Phase2 Project: Airline Booking Portal

Name :Regina

Surname : Machava

Full Stack Developer

Source Code:

Classes:

**package** com.airline.;

**public** **class** Admin {

**private** **int** adminId;

**private** String firstName;

**private** String lastName;

**private** String email;

**private** String password;

**public** **int** getAdminId() {

**return** adminId;

}

**public** **void** setAdminId(**int** adminId) {

**this**.adminId = adminId;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

@Override

**public** String toString() {

**return** "Admin [adminId=" + adminId + ", firstName=" + firstName + ", lastName=" + lastName + ", email=" + email

+ ", password=" + password + "]";

}

}

**package** com.airline;

**public** **class** Airport {

**private** String airportCode;

**private** String airport;

**private** String countryCode;

**private** String country;

**public** String getAirportCode() {

**return** airportCode;

}

**public** **void** setAirportCode(String airportCode) {

**this**.airportCode = airportCode;

}

**public** String getAirport() {

**return** airport;

}

**public** **void** setAirport(String airport) {

**this**.airport = airport;

}

**public** String getCountryCode() {

**return** countryCode;

}

**public** **void** setCountryCode(String countryCode) {

**this**.countryCode = countryCode;

}

**public** String getCountry() {

**return** country;

}

**public** **void** setCountry(String country) {

**this**.country = country;

}

@Override

**public** String toString() {

**return** "Airport [airportCode=" + airportCode + ", airport=" + airport + ", countryCode=" + countryCode

+ ", country=" + country + "]";

}

}

**public** **class** Customer {

**private** **int** customerId;

**private** String firstName;

**private** String lastName;

**private** String email;

**private** String password;

**private** String phone;

**public** **int** getCustomerId() {

**return** customerId;

}

**public** **void** setCustomerId(**int** customerId) {

**this**.customerId = customerId;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getPhone() {

**return** phone;

}

**public** **void** setPhone(String phone) {

**this**.phone = phone;

}

@Override

**public** String toString() {

**return** "Customer [customerId=" + customerId + ", firstName=" + firstName + ", lastName=" + lastName + ", email="

+ email + ", password=" + password + ", phone=" + phone + "]";

}

}

**public** **class** Fare {

**private** **int** flightNumber;

**private** String travelClass;

**private** **double** fare;

**public** **int** getFlightNumber() {

**return** flightNumber;

}

**public** **void** setFlightNumber(**int** flightNumber) {

**this**.flightNumber = flightNumber;

}

**public** String getTravelClass() {

**return** travelClass;

}

**public** **void** setTravelClass(String travelClass) {

**this**.travelClass = travelClass;

}

**public** **double** getFare() {

**return** fare;

}

**public** **void** setFare(**double** fare) {

**this**.fare = fare;

}

@Override

**public** String toString() {

**return** "Fare [flightNumber=" + flightNumber + ", travelClass=" + travelClass + ", fare=" + fare + "]";

}

}

**public** **class** Flight {

**private** **int** flightNumber;

**private** String airline;

**private** String weekdays;

**private** String source;

**private** String destination;

**public** **int** getFlightNumber() {

**return** flightNumber;

}

**public** **void** setFlightNumber(**int** flightNumber) {

**this**.flightNumber = flightNumber;

}

**public** String getAirline() {

**return** airline;

}

**public** **void** setAirline(String airline) {

**this**.airline = airline;

}

**public** String getWeekdays() {

**return** weekdays;

}

**public** **void** setWeekdays(String weekdays) {

**this**.weekdays = weekdays;

}

**public** String getSource() {

**return** source;

}

**public** **void** setSource(String source) {

**this**.source = source;

}

**public** String getDestination() {

**return** destination;

}

**public** **void** setDestination(String destination) {

**this**.destination = destination;

}

@Override

**public** String toString() {

**return** "Flight [flightNumber=" + flightNumber + ", airline=" + airline + ", weekdays=" + weekdays + ", source="

+ source + ", destination=" + destination + "]";

}

}

**package** com.arline;

**import** java.util.Date;

**public** **class** Reservation {

**private** **int** bookingId;

**private** **int** flightNumber;

**private** String travelClass;

**private** Date travelDate;

**private** **int** passengers;

**private** **double** totalFare;

**private** **int** customerId;

**public** Reservation() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** Reservation(**int** flightNumber, String travelClass, Date travelDate, **int** passengers, **double** totalFare,

