Project Specification Document

Project objective:

To design and develop an airline booking portal named FlyAway.

Background of the problem statement:

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

The website will 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:

1. Login Page where user can login with his credentials and with two options called Register and change password.
2. Register page is a page where a new user can register and create an account. Filling and submitting will insert data to back-end table called logindetails.
3. Change password page is a page where a user can change his/her password. If the username and current password entered is there in the back-end table, then it will update that record's password with the new password which is entered in the new password field.
4. Upon logging in with valid credentials, applications validate whether username and password entered are available in the table or not, if record is available, then it will user to Airlines portal page.
5. AirLine portal page will ask user to enter Date of travel, source, destination and number of passengers. Based on those details, the application will fetch the available flight details.
6. After clicking on the book seats button, the user will be taken to enter the passenger details.
7. Based on the number of passengers entered on AirLiines portal page, the passenger details filling page is iterated to ask for passenger details.
8. If the user entered 2 passengers, it would ask to fill in the passenger details. After clicking on continue, the application will insert the passenger details into the booking table and take to a page where he can click on the link to fill 2nd passenger details.
9. After filling in the 2nd passenger details, it will ask to book or pay to book the tickets.
10. After clicking on the book button, the application will take us to the confirmation page where he/she can find booking ID, price etc.
11. After getting the confirmation, the application will go and update the backend table flights to reduce the number of seats available.
12. Logout option to take users to login page again.
13. In AirLines portal page, user will have an option to check the booked ticket details with booking id.

Logic:

1. If user entered 3 passengers, then wrote the logic to iterate the details filling page for 3 times and insert booking id same for all 3 passengers.
2. Seat numbers are allocated in an incremental manner.
3. After clicking on book button, application will update the backend table flights, I.e. it reduces the seats available based on the passengers entered to book the tickets.

Concepts Used:

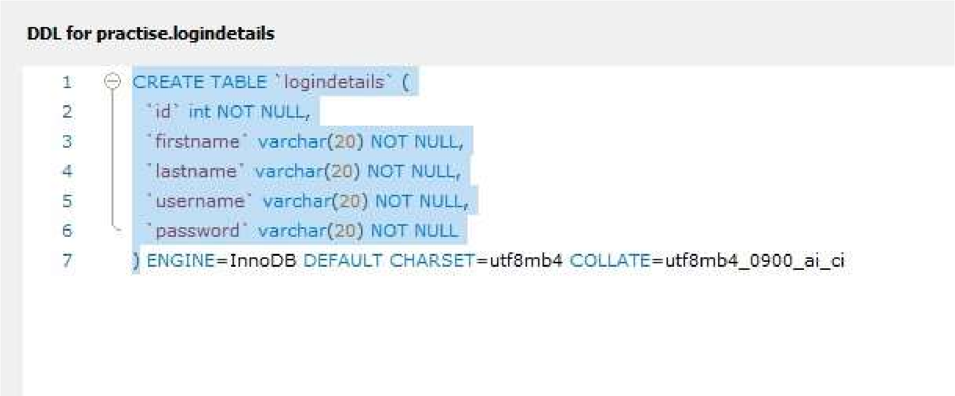
1. Java Core concepts
2. Servlets
3. Session login and logout
4. Hibernate XML concept
5. Persistence class
6. Mapping xml
7. Configuration xml
8. Servlet context
9. Servlet Request dispatcher.
10. Exceptions
11. HttpServlet Request
12. HttpServlet Response
13. SQL
14. HQL
15. DDL statements in SQL
16. Tomcat

Tables used:

Logindetails: Table is used to insert the data from register page.

Update the records if user wants to change his/her password.

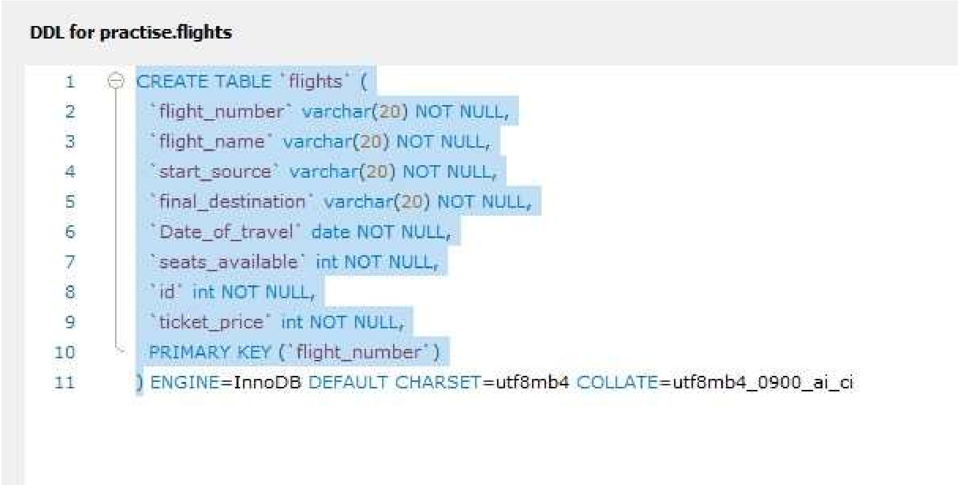
Extract and validate for login.



