**ACKNOWLEDGEMENT**

I wish to express my deep sense of indebtedness and sincerest gratitude to my guide, Mr. Sunil Kaundal, trainer at **Spectrum Academy of Computer Science, Bhareri, Hamirpur (H. P.)** for his invaluable guidance and constructive criticism throughout this dissertation. He has displayed unique tolerance and understanding at every step of progress and encourages me. I deem it my privilege to have carried out my Dissertation work under his able guidance.

I would especially like to thank Er. Ravi Kumar (Associate Professor and Head), Department of Computer Science Engineering, A.B.V.G.I.E.T PragatiNagar, Shimla, without whom, this work would not have been as it is now.

As a Final Personal Note, I am grateful to my parents, who are inspirational to me in their understanding, patience and constant encouragement.

**Abhishek**

**Roll No : 02BTD5050002**

**PREFACE**

The computers have gained a lot of importance in the past five decades. Most of our day-to-day jobs are being influenced by the use of computers. Now a day, computers are used for performing almost every function, which were performed by humans in the past. In some areas such as science and technology, targets can’t be achieved without the use of computers. The characteristics that make the computer so important include its extra ordinary speed, large storage capacity, accuracy and consistency.

Today computers play a great role in various industries and a large number of industries are using computers for various application such as maintaining cashbook, sales book, purchase book and other books of accounts. Computers can also be used for the designing of various products. Computers provide many options for the designing of products.

In this project report, steps have been taken for computerizing Bus Ticket Reservation System. The analysis of the project has been undertaken with utmost sincerity and honesty and we will be extremely satisfied if the effort is appreciated.

**INTRODUCTION**

#### **PROBLEM STATEMENT:**

This project aims at the automation of the existing Bus Ticket Reservation system and record registration. It is very difficult to maintain the record in the existing system because all the work is to be done manually. Information retrieval is very difficult since huge amount of data is to be processed manually. Register maintenance is very difficult in the existing system, because the data is in scattered manner.

**Existing System:-**

Processing and maintenance of the records of the Bus Tickets are done manually. The records are stored on the registers. The records of each category have been assigned a different register.

**Drawbacks of Current System:-**

There is no exclusive package for all of the above-mentioned activities. The registration details are entered in their processing systems. Efforts have been made to develop the integrated system covering all the sub modules.

1. **No proper Record of Passenger Details.**

There is no proper record of Passenger and his ticket details.

1. **Extensive use of paper work.**

There is extensive use of paper work. A lot of communication is there which needs a lot of difficulty to get data about a particular passenger’s ticket details.

1. **Information Retrieval is very difficult.**

As data is scattered, the information retrieval about any passenger’s ticket detail is very difficult required at any point of time.

**Requirement Analysis**

In purposed systems all the activities of system have been automated and efforts have been made to minimize the manual work.

**Benefits of Purposed System:-**

1. **Less Paper Work**

The paper work is reduced to minimal level. Computer is used to perform operations on data of passenger and his ticket details.

1. **No Manual Work.**

There is no manual work. All the processes are done through computer.

1. **Database Maintenance is Easier**

Database can now easily be maintained.

1. **Data Is Not Scattered**

Data is now stored at one place. Any information regarding anything is easily available to the user.

1. **User-friendly Software**

The software is GUI based and is very easy to use.

**OBJECTIVES OF PROJECT**

The main objectives of the projects involves the following features:

* Bus Ticket Reservation System is highly versatile, user-friendly and covers most aspects of Passenger and his journey and ticket details.
* All options in the system have user level privilege. So there is more security to the data being altered by unauthorized users.
* By using the “Bus Reservation Portal” we can reserve a ticket for a bus by going through the Travelling Details and Passenger Details section.
* By using the “See Details” we can access our ticket details at any time by entering our email and can also cancel the ticket easily
* The GUI provides the user to make use of the mouse, which makes the navigation throughout the software very easy and comfortable.
* High processing speed.

**FEASIBILITY STUDY**

During the course of completion of this project work, the complete analysis of proposed system was done. In the analysis task, a complete care about the feasibility of the proposed system was taken. The following feasibility analyses were carried out during the course of this project work:

1. Economic feasibility
2. Technical feasibility
3. Operational feasibility

**Economic Feasibility:**

Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system. The proposed system is economically feasible because the benefits and the savings that are expected from a candidate system outweigh the cost incurred. In this case we are getting the intangible benefits in terms of low cost of maintenance of data, less redundancy and getting the quick results.

**Technical Feasibility:**

The existing Hardware and Software facilities support the proposed system. Computer and storage media are available and software can be developed.

**Hardware configuration**:

a) Processor : Intel Pentium or Greater

b) Memory : 1 GB RAM or More

c) HD capacity : 500 MB or more

### Software configuration:-

a) Operating system : Windows XP or later

6b) Back end : MY-SQL

c) Front end : Java

There is nothing which is not technically feasible.

**Operational Feasibility:**

As in the case of present system the entire work is being done manually. So the data being scattered, information retrieval becomes difficult and maintaining database is also very tedious. In case of proposed system, entire work will be done automatically. So the above details regarding the feasibility study show that the design of the proposed system is very effective.

