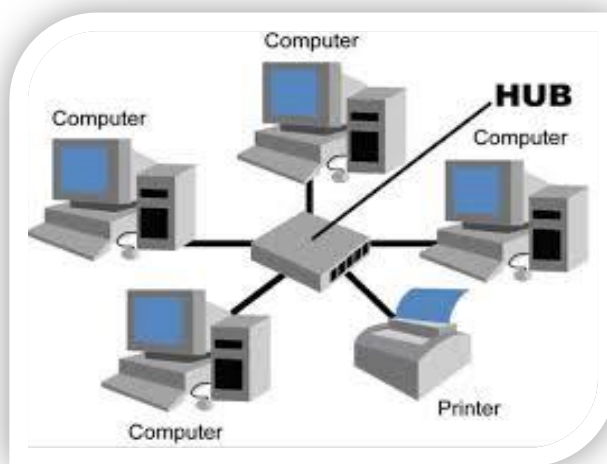


Network topology is the arrangement of the various elements (links, nodes, etc.) of a computer **network**. Essentially, it is the **topological** structure of a **network** and may be depicted physically or logically.

There are a number of **different types of network topologies**, including point-to-point, bus, star, ring, mesh, tree and hybrid.

Topology being used in LBS School Computer Lab

Star Topology: *This topology consists of a central node to which all other nodes are connected by a single path. It is the topology used in most existing information networks involving data processing or voice communications. The following figure shows the Star topology pattern.*



Advantages:

- Ease of Service.
- One device per connection.
- Centralized control.

Disadvantages:

- Long cable length.
- Difficult to expand.
- Central node dependency.

INTRODUCTION

OVERVIEW

Railway Reservation System is a fully menu driven package system. It has been developed keeping in view the present system of railway reservation prevailing in our country, as well as in all over the world. The main objective of this package is to provide effective better services to the passengers. The main utilities are as under

➤ Quick and efficient services to the passengers:

Since it is a computerized, complete menu driven package thus the time taken by the system to perform the reservation or cancellation is very less, so as to quick service can be maintained.

➤ Accuracy in data entries:

Full effort has been taken to maintain database accurately. So the data of the passengers can be safely preserved.

➤ Reduced workload:

Now-a-day due to the vast multitudes of passengers, there is a great strain of workload on working staff & the services to the passengers is also being delayed. Using this package, the heavy workload can be reduced as it is efficient to provide quick & effective services in the lesser time, thus reducing the workload.

TOOLS AND PLATFORM USED

Platform – Microsoft Windows 07

Tools used – Java Netbeans 6.5.1 (Front end application)

RDBMS – Mysql essential – 5.1-win 32 (Back end)

Interface used for connection – Java Netbeans

❖ Java Netbeans 6.5.1

1. It is both a programming language and platform.
2. Java programs need to be written once, which can be run on different platforms without making changes in the java program.
3. With Java, big and useful applications can also be treated with very light code.
4. It offers many security features to make its program safe and secure.
5. It is object oriented programming language, very near to the real world.
6. It is platform independent. Change of platform does not affect the original java program.

Further it provides an “Interactive Display Environment (IDE)”, i.e. it allows the user to add graphical entities & illustrations rather than writing numerous lines of codes to describe the appearance of the forms used in the project. It also provides Rapid Application Development (RAD).

❖ Mysql essential -5.1.33-win 32

1. It runs very fast, if the server is optimal.
2. It supports clustered servers for demanding applications.
3. It is a high performance, relatively simple database system.
4. Mysql is available free of cost & is an “open source” database.
5. It understands query based on SQL (Structured Query Language).
6. It can easily use multiple CPUs if they are available.
7. Mysql offers a privilege and password system that is very flexible and secure, and allows host-based verification.
8. It can handle large databases. Some real life Mysql databases contains 50 million records, some have up to 60,000 tables and about 5,000,000,000 rows.

Further it provides several client an utility programs. These both command line programs such as Mysql Administrator and Mysql Query Browser. Mysql server has built in support for SQL statements to check, optimize and repair tables.

MENU SYSTEM

This is a menu driven program, as soon as this package is initiated, it will show the following options after login:

1. Services
2. Enquiry
3. My Profile
4. General
5. Exit

According to the requirement of the user, anyone of the options may be selected. This system is developed keeping in mind the needs of the user. The options are described below:

- **Services**

It contains two menu items as follows:

- ❖ Plan My Travel- It allows the user to plan his/her travel and by clicking on this option, user can go to the ticket Booking.
- ❖ Cancel Ticket- By clicking on it, user can cancel a previously booked ticket.

- **Enquiry**

It contains the following items:

- ❖ Print Ticket- By clicking on it, user can print his/her ticket after booking the ticket.
- ❖ Check PNR status- By clicking on this option, user can check his/her pnr status by entering his/her pnr number.
- ❖ Check Passenger List- It allows the user to check the passenger list by entering the class type, train no and the date of journey.

- **My profile**

- ❖ Edit profile- It allows the user to edit his/her profile which he/she enters during account creation.

- **General**
 - ❖ Check Train Schedule- By clicking on this option, user can check the schedule of the train by entering the train number.
- **Exit**
 - ❖ Exit the application- User can exit the railway reservation system by clicking on it.

Table Structure

➤ Custom_details

```
mysql> desc custom_details;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(30)	NO		NULL	
Age	int(3)	YES		NULL	
Sex	varchar(8)	YES		NULL	
Mobile_No	bigint(20)	YES	UNI	NULL	
Gmail_Id	varchar(30)	YES		NULL	
Login_Id	varchar(30)	NO	PRI	NULL	
Password	varchar(30)	YES		NULL	

Sample data-

```
mysql> select * from custom_details;
```

Name	Age	Sex	mobile_no	Gmail_ID	Login_ID	Password
Ajay kumar	22	Male	9410765647	ajaykum11@gmail.com	ajay22	trustme
Divya goyal	25	Female	7053734587	divya09@gmail.com	divya25	kenttech
jatin mittal	28	male	9410932578	mittaljatin28@yahoo.com	jatin28	specimen
keshav kumar	24	male	9876467368	kumkeshav@gmail.com	keshav24	javmkeshav
Manoj rawat	19	Male	9714874452	rawatmanoj19@gmail.com	manoj19	rockmanoj
mayank gupta	17	male	9987877545	mayankg021@gmail.com	mayank17	railway

➤ Train_details

```
mysql> desc train_details;
```

Field	Type	Null	Key	Default	Extra
Train_no	int(5)	NO	PRI	NULL	
Train_name	varchar(25)	YES		NULL	
Source	varchar(25)	YES		NULL	
Departure_time	varchar(8)	YES		NULL	
Destination	varchar(25)	YES		NULL	
Arrival_time	varchar(8)	YES		NULL	
Distance_km	double(7,1)	YES		NULL	
Base_fare	double(7,2)	YES		NULL	
Journey_time	varchar(8)	YES		NULL	

Sample data-