**int** customerId) {

**super**();

**this**.flightNumber = flightNumber;

**this**.travelClass = travelClass;

**this**.travelDate = travelDate;

**this**.passengers = passengers;

**this**.totalFare = totalFare;

**this**.customerId = customerId;

}

**public** **int** getBookingId() {

**return** bookingId;

}

**public** **void** setBookingId(**int** bookingId) {

**this**.bookingId = bookingId;

}

**public** **int** getFlightNumber() {

**return** flightNumber;

}

**public** **void** setFlightNumber(**int** flightNumber) {

**this**.flightNumber = flightNumber;

}

**public** String getTravelClass() {

**return** travelClass;

}

**public** **void** setTravelClass(String travelClass) {

**this**.travelClass = travelClass;

}

**public** Date getTravelDate() {

**return** travelDate;

}

**public** **void** setTravelDate(Date travelDate) {

**this**.travelDate = travelDate;

}

**public** **int** getPassengers() {

**return** passengers;

}

**public** **void** setPassengers(**int** passengers) {

**this**.passengers = passengers;

}

**public** **double** getTotalFare() {

**return** totalFare;

}

**public** **void** setTotalFare(**double** totalFare) {

**this**.totalFare = totalFare;

}

**public** **int** getCustomerId() {

**return** customerId;

}

**public** **void** setCustomerId(**int** customerId) {

**this**.customerId = customerId;

}

@Override

**public** String toString() {

**return** "Reservation [bookingId=" + bookingId + ", flightNumber=" + flightNumber + ", travelClass=" + travelClass

+ ", travelDate=" + travelDate + ", passengers=" + passengers + ", totalFare=" + totalFare

+ ", customerId=" + customerId + "]";

}

}

Servlets:

package AdminSev;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import com.airlinel.Admin;

import com.airline.Fare;

import com.airline.Flight;

public class Admin {

private Connection con = null;

private PreparedStatement pst = null;

public int addFlight(Flight flight) {

int flightNumber = 0;

String sql = "insert into flight (airline , weekdays , src\_airport\_code , dest\_airport\_code) "

+ " values(? , ? ,? , ?)";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql , Statement.RETURN\_GENERATED\_KEYS);

pst.setString(1, flight.getAirline());

pst.setString(2, flight.getWeekdays());

pst.setString(3, flight.getSource());

pst.setString(4, flight.getDestination());

if (pst.executeUpdate() == 1) {

ResultSet rs = pst.getGeneratedKeys();

if(rs.next()) {

flightNumber = rs.getInt(1);

}

}else {

flightNumber = 0;

}

} catch (SQLException e) {

flightNumber = 0;

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return flightNumber;

}

public String addFare(Fare fare) {

String status = "";

String sql = "insert into fare (flight\_number , class , fare) "

+ " values(? , ? ,?)";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, fare.getFlightNumber());

pst.setString(2, fare.getTravelClass());

pst.setDouble(3, fare.getFare());

if (pst.executeUpdate() == 1) {

status = "SUCCESS";

}else {

status = "FAIL";

}

} catch (SQLException e) {

status = "FAIL";

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return status;

}

public String updatePasswordAdmin(int Id , String password) {

String sql = "update admin set password=? WHERE admin\_id = ?";

String status = "";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, password);

pst.setInt(2, Id);

if (pst.executeUpdate() == 1) {

status = "SUCCESS";

} else

status = "FAIL";

} catch (Exception e) {

status = "FAIL";

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return status;

}

public int adminLogin(String email , String password) {

String sql = "select \* from admin where email = ? and password = ?";

int adminId = 0;

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, email);

pst.setString(2, password);

ResultSet rs = pst.executeQuery();

if(rs.next()) {

adminId = rs.getInt(1);

}else {

adminId = 0;

}

} catch (SQLException e) {

adminId = 0;

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return adminId;

}

public Admin getAdmin(int adminId) {

String sql = "select \* from admin where admin\_id = ?";

Admin admin = new Admin();

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, adminId);