**FLOW CHART**

**Bus Ticket Reservation System**

**Final Details Form**

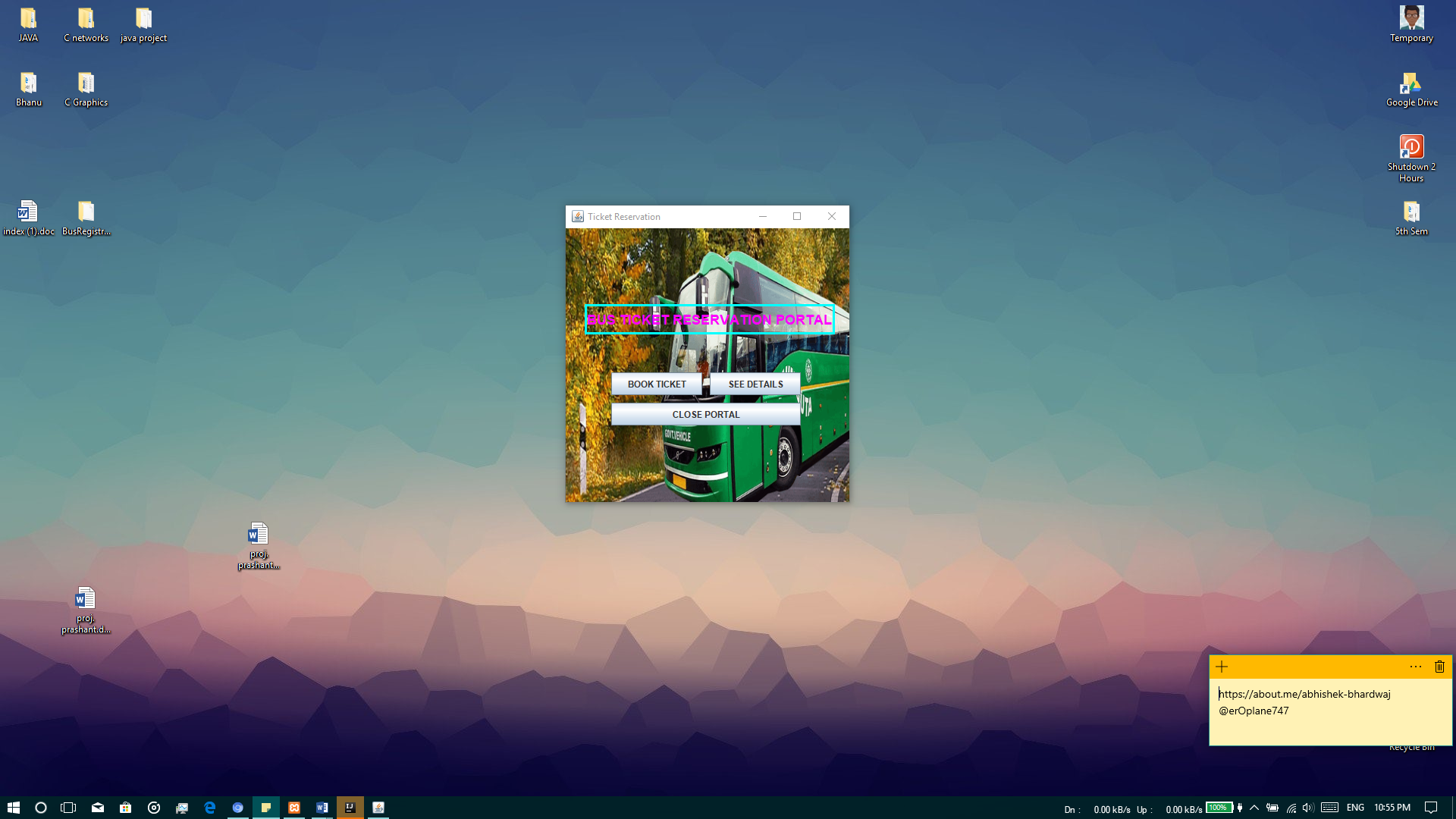
**Book Ticket Form**

**Travel Details Form**

**Personal Details Form**

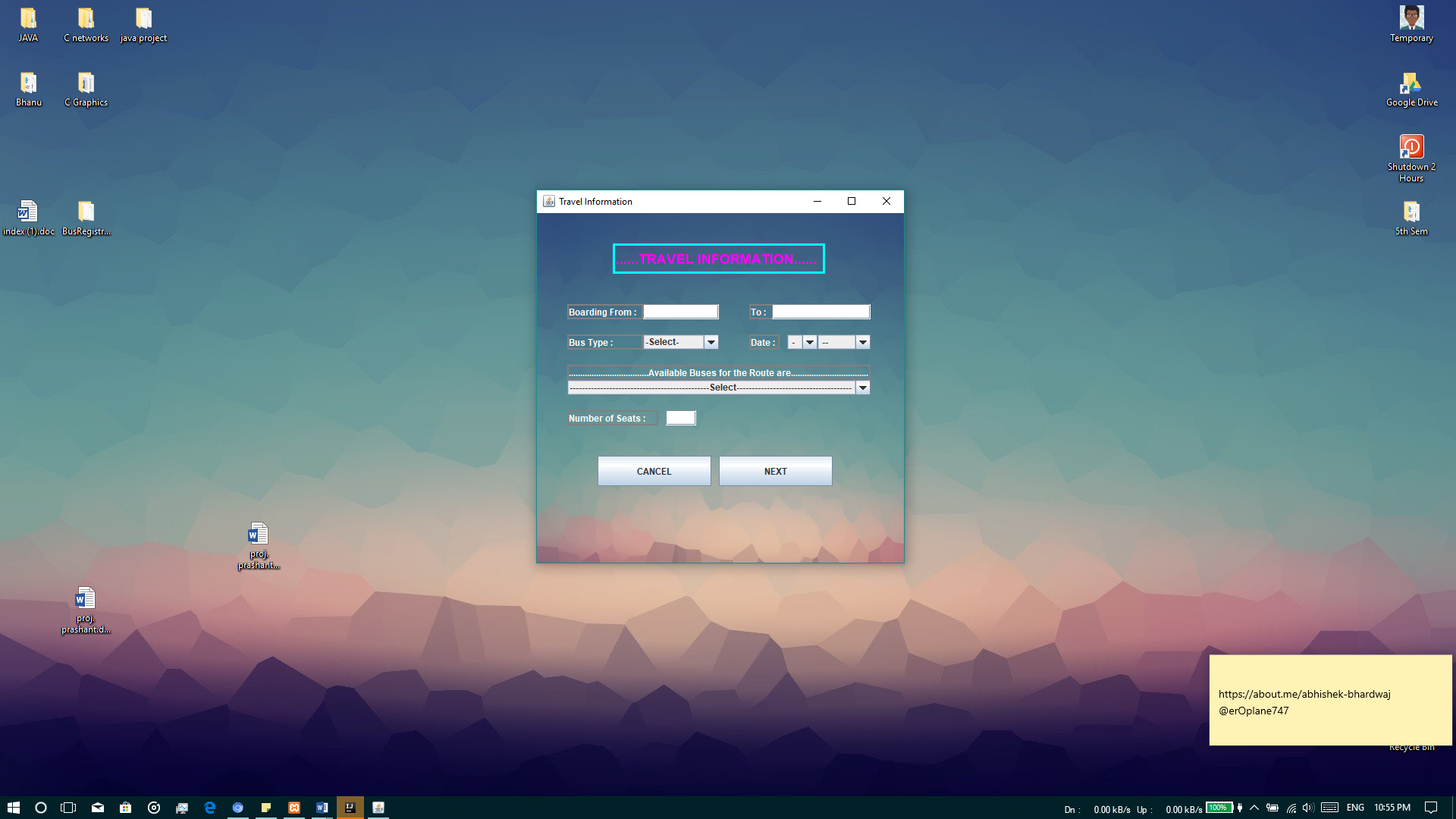
**DESIGN SCREENS**

* **Ticket Reservation Form**



This is a form, which is displayed at the beginning of the project. It shows three buttons, “BOOK TICKET” which takes us to the next form where we can add details related to our journey and personal details, “SEE DETAILS” through which we can see all our ticket details easily once we book a ticket and “CLOSE PORTAL” closes the form.

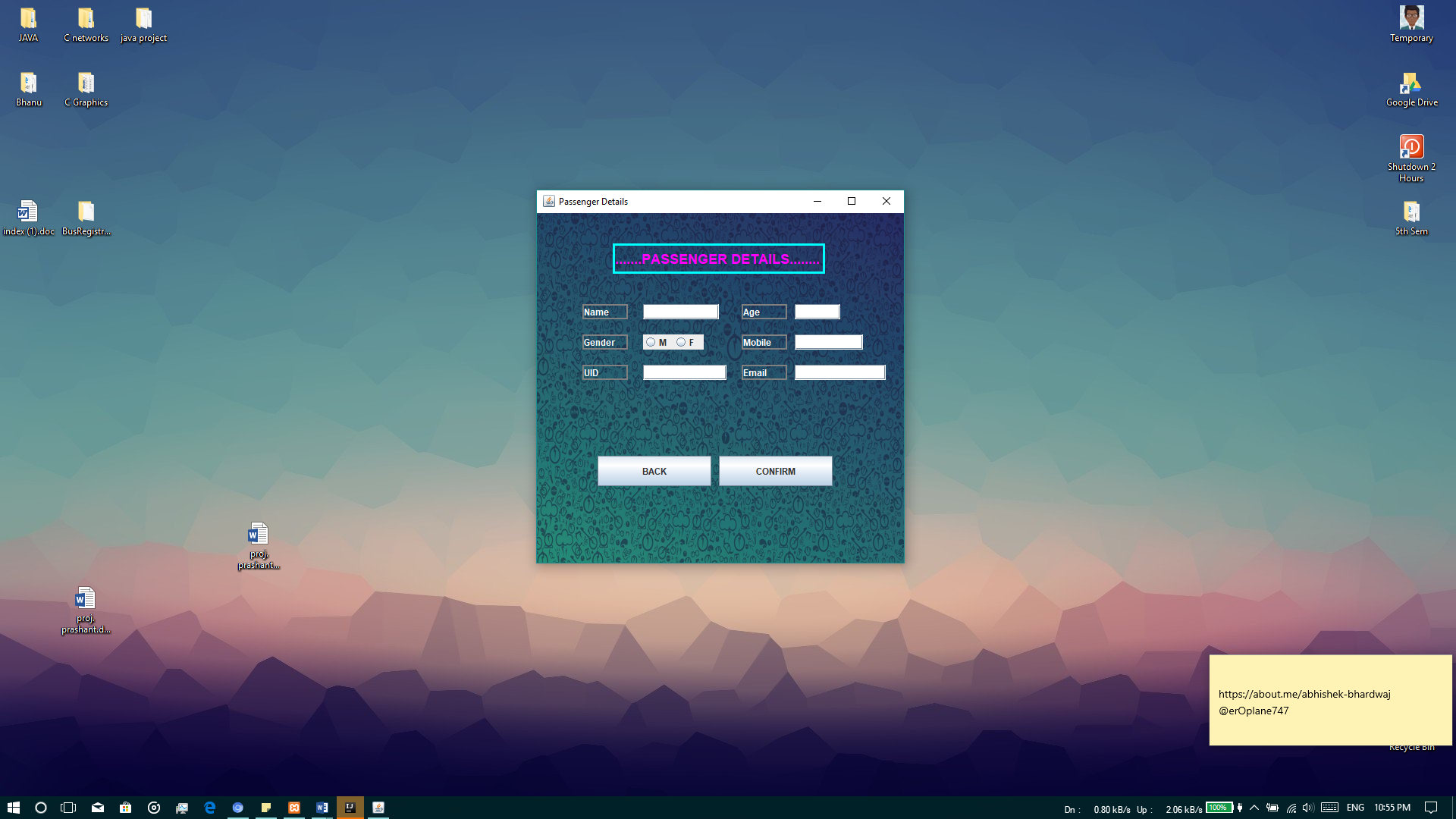
* **Travel Information Form**



This is the second form which will appear when we click on “BOOK TICKET” button from the main form.