```
mysql> select * from train_details;
```

Train_no	Train_name	Source	Departure_time	Destination	Arrival_time	Distance_km	Base_fare	Journey_time
12204	Garibrath express	New Delhi-NDLS	11:15	Lucknow-LKN	18:40	490.0	408.00	7:25
12420	Gomti express	New Delhi-NDLS	12:30	Lucknow-LKN	21:25	510.0	430.00	8:55
12474	Sarvodaya express	New Delhi-NDLS	21:50	Ahmedabad-ADI	14:15	820.0	580.00	16:25
12723	Andhra Pradesh express	Hyderabad Decan-HYB	06:25	New Delhi-NDLS	09:05	1677.0	795.00	50:40
12724	Andhra Pradesh express	New Delhi-NDLS	17:25	Hyderabad Decan-HYB	19:50	1677.0	805.00	50:35
12915	Ashram express	Ahmedabad-ADI	18:30	New Delhi-NDLS	10:10	815.0	530.00	15:40

➤ Class_seats

```
mysql> desc class_Seats;
```

Field	Type	Null	Key	Default	Extra
Train_no	int(6)	NO	PRI	NULL	
Train_name	varchar(25)	YES		NULL	
Sl_class	int(3)	YES		NULL	
AC1_class	int(4)	YES		NULL	
AC2_class	int(4)	YES		NULL	
AC3_class	int(4)	YES		NULL	

Sample data-

mysql> select * from class_seats;

Train_no	Train_name	Sl_class	AC1_class	AC2_class	AC3_class
12204	Garibrath express	300	151	119	79
12420	Gomti express	280	168	100	90
12474	Sarvodaya express	318	148	175	200
12723	Andhra Pradesh express	350	201	150	110
12724	Andhra Pradesh express	358	189	150	80
12915	Ashram express	300	171	180	160

➤ Ac1passenger

mysql> desc ac1passenger;

Field	Type	Null	Key	Default	Extra
passengerName	varchar(25)	NO		NULL	
age	int(2)	YES		NULL	
sex	varchar(7)	YES		NULL	
MobNo	bigint(12)	NO	PRI	NULL	
train_no	int(6)	NO		NULL	
PNR_no	int(12)	YES	UNI	NULL	
Seat_no	int(4)	YES	UNI	NULL	
Class	varchar(7)	YES		NULL	
Journey_Date	date	YES		NULL	

Sample data-

mysql> select * from ac1passenger order by seat_no;

passengerName	age	sex	MobNo	train_no	PNR_no	Seat_no	Class	Journey_Date
mayank	17	Male	9990478590	12420	1400548138	1	ac1	2017-02-02
amit	18	Male	9989887476	12420	1399957024	2	ac1	2017-02-02
keshav	16	Male	9826764590	12420	1236834138	3	ac1	2017-02-02
sanjana	25	Female	9913925414	12420	1323994962	4	ac1	2017-02-02
kunal	18	Male	9876569786	12204	1286635212	5	ac1	2017-01-01
Ankit	22	Male	9895675768	12724	1305741198	6	ac1	2017-06-01

➤ Ac2passenger

```
mysql> desc ac2passenger;
```

Field	Type	Null	Key	Default	Extra
passengerName	varchar(25)	NO		NULL	
age	int(2)	YES		NULL	
sex	varchar(7)	YES		NULL	
MobNo	bigint(12)	NO	PRI	NULL	
train_no	int(6)	NO		NULL	
PNR_no	int(12)	YES	UNI	NULL	
Seat_no	int(4)	YES	UNI	NULL	
Class	varchar(7)	YES		NULL	
Journey_Date	date	YES		NULL	

Sample data-

```
mysql> select * from ac2passenger order by seat_no;
```

passengerName	age	sex	MobNo	train_no	PNR_no	Seat_no	Class	Journey_Date
Aditya	22	Male	9977876560	12474	1387941990	1	ac2	2017-03-05
Disha	19	Female	9999675690	12724	1409741117	2	ac2	2017-03-06
Arpit	35	Male	9895629875	12724	1305695318	3	ac2	2017-04-03

➤ Ac3passenger

```
mysql> desc ac3passenger;
```

Field	Type	Null	Key	Default	Extra
passengerName	varchar(25)	NO		NULL	
age	int(2)	YES		NULL	
sex	varchar(7)	YES		NULL	
MobNo	bigint(12)	NO	PRI	NULL	
train_no	int(6)	NO		NULL	
PNR_no	int(12)	YES	UNI	NULL	
Seat_no	int(4)	YES	UNI	NULL	
Class	varchar(7)	YES		NULL	
Journey_Date	date	YES		NULL	

Sample data-

```
mysql> select * from ac3passenger order by seat_no;
```

passengerName	age	sex	MobNo	train_no	PNR_no	Seat_no	Class	Journey_Date
Sanjay	35	Male	9890547685	12204	1300613128	1	ac3	2017-04-12
Aarti	23	Female	9997565890	12204	1407631321	2	ac3	2017-03-06

sl-passenger

```
mysql> desc slpassenger;
```

Field	Type	Null	Key	Default	Extra
passengerName	varchar(25)	NO		NULL	
age	int(2)	YES		NULL	
sex	varchar(7)	YES		NULL	
MobNo	bigint(12)	NO	PRI	NULL	
train_no	int(6)	NO		NULL	
PNR_no	int(12)	YES	UNI	NULL	
Seat_no	int(4)	YES	UNI	NULL	
Class	varchar(7)	YES		NULL	
Journey_Date	date	YES		NULL	

Sample data-

```
mysql> select * from slpassenger order by seat_no;
```

passengerName	age	sex	MobNo	train_no	PNR_no	Seat_no	Class	Journey_Date
Gourav	28	Male	9999786514	12723	1409851950	1	sl	2017-01-04
Jhanvi	22	Female	9899568952	12723	1309634382	2	sl	2017-02-04

(1) Begin Page (Railway Reservation System)



Coding:

```
private void beginActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(this,"Welcome to Railway  
    Reservation System");  
    new accountLog().setVisible(true);  
}
```

Import command:

```
import javax.swing.JOptionPane;
```


(2)Login Page



Coding:

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import javax.swing.JOptionPane;
```

```
public class accountLog extends javax.swing.JFrame {  
    public static String name;  
    public static int age;  
    public static String sex;  
    public static Long mob;  
    public static String gmail;  
    public static String login;  
    public static String pass;  
    public accountLog() {  
        initComponents();  
    }  
    private void loginBTNActionPerformed(java.awt.event.ActionEvent  
    evt) {
```

```
try {
    Class.forName("java.sql.Driver");
    Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",
    "tiger");
    Statement stment = conn.createStatement();
    String iduse = ID.getText();
    String pwd = new String(PASS.getPassword());
    String qry = " select * from custom_details where Login_ID = '" +iduse+"'";
    ResultSet rs = stment.executeQuery(qry);
    if(rs.first()) {
        login = rs.getString("login_id");
        pass = rs.getString("password");
        gmail=rs.getString("gmail_id");
        mob=rs.getLong("mobile_no");
        sex=rs.getString("sex");
        name=rs.getString("name");
        age=rs.getInt("age");
    }
    if (iduse.equalsIgnoreCase(login)==true) {
        if(pwd.equals(pass)==true) {
            JOptionPane.showMessageDialog(this, "Welcome to railway site");
            new PageMain().setVisible(true);
        }
        else
            JOptionPane.showMessageDialog(this, "Your Password is wrong");
    }
    else
        JOptionPane.showMessageDialog(this, "your Login Id doesn't exist... \n
pls Signup or   enter correct Login Id");
    conn.close();
    stment.close();
    rs.close();
}
catch(Exception exp) {
    JOptionPane.showMessageDialog(null,"erorr!!! " +exp.getMessage());
}
```

```
}  
  
private void ForgotPasActionPerformed(java.awt.event.ActionEvent  
evt) {  
    try {  
        Class.forName("java.sql.Driver");  
        Connection  
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",  
        "tiger");  
        Statement stment = conn.createStatement();  
        gmail = JOptionPane.showInputDialog("Enter your Gmail_ID");  
        login = JOptionPane.showInputDialog("Enter your login_ID");  
        String sql="Select * from custom_details where gmail_ID='"+gmail+"'and  
login_id=' '+login+' ' ";  
        ResultSet rs = stment.executeQuery(sql);  
        if (rs.first()) {  
            pass = rs.getString("password");  
            JOptionPane.showMessageDialog(this,"Your Password is " +pass);  
        }  
        else  
            JOptionPane.showMessageDialog(this,"Enter correct Login_ID and  
Gmail_ID");  
    }  
    catch(Exception exp) {  
        JOptionPane.showMessageDialog(null,"error!!!" +exp.getMessage());  
    }  
}  
  
private void  
signUpBTNActionPerformed(java.awt.event.ActionEvent evt) {  
    this.dispose();  
    new CreateAcc().setVisible(true);  
}
```

(3) Sign up Page

Coding:

```
private void RegisterActionPerformed(java.awt.event.ActionEvent
evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",
    "tiger");
        Statement stment = conn.createStatement();
        String cfname = fnameTF.getText();
        String clname = lnameTF.getText();
        int cage = Integer.parseInt(ageTF.getText());
        String csex = sexTF.getText();
        long mobno =Long.parseLong(mobTF.getText());
        String gid = gmailTF.getText();
        String pass = JOptionPane.showInputDialog(this,"Create password");
        String login = cfname+cage;
```



```

    int ans=JOptionPane.showConfirmDialog(this,"Do you really want to create
your account");
    if(ans==0) {
        loginTF.setText(login);
        JOptionPane.showMessageDialog(this, "your login Id is "+ login);
        String qry="Insert into custom_details values('"+
        cname+"
"+cname+"','"+cage+"','"+csex+"','"+mobno+"','"+gid+"','"+login+"','"+pass+"')";
        stment.executeUpdate(qry);
        JOptionPane.showMessageDialog(this,"Your account has been created");
        this.dispose();
        new accountLog().setVisible(true);
    }
    conn.close();
    stment.close();
}
catch(Exception exp) {
    JOptionPane.showMessageDialog(null,"erorr!!!" +exp.getMessage());
}
}

private void BackActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose();
new accountLog().setVisible(true);
}

private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);
}

```

Import command:

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.JOptionPane;

```

(4)Main Page



Coding:

```
private void travelPlanActionPerformed(java.awt.event.ActionEvent  
evt) {  
    new ticketBooking().setVisible(true);  
}  
  
private void cancelActionPerformed(java.awt.event.ActionEvent evt)  
{  
    new CancelBooking().setVisible(true);  
}  
  
private void printActionPerformed(java.awt.event.ActionEvent evt) {  
    PrintTicket ticket=new PrintTicket();  
    PrintTicket.journeyDate.setText(JourneyDetail.journey);
```

```
PrintTicket.train_no.setText(""+JourneyDetail.tno);
PrintTicket.to.setText(JourneyDetail.to);
PrintTicket.pnrTF.setText(""+JourneyDetail.pnr);
PrintTicket.tname.setText(JourneyDetail.tname);
PrintTicket.pname.setText(JourneyDetail.pname);
PrintTicket.from.setText(JourneyDetail.from);
PrintTicket.sexTF.setText(JourneyDetail.psex);
PrintTicket.ageTF.setText(""+JourneyDetail.page);
PrintTicket.snoTF.setText(""+JourneyDetail.sno);
PrintTicket.classType.setText(JourneyDetail.trainClass);
PrintTicket.mobile_no.setText(""+JourneyDetail.mobno);
PrintTicket.amountTF.setText("Rs."+BookTrain.totalFare);
PrintTicket.timeTF.setText(""+BookTrain.deptTime);
ticket.setVisible(true);
}
```

```
private void CheckPNRActionPerformed(java.awt.event.ActionEvent
evt) {
new CheckPNR().setVisible(true);
}
```

```
private void
passengerListActionPerformed(java.awt.event.ActionEvent evt) {
new passengerList().setVisible(true);
}
```

```
private void profileEditActionPerformed(java.awt.event.ActionEvent
evt) {
my_profile.cnameTF.setText(accountLog.name);
my_profile.cageTF.setText(""+accountLog.age);
my_profile.mobTF.setText(""+accountLog.mob);
my_profile.gmailTF.setText(accountLog.gmail);
my_profile.sexTF.setText(accountLog.sex);
new my_profile().setVisible(true);
}
```

```
private void  
trainScheduleActionPerformed(java.awt.event.ActionEvent evt) {  
    new trainSchedule().setVisible(true);  
}  
  
private void exitActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

(5)Journey Booking(Step-1)

Start Booking


FROM

TO

Date of journey

Ticket type ☒ e-ticket

☒ I am agree with all terms and conditions



Train no.	Train name	From	Depart time	To	Arrival time	Distt.(Km)	Journey Time
12724	Andhra Pradesh e...	New Delhi-NDLS	17:25	Hyderabad Decan-...	19:50	1677.0	50:35

Coding:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class ticketBooking extends javax.swing.JFrame {
    public static String yr;
    public static String mon;
    public static String day;
    public static String journeyDate;
    public ticketBooking() {
        initComponents();
    }
}
```