ResultSet rs = pst.executeQuery();

rs.next();

admin.setAdminId(rs.getInt(1));

admin.setFirstName(rs.getString(2));

admin.setLastName(rs.getString(3));

admin.setEmail(rs.getString(4));

admin.setPassword(rs.getString(5));

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return admin;

}

}

package com.AdminSev;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import com.airline.Airport;

import com.airline.Customer;

import com.airline.Fare;

import com.airline.Flight;

import com.airline.Reservation;

public class Customer {

private Connection con = null;

private PreparedStatement pst = null;

public int addCustomer(Customer custBean) {

int customerId = 0;

String sql = "insert into customer (first\_name , last\_name , email , password , phone)"

+ " values(? , ? , ? , ? , ?)";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql , Statement.RETURN\_GENERATED\_KEYS);

pst.setString(1, custBean.getFirstName());

pst.setString(2, custBean.getLastName());

pst.setString(3, custBean.getEmail());

pst.setString(4, custBean.getPassword());

pst.setString(5, custBean.getPhone());

if(pst.executeUpdate() == 1) {

ResultSet rs = pst.getGeneratedKeys();

if(rs.next()) {

customerId = rs.getInt(1);

}

}else {

customerId = 0;

}

} catch (SQLException e) {

customerId = 0;

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return customerId;

}

public String updatePassword(int Id , String password) {

String sql = "update customer set password=? WHERE customer\_id = ?";

String status = "";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, password);

pst.setInt(2, Id);

if (pst.executeUpdate() == 1) {

status = "SUCCESS";

} else

status = "FAIL";

} catch (Exception e) {

status = "FAIL";

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return status;

}

public int customerLogin(String email , String password) {

String sql = "select \* from customer where email = ? and password = ?";

int customerId = 0;

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, email);

pst.setString(2, password);

ResultSet rs = pst.executeQuery();

if(rs.next()) {

customerId = rs.getInt(1);

}else {

customerId = 0;

}

} catch (SQLException e) {

customerId = 0;

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return customerId;

}

public Customer getCustomer(int customerId) {

String sql = "select \* from customer where customer\_id = ?";

Customer customer = new Customer();

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, customerId);

ResultSet rs = pst.executeQuery();

rs.next();

customer.setCustomerId(rs.getInt(1));

customer.setFirstName(rs.getString(2));

customer.setLastName(rs.getString(3));

customer.setEmail(rs.getString(4));

customer.setPassword(rs.getString(5));

customer.setPhone(rs.getString(6));

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return customer;

}

public int addReservation(Reservation resBean) {

int bookingId = 0;

String sql = "insert into reservations (flight\_number,class,travel\_date,"

+ "no\_of\_passengers,total\_fare,customer\_id) values(?,?,?,?,?,?)";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql , Statement.RETURN\_GENERATED\_KEYS);

pst.setInt(1, resBean.getFlightNumber());

pst.setString(2, resBean.getTravelClass());

pst.setDate(3, getSQLDate(resBean.getTravelDate()));

pst.setInt(4, resBean.getPassengers());

pst.setDouble(5, resBean.getTotalFare());

pst.setInt(6, resBean.getCustomerId());

if (pst.executeUpdate() == 1) {

ResultSet rs = pst.getGeneratedKeys();

if(rs.next()) {

bookingId = rs.getInt(1);

}

}else {

bookingId = 0;

}

} catch (SQLException e) {

bookingId = 0;

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return bookingId;

}

public List<Reservation> showReservations(int customerId) {

List<Reservation> lrev = new ArrayList<>();

String sql = "select r.booking\_id, r.flight\_number, r.class, r.travel\_date,"

+ " r.no\_of\_passengers , r.total\_fare , r.customer\_id from "

+ " reservations r where r.customer\_id = ? ";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, customerId);

ResultSet rs = pst.executeQuery();

while(rs.next()) {

Reservation rserv = new Reservation();

rserv.setBookingId(rs.getInt(1));

rserv.setFlightNumber(rs.getInt(2));

rserv.setTravelClass(rs.getString(3));

rserv.setTravelDate(rs.getDate(4));

rserv.setPassengers(rs.getInt(5));

rserv.setTotalFare(rs.getDouble(6));

rserv.setCustomerId(rs.getInt(7));

lrev.add(rserv);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return lrev;