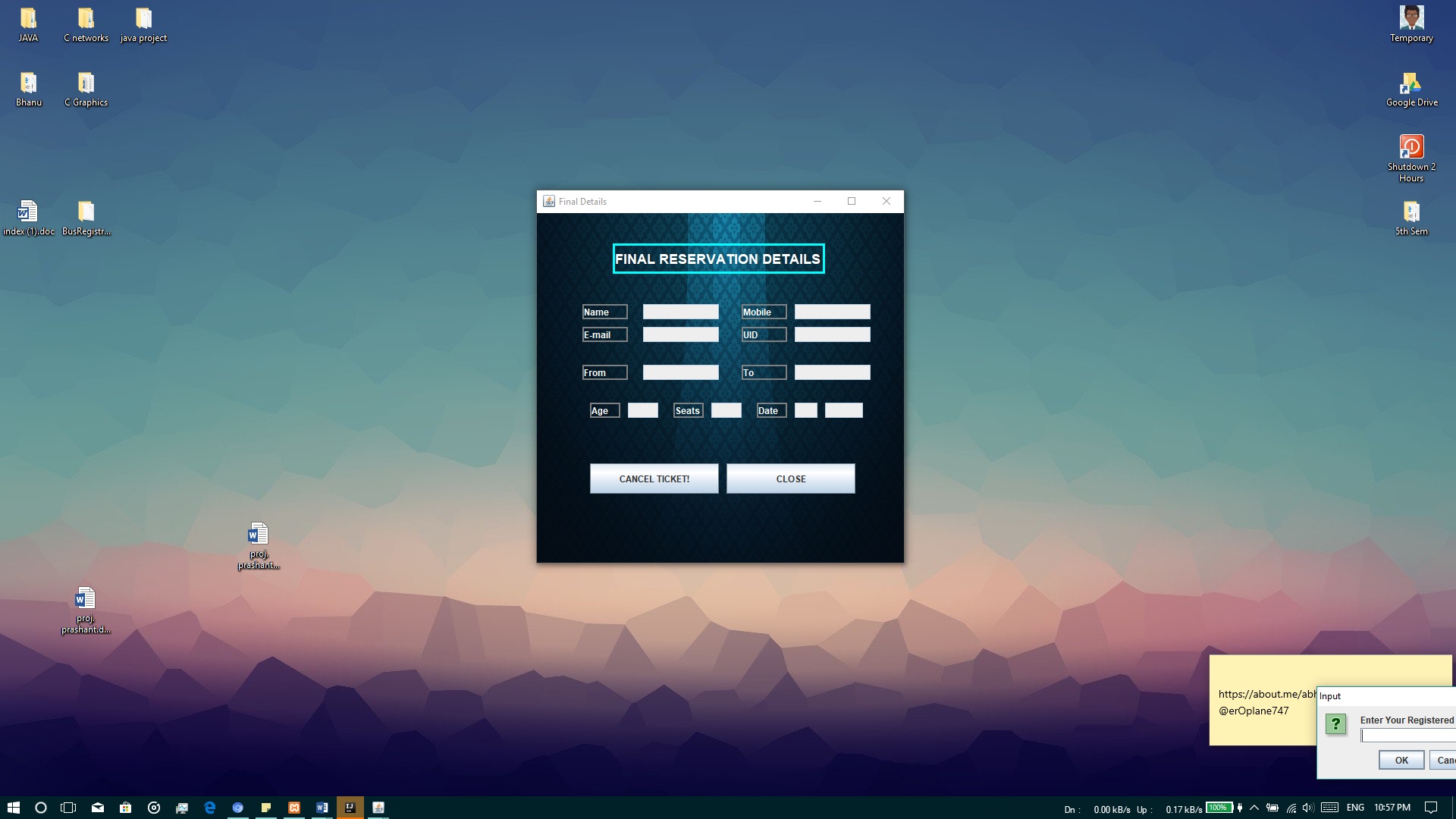
Here we can add various journey related details, like boarding from, boarding to, and it also facilitates to select the bus type. When all the fields are filled only then we can move to the next frame. Here all the fields are mandatory. The user can also cancel the process at any time by clicking “CANCEL” button, which will take user to the previous form. When user has filled all the fields then by clicking “NEXT” will take him to next frame. Also the user can modify the Travel details at any time until he has confirmed all of the provided details.

* **Passenger Details Form**



This is the third form in booking process. Here the user need to add his personal details like name, age, gender, mobile number, Unique ID number (aadhaar card), and email. Here all the fields are mandatory. Thus user can only proceed further if all the fields are filled. Also the email here is considered as a username, by which the user can access his ticket details at any time by just entering his email address. As each person has a unique email address, then there is no worry about duplicate details. Also if the user want to modify the Travel details, then he can click “BACK” button which will take him back to the second frame where the user can modify the travel details. After all the required field are filled, then user needs to click on “CONFIRM” which will save all the details provided by the user in database and then a message box will appear which will indicate the booking has been done successfully and then the user will be forwarded to next frame.