```
private void submitActionPerformed(java.awt.event.ActionEvent evt)
{
DefaultTableModel dtm1 = (DefaultTableModel)trainTABLE.getModel();
try {
String from = (String)fromCOMBO.getSelectedItem();
String to = (String)toCOMBO.getSelectedItem();
Class.forName("java.sql.Driver");
Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",
"tiger");
Statement stment = conn.createStatement();
String sql = "select * from train_details where source = '"+from+"'&&
destination = '"+to+"'";
ResultSet rs = stment.executeQuery(sql);
{//To wipe the previous data
int r = dtm1.getRowCount();
for (int i = 0;i<r;++i )
dtm1.removeRow(0);
}
while(rs.next()) {
int tno = rs.getInt("Train_no");
String tname = rs.getString("Train_name");
String deptTime = rs.getString("departure_time");
String arriveTime = rs.getString("arrival_time");
double dist = rs.getDouble("distance_km");
String journey=rs.getString("journey_time");
Object ob1[] = {tno,tname,from,deptTime,to,arriveTime,dist,journey};
dtm1.addRow(ob1);
}
furtherDetails.setEnabled(true);
conn.close();
stment.close();
rs.close();
}
catch(Exception exp) {
JOptionPane.showMessageDialog(null,"erorr!!!" +exp.getMessage());
}
```



```
} }
```

```
private void resetActionPerformed(java.awt.event.ActionEvent evt) {  
    submit.setEnabled(false);  
    condition.setSelected(false);  
    furtherDetails.setEnabled(false);  
    fromCOMBO.setSelectedItem("New Delhi-NLDS");  
    toCOMBO.setSelectedItem("New Delhi-NLDS");  
}
```

```
private void
```

```
furtherDetailsActionPerformed(java.awt.event.ActionEvent evt) {  
    JourneyDetail journey=new JourneyDetail();  
    yr=(String)yearCOMBO.getSelectedItem();  
    mon=(String)monthCOMBO.getSelectedItem();  
    day=(String)dayCOMBO.getSelectedItem();  
    int mon_no;  
    if(mon.equalsIgnoreCase("jan"))  
        mon_no=01;  
    else if(mon.equalsIgnoreCase("feb"))  
        mon_no=02;  
    else if(mon.equalsIgnoreCase("march"))  
        mon_no=03;  
    else if(mon.equalsIgnoreCase("april"))  
        mon_no=04;  
    else if(mon.equalsIgnoreCase("may"))  
        mon_no=05;  
    else if(mon.equalsIgnoreCase("june"))  
        mon_no=06;  
    else if(mon.equalsIgnoreCase("july"))  
        mon_no=07;  
    else if(mon.equalsIgnoreCase("aug"))  
        mon_no =8;  
    else if(mon.equalsIgnoreCase("sept"))  
        mon_no=9;  
    else if(mon.equalsIgnoreCase("oct"))  
        mon_no=10;
```

```
else if(mon.equalsIgnoreCase("nov"))
    mon_no=11;
else
    mon_no=12;
journeyDate=yr+"-"+mon_no+"-"+day;
JourneyDetail.dateJourney.setText(ticketBooking.journeyDate);
journey.setVisible(true);
}

private void conditionActionPerformed(java.awt.event.ActionEvent
evt) {
submit.setEnabled(true);
}

private void backActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose();
new PageMain().setVisible(true);
}
```


(6)Journey Booking(Step-2)

Passenger Details Entry

Journey Details

Train no. 12420 To Station Lucknow-LKN

Train name Gomti express Journey Date 2017-2-02 (yyyy-mm-dd)

From Station New Delhi-NDLS PNR number 1400548138

Save Journey Details

Next

Passenger Details

Reset Passenger Deatils

Back

Message

i Details Added successfully

OK

AVAILABLE SEATS 175

Select Class

☐ SL ☒ AC1 ☐ AC2 ☐ AC3

Coding:

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import javax.swing.JOptionPane;
```

```
public class JourneyDetail extends javax.swing.JFrame {  
    public static String yr;  
    public static String mon;  
    public static String day;  
    public static String trainClass;  
    public static String journey;  
    public static int page;  
    public static String pname;  
    public static String psex;  
    public static int tno;  
    public static int pnr;  
    public static String from;
```

```
public static String to;
public static String tname;
public static long mobno;
public static int sno;
public static int seatsRemain;
public JourneyDetail() {
    initComponents();
}
}
```

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
try{
Class.forName("java.sql.Driver");
Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",
"tiger");
Statement stment = conn.createStatement();
tno = Integer.parseInt(JOptionPane.showInputDialog("Enter train to book
for....."));
tnoTF.setText(""+tno);
String sql = "select * from train_details where train_no="+tno;
ResultSet rs = stment.executeQuery(sql);
if(rs.first())
{
from = rs.getString("source");
to = rs.getString("destination");
tname = rs.getString("Train_name");
tnameTF.setText(tname);
toTF.setText(to);
fromTF.setText(from);
}
conn.close();
stment.close();
rs.close();
}
catch(Exception exp){
JOptionPane.showMessageDialog(null,"erorr!!!" +exp.getMessage());
}
```

```
}  
}
```

```
private void saveActionPerformed(java.awt.event.ActionEvent evt) {  
try{  
Class.forName("java.sql.Driver");  
Connection  
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",  
"tiger");  
Statement stment = conn.createStatement();  
pname = name.getText();  
page = Integer.parseInt(age.getText());  
psex = (String)JourneyDetail.sex.getSelectedItemAt();  
mobno = Long.parseLong(mobile.getText());  
tno = Integer.parseInt(tnoTF.getText());  
from=fromTF.getText();  
to=toTF.getText();  
tname=tnameTF.getText();  
journey=dateJourney.getText();  
int slseats,ac1seats,ac2seats,ac3seats;  
if(sl.isSelected()==true)  
{  
trainClass="sl";  
String sql1="select * from slpassenger order by seat_no";  
ResultSet rs1=stment.executeQuery(sql1);  
if(rs1.last()){  
sno=rs1.getInt("seat_no");  
sno+=1;  
}  
else  
sno=1;  
String qry1="select * from class_seats where train_no="+tno;  
ResultSet rs = stment.executeQuery(qry1);  
if(rs.first()) {  
slseats= rs.getInt("sl_class");  
seatsRemain=slseats-1;
```

```
String query1="update class_seats set "+trainClass+"_class
="+seatsRemain+" where train_no="+tno;
stment.executeUpdate(query1);
rs1.close();
rs.close();
}
}
else if(ac1.isSelected()==true)
{
trainClass="ac1";
String sql2="select * from ac1passenger order by seat_no";
String qry2="select * from class_seats where train_no="+tno;
ResultSet rs2=stment.executeQuery(sql2);
if(rs2.last())
{
sno=rs2.getInt("seat_no");
sno+=1;
}
else
sno=1;
ResultSet rs5 = stment.executeQuery(qry2);
if(rs5.first())
{
ac1seats= rs5.getInt("ac1_class");
seatsRemain=ac1seats-1;
String query2="update class_seats set "+trainClass+"_class
="+seatsRemain+" where train_no="+tno;
stment.executeUpdate(query2);
rs2.close();
rs5.close();
}
}
else if(ac2.isSelected()==true)
{
trainClass="ac2";
String sql3="select * from ac2passenger order by seat_no";
String qry3="select * from class_seats where train_no="+tno;
```

