**DOCUMENTATION**

**Project Title : Airlines Booking System**

## Overview

The Airlines Booking System is a web application designed to facilitate flight booking for passengers. This system allows users to search for flights, view available seats, and complete booking transactions. The application provides an intuitive interface for users and a back-end system for airline management.

## Features

* **User Registration**: Passengers can sign up and log into the system.
* **Flight Search**: Users can search flights based on destination, dates, and other preferences.
* **Booking Management**: Users can book, modify, or cancel flights.
* **Payment Gateway Integration**: Secure payment integration to complete booking.
* **Admin Panel**: Airline administrators can manage flights, view booking details, and generate reports.
* **Seat Reservation**: Users can select and reserve seats on available flights.
* **Flight Scheduling**: Admins can add, edit, and delete flight schedules.

## Technologies Used

* **Frontend**:
  + HTML, CSS, JavaScript
  + JSP (JavaServer Pages) for dynamic content rendering
* **Backend**:
  + Java (JSP, Servlet)
  + JDBC (Java Database Connectivity) for database interaction
* **Database**:
  + MySQL or PostgreSQL for data storage
* **Others**:
  + Apache Tomcat for server-side processing

## Installation

### Prerequisites

* **JDK** (Java Development Kit) installed on your machine.
* **Apache Tomcat** server.
* **MySQL Database** setup for storing user and booking information.

### Steps to Run the Application

1. Clone the repository:

git clone https://github.com/vanshikagupta80455/localrepo.git

2.Configure the MySQL database:

* 1. Create a database named airlines\_system.
  2. Run the SQL scripts provided in the database/ folder to create necessary tables.

3. Configure JDBC in your application by modifying the dbconfig.properties file to connect to your MySQL database.

4. Deploy the application on Apache Tomcat:

* 1. Copy the build/web/ folder contents into the Tomcat's webapps directory.
  2. Start the Tomcat server.

1. Access the application at http://localhost:8080/airlines\_system.

## File Structure

Here is the basic structure of the project:

/airlines\_system

│

├── /build

│ ├── /web

│ │ ├── home.jsp # Main homepage for users

│ │ ├── login.jsp # Login page

│ │ ├── booking.jsp # Flight booking page

│ │ └── ... # Other JSP pages

│ └── ... # Compiled project files

│

├── /src

│ ├── /com

│ │ ├── /controller # Servlets for handling user requests

│ │ ├── /dao # Data Access Objects for database interaction

│ │ └── /model # Java classes representing database entities

│ └── ...

│

└── /database

├── create\_tables.sql # SQL script to create necessary tables

└── dbconfig.properties # Configuration for database connection

## Key Components

### 1. Home Page (home.jsp)

This page is the entry point of the application. It allows users to search for flights based on their travel dates, destination, and other preferences. The main features of the homepage include:

* **Flight Search Form**: Users can input their flight details, such as origin, destination, travel date, and number of passengers.
* **Flight Listings**: After submitting the search form, users are shown available flights matching their criteria.

### 2. Login and Registration (login.jsp, register.jsp)

These pages allow users to create a new account or log in to an existing account. Registered users can access their booking history, while new users can sign up for an account to start booking flights.

### 3. Flight Booking (booking.jsp)

Once a user selects a flight from the search results, they are redirected to this page to confirm their booking, select seats, and enter payment details.

### 4. Admin Panel

The admin panel allows airline administrators to:

* View and manage flight schedules.
* Add, edit, and delete flights.
* View booking history and manage reservations.

## Database Schema

Here is the database schema used by the Airlines Booking System:

* **Users**: Stores user details like name, email, phone number, etc.
* **Flights**: Stores details of each flight such as flight number, origin, destination, date, and available seats.
* **Bookings**: Records of booked flights, including user ID, flight ID, booking date, and payment status.

CREATE TABLE Users (

user\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

phone VARCHAR(15),

password VARCHAR(255)

);

CREATE TABLE Flights (

flight\_id INT PRIMARY KEY AUTO\_INCREMENT,

flight\_number VARCHAR(20),

origin VARCHAR(50),

destination VARCHAR(50),

departure\_date DATETIME,

available\_seats INT

);

CREATE TABLE Bookings (

booking\_id INT PRIMARY KEY AUTO\_INCREMENT,

user\_id INT,

flight\_id INT,

booking\_date DATETIME,

payment\_status VARCHAR(20),

FOREIGN KEY (user\_id) REFERENCES Users(user\_id),

FOREIGN KEY (flight\_id) REFERENCES Flights(flight\_id)

);

## Contributing

If you would like to contribute to this project, feel free to fork the repository and create a pull request. Please ensure that your code adheres to the project's coding standards and passes any existing unit tests.

## License

This project is licensed under the MIT License - see the LICENSE file for details.