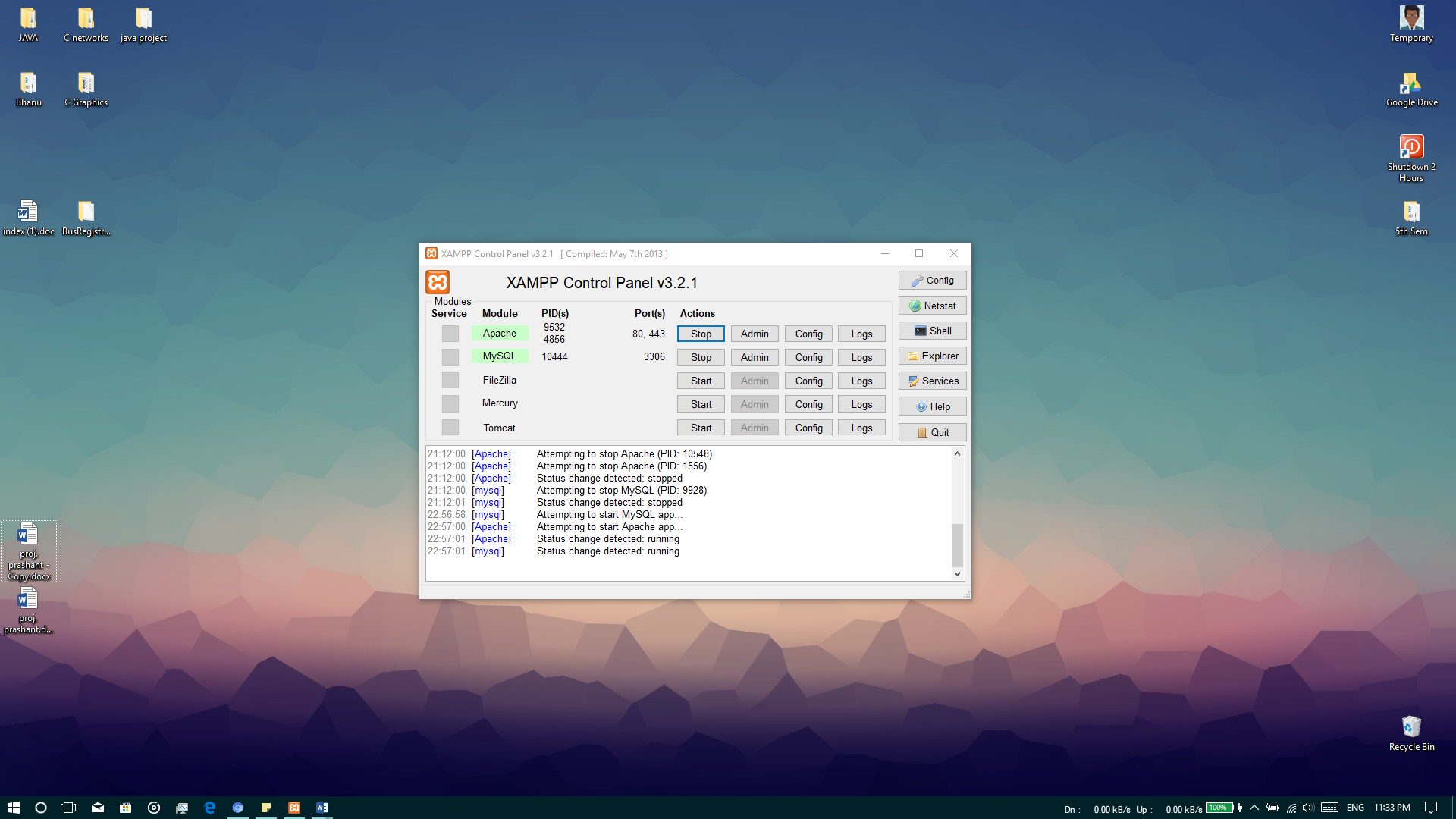
* **Final Details Form**



This is the last frame of the ticket booking process. After the user has confirmed all the travel and personal details by clicking on “CONFIRM” in the previous form, the user is then directed to this frame. Here the user can see his details related to the ticket. The fields here are not editable which means the user here is provided only with read only privilege. The user cannot modify the details that he has entered now. He is only able to read the details now. Here a button “CANCEL TICKET!” where upon clicking the user can cancel the ticket and his record will be deleted from the system database. The user is asked for confirmation through a confirmation pane that if the user really wants to cancel the ticket? If so then the user can click “YES” and his ticket will be canceled and then the user will be directed to first frame to the project. Else he can click “NO” and the user details will not be affected. There is a second button “CLOSE” which closes the current frame and then direct the user to the main frame.

**DATABASE DESIGN**

* **Connection with Xampp Server**

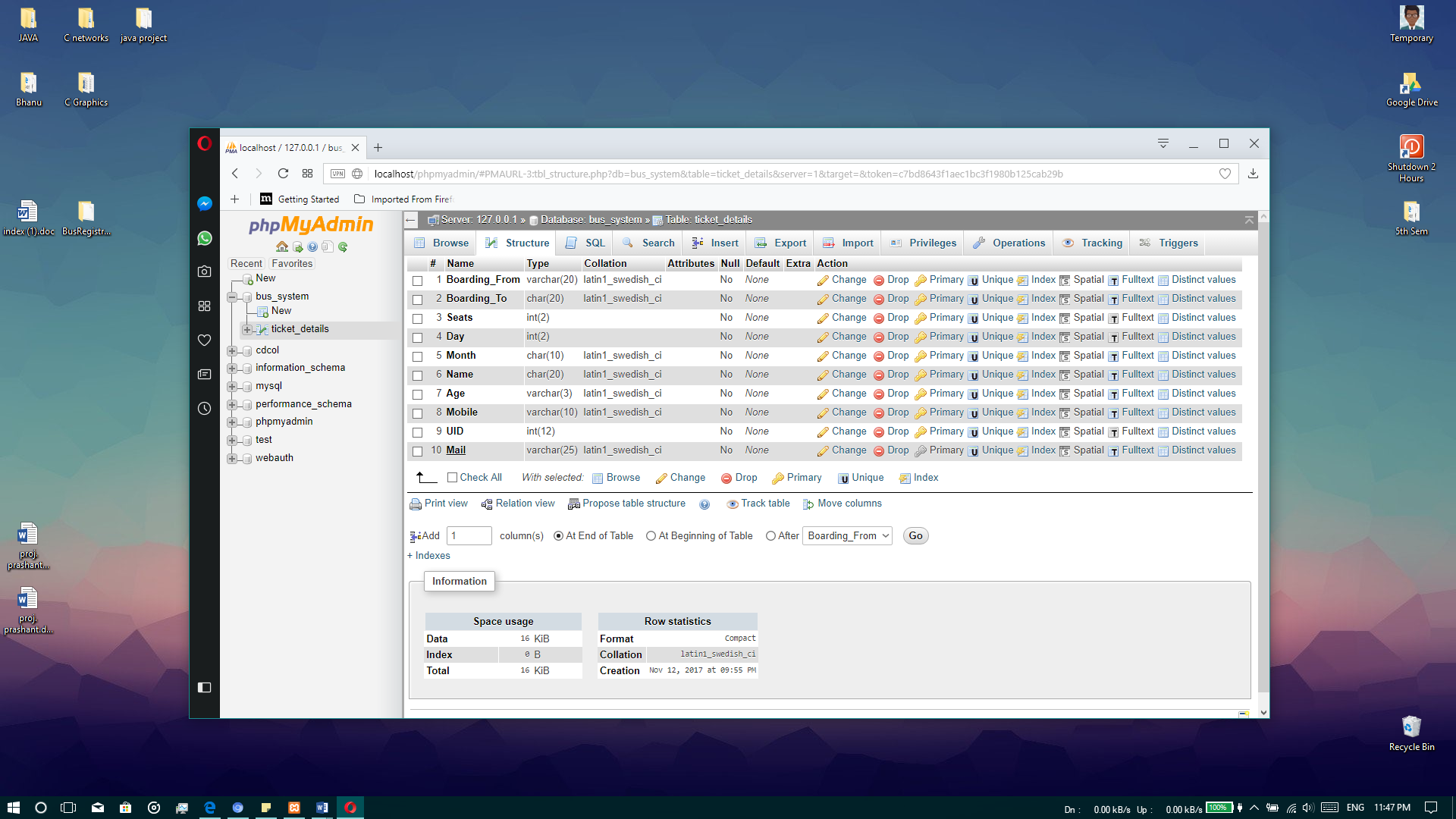
****

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P), and Perl (P). XAMPP is a free and open source [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack)package developed by Apache Friends. It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file.

XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

* **Database Design**

The database is the most important thing in a software as it is the backbone to store the data permanently on computer. This software also includes a database named “bus\_system”. In this database we have created table named “ticker\_details” to maintain records of passenger journey details for ticket. The structure of table is:



**SYSTEM TESTING**

System testing and implementation is a critical process that can even consume more than fifty percent of the development time.

* **TESTING**

The system development life cycle involves the phases of testing and debugging after the requirement analysis, designing and coding. The project has been debugged and implemented successfully.

Two strategies of software testing adopted for the System are as follows:

* Code Testing
* Specification Testing
* **CODE TESTING**

Code Testing was carried out to see the correctness of the logic involved and the correctness of the modules. Tests were conducted based upon sample data, live data and artificial data. All the modules are checked separately for assuring the correctness and accuracy in all calculations.

* **SPECIFICATION TESTING**

It examines the specifications stating what the program should do and how it should perform under various conditions. This testing strategy is a better strategy since it focuses on the way the software is expected to work.

* **UNIT TESTING**

In this testing, each module was tested individually for all tactical units to rectify errors and finally all the modules are integrated and tested. Sample data are used for testing.

* **INTEGRATION TESTING**

The relationship between different modules was checked under this testing and all the modules are integrated into a single system that the system is tested with sample data.

* **USER ACCEPTANCE TESTING**

Once the above testing is successful, the system group tests the developed prototype of the project and their suggestions are incorporated in the prototype to form the overall system.

In this way, Bus ticket reservation system was tested to yield appropriate and efficient results.

