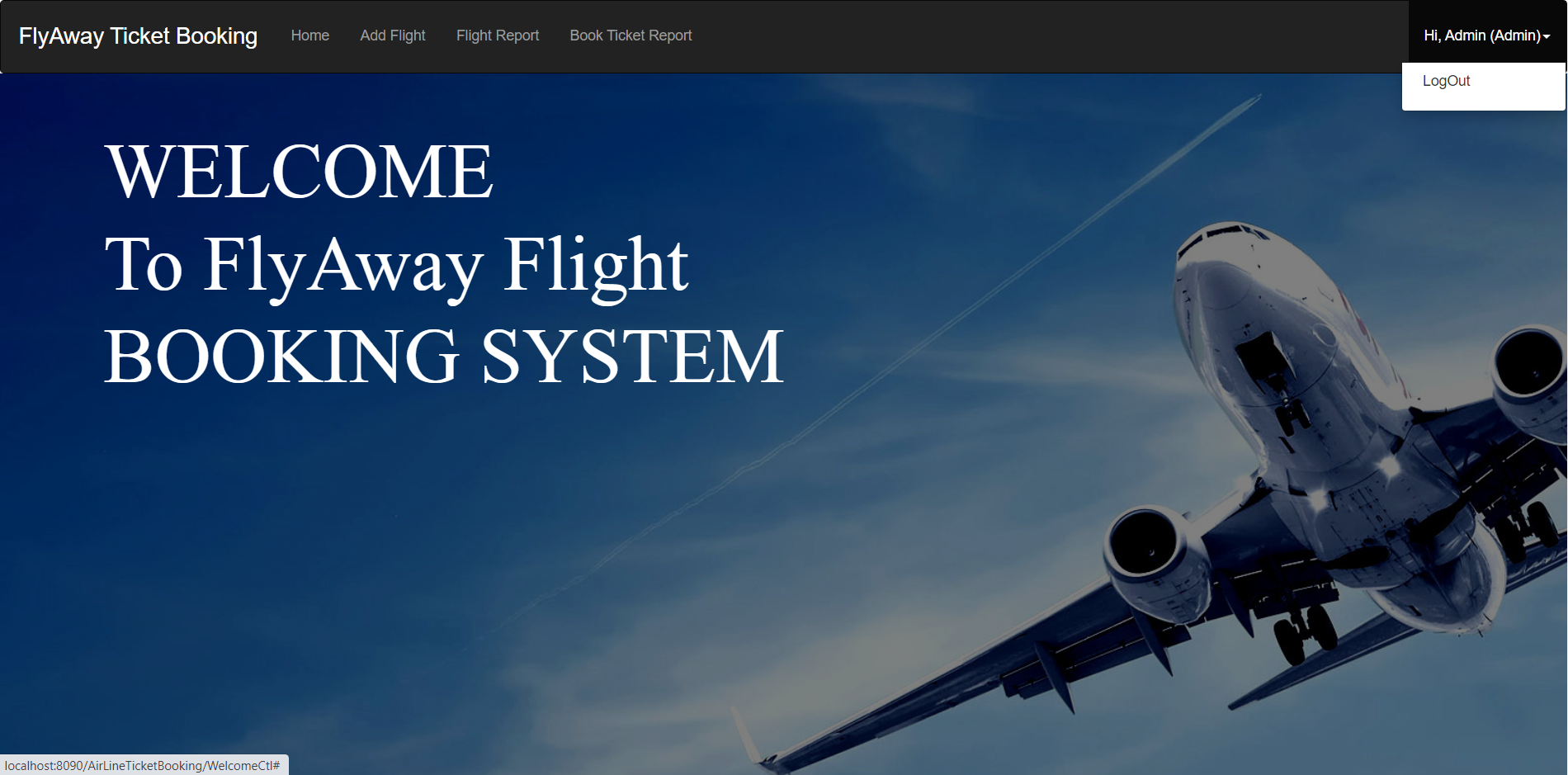
The booked report of the all the users have been listed out for the Admin to view the bookings made.

* If there are any details of the bookings then the bookings are listed out with the details of the user along with the Flight details booked
* If there are no records booked yet by any user then the message No Records Found message

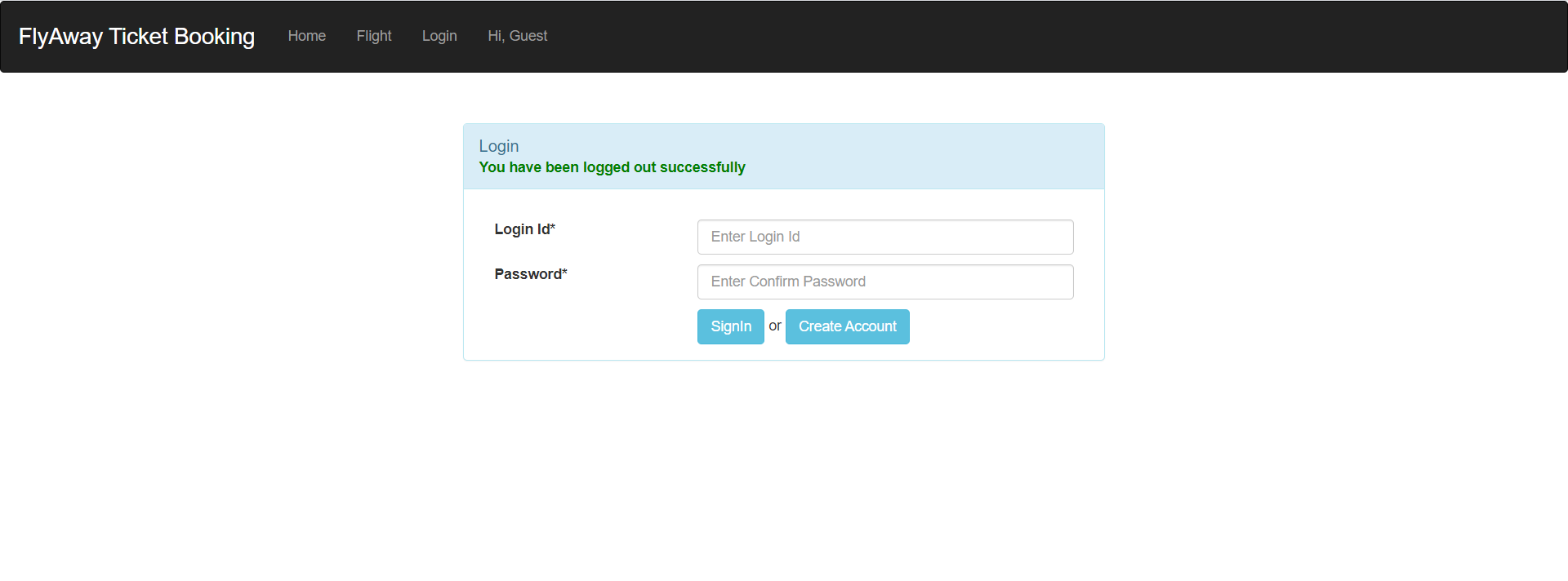
Below are the Images to show the view of both the views