```
        ResultSet rs3=stment.executeQuery(sql3);
        if(rs3.last()){
            sno=rs3.getInt("seat_no");
            sno+=1;
        }
        else
            sno=1;
        ResultSet rs6 = stment.executeQuery(qry3);
        if(rs6.first()) {
            ac2seats= rs6.getInt("ac2_class");
            seatsRemain=ac2seats-1;
            String query3="update class_seats set "+trainClass+"_class
            =" +seatsRemain+" where train_no="+tno;
            stment.executeUpdate(query3);
            rs3.close();
            rs6.close();
        }
        else {
            trainClass="ac3";
            String sql4="select * from ac3passenger order by seat_no";
            String qry4="select * from class_seats where train_no="+tno;
            ResultSet rs4=stment.executeQuery(sql4);
            if(rs4.last()){
                sno=rs4.getInt("seat_no");
                sno+=1;
            }
            else
                sno=1;
            ResultSet rs7 = stment.executeQuery(qry4);
            if(rs7.first()) {
                ac3seats= rs7.getInt("ac3_class");
                seatsRemain=ac3seats-1;
                String query4="update class_seats set "+trainClass+"_class
                =" +seatsRemain+" where train_no="+tno;
                stment.executeUpdate(query4);
                rs4.close();
```

```
rs7.close();
}
}
pnr = (int) (page+mobno);
int mess = JOptionPane.showConfirmDialog(null,"your PNR no. is "+pnr);
if(mess==0){
if(seatsRemain>0)
{
    String sql = "insert into
"+trainClass+"passengervalues("+pname+", "+page+", "+psex+", "
    +mobno+", "+tno+", "+pnr+", "+sno+", "+trainClass+", "+journey+"))";
    stment.executeUpdate(sql);
    pnrTF.setText(""+pnr);
    seatNo.setText(""+sno);
    JOptionPane.showMessageDialog(this,"Details Added successfully");
    next.setEnabled(true);
}
else
JOptionPane.showMessageDialog(this,"No seats Available");
}
conn.close();
stment.close();
}
catch(Exception exp){
JOptionPane.showMessageDialog(null,"error!!!" +exp.getMessage());
}
}
```

```
private void nextActionPerformed(java.awt.event.ActionEvent evt) {
BookTrain bt1 = new BookTrain();
BookTrain.tnoTF.setText(JourneyDetail.tnoTF.getText());
BookTrain.pnrTF.setText(JourneyDetail.pnrTF.getText());
BookTrain.typeClass.setText(JourneyDetail.trainClass);
bt1.setVisible(true);
}
```

```
private void resetActionPerformed(java.awt.event.ActionEvent evt) {  
    name.setText("");  
    age.setText("");  
    mobile.setText("");  
    seatNo.setText("");  
    pnrTF.setText("");  
    sl.setSelected(true);  
}
```

```
private void backActionPerformed(java.awt.event.ActionEvent evt) {  
    this.dispose();  
    new ticketBooking().setVisible(true);  
}
```

```
private void ac1ActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        Class.forName("java.sql.Driver");  
        Connection  
        conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",  
        "tiger");  
        Statement stment = conn.createStatement();  
        tno = Integer.parseInt(tnoTF.getText());  
        String qry = "Select * from class_seats where train_no =" + tno;  
        ResultSet rs = stment.executeQuery(qry);  
        if(rs.first()) {  
            int seatsAc1= rs.getInt("ac1_class");  
            seats.setText("" + seatsAc1);  
        }  
        conn.close();  
        stment.close();  
        rs.close();  
    }  
    catch(Exception exp) {  
        JOptionPane.showMessageDialog(null,"erorr!!!" + exp.getMessage());  
    }  
}
```

```
private void ac2ActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        Class.forName("java.sql.Driver");  
        Connection  
        conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",  
        "tiger");  
        Statement stment = conn.createStatement();  
        tno = Integer.parseInt(tnoTF.getText());  
        String qry = "Select * from class_seats where train_no =" + tno;  
        ResultSet rs = stment.executeQuery(qry);  
        if(rs.first()) {  
            int seatsAc2= rs.getInt("ac2_class");  
            seats.setText("" + seatsAc2);  
        }  
        conn.close();  
        stment.close();  
        rs.close();  
    }  
    catch(Exception exp) {  
        JOptionPane.showMessageDialog(null,"erorr!!!" + exp.getMessage());  
    }  
}
```



```
private void ac3ActionPerformed(java.awt.event.ActionEvent evt) {  
try {  
Class.forName("java.sql.Driver");  
Connection  
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",  
"tiger");  
Statement stment = conn.createStatement();  
tno = Integer.parseInt(tnoTF.getText());  
String qry = "Select * from class_seats where train_no =" + tno;  
ResultSet rs = stment.executeQuery(qry);  
if(rs.first()) {  
int seatsAc3= rs.getInt("ac3_class");  
seats.setText("" + seatsAc3);  
}  
conn.close();  
stment.close();  
rs.close();  
}  
catch(Exception exp) {  
JOptionPane.showMessageDialog(null,"erorr!!!" + exp.getMessage());  
}  
}
```

```
private void slActionPerformed(java.awt.event.ActionEvent evt) {  
try {  
Class.forName("java.sql.Driver");  
Connection  
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",  
"tiger");  
Statement stment = conn.createStatement();  
tno = Integer.parseInt(tnoTF.getText());  
String qry = "Select * from class_seats where train_no =" + tno;  
ResultSet rs = stment.executeQuery(qry);  
if(rs.first())  
{  
int seatsSL= rs.getInt("sl_class");
```