**IMPLEMENTATION**

Implementation of a process includes all those activities that take place to convert from old system to the new system. The new system may be totally new, replacing an existing manual or automated system or may be a major modification to an existing system. Proper implementation is essential to provide a reliable system to meet to organizational requirements.

During the implementation stage, a live demo was undertaken and made in front end users. The various features provided in the system are discussed during implementation. Doubts and clarifications were cleared immediately.

* **SYSTEM SPECIFICATIONS**

Hardware Specifications:

The following is the hardware specification of the system on which the software has been developed:-

Operating System : Windows 10

Machine Used : Intel Core i5 2.4 GHz, 8GB RAM, 1TB Hard Disk, 4GB Graphics

Software Specifications:

IDE Used : Intellij Idea by JetBrains

Front end technology : Java

Back end technology : My-SQL

**EXCEPTIONAL TOOLS IN JAVA**

* **Data Base Environment Designer**

The Data Base Environment designer provides an interactive, design –time environment for creating DAO objects. These can be used as a data source for data-base oriented objects on a form or report or accessed programmatically The Data Environment designer supports all the functionality of Visual Basic’s User Connection designer, as well as additional features, including drag and drop, hierarchies, grouping and aggregates.

* **Data Bound Grid Control**

The Data Bound Grid control is a spreadsheet-like bound control that displays a series of rows and columns representing records and fields from a Record set object. We can use the DataBound Grid control to create an application that allows the end user to read and write the most databases. The Data Grid control can be quickly configured at design time with little or no code. When we set the Data Grid control’s Data Source property at design time, the control is automatically filled and its column header are automatically from the data source’s record set. We can then edit the grid’s columns, delete, rearrange, add column headers to, or adjust any column’s width. At run time, the Data Source can be programmatically switched to view a different table, or we can modify the query of the current database to return a different set of records.

* **Java Database Connectivity JDBC**

Java Database Connectivity (JDBC) is an application program interface ([API](http://searchexchange.techtarget.com/definition/application-program-interface)) specification for connecting programs written in [Java](http://searchsoa.techtarget.com/definition/Java) to the data in popular [database](http://searchsqlserver.techtarget.com/definition/database)s. The application program interface lets you encode access request statements in Structured Query Language ([SQL](http://searchsqlserver.techtarget.com/definition/SQL)) that are then passed to the program that manages the database. It returns the results through a similar interface. JDBC is very similar to the SQL Access Group's Open Database Connectivity ([ODBC](http://searchoracle.techtarget.com/definition/Open-Database-Connectivity)) and, with a small "bridge" program, you can use the JDBC interface to access databases through the ODBC interface. For example, you could write a program designed to access many popular database products on a number of system platforms. When accessing a database on a PC running Microsoft's Windows 2000 and, for example, a Microsoft Access database, your program with JDBC statements would be able to access the Microsoft Access database.

**CONCLUSION**

Software for Bus Ticket Reservation is suitable for automating the ticket reservation process in transportation field. In this project we find all details about the journey that are needed during the reservation of ticket. It helps people to reserve the ticket for various buses at any place like their home, office etc. instead of waiting in a long queue for ticket reservation at booking counters. Anyone can book the ticket at any time and can cancel it at any moment. This provide the user a comfortable way of doing the work at any time and at any place.

All the modules and procedures are made simple as possible in order to make the system self-explanatory so that the user of any age group can operate on this software. All the specified objectives of the system is achieved through the complete analysis, proper design and development of system.

The system is designed in such a way that any future enhancements in the system can be made easily.

**SOURCE CODE OF PROJECT**

* **Ticket Reservation Form**

**import** javax.swing.\*;  
**import** javax.swing.border.Border;  
**import** java.awt.\*;  
**import** java.util.\*;  
**import** java.awt.event.\*;  
  
  
**public class** Portal **extends** JFrame{  
 JLabel heading, background;  
 JButton Book\_Ticket, See\_Details, Exit;  
 ImageIcon Portal\_Background;  
 Font font;  
  
 Portal()  
 {  
 setTitle(**"Ticket Reservation"**); *//frame title* setLayout(**null**);  
 setBounds(800,330,390,400);  
 setVisible(**true**);  
 setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);  
  
 font = **new** Font(**null**, Font.BOLD,18);  
 Border border = BorderFactory.createLineBorder(Color.cyan,3);  
  
 *//Portal Heading Label* heading = **new** JLabel(**"BUS TICKET RESERVATION PORTAL"**);  
 add(heading);  
 heading.setBounds(25,100,330,40);  
 heading.setBorder(border);  
 heading.setFont(font);  
 heading.setForeground(Color.magenta);  
  
  
 *//Book\_Ticket Button* Book\_Ticket = **new** JButton(**"BOOK TICKET"**);  
 add(Book\_Ticket);  
 Book\_Ticket.setBounds(60,190,120,30);  
  
 *//See\_Details Button* See\_Details = **new** JButton(**"SEE DETAILS"**);  
 add(See\_Details);  
 See\_Details.setBounds(190,190,120,30);  
  
 *//Exit Button* Exit= **new** JButton(**"CLOSE PORTAL"**);  
 add(Exit);  
 Exit.setBounds(60,230,250,30);  
  
 *//Portal Background* Portal\_Background = **new** ImageIcon(getClass().getResource(**"PortalBackground.png"**));  
 background = **new** JLabel();  
 background.setBounds(0,0,390,400);  
 add(background);  
 background.setIcon(**new** ImageIcon(Portal\_Background.getImage().getScaledInstance(390,400, Image.SCALE\_AREA\_AVERAGING)));  
  
  
 *//Action listener for BookTicket Button* Book\_Ticket.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 **new** Travel\_Information();  
 dispose();  
 }  
 });  
  
  
 *//Acton listener for SeeDetails Button* See\_Details.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 **new** See\_Details();  
 dispose();  
 }  
 });  
  
 Exit.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 dispose();  
 }  
 });  
 }  
  
 *// main function* **public static void** main(String[] args) {  
 **new** Portal();  
 }  
  
  
}

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* **Travel Information Form**

**import** java.awt.\*;  
**import** java.util.\*;  
**import** java.awt.event.\*;  
**import** javax.swing.\*;  
**import** javax.swing.border.Border;  
  
  
**public class** Travel\_Information **extends** JFrame {  
  
 JLabel **Heading**, **Boarding\_From**, **Boarding\_To**, **Bus\_Type**, **Date**, **Available\_Buses**, **No\_of\_seats**, **background**;  
 JTextField **From**, **To**, **SEATS**;  
 JComboBox **Btype**, **day**, **month**, **buses**;  
 JButton **Cancel**, **Next**;  
 ImageIcon **Travel\_Background**;  
 Font **font**;  
  
 **public** String **FROM**, **TO**, **SEAT**, **BUSTYPE**, **DAY**, **MONTH**, **BUSES**;  
  
 Travel\_Information()  
 {  
 **font** = **new** Font(**null**, Font.***BOLD***,18);  
 Border border1 = BorderFactory.*createLineBorder*(Color.***cyan***,3); *// frame border* Border border2 = BorderFactory.*createLineBorder*(Color.***gray***,2); *// label border* setTitle(**"Travel Information"**); *//frame title* setLayout(**null**);  
 setBounds(700,250,500,500);  
 setVisible(**true**);  
 setDefaultCloseOperation(WindowConstants.***EXIT\_ON\_CLOSE***);  
  
  
 *//Heading label* **Heading** = **new** JLabel(**"......TRAVEL INFORMATION......"**);  
 **Heading**.setBounds(100,40,280,40);  
 **Heading**.setBorder(border1);  
 **Heading**.setFont(**font**);  
 **Heading**.setForeground(Color.***magenta***);  
 add(**Heading**);  
  
  
  
 *//Booking From Label and Text Field* **Boarding\_From** = **new** JLabel(**"Boarding From : "**);  
 **Boarding\_From**.setBounds(40,120,100,20);  
 **Boarding\_From**.setForeground(Color.***white***);  
 **Boarding\_From**.setBorder(border2);  
 add(**Boarding\_From**);  
 **From** = **new** JTextField(15);  
 **From**.setBounds(140,120,100,20);  
 add(**From**);  
  