****

****

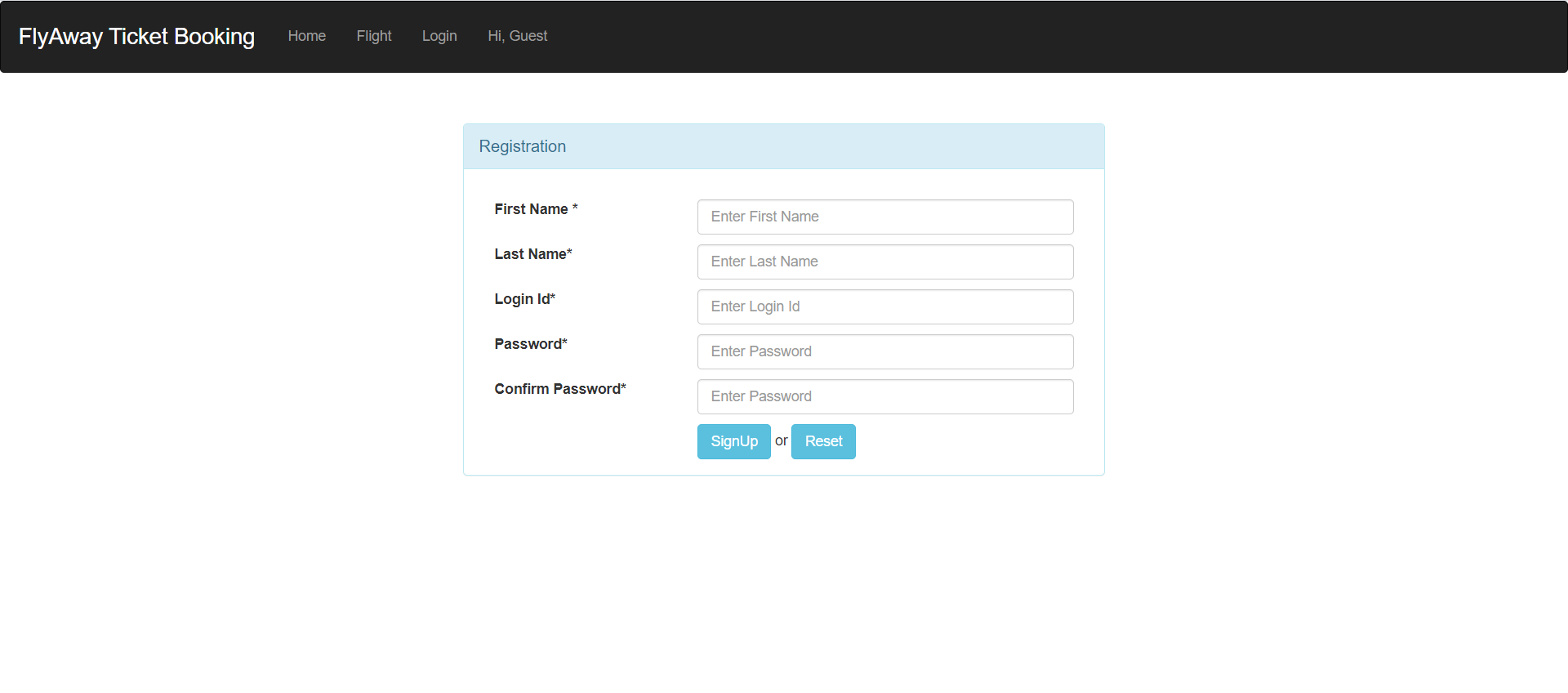
**Log out as Admin:**

**Logged out successfully:**

****

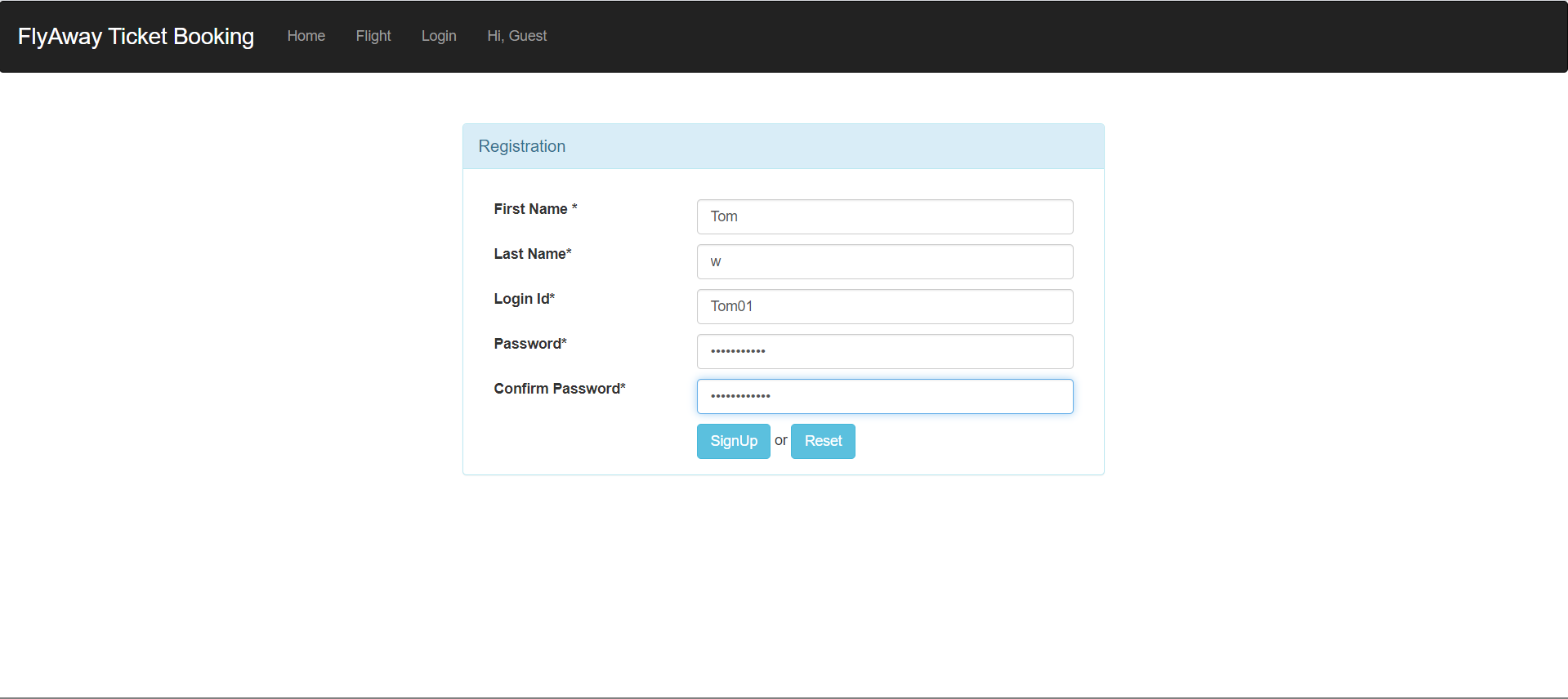
**Login as Guest:**

As soon as a Guest loads into the application, A Guest they have to login to the application or has to create an account in order to book the tickets of flights from the Flyaway Reservation system