}

public Reservation getReservation(int bookingId ) {

Reservation reservation = new Reservation();

String sql = "select r.booking\_id, r.flight\_number, r.class, r.travel\_date,"

+ " r.no\_of\_passengers , r.total\_fare , r.customer\_id from "

+ " reservations r where r.booking\_id = ? ";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, bookingId);

ResultSet rs = pst.executeQuery();

rs.next();

reservation.setBookingId(rs.getInt(1));

reservation.setFlightNumber(rs.getInt(2));

reservation.setTravelClass(rs.getString(3));

reservation.setTravelDate(rs.getDate(4));

reservation.setPassengers(rs.getInt(5));

reservation.setTotalFare(rs.getDouble(6));

reservation.setCustomerId(rs.getInt(7));

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return reservation;

}

public double getFare(int flightNumber , String travelClass) {

double fare = 0.0;

String sql = "select fare from fare where flight\_number = ? and class = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, flightNumber);

pst.setString(2,travelClass);

ResultSet st = pst.executeQuery();

st.next();

fare = st.getDouble(1);

} catch (SQLException e) {

e.printStackTrace();

System.out.println("Cannot find fare");

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return fare;

}

public double calculateFare(int flightNumber , String travelClass , int passengers) {

double totalFare = 0.0;

double fare = 0.0;

fare = getFare(flightNumber , travelClass);

totalFare = fare \* passengers;

return totalFare;

}

public List<Flight>flightList(String src, String dest , Date travelDate){

List<Flight> flightList = new ArrayList<Flight>();

SimpleDateFormat sdf = new SimpleDateFormat("E");

String day = sdf.format(travelDate);

String sql = "select f.flight\_number , f.airline , f.weekdays , f.src\_airport\_code , "

+ " f.dest\_airport\_code from flight f where"

+ " f.src\_airport\_code = ? and"

+ " f.dest\_airport\_code = ? and (find\_in\_set(? , replace(f.weekdays , '\_' , ',')) > 0) ";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, src);

pst.setString(2, dest);

pst.setString(3, day);

ResultSet rs = pst.executeQuery();

while(rs.next()) {

Flight flight = new Flight();

flight.setFlightNumber(rs.getInt(1));

flight.setAirline(rs.getString(2));

flight.setWeekdays(rs.getString(3));

flight.setSource(rs.getString(4));

flight.setDestination(rs.getString(5));

flightList.add(flight);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return flightList;

}

public List<Flight>showFlightList(){

List<Flight> flightList = new ArrayList<Flight>();

String sql = "select f.flight\_number , f.airline , f.weekdays , f.src\_airport\_code , "

+ " f.dest\_airport\_code from flight f ";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

while(rs.next()) {

Flight flight = new Flight();

flight.setFlightNumber(rs.getInt(1));

flight.setAirline(rs.getString(2));

flight.setWeekdays(rs.getString(3));

flight.setSource(rs.getString(4));

flight.setDestination(rs.getString(5));

flightList.add(flight);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return flightList;

}

public Flight getFlight(int flightNumber){

Flight flight = new Flight();

String sql = "select f.flight\_number , f.airline , f.weekdays , f.src\_airport\_code , "

+ " f.dest\_airport\_code from flight f where"

+ " f.flight\_number = ? ";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, flightNumber);

ResultSet rs = pst.executeQuery();

rs.next();

flight.setFlightNumber(rs.getInt(1));

flight.setAirline(rs.getString(2));

flight.setWeekdays(rs.getString(3));

flight.setSource(rs.getString(4));

flight.setDestination(rs.getString(5));

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return flight;

}

public List<Fare>showFareList(int flightNumber){

List<Fare> fareList = new ArrayList<>();

String sql = "select \* from fare where flight\_number = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, flightNumber);

ResultSet rs = pst.executeQuery();

while(rs.next()) {

Fare fare = new Fare();

fare.setFlightNumber(rs.getInt(1));

fare.setTravelClass(rs.getString(2));

fare.setFare(rs.getDouble(3));

fareList.add(fare);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return fareList;

}

public Fare getFareRecord(int flightNumber , String travelClass){

Fare fare = new Fare();

String sql = "select \* from fare where flight\_number = ? and class = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, flightNumber);

pst.setString(2, travelClass);