  
 *//Boarding To Label and Text Field* **Boarding\_To** = **new** JLabel(**"To : "**);  
 **Boarding\_To**.setBounds(280,120,60,20);  
 **Boarding\_To**.setForeground(Color.***white***);  
 **Boarding\_To**.setBorder(border2);  
 add(**Boarding\_To**);  
 **To** = **new** JTextField(15);  
 **To**.setBounds(310,120,130,20);  
 add(**To**);  
  
  
 *//BusType label and combo box* **Bus\_Type**= **new** JLabel(**"Bus Type : "**);  
 **Bus\_Type**.setBounds(40,160,100,20);  
 **Bus\_Type**.setForeground(Color.***white***);  
 **Bus\_Type**.setBorder(border2);  
 add(**Bus\_Type**);  
 *//combo box for bus type* String type [] = {**"-Select-"**, **"Ordinary"**, **"AC Bus"**, **"Non-AC Bus"**,**"Volvo"**}; *//list of buses* **Btype** = **new** JComboBox();  
 **for**(**int** i = 0; i<type.**length**; i ++)  
 {  
 **Btype**.addItem(type[i]);  
 }  
 **Btype**.setBounds(140,160,100,20);  
 add(**Btype**);  
  
  
 *//Date label and combo box* **Date** = **new** JLabel(**"Date : "**);  
 **Date**.setBorder(border2);  
 **Date**.setForeground(Color.***white***);  
 **Date**.setBounds(280,160,40,20);  
 add(**Date**);  
 String Day [] = {**" - "**,**"1"**,**"2"**,**"3"**,**"4"**,**"5"**,**"6"**,**"7"**,**"8"**,**"9"**,**"10"**,**"11"**,**"12"**,**"13"**,**"14"**,**"15"**,**"16"**,**"17"**,**"18"**,**"19"**,**"20"**,**"21"**,**"22"**,**"23"**,**"24"**,**"25"**,**"26"**,**"27"**,**"28"**,**"29"**,**"30"**,**"31"**};  
 String Month [] = {**" -- "**,**"January"**, **"February"**, **"March"**, **"April"**, **"May"**, **"June"**, **"July"**, **"August"**, **"September"**, **"October"**, **"November"**, **"December"**};  
 *//day combo box* **day** = **new** JComboBox();  
 **for**(**int** d = 0; d<Day.**length**; d++)  
 {  
 **day**.addItem(Day[d]);  
 }  
 **day**.setBounds(330,160,40,20);  
 add(**day**);  
 *//month combo box* **month** = **new** JComboBox();  
 **for**(**int** m = 0; m<Month.**length**; m++)  
 {  
 **month**.addItem(Month[m]);  
 }  
 **month**.setBounds(370,160,70,20);  
 add(**month**);  
  
  
 *//Availabe buses label and combo box* **Available\_Buses** = **new** JLabel(**"...................................Available Buses for the Route are..................................."**);  
 **Available\_Buses**.setForeground(Color.***white***);  
 **Available\_Buses**.setBorder(border2);  
 **Available\_Buses**.setBounds(40,200,400,20);  
 add(**Available\_Buses**);  
 *//combo box for availabe buses* String avail[] = {**"----------------------------------------------Select--------------------------------------"**,  
 **"Hamirpur-Bilaspur-Shimla-Rohru"**,  
 **"Hamirpur-Bilaspur-Chandigarh"**,  
 **"Bilaspur-Mandi-Manali"**,  
 **"Shimla-Bilaspur-Hamirpur-Kangra"**,  
 **"Sarkaghat-Ghumarwin-Bilaspur-Shimla-Theyog-Hatkoti-Rohru"**,  
 **"Manali-Mandi-Bilaspur-Chandigarh-Delhi"**,  
 **"Mandi-Sundernagar-Shimla-Rohru"**,  
 **"Keylong-Kullu-Mandi-Bilaspur-Chandigarh-Delhi"**,  
 **"Palampur-Kangra-Shimla"**,  
 **"Mandi-Kullu-Manali-Rohtang-Keylong-Leh"**,  
 **"Shimla-Delhi-Jaipur"**};  
  
 **buses** = **new** JComboBox();  
 **for** (**int** b = 0; b< avail.**length**; b++)  
 {  
 **buses**.addItem(avail[b]);  
 }  
 **buses**.setBounds(40,220,400,20);  
 add(**buses**);  
  
 *//no. of seats* **No\_of\_seats** = **new** JLabel(**"Number of Seats : "**);  
 **No\_of\_seats**.setBorder(border2);  
 **No\_of\_seats**.setForeground(Color.***white***);  
 **No\_of\_seats**.setBounds(40,260,120,20);  
 add(**No\_of\_seats**);  
 **SEATS** = **new** JTextField(5);  
 **SEATS**.setBounds(170,260,40,20);  
 add(**SEATS**);  
  
  
 *//cancel button* **Cancel** = **new** JButton(**"CANCEL"**);  
 **Cancel**.setBounds(80,320,150,40);  
 add(**Cancel**);  
  
 *//next button* **Next** = **new** JButton(**"NEXT"**);  
 **Next**.setBounds(240,320,150,40);  
 add(**Next**);  
  
  
  
 *//action listeener for cancel button* **Cancel**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 **new** Portal();  
 dispose();  
 }  
 });  
  
  
 *//action listeener for next button* **Next**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
  
 **if** (**From**.equals(**""**) || **SEATS**.equals(**""**) || **To**.equals(**""**) || **Btype**.getSelectedItem().equals(**"-Select-"**) || **day**.getSelectedItem().equals(**" - "**) || **month**.getSelectedItem().equals(**" -- "**) || **buses**.getSelectedItem().equals(**"-------------"**)) {  
  
 JOptionPane.*showMessageDialog*(**null**, **"All fields are Mandatory!"**);  
 } **else** {  
 *// saving data from label in to variable* String from, to, Seats, Day, Month;  
  
  
 from = **From**.getText();  
 to = **To**.getText();  
 Seats = **SEATS**.getText();  
 Day = (String) **day**.getSelectedItem();  
 Month = (String) **month**.getSelectedItem();  
  
**new** Passenger\_Details(from, to, Seats, Day, Month).setVisible(**true**);  
 dispose();  
 }  
  
 }  
 });  
  
 *//Background* **Travel\_Background** = **new** ImageIcon(getClass().getResource(**"TravelBackground.jpg"**));  
 **background** = **new** JLabel();  
 **background**.setBounds(0, 0, 600, 600);  
 add(**background**);  
 **background**.setIcon(**new** ImageIcon(**Travel\_Background**.getImage().getScaledInstance(600, 600, Image.***SCALE\_AREA\_AVERAGING***)));  
  
}  
 *//main function* **public static void** main(String[] args) {  
 **new** Travel\_Information();  
 }  
}

* **Passenger Details Form**

**import** javafx.scene.control.RadioButton;  
  
**import** java.awt.\*;  
**import** java.util.\*;  
**import** java.awt.event.\*;  
**import** javax.swing.\*;  
**import** javax.swing.border.Border;  
  
*//for database***import** com.mysql.jdbc.Connection;  
**import** java.sql.\*;  
  
  
  
**public class** Passenger\_Details **extends** JFrame  
{  
 JLabel **Headings**, **Name**, **Age**, **Gender**, **Mobile**, **UID**, **Email**, **background**;  
 JTextField **name**, **age**, **mob**, **uid**, **mail**;  
 JRadioButton **Male**, **Female**;  
 JButton **Cancel**, **Confirm**;  
 ImageIcon **Passenger\_Background**;  
 Font **font**;  
  
  
 *//For DataBase* String **url** = **"jdbc:mysql://localhost:3306/bus\_system"**;  
 String **user** = **"root"**;  
 String **password** = **""**  
  
 Passenger\_Details(String FROM, String TO, String SEATS, String DAY, String MONTH){  
  
 **font** = **new** Font(**null**, Font.***BOLD***,18);  
 Border border1 = BorderFactory.*createLineBorder*(Color.***cyan***,3); *// heading bord* Border border2 = BorderFactory.*createLineBorder*(Color.***gray***,2); *// label border* setTitle(**"Passenger Details"**); *//frame title* setLayout(**null**);  
 setBounds(700,250,500,500);  
 setVisible(**true**);  
 setDefaultCloseOperation(WindowConstants.***EXIT\_ON\_CLOSE***);  
  
