

Online Flight Booking Management Portal

Background:

Airline Reservation System is a computerized system used to store and retrieve information and conduct transactions related to air travel. The project is aimed at exposing the relevance and importance of Airline Reservation Systems. It is projected towards enhancing the relationship between customers and airline agencies, and thereby making it convenient for the customers to book the flights as when they require such that they can utilize this software to make reservations. It incorporates airline schedules, fare tariffs, passenger reservations and ticket records. It is the primary form of travel agency computerization in the world. These systems manage the millions of reservation requests and cancellations, fare, and reservation pricing requests that are initiated by travel agencies using these systems—not to mention the thousands of database changes that occur daily.

Mission Statements / Objectives:

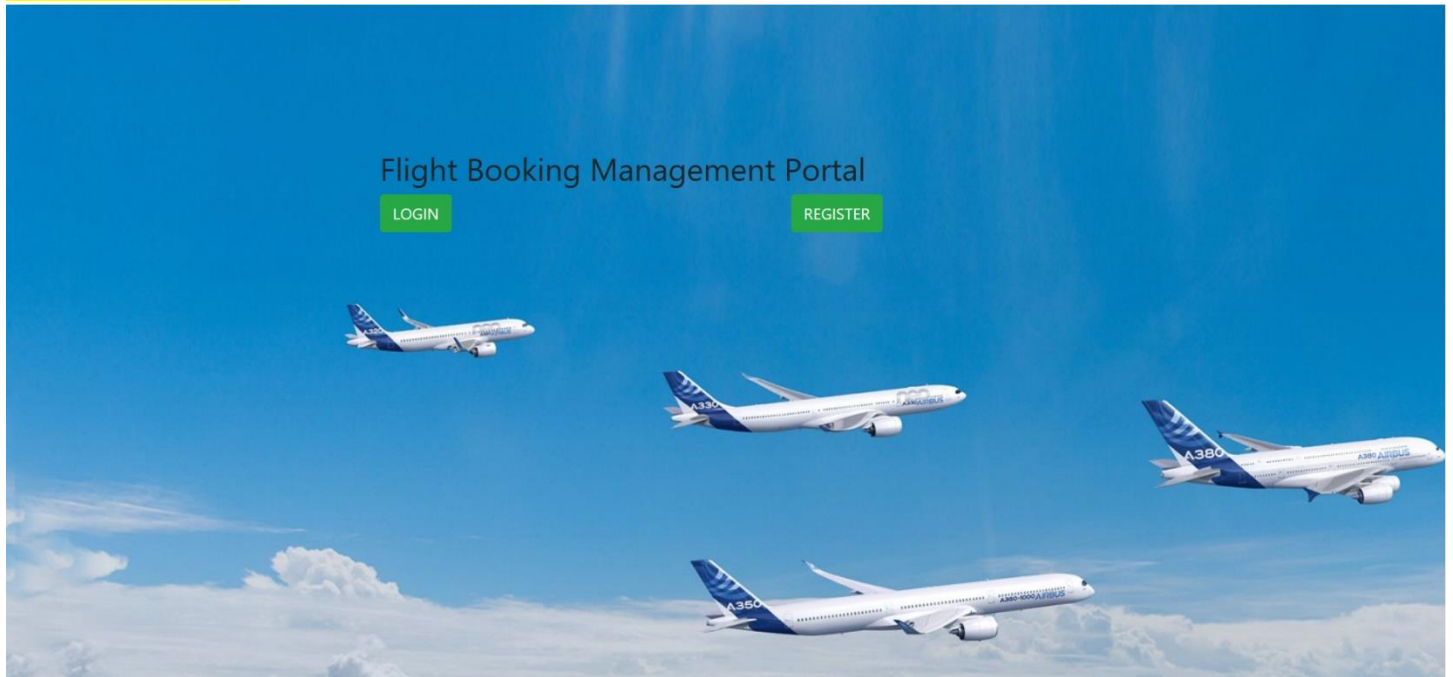
- Airline Reservation System is an implementation of a general Airline Ticketing website which helps the customers to search the availability and prices of various airline tickets, along with the different packages available with the reservations.
- The project has been planned to be having the view of distributed database architecture, with centralized storage of the database.
- It deals with monitoring the information and transactions of booking counter.
- The system allows the airline passenger to search for flights that are available between the two travel cities, namely the “Departure city” and “Arrival city” for a departure and arrival dates. The system displays all the flight’s details such as flight no, price
- After search, the system displays the list of available flights and allows customer to book the flight. Then the system checks for the availability of seats on the flight. If the seats are available, then the system allows the passenger to book a seat.