**Create an account for the user:** ****

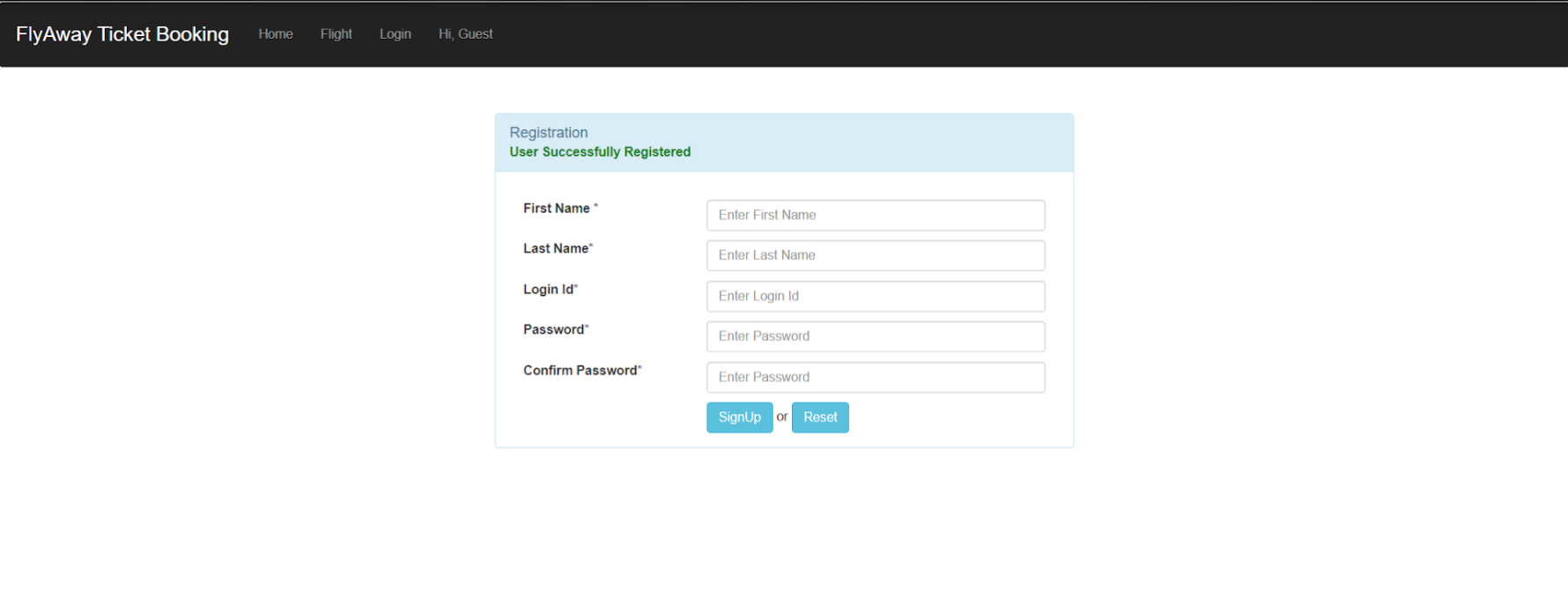
**Entered user details**

The User details has been entered with the details of the user sufficient to identify unique the user and have to signup to add the user details to the database

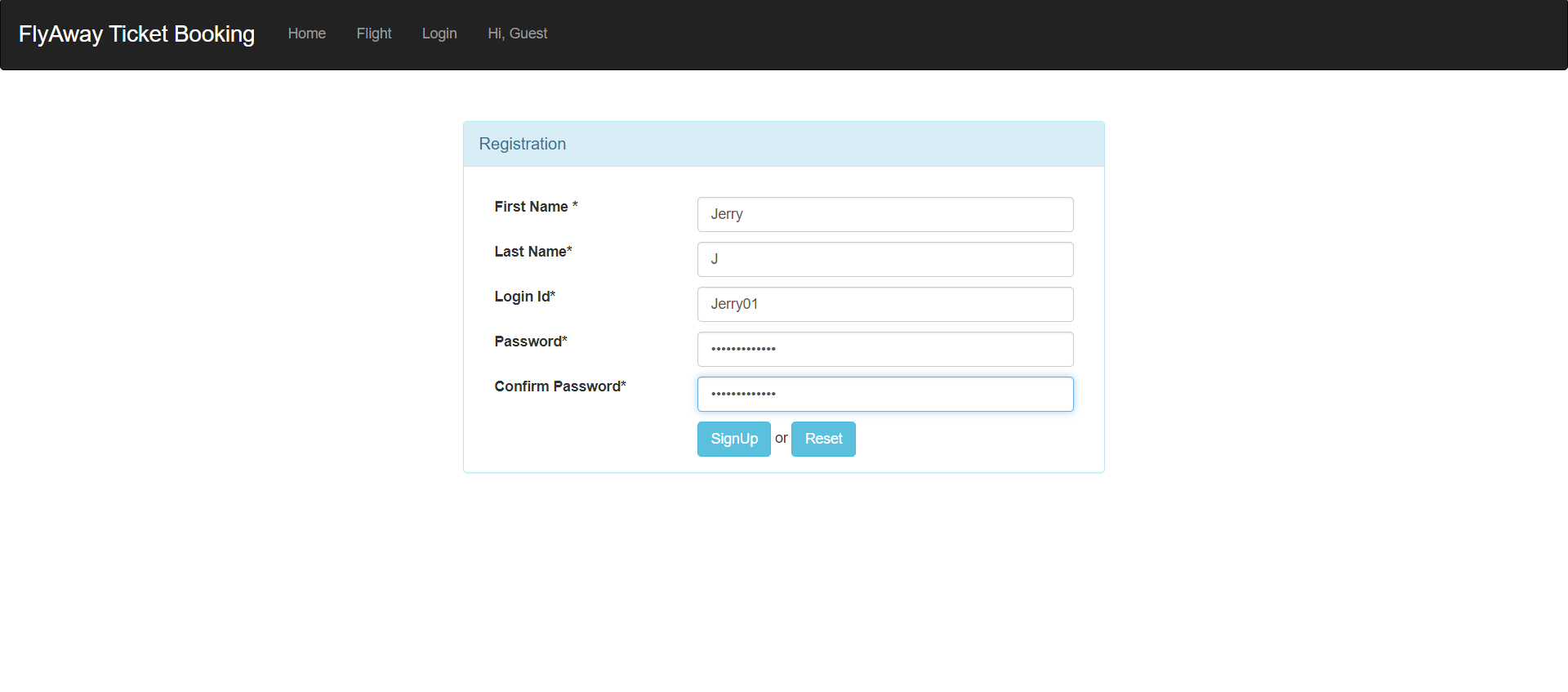


**User successfully added:**

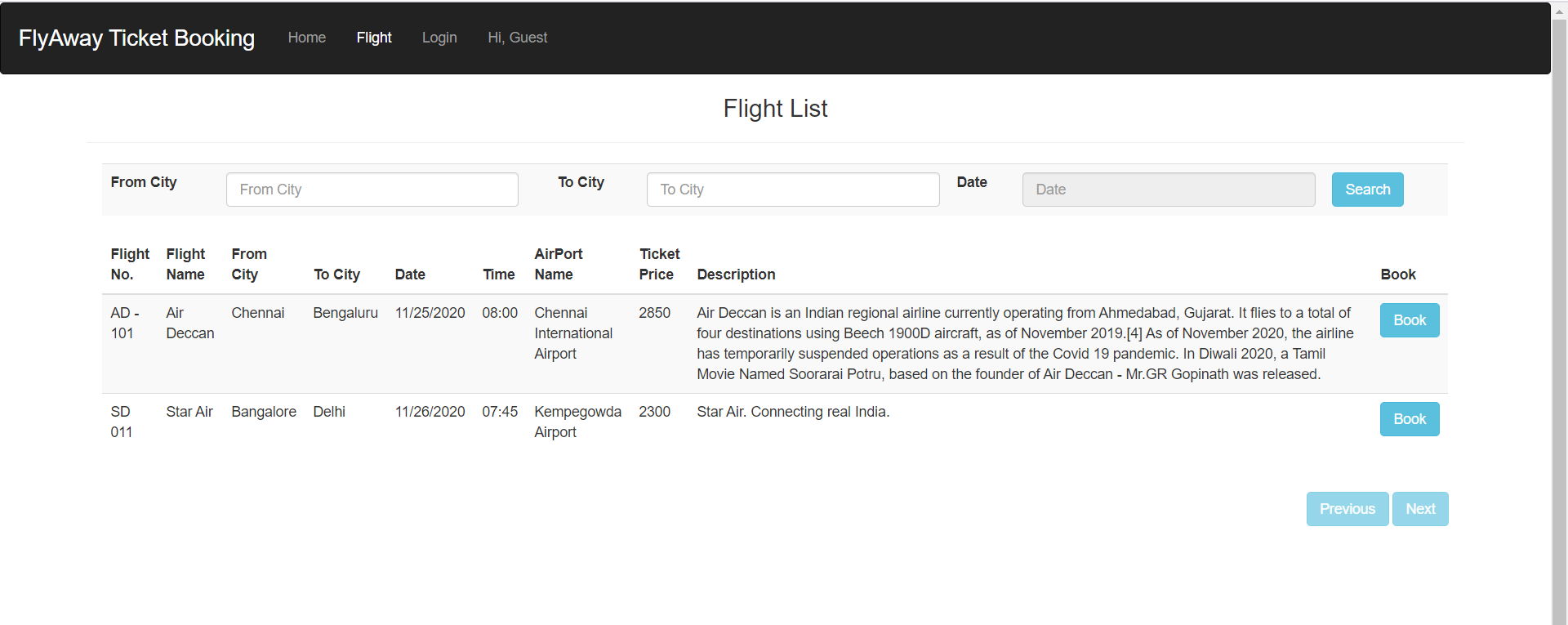
After the User successfully Signup with the details entered as per the validations then User Successfully Registered message appears and the details are successfully Added