 *//heading* **Headings** = **new** JLabel(**".......PASSENGER DETAILS........"**);  
 **Headings**.setBounds(100,40,280,40);  
 **Headings**.setBorder(border1);  
 **Headings**.setFont(**font**);  
 **Headings**.setForeground(Color.***magenta***);  
 add(**Headings**);  
 *//NAME LABEL TEXTFIELD* **Name**= **new** JLabel(**"Name "**);  
 **Name**.setBounds(60,120,60,20);  
 **Name**.setForeground(Color.***white***);  
 **Name**.setBorder(border2);  
 add(**Name**);  
 **name** = **new** JTextField(15);  
 **name**.setBounds(140,120,100,20);  
 add(**name**);  
 *//age label textfield* **Age**= **new** JLabel(**"Age "**);  
 **Age**.setBounds(270,120,60,20);  
 **Age**.setForeground(Color.***white***);  
 **Age**.setBorder(border2);  
 add(**Age**);  
 **age** = **new** JTextField(5);  
 **age**.setBounds(340,120,60,20);  
 add(**age**);  
 *//ButtonGroup for gender* **Gender** = **new** JLabel(**"Gender"**);  
 **Gender**.setBounds(60,160,60,20);  
 **Gender**.setBorder(border2);  
 **Gender**.setForeground(Color.***white***);  
 add(**Gender**);  
 ButtonGroup Gender = **new** ButtonGroup();  
 **Male** = **new** JRadioButton(**"M"**);  
 **Male**.setBounds(140,160,40,20);  
 add(**Male**);  
 **Female**= **new** JRadioButton(**"F"**);  
 **Female**.setBounds(180,160,40,20);  
 add(**Female**);  
 Gender.add(**Male**); *// adding into group* Gender.add(**Female**);  
 *//mob label and textbox* **Mobile** = **new** JLabel(**"Mobile"**);  
 **Mobile**.setBorder(border2);  
 **Mobile**.setForeground(Color.***white***);  
 **Mobile**.setBounds(270,160,60,20);  
 add(**Mobile**);  
 **mob** = **new** JTextField(10);  
 **mob**.setBounds(340,160,90,20);  
 add(**mob**);  
 *// Adhaar label and text box* **UID** = **new** JLabel(**"UID"**);  
 **UID**.setBorder(border2);  
 **UID**.setForeground(Color.***white***);  
 **UID**.setBounds(60,200,60,20);  
 add(**UID**);  
 **uid** = **new** JTextField(12);  
 **uid**.setBounds(140,200,110,20);  
 add(**uid**);  
 *//email label text box'* **Email** = **new** JLabel(**"Email"**);  
 **Email**.setBorder(border2);  
 **Email**.setForeground(Color.***white***);  
 **Email**.setBounds(270,200,60,20);  
 add(**Email**);  
 **mail** = **new** JTextField(20);  
 **mail**.setBounds(340,200,120,20);  
 add(**mail**);  
 *//cancel button* **Cancel** = **new** JButton(**"BACK"**);  
 **Cancel**.setBounds(80,320,150,40);  
 add(**Cancel**);  
 *//next button* **Confirm**= **new** JButton(**"CONFIRM"**);  
 **Confirm**.setBounds(240,320,150,40);  
 add(**Confirm**);  
 *//Background* **Passenger\_Background**= **new** ImageIcon(getClass().getResource(**"PassengerBackground.jpg"**));  
 **background** = **new** JLabel();  
 **background**.setBounds(0,0,600,600);  
 add(**background**);  
 **background**.setIcon(**new** ImageIcon(**Passenger\_Background**.getImage().getScaledInstance(600,600, Image.***SCALE\_AREA\_AVERAGING***)));  
 *//action listener for cancel button* **Cancel**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
  
 **new** Travel\_Information();  
 dispose();  
 }  
 });  
  
 **Confirm**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
  
 **if**(**name**.equals(**""**) || **age**.equals(**""**) || **mob**.equals(**""**) || **uid**.equals(**""**) || **mail**.equals(**""**) || Gender.isSelected(**null**))  
 {  
  
 JOptionPane.*showMessageDialog*(**null**, **"All fields are Mandatory!! Kindly fill them in order to continue!"**);  
 }  
  
 **else** {  
 String NAME, AGE, MOB, UID, MAIL;  
  
 NAME = **name**.getText();  
 AGE = **age**.getText();  
 MOB = **mob**.getText();  
 UID = **uid**.getText();  
 MAIL = **mail**.getText();  
**try** {  
 *//1. get a connection to database* Connection myconnection = (Connection) DriverManager.*getConnection*(**url**, **user**, **password**);  
  
 myconnection.setAutoCommit(**true**);  
PreparedStatement pState = (PreparedStatement) myconnection.prepareStatement(**"insert into ticket\_details (Boarding\_From, Boarding\_To, Seats, Day, Month, Name, Age, Mobile, UID, Mail)"** + **" values (?, ?, ?, ? ,?, ?, ?, ?, ?, ?)"**);  
 pState.setString(1,FROM);  
 pState.setString(2,TO);  
 pState.setString(3,SEATS);  
 pState.setString(4,DAY);  
 pState.setString(5,MONTH);  
 pState.setString(6,NAME);  
 pState.setString(7,AGE);  
 pState.setString(8,MOB);  
 pState.setString(9,UID);  
 pState.setString(10,MAIL);  
  
 *//3. execute query* **int** r = pState.executeUpdate();  
  
 JOptionPane.*showMessageDialog*(**null**,**"Booking Successfully Done!"**);  
 **new** See\_Details();  
  
  
 }  
  
 **catch** (Exception ex)  
 {  
 JOptionPane.*showMessageDialog*(**null**,**"Details Already Exist! Enter New Details!"**);  
 ex.printStackTrace();  
 }  
  
 }  
  
 }  
 });  
 }  
  
 **public** Passenger\_Details() {  
  
 }  
  
  
 **public static void** main(String[] args) {  
 **new** Passenger\_Details();  
 }  
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **Final Details Form**

**import** java.awt.\*;  
**import** java.util.\*;  
**import** java.awt.event.\*;  
**import** javax.swing.\*;  
**import** javax.swing.border.Border;  
  
*//for database***import** com.mysql.jdbc.Connection;  
**import** jdk.nashorn.internal.scripts.JO;  
  
**import** java.sql.\*;  
**public class** See\_Details **extends** JFrame {  
  
 JLabel **Heading**, **Name**, **Age**, **Mail**, **UID**, **Mobile**, **From**, **To**, **Seat**, **Date**, **background**;  
 JTextField **name**, **age**, **mail**, **uid**, **mobile**, **from**, **to**, **seat**, **day**, **month**;  
 JButton **Close**, **Ticket**;  
 ImageIcon **Details\_Background**;  
 Font **font**;  
  
 *//For DataBase* String **url** = **"jdbc:mysql://localhost:3306/bus\_system"**;  
 String **user** = **"root"**;  
 String **password** = **""**;  
  