ResultSet rs = pst.executeQuery();

rs.next();

fare.setFlightNumber(rs.getInt(1));

fare.setTravelClass(rs.getString(2));

fare.setFare(rs.getDouble(3));

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return fare;

}

public String getAirport(String airportCode) {

String airport = "";

String sql = "select a.airport from airport a where a.airport\_code = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, airportCode);

ResultSet rs = pst.executeQuery();

rs.next();

airport = rs.getString(1);

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return airport;

}

public String getCountryCode(String airportCode) {

String countryCode = "";

String sql = "select a.country\_code from airport a where a.airport\_code = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, airportCode);

ResultSet rs = pst.executeQuery();

rs.next();

countryCode = rs.getString(1);

} catch (SQLException e) {

e.printStackTrace();

}

return countryCode;

}

public String getCountry(String airportCode) {

String country = "";

String sql = "select a.country from airport a where a.airport\_code = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, airportCode);

ResultSet rs = pst.executeQuery();

rs.next();

country = rs.getString(1);

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return country;

}

public Date getDate(String date) {

Date theDate = new Date();

try {

theDate = DateUtils.parseDate(date);

} catch (ParseException e) {

e.printStackTrace();

}

return theDate;

}

public java.sql.Date getSQLDate(Date date){

return new java.sql.Date(date.getTime());

}

public List<Airport> listAirport(){

List<Airport> airportList = new ArrayList<>();

String sql = "select \* from airport";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

while(rs.next()){

Airport airport = new Airport();

airport.setAirportCode(rs.getString(1));

airport.setAirport(rs.getString(2));

airport.setCountryCode(rs.getString(3));

airport.setCountry(rs.getString(4));

airportList.add(airport);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return airportList;

}

public List<String> listAirportCode(){

List<String> codeList = new ArrayList<>();

String sql = "select airport\_code from airport";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

while(rs.next()){

String code = rs.getString(1);

codeList.add(code);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return codeList;

}

public List<String> listAirline(){

List<String> airList = new ArrayList<>();

String sql = "select \* from airline";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

while(rs.next()){

String airline = rs.getString(1);

airList.add(airline);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return airList;

}

public Airport getAirportObject(String airportCode){

String sql = "select \* from airport where airport\_code = ?";

Airport airport = new Airport();

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, airportCode);

ResultSet rs = pst.executeQuery();

rs.next();

airport.setAirportCode(rs.getString(1));

airport.setAirport(rs.getString(2));

airport.setCountryCode(rs.getString(3));

airport.setCountry(rs.getString(4));

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return airport;

}

public String getDay(Date theDate) {

SimpleDateFormat sdf = new SimpleDateFormat("EEEE");

String day = sdf.format(theDate);

return day;

}

}

Database Connection

package AdminServ;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DbConnection {

public static Connection getConnection() throws SQLException {

Connection conObj = null;

try

{

Class.forName("com.mysql.cj.jdbc.Driver");

conObj = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase","root", "Al1c$@95");

} catch (Exception e) {

// TODO: handle exception

System.out.println(e);

}

return conObj;

}

/\*

public static void main(String[] args) {

getConnection();

System.out.println("Connection Established");

}

\*/

}

package com.AdminServ;

import java.util.ArrayList;

import java.util.LinkedHashMap;

import java.util.List;

import java.util.Map;

public class Utils {

public static Map<String,String> getWeekDays() {

Map<String,String>weekMap = new LinkedHashMap<>();

weekMap.put("Sun", "Sunday");

weekMap.put("Mon", "Monday");

weekMap.put("Tue", "Tuesday");

weekMap.put("Wed", "Wednesday");

weekMap.put("Thu", "Thursday");

weekMap.put("Fri", "Friday");

weekMap.put("Sat", "Saturday");

return weekMap;

}

public static List<String> getClasses(){

List<String> classList = new ArrayList<>();

classList.add("Economy");

classList.add("Premium");

classList.add("Business");

return classList;

}

}

package com.AdminServ;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class DateUtils {

private static SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");

// read a date string and parse/convert to a date

public static Date parseDate(String dateStr) throws ParseException {

Date theDate = formatter.parse(dateStr);

return theDate;

}

// read a date and format/convert to a string

public static String formatDate(Date theDate) {

String result = null;

if (theDate != null) {

result = formatter.format(theDate);