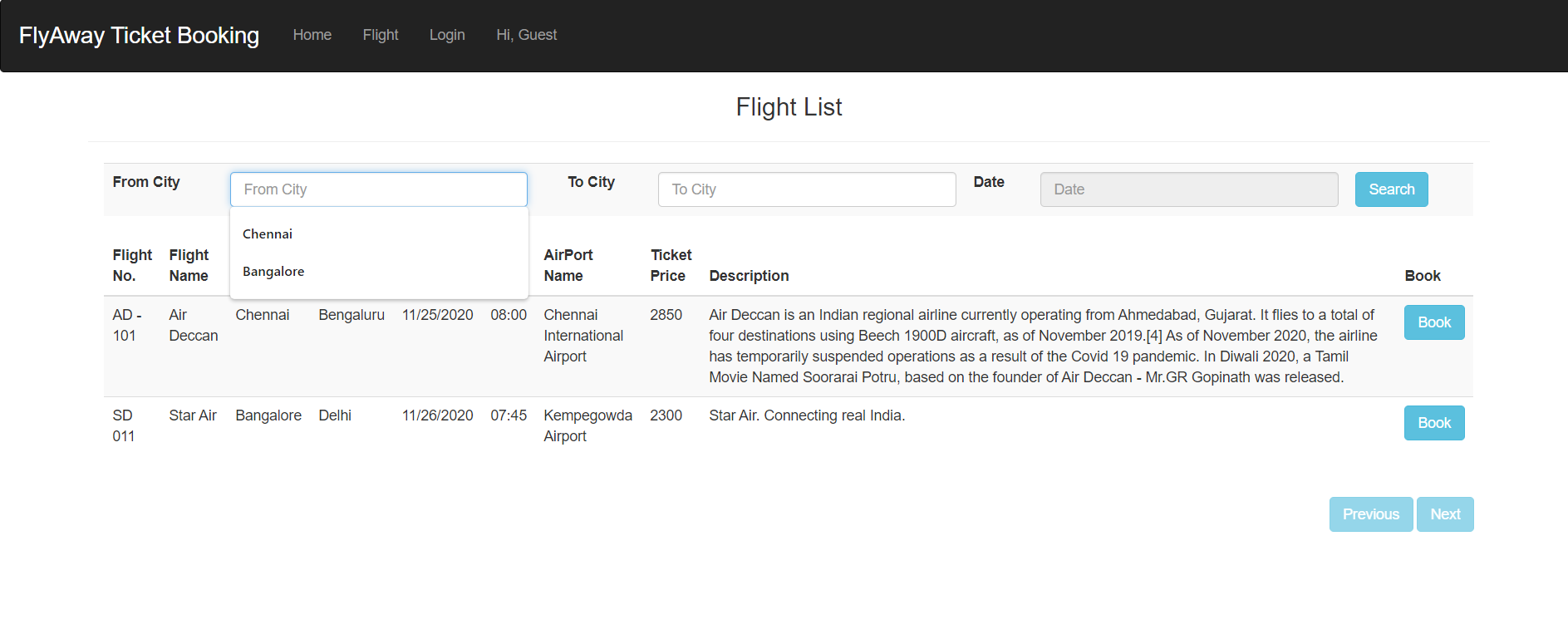
**Login as User:**

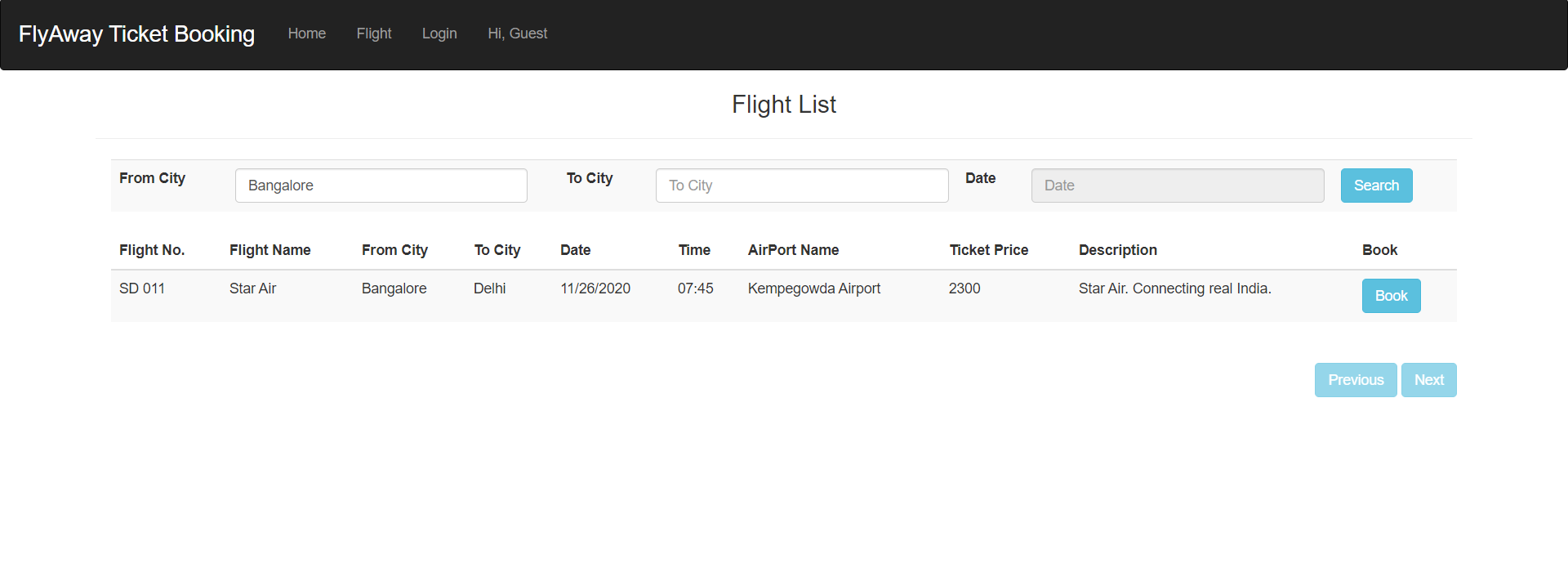
****

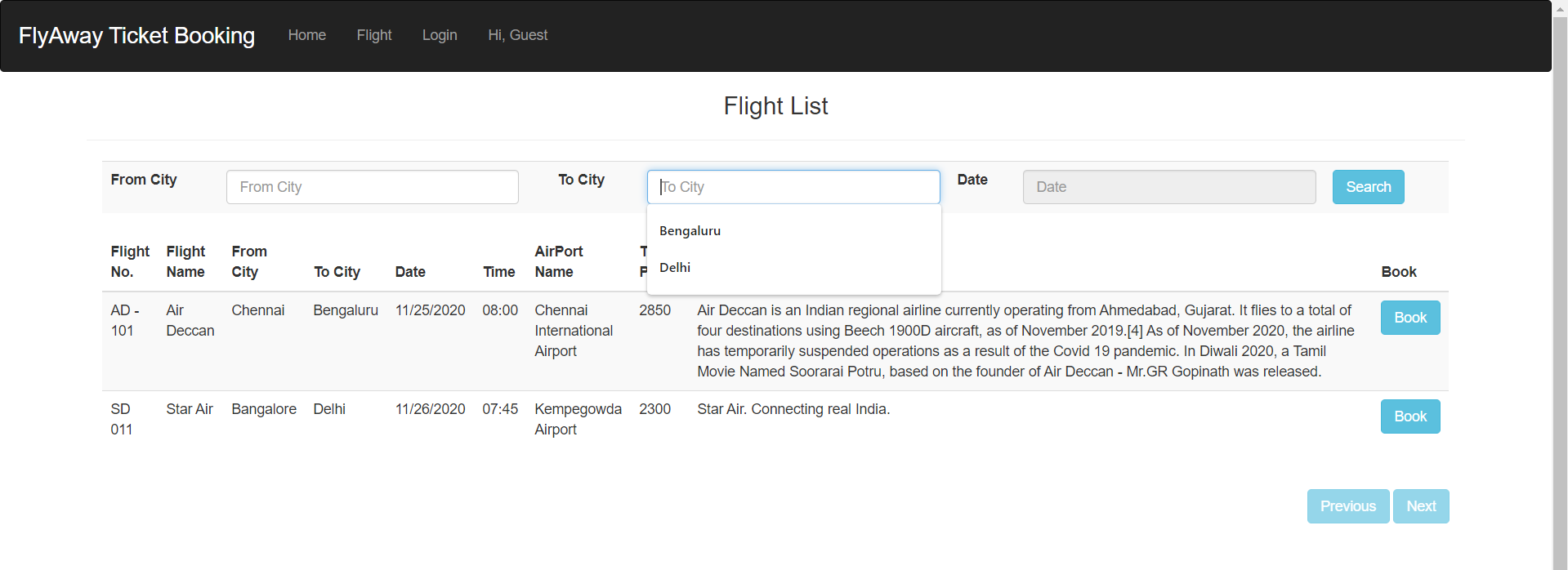
**User views the flight list:**

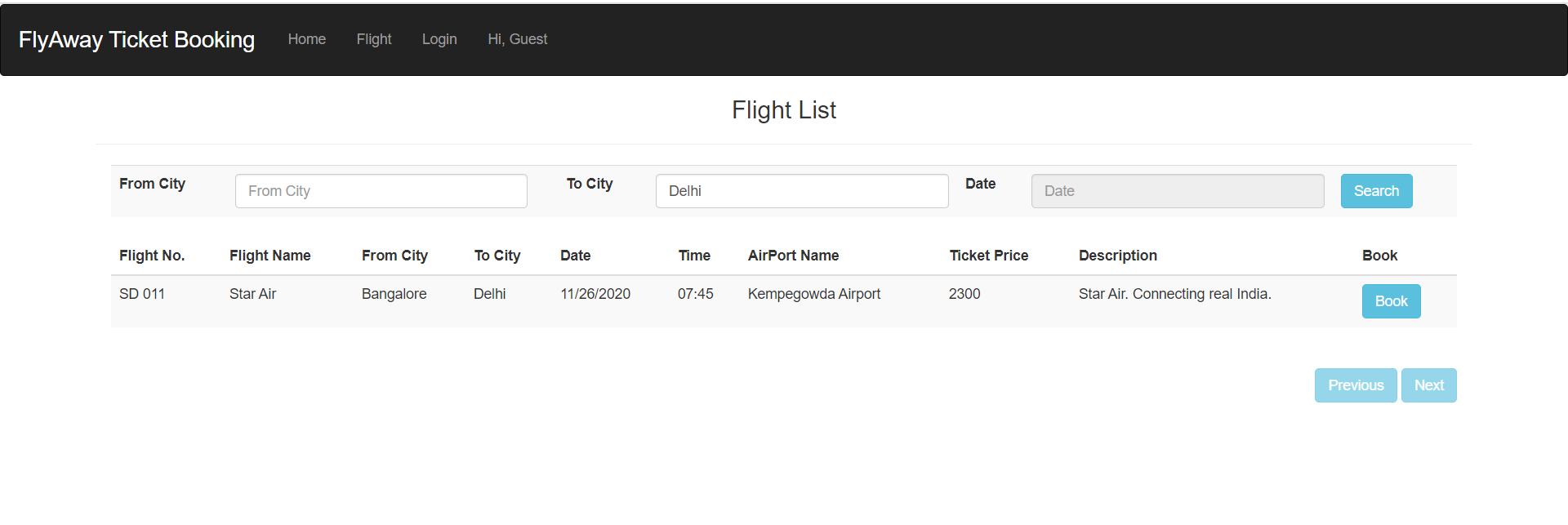