Technologies and Skills used to build the project:

- Java SE, Java EE, Spring MVC, Maven, Hibernate
- Database, SQL, MySQL Workbench
- HTML, CSS, Bootstrap
- Eclipse IDE, STS
- XML
- POJO, DAO, Controllers
- .xml and .hbm files for configuration

WEB Application Flow in terms of Web Pages

Landing Page:



Register Account Page:

The register account page features a blue sky background with four Airbus A350-900 aircraft flying. The registration form is located on the left side of the page. It includes fields for Phone, First Name, Last Name, Email, and Password, each with a corresponding label. A green "REGISTER" button is positioned below the Password field.

Phone:
8578694970

First Name:
Sagar

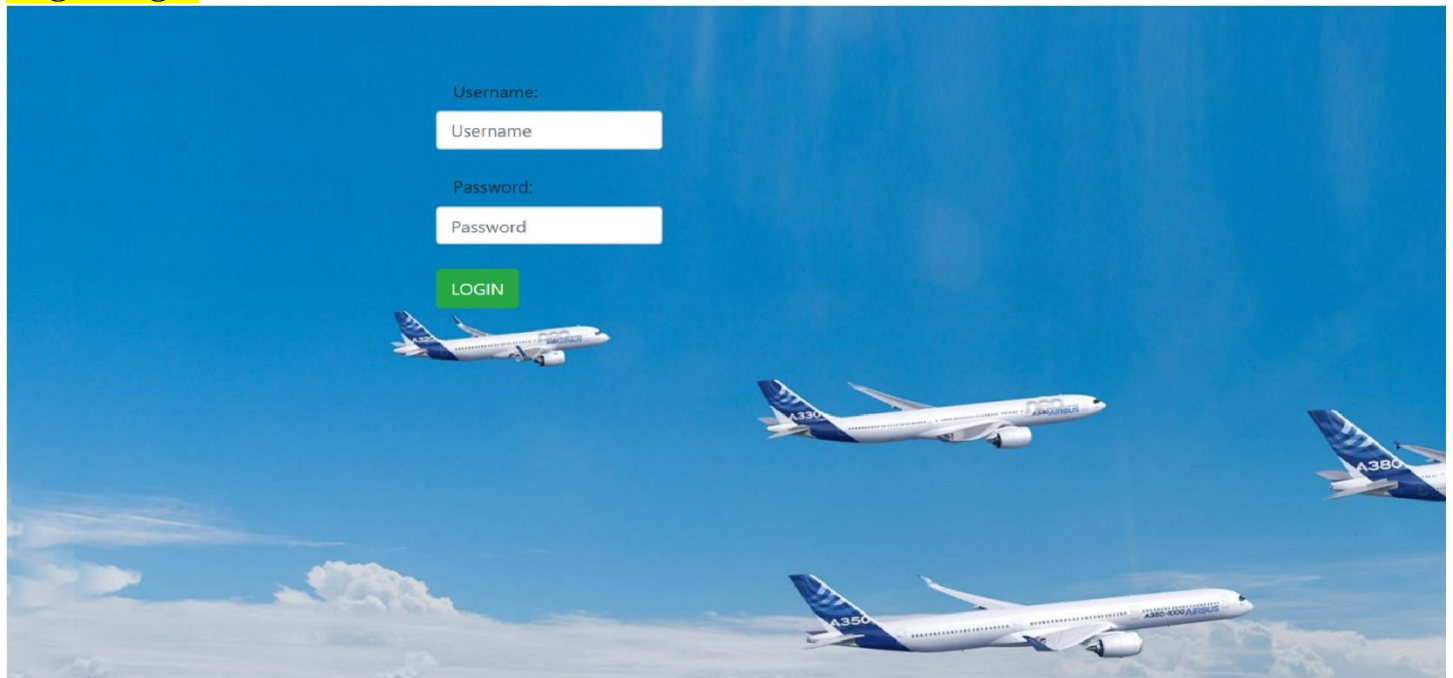
Last Name:
Shah

Email:
shah@neu.edu

Passsword:
•••••

REGISTER

Login Page:

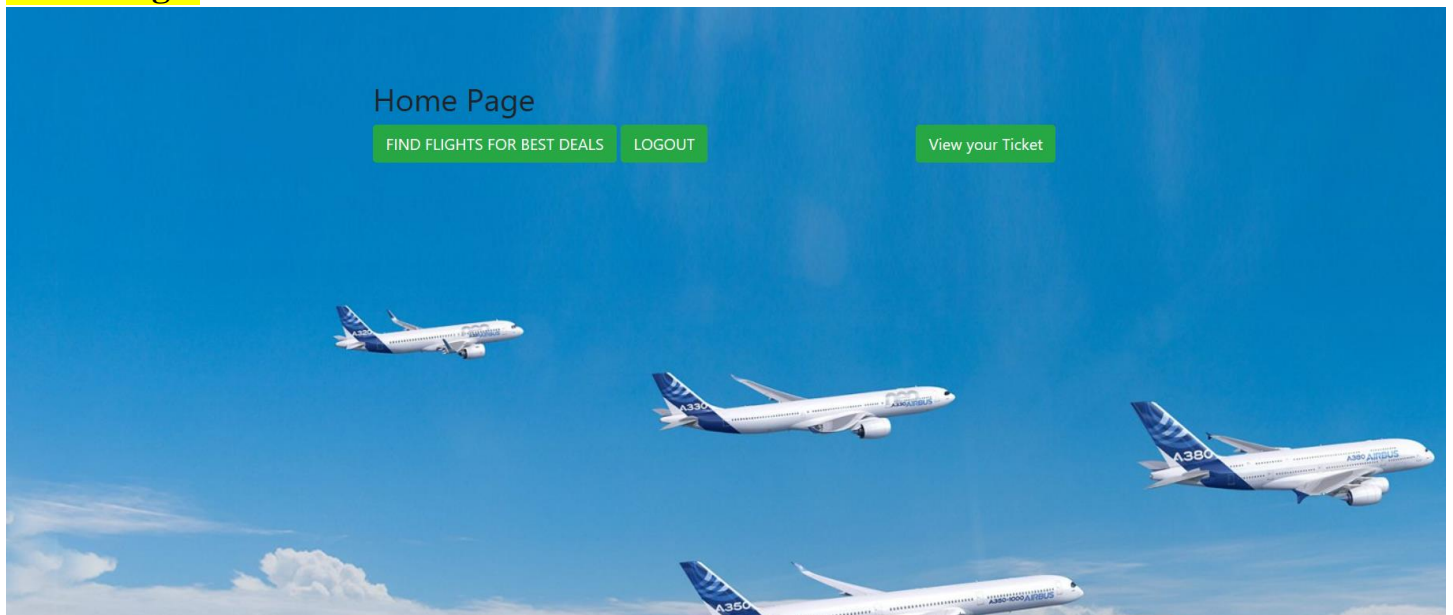
The background of the login page is a vibrant blue sky with scattered white clouds. Several white Airbus A380-800 aircraft with blue and white livery are shown in flight, angled upwards towards the right. The aircraft have 'A380' and 'AIRBUS A380-800' visible on their fuselages.

Username:

Password:

LOGIN

Home Page:

The background of the home page is identical to the login page, featuring a blue sky with white clouds and several Airbus A380-800 aircraft in flight.

Home Page

FIND FLIGHTS FOR BEST DEALS

LOGOUT

View your Ticket

Search Flights Page:

Search Flights by filling up the following parameters

Flying From

City or Airport

Flying To

City or Airport

Travel Date

Enter date in MM/DD/YYYY

Number of tickets

1

Serach Flights

LOGOUT

Search Results to display Flight Table:

Get flights with best deals

Search Results based on the filters

Flight Number	Source	Destination	Price
AE786	Boston	NY	500
BE786	Boston	NY	500

Enter Flight number :

Flight No.

Enter number of seats:

Seats

Enter phone number :

Phone No.