DDL for practise.logindetails

```
CREATE TABLE `logindetails` (
  `id` int NOT NULL,
  `firstname` varchar(20) NOT NULL,
  `lastname` varchar(20) NOT NULL,
  `username` varchar(20) NOT NULL,
  `password` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

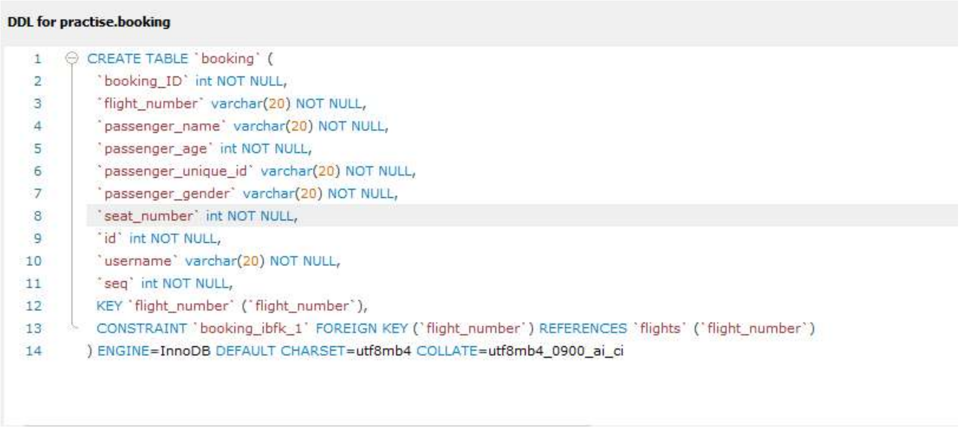
Flights: Table is used to store the flights details from where application fetches the data to show available flights bases on user selection of source and destination.



DDL for practise.flights

```
CREATE TABLE `flights` (
  `flight_number` varchar(20) NOT NULL,
  `flight_name` varchar(20) NOT NULL,
  `start_source` varchar(20) NOT NULL,
  `final_destination` varchar(20) NOT NULL,
  `Date_of_travel` date NOT NULL,
  `seats_available` int NOT NULL,
  `id` int NOT NULL,
  `ticket_price` int NOT NULL,
  PRIMARY KEY (`flight_number`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Booking: Table is used to store the passenger details.

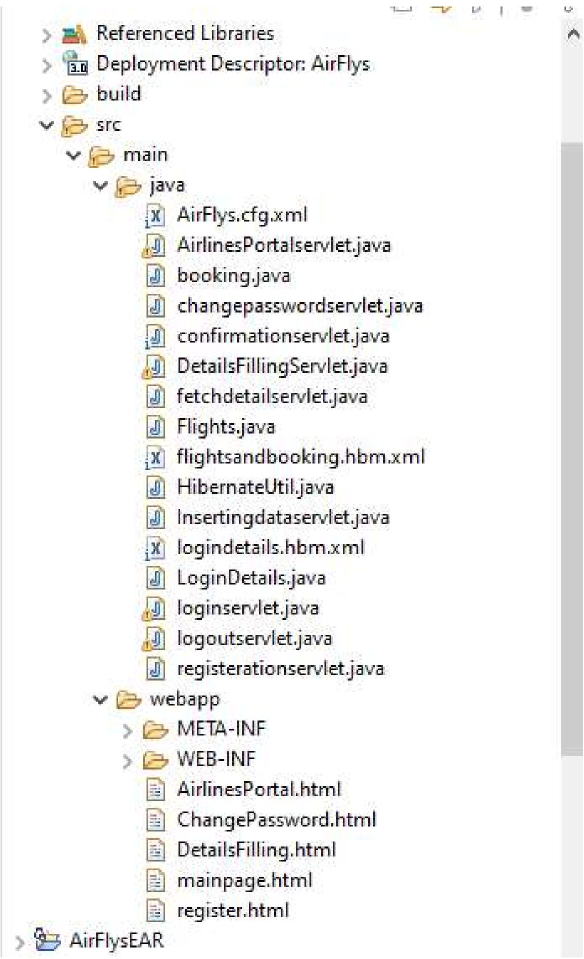


DDL for practise.booking

```
CREATE TABLE `booking` (
  `booking_ID` int NOT NULL,
  `flight_number` varchar(20) NOT NULL,
  `passenger_name` varchar(20) NOT NULL,
  `passenger_age` int NOT NULL,
  `passenger_unique_id` varchar(20) NOT NULL,
  `passenger_gender` varchar(20) NOT NULL,
  `seat_number` int NOT NULL,
  `id` int NOT NULL,
  `username` varchar(20) NOT NULL,
  `seq` int NOT NULL,
  KEY `flight_number` (`flight_number`),
  CONSTRAINT `booking_ibfk_1` FOREIGN KEY (`flight_number`) REFERENCES `flights` (`flight_number`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Attaching DDL scripts in the code section.

HTML pages and servlets:



```
> Referenced Libraries
> Deployment Descriptor: AirFlys
> build
v src
  v main
    v java
        AirFlys.cfg.xml
        AirlinesPortalservlet.java
        booking.java
        changepasswordservlet.java
        confirmationservlet.java
        DetailsFillingServlet.java
        fetchdetailservlet.java
        Flights.java
        flightsandbooking.hbm.xml
        HibernateUtil.java
        Insertingdataservlet.java
        logindetails.hbm.xml
        LoginDetails.java
        loginservlet.java
        logoutservlet.java
        registerationservlet.java
    v webapp
      > META-INF
      > WEB-INF
        AirlinesPortal.html
        ChangePassword.html
        DetailsFilling.html
        mainpage.html
        register.html
> AirFlysEAR
```

Link to the GitHub repository

https://github.com/rvanjali007/AirFlys.git

Conclusion:

Flyaway airlines portal for booking the flight tickets. Application is comfortable for the user and easy to understand. Application have features like login, register for a new account, Change password for an existing account. Application will fetch the details and display the available flights based on the user selection. Application will ask for passenger details and store them in the table to auto generate the same booking id for all passengers booked by one user. At the end after booking, the user will get confirmation with the booking id and will have an option to logout. User can also look for already booked ticket details from Airlines portal page with the booking id.