****

**User to filter flight details with City name:**

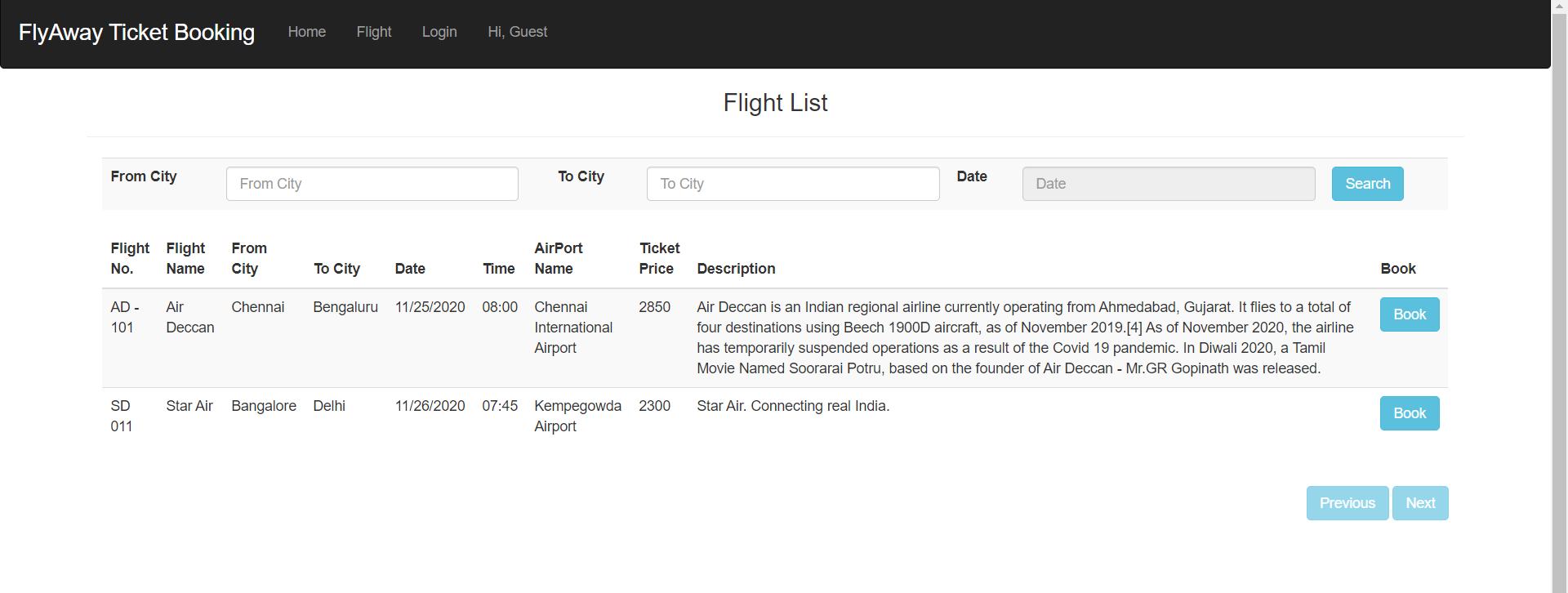
****

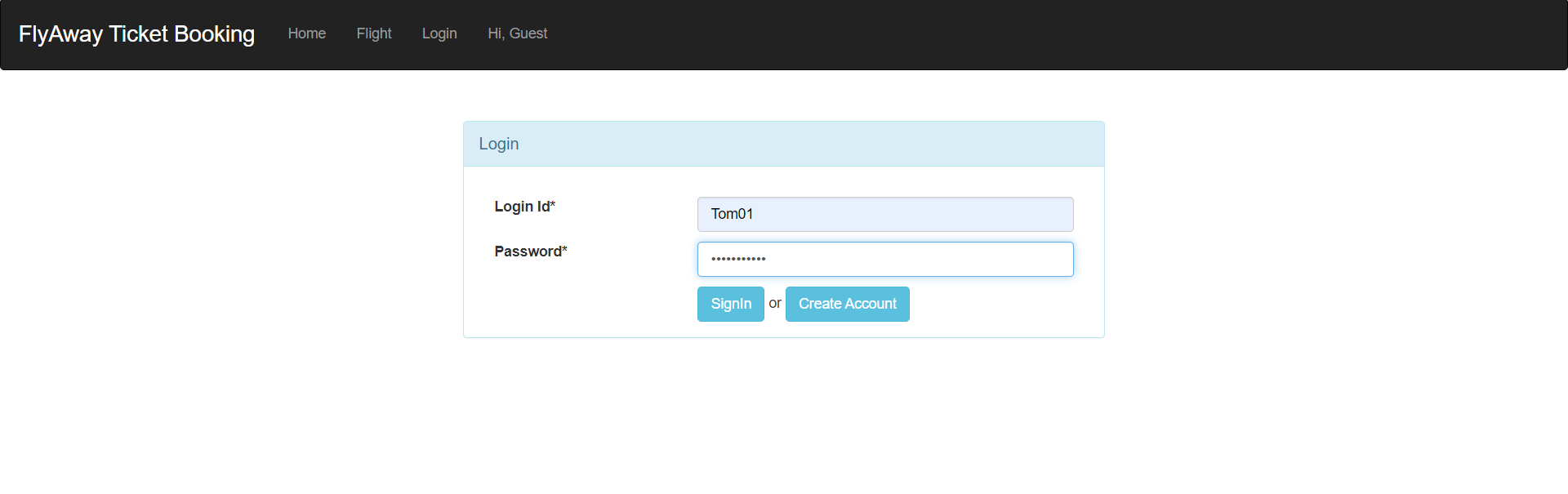
****



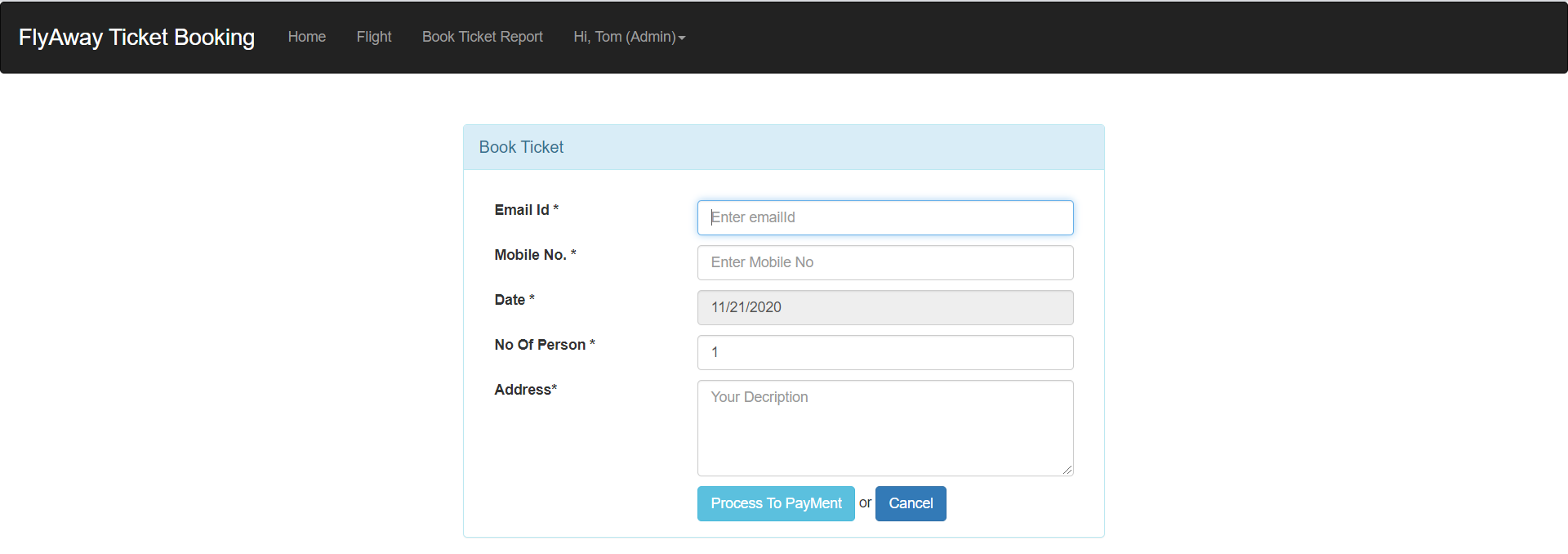