```
seats.setText(""+seatsSL);
}
conn.close();
stment.close();
rs.close();
}
catch(Exception exp) {
JOptionPane.showMessageDialog(null,"erorr!!!" +exp.getMessage());
}
}
```

(7)Journey Booking(Final step)

Train no. PNR number

Base Fare	Reservation fare	Service Tax	Total fare
430.0	800.0	43	1273.0

CLASS

Message

Your ticket has been Confirmed...Thanks for using e-Ticket software.
Print your ticket from options in main page

Train no.	Train name	From	Depart time	To	Arrive time	Distance(km.)	Journey Time
12420	Gomti express	New Delhi-NDLS	12:30	Lucknow-LKN	21:25	510.0	8:55

Coding:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class BookTrain extends javax.swing.JFrame {
    static int tno;
    public static String cls;
    public static double totalFare;
    public static String deptTime;
    public BookTrain() {
        initComponents();
    }
}

private void formWindowOpened(java.awt.event.WindowEvent evt) {
```

```
DefaultTableModel dtm1 = (DefaultTableModel)trainDTL.getModel();
DefaultTableModel dtm2 = (DefaultTableModel)fareTBL.getModel();
try{
Class.forName("java.sql.Driver");
Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",
"tiger");
Statement stment = conn.createStatement();
tno = Integer.parseInt(tnoTF.getText());
String sql = "select * from train_details where train_no="+tno;
ResultSet rs = stment.executeQuery(sql);
{//To wipe the previous data
int r = dtm1.getRowCount();
for (int i = 0;i<r;++i )
dtm1.removeRow(0);
}
if(rs.first()){
String from = rs.getString("source");
String to = rs.getString("destination");
String tname = rs.getString("Train_name");
deptTime = rs.getString("departure_time");
String arriveTime = rs.getString("arrival_time");
double dist = rs.getDouble("distance_km");
String journey=rs.getString("journey_time");
Object ob1[] = {tno,tname,from,deptTime,to,arriveTime,dist,journey};
dtm1.addRow(ob1);
{//To wipe the previous data
int r = dtm2.getRowCount();
for (int i = 0;i<r;++i )
dtm2.removeRow(0);
}
double baseFare = rs.getDouble("base_fare");
int serviceTAX = (int) (0.10*baseFare);
double reservationFare = 0;
cls=typeClass.getText();
if(cls.equalsIgnoreCase("sl"))
reservationFare=180;
```

```
else if(cls.equalsIgnoreCase("ac1"))
reservationFare=800;
else if(cls.equalsIgnoreCase("ac2"))
reservationFare=600;
else if(cls.equalsIgnoreCase("ac3"))
reservationFare=350;
totalFare =reservationFare+baseFare+serviceTAX;
Object obj1[] = {baseFare,reservationFare,serviceTAX,totalFare};
dtm2.addRow(obj1);
}
conn.close();
stment.close();
rs.close();
}
catch(Exception exp)
{
JOptionPane.showMessageDialog(null,"erorr!!!" +exp.getMessage());
}
}
```

```
private void BookNOWActionPerformed(java.awt.event.ActionEvent
evt) {
JOptionPane.showMessageDialog(null,"Your ticket has been
Confirmed..Thanks for using "
+"e-Ticket software."+"\n"+"  Print your ticket from options in main page");
new PageMain().setVisible(true);
}
```

```
private void backActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose();
new ticketBooking().setVisible(true);
}
```

(8)Print Ticket

Print Ticket

PNR No:- 4568 Date of Journey:- 04-08-2017

Train No:- 3489 From:- NDLS

Train Name:- Durgam To:- HYB

Passenger Details

Passenger Name	Age	Sex	Mobile No.	Seat No.
SaiAshish	16	Male	985434979	7

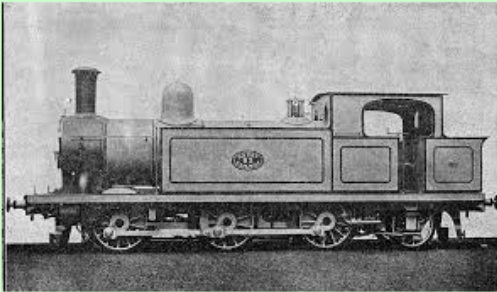
Print Ticket

CLOSE

LOG OUT

Departure time

01:00



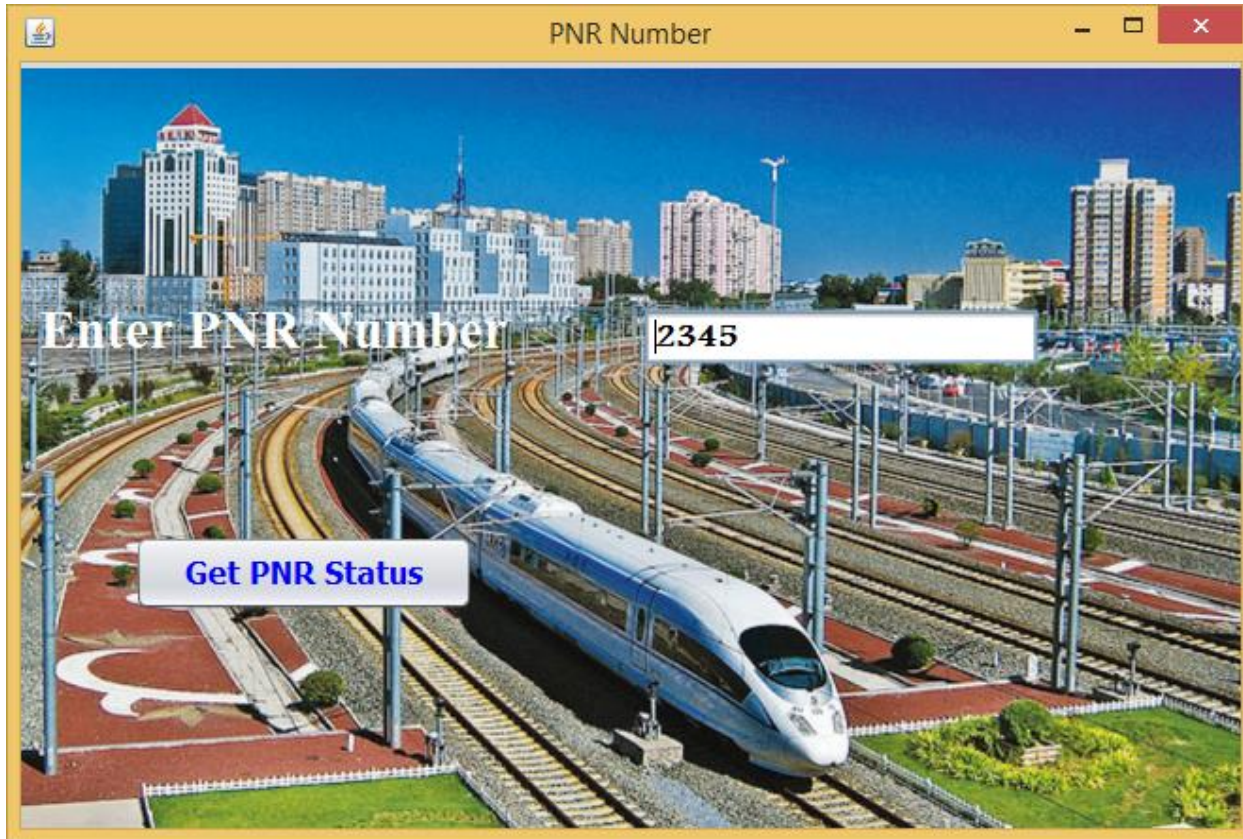
Coding:

```
private void printTicketActionPerformed(java.awt.event.ActionEvent  
evt) {  
JOptionPane.showMessageDialog(this,"Your ticket has been succesfully  
printed");  
logOut.setEnabled(true);  
}
```

```
private void ExitActionPerformed(java.awt.event.ActionEvent evt) {  
JOptionPane.showMessageDialog(this,"Thanks for using e-software...");  
System.exit(0);  
}
```

```
private void logOutActionPerformed(java.awt.event.ActionEvent evt)
{
this.dispose();
new accountLog().setVisible(true);
}
```