 See\_Details()  
 {  
 **font** = **new** Font(**null**, Font.***BOLD***,18);  
 Border border1 = BorderFactory.*createLineBorder*(Color.***cyan***,3); *// heading border* Border border2 = BorderFactory.*createLineBorder*(Color.***gray***,2); *// label border* setTitle(**"Final Details"**); *//frame details* setLayout(**null**);  
 setBounds(700,250,500,500);  
 setVisible(**true**);  
 setDefaultCloseOperation(WindowConstants.***EXIT\_ON\_CLOSE***);  
 *//heading* **Heading** = **new** JLabel(**"FINAL RESERVATION DETAILS"**);  
 **Heading**.setBounds(100,40,280,40);  
 **Heading**.setFont(**font**);  
 **Heading**.setBorder(border1);  
 **Heading**.setForeground(Color.***white***);  
 add(**Heading**);  
 *//Name Label and text field* **Name** = **new** JLabel(**"Name "**);  
 **Name**.setBounds(60,120,60,20);  
 **Name**.setForeground(Color.***white***);  
 **Name**.setBorder(border2);  
 add(**Name**);  
 **name** = **new** JTextField(15);  
 **name**.setBounds(140,120,100,20);  
 **name**.setEditable(**false**);  
 add(**name**);  
 *//mobile label and text field* **Mobile**= **new** JLabel(**"Mobile"**);  
 **Mobile**.setBounds(270,120,60,20);  
 **Mobile**.setForeground(Color.***white***);  
 **Mobile**.setBorder(border2);  
 add(**Mobile**);  
 **mobile**= **new** JTextField(5);  
 **mobile**.setBounds(340,120,100,20);  
 **mobile**.setEditable(**false**);  
 add(**mobile**);  
 *//mail label and text field* **Mail** = **new** JLabel(**"E-mail "**);  
 **Mail**.setBorder(border2);  
 **Mail**.setBounds(60,150,60,20);  
 **Mail**.setForeground(Color.***white***);  
 add(**Mail**);  
 **mail** = **new** JTextField(20);  
 **mail**.setEditable(**false**);  
 **mail**.setBounds(140,150,100,20);  
 add(**mail**);  
 *//uid label and text field* **UID** = **new** JLabel(**"UID"**);  
 **UID**.setBorder(border2);  
 **UID**.setForeground(Color.***white***);  
 **UID**.setBounds(270,150,60,20);  
 add(**UID**);  
 **uid** = **new** JTextField(12);  
 **uid**.setBounds(340,150,100,20);  
 **uid**.setEditable(**false**);  
 add(**uid**);  
 *//From label and text field* **From**= **new** JLabel(**"From "**);  
 **From**.setBounds(60,200,60,20);  
 **From**.setForeground(Color.***white***);  
 **From**.setBorder(border2);  
 add(**From**);  
 **from** = **new** JTextField(10);  
 **from**.setBounds(140,200,100,20);  
 **from**.setEditable(**false**);  
 add(**from**);  
 *//to label and text field* **To**= **new** JLabel(**"To "**);  
 **To**.setBounds(270,200,60,20);  
 **To**.setForeground(Color.***white***);  
 **To**.setBorder(border2);  
 add(**To**);  
 **to** = **new** JTextField(10);  
 **to**.setBounds(340,200,100,20);  
 **to**.setEditable(**false**);  
 add(**to**);  
 *//age label and text field* **Age**= **new** JLabel(**"Age "**);  
 **Age**.setBounds(70,250,40,20);  
 **Age**.setForeground(Color.***white***);  
 **Age**.setBorder(border2);  
 add(**Age**);  
 **age** = **new** JTextField(5);  
 **age**.setBounds(120,250,40,20);  
 **age**.setEditable(**false**);  
 add(**age**);  
 *//seat label and text field* **Seat** = **new** JLabel(**"Seats"**);  
 **Seat**.setBounds(180,250,40,20);  
 **Seat**.setForeground(Color.***white***);  
 **Seat**.setBorder(border2);  
 add(**Seat**);  
 **seat** = **new** JTextField(5);  
 **seat**.setEditable(**false**);  
 **seat**.setBounds(230,250,40,20);  
 add(**seat**);  
 *//day and month label and text field;* **Date** = **new** JLabel(**"Date"**);  
 **Date**.setBorder(border2);  
 **Date**.setForeground(Color.***white***);  
 **Date**.setBounds(290,250,40,20);  
 add(**Date**);  
 **day** = **new** JTextField(5);  
 **day**.setEditable(**false**);  
 **day**.setBounds(340,250,30,20);  
 add(**day**);  
 **month** = **new** JTextField(10);  
 **month**.setEditable(**false**);  
 **month**.setBounds(380,250,50,20);  
 add(**month**);  
 *//close button* **Close**= **new** JButton(**"CLOSE"**);  
 **Close**.setBounds(250,330,170,40);  
 add(**Close**);  
 *//Ticket Button* **Ticket** = **new** JButton(**"CANCEL TICKET!"**);  
 **Ticket**.setBounds(70,330,170,40);  
 add(**Ticket**);  
 *//Background* **Details\_Background**= **new** ImageIcon(getClass().getResource(**"detailsBackground.jpg"**));  
 **background** = **new** JLabel();  
 **background**.setBounds(0,0,500,500);  
 add(**background**);  
 **background**.setIcon(**new** ImageIcon(**Details\_Background**.getImage().getScaledInstance(500,500, Image.***SCALE\_AREA\_AVERAGING***)));  
 *//JoptionPane for email address* String Emailaddress = JOptionPane.*showInputDialog*(**null**,**"Enter Your Registered Email Address : "**); *//database fetching* **try**{  
 Connection mycon = (Connection) DriverManager.*getConnection*(**url**,**user**,**password**);  
  
 String sql = **"SELECT Boarding\_From, Boarding\_To, Seats, Day, Month, Name, Age, Mobile, UID, Mail FROM ticket\_details WHERE Mail = ?"**;  
  
  
 PreparedStatement pst = mycon.prepareStatement(sql);  
  
 pst.setString(1,Emailaddress);  
  
 ResultSet rs = pst.executeQuery();  
  
 **if** (rs.next())  
 {  
 String From = rs.getString(**"Boarding\_From"**);  
 **from**.setText(From);  
  
 String To = rs.getString(**"Boarding\_To"**);  
 **to**.setText(To);  
  
 String Seats = rs.getString(**"Seats"**);  
 **seat**.setText(Seats);  
  
 String Day = rs.getString(**"Day"**);  
 **day**.setText(Day);  
  
 String Month = rs.getString(**"Month"**);  
 **month**.setText(Month);  
  
 String Name = rs.getString(**"Name"**);  
 **name**.setText(Name);  
  
 String Age = rs.getString(**"Age"**);  
 **age**.setText(Age);  
  
 String Mobile = rs.getString(**"Mobile"**);  
 **mobile**.setText(Mobile);  
  
 String UID = rs.getString(**"UID"**);  
 **uid**.setText(UID);  
  
 String Mail = rs.getString(**"Mail"**);  
 **mail**.setText(Mail);  
 }  
  
 **else** {  
  
 JOptionPane.*showMessageDialog*(**null**,**"Details Not Found!"**);  
 dispose();  
 **new** Portal();  
  
 }  
 }  
  
 **catch** (Exception ex)  
 {  
 ex.printStackTrace();  
  
 }  
  
  
 *//action listener for Close button* **Close**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
  
 dispose();  
 **new** Portal();  
 }  
 });  
  
  
  
  
 *//action listener for Ticket Button* **Ticket**.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
  
 *// first of all, asking for confirmation from user* **int** YorN = JOptionPane.*showConfirmDialog*(**null**,**"Do you really want to cancel ticket?"**, **"Cancel Ticket!"**, JOptionPane.***YES\_NO\_OPTION***);  
  
 **if**(YorN == 0) *// if yes is pressed* {  
 **try** {  
  
 Connection myconn = (Connection) DriverManager.*getConnection*(**url**, **user**, **password**);  
  
 Statement stat = myconn.createStatement();  
  
 PreparedStatement pst = myconn.prepareStatement(**"DELETE FROM ticket\_details WHERE Mail = ?"**); *// delete statement* pst.setString(1, Emailaddress); *//passing primary key value* **int** rs = pst.executeUpdate(); *//executing update instead of executeQuery()* JOptionPane.*showMessageDialog*(**null**,**"Ticket Has Been Canceled Successfully"**); *//Showing msg to user after the deletion of ticket details  
  
 //now erasing the data from all of text fields* **name**.setText(**""**); **uid**.setText(**""**);  
 **age**.setText(**""**); **mobile**.setText(**""**);  
 **mail**.setText(**""**); **from**.setText(**""**);  
 **to**.setText(**""**); **seat**.setText(**""**);  
 **day**.setText(**""**); **month**.setText(**""**);  
  
 dispose(); *// now closing the See\_Details Frame* **new** Portal(); *//and opening the Portal again* } **catch** (Exception exc) {  
 exc.printStackTrace();  
 }  
 }  
  
  
 }  
 });  
  
  
  
 }  
  
 *//main function* **public static void** main(String[] args) {  
 **new** See\_Details();  
 }  
  
}