**Book a flight:**

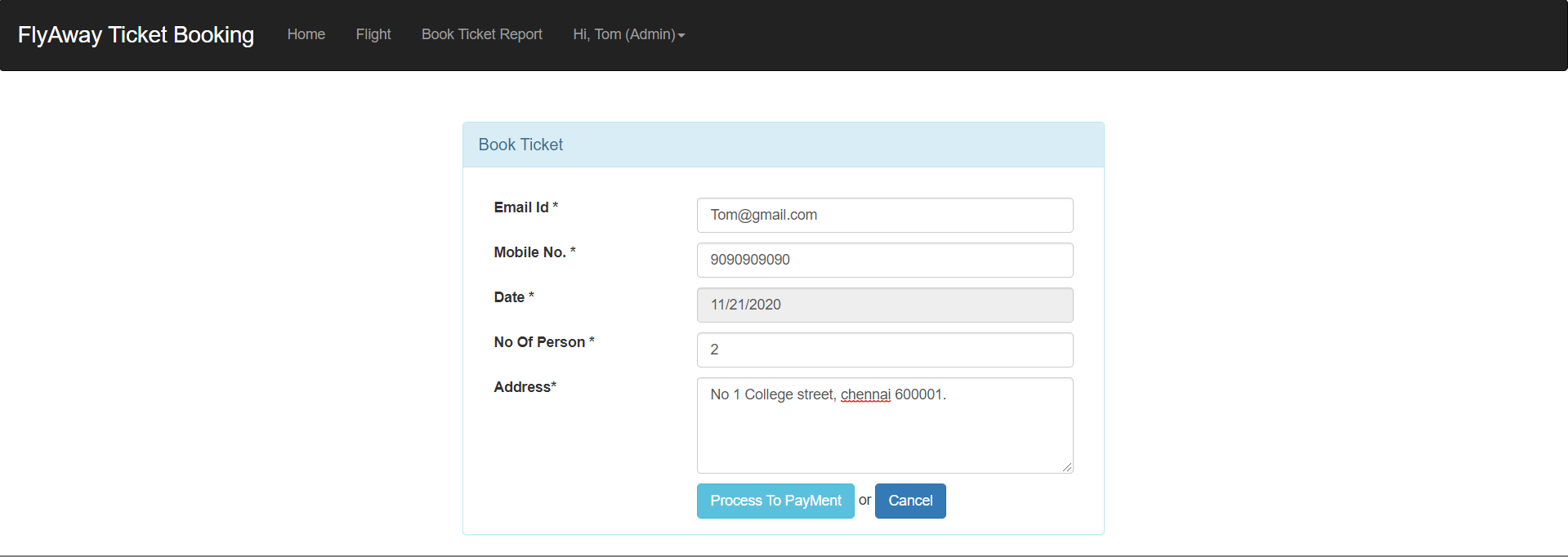
****

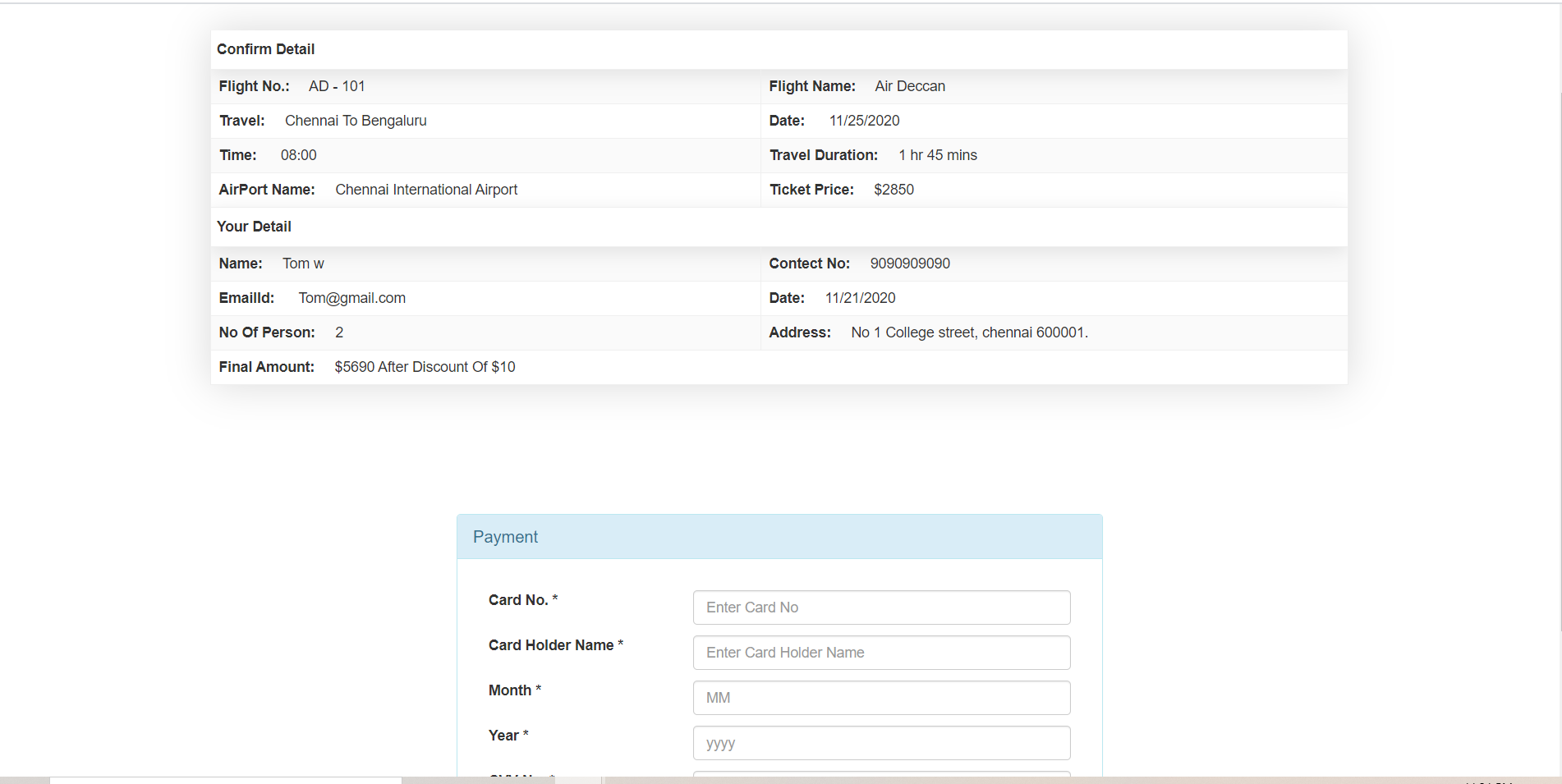
**User logs in to book flight ticket:**

**More user details to be entered:**

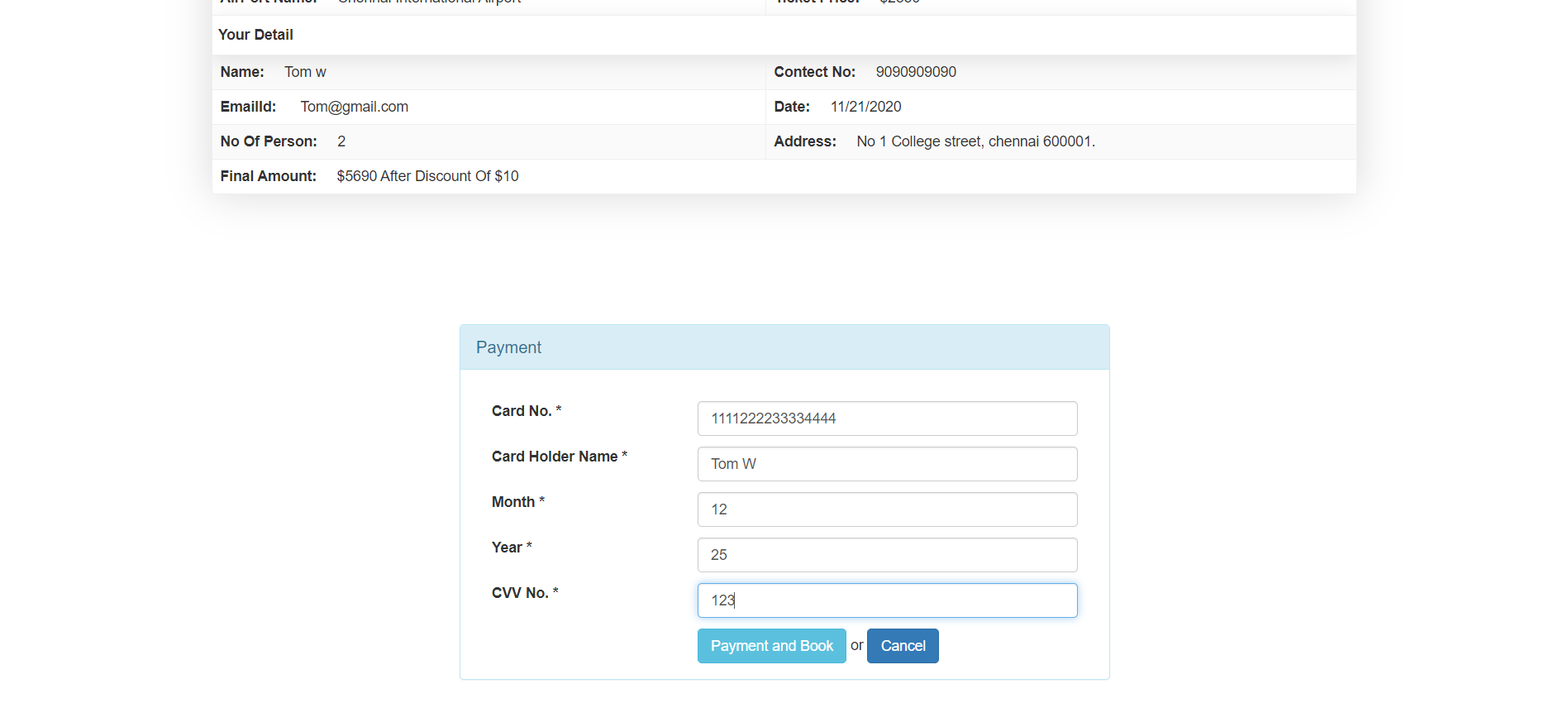
****

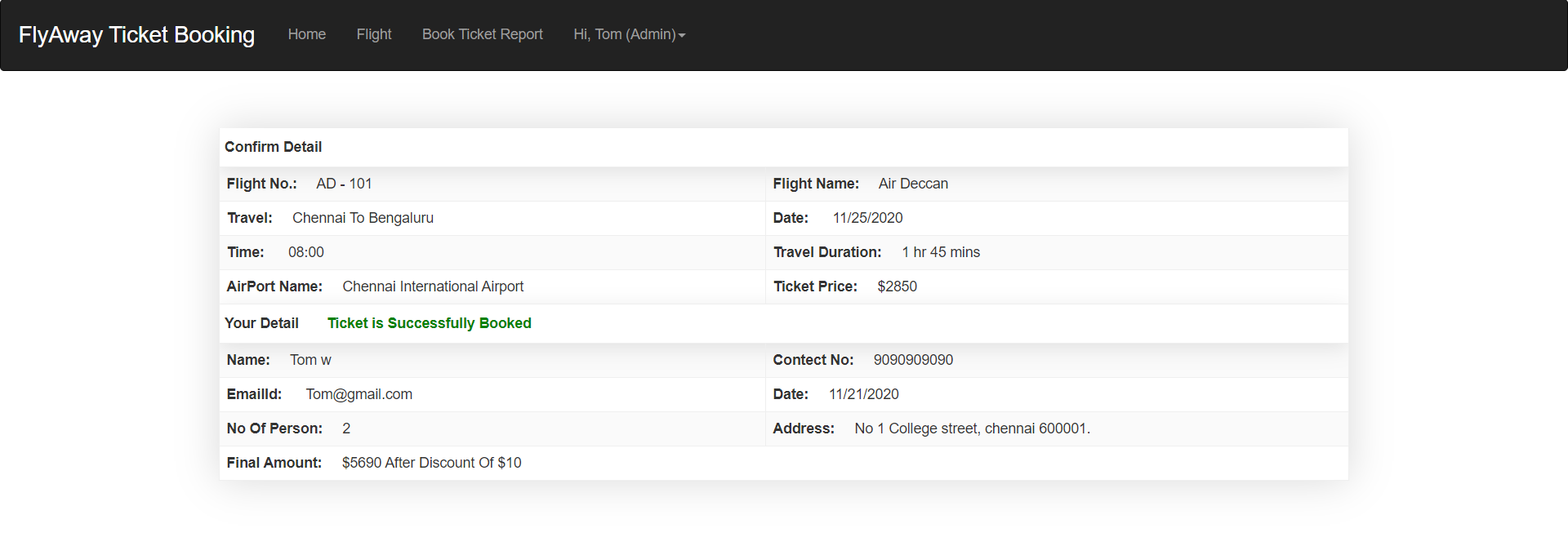