HOME

Book Flight

LOGOUT

View Ticket:

View your Ticket

Phone Number	First Name	Last Name	Flight Number	Departure from	Arrival at	Number of seats	Departure Date
1	Sagar	Shah	AE786	Boston	NY	2	05/23/2020

HOME

Web Application Flow in terms of Controller:

Home Controller.java: Handles requests for the application home page

```
package com.me.demo;

import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

/**
 * Handles requests for the application home page.
 */
@Controller
public class HomeController {

    private static final Logger logger = LoggerFactory.getLogger(HomeController.class);
    /**
     * Simply selects the home view to render by returning its name.
     */
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home(Locale locale, Model model) {
        logger.info("Welcome home! The client locale is {}.", locale);

        Date date = new Date();
```

```

        DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG,
DateFormat.LONG, locale);

        String formattedDate = dateFormat.format(date);

        model.addAttribute("serverTime", formattedDate );

        return "home";
    }
}

```

UserController.java: Handles Login and Registration of Users on the portal.Also maps various JSPs used accross various Controllers

```

package com.me.demo;

import java.util.List;

import javax.persistence.criteria.CriteriaBuilder;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.SimpleEmail;
import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Restrictions;
import org.hibernate.hql.internal.ast.tree.RestrictableStatement;
import org.hibernate.query.Query;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.SessionAttributes;

import com.me.pojo.User;
import com.me.dao.MessageDAO;
import com.me.dao.UserDAO;
import com.me.pojo.Message;

/**
 * Handles Login and Registration of Users on the portal.Also maps various JSPs used accross
 various Controllers
 */

@Controller
public class UserController {
    @RequestMapping(value = "/option.htm", method = RequestMethod.GET)
    public String showOptionPage() {
        return "view-option";
    }

    @RequestMapping(value = "/home.htm", method = RequestMethod.GET)
    public String showHomePage() {

```

```

        return "home";
    }

    @RequestMapping(value = "/user.htm", method = RequestMethod.GET)
    public String showLogin() {
        return "user-login";
    }

    @RequestMapping(value = "/search_flight.htm", method = RequestMethod.GET)
    public String showFlightPage() {
        return "user-view";
    }

    @RequestMapping(value = "/view.htm", method = RequestMethod.GET)
    public String enterNumberToShowTicket() {
        return "enter-number";
    }

    @RequestMapping(value = "/register.htm", method = RequestMethod.GET)
    public String showRegister() {
        return "user-register";
    }

    @RequestMapping(value = "/user.htm", method = RequestMethod.POST)
    public String handleRequest(HttpServletRequest request, UserDao userDao, MessageDAO msgdao)
    {
        String username = request.getParameter("un");
        String userpswd = request.getParameter("up");
        User user = userDao.getUser(username, userpswd);
        if(user == null) {
            request.setAttribute("errmsg", "User/Password combination error");
            return "login-error";
        }
        List<Message> msglist = msgdao.getMessageList(user.getUserName());
        request.setAttribute("msglist", msglist);
        request.setAttribute("user", user);
        return "view-option";
    }

    @RequestMapping(value = "/register.htm", method = RequestMethod.POST)
    public String handleRequest(HttpServletRequest request, UserDao userDao) {
        long phone_number = Long.parseLong(request.getParameter("ph"));
        String username = request.getParameter("un");
        String lastname = request.getParameter("ln");
        String userpswd = request.getParameter("up");
        String email = request.getParameter("em");
        int user = userDao.addUser(phone_number, username, lastname, userpswd, email);
        if(user == 1) {
            return "registration-successfull";
        }else {
            request.setAttribute("errmsg", "Enter username already exists");
            return "error-page";
        }
    }
}

```

SearchFlightController.java: Searches various Flights based on various parameters

```

package com.me.demo;
import java.util.List;

import javax.persistence.criteria.CriteriaBuilder;
import javax.servlet.http.HttpServletRequest;
import javax.swing.JOptionPane;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Restrictions;
import org.hibernate.hql.internal.ast.tree.RestrictableStatement;
import org.hibernate.query.Query;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.me.pojo.User;
import com.me.dao.FlightDAO;
import com.me.dao.MessageDAO;
import com.me.dao.UserDAO;
import com.me.pojo.Flight;
import com.me.pojo.Message;

/**
 * Searches various Flights based on various parameters
 */

@Controller
public class SearchFlightController {

    @RequestMapping(value = "/search_flight.htm", method = RequestMethod.POST)
    public String handleRequest(HttpServletRequest request, FlightDAO flightdao) {
        String source = request.getParameter("source");
        System.out.println(source);
        String destination = request.getParameter("destination");
        String flight_date = request.getParameter("travel_date");
        int seat_count = Integer.parseInt(request.getParameter("ticket_count"));
        System.out.println("source: " + source + " destination: " + destination + " flight_date: "
            + flight_date + " seat_count: " + seat_count);
        List<Flight> flightList = flightdao.searchFlights(source, destination, flight_date);
        System.out.println(flightList);
        if(flightList == null) {
            request.setAttribute("errmsg", "No flight found for search criteria");
            return "error-page";
        } else {
            for(Flight flg : flightList) {
                System.out.println(flg.getFlight_number() + " ");
            }
            request.setAttribute("searchflightList", flightList);
            return "search-results";
        }
    }
}