}

return result;

}

}

JavaCode:

package com.controller;

import java.io.IOException;

import java.util.Enumeration;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.AdminServ.Customer;

import com.airlinet;

/\*\*

\* Servlet implementation class Login

\*/

@WebServlet("/login")

public class Login extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String email = request.getParameter("emailaddress");

String password = request.getParameter("password");

RequestDispatcher rd;

CustomerDAO cust = new CustomerDAO();

if((email != null && email.trim() != "") && (password != null && password.trim() != "")) {

int customerId = cust.customerLogin(email, password);

if(customerId != 0) {

HashMap<String, Object> map = new HashMap<String, Object>();

HttpSession oldSession = request.getSession(false);

HttpSession newSession = null;

if (oldSession != null) {

Enumeration keys = oldSession.getAttributeNames();

while(keys.hasMoreElements()) {

String key = (String)keys.nextElement();

map.put(key, oldSession.getAttribute(key));

oldSession.removeAttribute(key);

}

oldSession.invalidate();

newSession = request.getSession();

for(Map.Entry<String , Object> m : map.entrySet()) {

newSession.setAttribute((String)m.getKey(), m.getValue());

map.remove(m);

}

}else if(oldSession == null) {

newSession = request.getSession();

}

Flight flight = (Flight)newSession.getAttribute("flightobject");

if (flight == null) {

newSession.setAttribute("customerId", customerId);

rd = getServletContext().getRequestDispatcher("/customerdetails.jsp");

rd.forward(request, response);

}else {

newSession.setAttribute("customerId", customerId);

rd = getServletContext().getRequestDispatcher("/confirmbooking.jsp");

rd.forward(request, response);

}

}else {

request.setAttribute("loginerr", "Incorrect email or Password");

rd = getServletContext().getRequestDispatcher("/login.jsp");

rd.forward(request, response);

}

}else {

request.setAttribute("loginerr1", "Error Occured while login in");

rd = getServletContext().getRequestDispatcher("/login.jsp");

rd.forward(request, response);

}

}

}

package com.controller;

import java.io.IOException;

import java.util.Enumeration;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.AdminServ.Customer;

import airline.Customer;

/\*\*

\* Servlet implementation class Register

\*/

@WebServlet("/register")

public class Register extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HashMap<String, Object> map = new HashMap<String, Object>();

HttpSession oldSession = request.getSession(false);

HttpSession newSession = null;

if (oldSession != null) {

Enumeration keys = oldSession.getAttributeNames();

while(keys.hasMoreElements()) {

String key = (String)keys.nextElement();

map.put(key, oldSession.getAttribute(key));

oldSession.removeAttribute(key);

}

oldSession.invalidate();

newSession = request.getSession();

for(Map.Entry<String , Object> m : map.entrySet()) {

newSession.setAttribute((String)m.getKey(), m.getValue());

map.remove(m);

}

}else if(oldSession == null) {

newSession = request.getSession();

}

String firstName = request.getParameter("firstname");

String lastName = request.getParameter("lastname");

String email = request.getParameter("emailaddress");

String password = request.getParameter("password");

String phone = request.getParameter("phone");

int customerId = 0;

CustomerDAO cust = new CustomerDAO();

Customer customer = new Customer();

if(firstName.trim() != "" && lastName.trim() != "" && email.trim() != ""

&& password.trim() != "" && phone.trim() != "" ) {

customer.setFirstName(firstName);

customer.setLastName(lastName);

customer.setEmail(email);

customer.setPassword(password);

customer.setPhone(phone);

customerId = cust.addCustomer(customer);

System.out.println(customerId);

if(customerId != 0) {

newSession.setAttribute("customerId", customerId);

RequestDispatcher rd = getServletContext().getRequestDispatcher("/confirmbooking.jsp");

rd.forward(request, response);

}else {

request.setAttribute("Error", "Error Occured while adding customer");

RequestDispatcher rd = getServletContext().getRequestDispatcher("/register.jsp");

rd.forward(request, response);

}

}else {

request.setAttribute("Error1", "Error Occured while adding customer");

RequestDispatcher rd = getServletContext().getRequestDispatcher("/register.jsp");

rd.forward(request, response);

}

}

}