(9)Check PNR Status(Step-1)



Coding:

```
public class CheckPNR extends javax.swing.JFrame {  
    public static int pnr;  
    public static int tno;  
    public static String pname;  
    public static int page;  
    public static String type;  
    public static String psex;  
    public static long mob;  
    public static String journey;  
    public static int sno;  
  
    public CheckPNR() {  
        initComponents();  
    }  
}
```



```
private void pnrStatusActionPerformed(java.awt.event.ActionEvent  
evt) {  
    pnr = Integer.parseInt(pnrNTF.getText());  
    StatusPNR stpnr=new StatusPNR();  
    StatusPNR.pnrTF.setText(""+pnr);  
    stpnr.setVisible(true);  
}
```

(10) Check PNR Status(Final Step)

PNR status

PNR No. Train no.

Train name

Departure time To Station

From Station Date of Journey:-

Passenger Details

Passenger Name	Age	Sex	Mobile No.	Seat No.
<input type="text" value="SaiAshish"/>	<input type="text" value="16"/>	<input type="text" value="Male"/>	<input type="text" value="123456789"/>	<input type="text" value="1"/>

Coding:

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import javax.swing.JOptionPane;
```

```
public class StatusPNR extends javax.swing.JFrame {  
    public static int tno;  
    public static String pname;  
    public static int page;  
    public static String type;  
    public static String psex;  
    public static long mob;  
    public static String journey;  
    public static int sno;
```

```
public static int pnr;  
public static String tname;  
public static String source;  
public static String destination;
```

```
public StatusPNR() {  
    initComponents();  
}  
}
```

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {  
    try {  
        Class.forName("java.sql.Driver");  
        Connection  
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",  
"tiger");  
        Statement stment = conn.createStatement();  
        pnr = Integer.parseInt(pnrTF.getText());  
        String sql1="select * from ac1passenger where pnr_no="+pnr;  
        ResultSet rs1=stment.executeQuery(sql1);  
        if(rs1.first())  
        {  
            pname = rs1.getString("passengerName");  
            page = rs1.getInt("age");  
            psex = rs1.getString("sex");  
            mob = rs1.getLong("mobno");  
            tno=rs1.getInt("train_no");  
            sno=rs1.getInt("seat_no");  
            journey=rs1.getString("journey_date");  
            type=rs1.getString("Class");  
            name.setText(pname);  
            age.setText(""+page);  
            sexTF.setText(psex);  
            mobile.setText(""+mob);  
            tnoTF.setText(""+tno);  
            seatNo.setText(""+sno);  
            journeyDate.setText(journey);  
        }  
    }  
}
```

```
classType.setText(type);
    String sql5="select * from train_details where train_no="+tno;
    ResultSet rs5=stment.executeQuery(sql5);
    if(rs5.first())
    {
        tname=rs5.getString("train_name");
        source=rs5.getString("source");
        destination =rs5.getString("destination");
        String deptTime=rs5.getString("departure_time");
        tnameTF.setText(tname);
        fromTF.setText(source);
        toTF.setText(destination);
        departTime.setText(deptTime);
        rs5.close();
    }
}
String sql2="select * from ac2passenger where pnr_no="+pnr;
ResultSet rs2=stment.executeQuery(sql2);
if(rs2.first())
{
    pname = rs2.getString("passengerName");
    page = rs2.getInt("age");
    psex = rs2.getString("sex");
    mob = rs2.getLong("mobno");
    tno=rs2.getInt("train_no");
    sno=rs2.getInt("seat_no");
    journey=rs2.getString("journey_date");
    type=rs2.getString("Class");
    name.setText(pname);
    age.setText(""+page);
    sexTF.setText(psex);
    mobile.setText(""+mob);
    tnoTF.setText(""+tno);
    seatNo.setText(""+sno);
    journeyDate.setText(journey);
    classType.setText(type);
    String sql6="select * from train_details where train_no="+tno;
```

```
ResultSet rs6=stment.executeQuery(sql6);
if(rs6.first())
{
tname=rs6.getString("train_name");
source=rs6.getString("source");
destination =rs6.getString("destination");
String deptTime=rs6.getString("departure_time");
tnameTF.setText(tname);
fromTF.setText(source);
toTF.setText(destination);
departTime.setText(deptTime);
rs6.close();
}
}
String sql3="select * from ac3passenger where pnr_no="+pnr;
ResultSet rs3=stment.executeQuery(sql3);
if(rs3.first())
{
pname = rs3.getString("passengerName");
page = rs3.getInt("age");
psex = rs3.getString("sex");
mob = rs3.getLong("mobno");
tno=rs3.getInt("train_no");
sno=rs3.getInt("seat_no");
journey=rs3.getString("journey_date");
type=rs3.getString("Class");
name.setText(pname);
age.setText(""+page);
sexTF.setText(psex);
mobile.setText(""+mob);
tnoTF.setText(""+tno);
seatNo.setText(""+sno);
journeyDate.setText(journey);
classType.setText(type);
String sql7="select * from train_details where train_no="+tno;
ResultSet rs7=stment.executeQuery(sql7);
if(rs7.first())
```

```
{
tname=rs7.getString("train_name");
source=rs7.getString("source");
destination =rs7.getString("destination");
    String deptTime=rs7.getString("departure_time");
tnameTF.setText(tname);
fromTF.setText(source);
toTF.setText(destination);
departTime.setText(deptTime);
rs7.close();
}
}
String sql4="select * from slpassenger where pnr_no="+pnr;
ResultSet rs4=stment.executeQuery(sql4);
if(rs4.first()){
pname = rs4.getString("passengerName");
page = rs4.getInt("age");
psex = rs4.getString("sex");
mob = rs4.getLong("mobno");
tno=rs4.getInt("train_no");
sno=rs4.getInt("seat_no");
journey=rs4.getString("journey_date");
type=rs4.getString("Class");
name.setText(pname);
age.setText(""+page);
sexTF.setText(psex);
mobile.setText(""+mob);
tnoTF.setText(""+tno);
seatNo.setText(""+sno);
journeyDate.setText(journey);
classType.setText(type);
    String sql8="select * from train_details where train_no="+tno;
    ResultSet rs8=stment.executeQuery(sql8);
if(rs8.first())
{
tname=rs8.getString("train_name");
source=rs8.getString("source");
```

```
destination =rs8.getString("destination");
    String deptTime=rs8.getString("departure_time");
tnameTF.setText(tname);
fromTF.setText(source);
toTF.setText(destination);
departTime.setText(deptTime);
rs8.close();
    }
}
conn.close();
stment.close();
rs1.close();
rs2.close();
rs3.close();
rs4.close();
}
catch(Exception exp)
{
JOptionPane.showMessageDialog(null,"erorr!!" +exp.getMessage());
}
}
```

```
private void backMainActionPerformed(java.awt.event.ActionEvent
evt) {
this.dispose();
new PageMain().setVisible(true);}
}
```

(11)Check Passenger List

The screenshot shows a Java Swing window titled "Passenger List" with a green forest background. The window contains the following elements:

- Train No.:** A text field containing "12420".
- Date:** A text field containing "2017-02-02".
- Class:** A dropdown menu showing "ac1".
- Display:** A button to display the passenger list.
- Back:** A button to return to the previous screen.