```

SearchResultsController.java: Displays Flight Table depending on the search criteria


```

package com.me.demo;

import java.util.List;
import javax.persistence.criteria.CriteriaBuilder;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.swing.JOptionPane;
import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Restrictions;
import org.hibernate.hql.internal.ast.tree.RestrictableStatement;
import org.hibernate.query.Query;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import com.me.pojo.User;
import com.me.dao.BookTicketDAO;
import com.me.dao.FlightDAO;
import com.me.dao.MessageDAO;
import com.me.dao.UserDAO;
import com.me.pojo.Flight;
import com.me.pojo.Message;
import com.me.pojo.TicketDetails;
import com.me.demo.UserController;

/**
 * Displays Flight Table depending on the search criteria
 */
@Controller
public class SearchResultsController {
    @RequestMapping(value = "/results.htm", method = RequestMethod.POST)
    public String handleRequest(HttpServletRequest request, UserDAO userdao, FlightDAO
flightdao, BookTicketDAO bookticketdao) {
        String flight_number = request.getParameter("flight_number");
        int number_of_seats = Integer.parseInt(request.getParameter("seats"));
        long phone_number = Long.parseLong(request.getParameter("ph_number"));
        User user = bookticketdao.getUser(phone_number);
        System.out.println("user: "+user);
        if(user == null) {
            request.setAttribute("errmsg", "Error occured return to home page and try
again");
            return "error-page";
        }
        String username = user.getUsername();
        String lastname = user.getLname();
        System.out.println("username: "+username+" "+lastname);
        Flight flight = bookticketdao.getFlightDetails(flight_number);
        System.out.println("flight: "+flight);
        if(flight == null) {
            request.setAttribute("errmsg", "Error occured return to home page and try
again");
            return "error-page";
        }
        String source = flight.getSource();
        String destination = flight.getDestination();

```

```

        String travel_date = flight.getFlight_date();
        System.out.println("flight_number: "+flight_number+" source: "+source);
        System.out.println("phone_number: "+phone_number+" username: "+username+" lastname: "+lastname+" flight_number: "+flight_number+" source: "+source+" destination: "+destination+" number_of_seats: "+number_of_seats+" travel_date: "+travel_date );

        int ticket_details = bookticketdao.addTicketDetails(phone_number, username, lastname, flight_number, source, destination, number_of_seats, travel_date);
        System.out.println("ticketdetails: "+ticket_details);
        if(ticket_details == 1) {
            List<TicketDetails> ticketdetails = bookticketdao.getTicketDetails(phone_number);
            request.setAttribute("ticketdetails", ticketdetails);
            return "view-ticket";
        }else {
            request.setAttribute("errmsg", "not able to book ticket return to home page");
            return "error-page";
        }
    }
}

```

ShowTicketController.java: Displays Flight ticket of the user based on the phone number

```

package com.me.demo;

import java.util.List;

import javax.persistence.criteria.CriteriaBuilder;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Restrictions;
import org.hibernate.hql.internal.ast.tree.RestrictableStatement;
import org.hibernate.query.Query;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.SessionAttributes;

import com.me.pojo.User;
import com.me.dao.BookTicketDAO;
import com.me.dao.MessageDAO;
import com.me.dao.UserDAO;
import com.me.pojo.Flight;
import com.me.pojo.Message;
import com.me.pojo.TicketDetails;

```

```

@Controller
public class ShowTicketController {

    @RequestMapping(value = "/ticket.htm", method = RequestMethod.POST)

```

```
public String handleRequest(HttpServletRequest request, BookTicketDAO bookticketdao) {  
    long phone_number = Long.parseLong(request.getParameter("ph_number"));  
    System.out.println("ph: "+phone_number);  
    List<TicketDetails> ticketdetails = bookticketdao.getTicketDetails(phone_number);  
    if(ticketdetails == null) {  
        request.setAttribute("errmsg", "Phone number incorrect");  
        return "error-page";  
    }else {  
        for(TicketDetails t : ticketdetails) {  
            System.out.println(t.getFlight_number()+" ");  
        }  
        request.setAttribute("ticketdetails", ticketdetails);  
        return "view-ticket";  
    }  
}  
  
}
```