**Details entered by the user:**

****

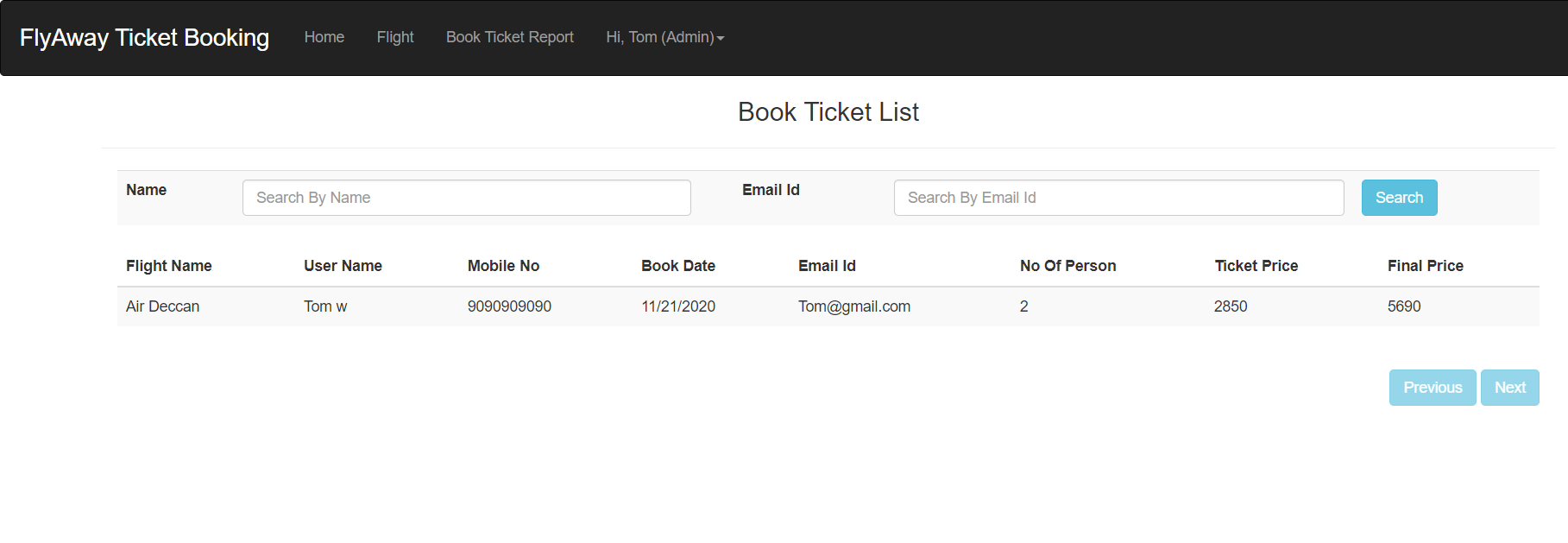
**Confirm details displayed:**

**Payment details entered:**

****

**Ticket is booked successfully:**

**Passenger with flight details:**

****

**Conclusion**

Though the system still containing lot of scope of improvement in it.

But its overall look and feel gives rough picture of on existing automation system and meets the company’s demand