A smaller "Passenger List" dialog box is open in the foreground, displaying a table with the following columns:

PNR No.	Passenger Name	Age	Sex	Seat No.

Coding:

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
    int tno=Integer.parseInt(JOptionPane.showInputDialog(this,"Enter train no."));
    String date=JOptionPane.showInputDialog(this,"Enter date (yyyy-mm-dd)");
    tnoTF.setText(""+tno);
    dateTF.setText(date);
}
```

```
private void
displayBTNActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel dtm1 = (DefaultTableModel)passengerTBL.getModel();
    try{
        int tno=Integer.parseInt(tnoTF.getText());
        String date=dateTF.getText();
```

```
String classType=(String)classCombo.getSelectedItem();
Class.forName("java.sql.Driver");
Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",
"tiger");
Statement stment = conn.createStatement();
String sql="select * from "+classType+"passenger where train_no="+tno
        +" && journey_Date='"+date+"' order by seat_no asc";
ResultSet rs=stment.executeQuery(sql);
//To wipe the previous data
int r = dtm1.getRowCount();
for (int i = 0;i<r;++i )
dtm1.removeRow(0);
}
while(rs.next()){
int pnr=rs.getInt("PNR_no");
String passenger=rs.getString("passengerName");
String sex=rs.getString("sex");
int age=rs.getInt("age");
int sno=rs.getInt("Seat_No");
Object ob1[] = {pnr,passenger,age,sex,sno};
dtm1.addRow(ob1);
}
conn.close();
stment.close();
rs.close();
}
catch(Exception exp)
{
JOptionPane.showMessageDialog(null,"erorr!!!" +exp.getMessage());
}
}

private void backActionPerformed(java.awt.event.ActionEvent evt) {
new PageMain().setVisible(true);
}
```

Import command:

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;
```

(12)Edit Profile



Name Sai Ashish

Age 16

Sex Male

Mobile No. 9875435678

Email ID sai@gmail.com

BACK **Update** **Change Password** **CLOSE**

Coding:

```
private void updateActionPerformed(java.awt.event.ActionEvent  
evt) {  
try{  
String login=accountLog.login;  
Class.forName("java.sql.Driver");  
Connection  
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",  
"tiger");  
Statement stment = conn.createStatement();  
String name = cnameTF.getText();  
int cage = Integer.parseInt(cageTF.getText());
```



```

String csex = sexTF.getText();
long mob = Long.parseLong(mobTF.getText());
String gmail = gmailTF.getText();
String qry="update custom_details set name='"+name+"',age='"+cage+
",sex='"+csex+"',mobile_no='"+mob+" where login_id='"+login+"'";
if(JOptionPane.showConfirmDialog(this,"Do you really want to make the
changes")==0)
{
stment.executeUpdate(qry);
JOptionPane.showMessageDialog(this,"Your profile has been updated
successfully");
}
conn.close();
stment.close();
}
catch(Exception exp) {
JOptionPane.showMessageDialog(this,"Error !! "+exp.getMessage());
}
}

```

private void

```

passwordBTNActionPerformed(java.awt.event.ActionEvent evt) {
try{
Class.forName("java.sql.Driver");
Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",
"tiger");
Statement stment = conn.createStatement();
String passOld=JOptionPane.showInputDialog(this,"Enter Old Password");
String gmail=gmailTF.getText();
String qry="select * from custom_details where gmail_id='"+gmail+"'";
ResultSet rs=stment.executeQuery(qry);
if(rs.first()) {
String pass=rs.getString("password");
if(passOld.equals(pass)==true)
{

```

```
String newPassword=JOptionPane.showInputDialog(this,"Enter New  
Password");  
String qry1="update custom_details set password='"+newPassword+"' where  
gmail_id='"+gmail+"'";  
JOptionPane.showMessageDialog(this,"Your password has been changed  
successfully");  
stmt.executeUpdate(qry1);  
}  
}  
conn.close();  
stmt.close();  
rs.close();  
}  
catch(Exception exp) {  
JOptionPane.showMessageDialog(null,"erorr!!!" +exp.getMessage());  
}  
}
```

```
private void back_mainActionPerformed(java.awt.event.ActionEvent  
evt) {  
new PageMain().setVisible(true);  
}
```

```
private void ExitActionPerformed(java.awt.event.ActionEvent evt) {  
System.exit(0);  
}
```

Import command:

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import javax.swing.JOptionPane;
```

(13)Train Schedule



Coding:

```
private void scheduleActionPerformed(java.awt.event.ActionEvent
evt) {
DefaultTableModel dtm1 = (DefaultTableModel)trainDTL.getModel();
try{
Class.forName("java.sql.Driver");
Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/railway","root",
"tiger");
Statement stment = conn.createStatement();
int tno = Integer.parseInt((String)trainNo.getSelectedItemAt());
String sql = "select * from train_details where train_no="+tno;
```



```
ResultSet rs = stment.executeQuery(sql);
{//To wipe the previous data
int r = dtm1.getRowCount();
for (int i = 0;i<r;++i )
dtm1.removeRow(0);
}
if(rs.next()){
    String from = rs.getString("source");
    String to = rs.getString("destination");
    String tname = rs.getString("Train_name");
    String deptTime = rs.getString("departure_time");
    String arriveTime = rs.getString("arrival_time");
    double dist = rs.getDouble("distance_km");
    String journey=rs.getString("journey_time");
    Object ob1[] = {tno,tname,from,deptTime,to,arriveTime,dist,journey};
    dtm1.addRow(ob1);
}
else
JOptionPane.showMessageDialog(null,"this train no."+tno +"does not exist");
conn.close();
stmt.close();
rs.close();
}
catch(Exception exp) {
JOptionPane.showMessageDialog(null,"erorr!!!" +exp.getMessage());
}
}

private void backBTNActionPerformed(java.awt.event.ActionEvent
evt) {
this.dispose();
new PageMain().setVisible(true);
}
```

Import command:

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;
```

BIBLIOGRAPHY

- **Informatics Practices – XI by Sumita Arora**
- **Informatics Practices – XII by Sumita Arora**
- **<http://www.irctc.in>**
- **Google Images**