

STADIUM SEAT BOOKING SYSTEM



A Course Project Report in the course

Object Oriented Programming Concepts through java

Computer Science & Engineering

By Batch-07

18K41A0539

NAHEER FATIMA

18K41A0550

SANA FARHEEN

18K41A0551

SAMBARI PRAVEEN KUMAR

Under the Guidance of

Mr. K. Ravi Chythanya
Assoc. Prof

Submitted on APRIL 2020

Submitted to Mr. K. Ravi Chythanya



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SR ENGINEERING COLLEGE

Approved by AICTE, New Delhi | Affiliated to JNTUH | Accredited by NAAC 'A' Grade
Ananthasagar (V), Hasanparthy (M), WARANGAL - 506 371
(An Autonomous Institution)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the **Object Oriented Programming through Java - Course Project** Report entitled “**STADIUM SEAT BOOKING SYSTEM**” a record of bonafide work carried out by the students Naheer Fatima, Sana Farheen, Praveen kumar bearing RollNo(s)18K41A0539, 18K41A0550, 18K41A0551 during the academic year 2019-20 in partial fulfillment of the award of the degree of *Bachelor of Technology* in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

Lab In-charge

Head of the Department

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere and deep sense of respect to our beloved principal **Dr. V Mahesh, S R Engineering College** for providing an excellent academic atmosphere in the institution. We express heartfelt thanks to the Head of Department, Computer Science & Engineering **Mrs. M.Sheshikala** for providing us with necessary infrastructure and there by giving us freedom to carry out the work. We also thank the other staff members and friends who assisted us. Finally, We thank our parents who inspired us always to do the best.

18K41A0539:NAHEER FATIMA

18K41A0550:SANA FARHEEN

18K41A0551:SAMBARI PRAVEEN KUMAR

TABLE OF CONTENTS:

S.NO	TITLE	PAGE NO
1.	OBJECTIVES OF PROJECT	1
2.	DEFINITIONS OF ELEMENTS USED IN THE PROJECT	2-4
3.	DESIGN	
3.1	SCREENS	5-9
4.	IMPLEMENTATION	
4.1	CODE	10-38
5.	RESULT SCREENS	39-44
6.	CONCLUSION	45

ABSTRACT

Stadium is a centerpiece for the hosting of major events especially sport events such as football. It is one thing to have a stadium, and it's another thing to manage it. Ticket acquisition is a vital aspect of stadium management. The system can also be considered to be a critical one, as the failure of it would result in significant financial loss for the company managing the stadium, and, in the case of the software failing, potentially thousands of angry fans stuck outside the ground, many of whom may decide not to return – against this would have implications for the managers of the stadium, and probably the software developers too. Therefore, the system must be created using well established.

The main aim of our project is to manage the details of seats, show times, match name, show date, cost. It manages the booking of above things. The project is totally built for users where as only the admin is guaranteed the access to update or change the details. The purpose of the project is to build an application program to reduce the manual work for managing the seats, booking. This project gives crystal clear information which can be easily understood by the customer.

1.OBJECTIVE OF THE PROJECT

It provides the searching facilities based on seats, show time, stadium names, show dates. The concept of files is used to store the information of registered customer details as well as reserved seats for specific show or a stadium. It provides efficiency and security of data.

2.DEFINATIONS OF THE ELEMENTS USED IN THE PROJECT

Mainly we have used the concept of swings. Files are used to store the data, if we store data in files, it will be safe and secured. The concepts we have used are:

- **Swings:**

It is used to create window-based application and is built on the top of AWT (Abstract Windowing Toolkit) API (Application Programming Interface) and is entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc. Below are the ones which we have used:

- **JLabel:**

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

- **JPanel:**

The JPanel is a simplest container class. It provides space in which an application can attach any other component. It inherits the JComponents class.

- **JTextField:**

The object of a JTextField class is a text component that allows the editing of a single line text.

- **JTextArea:**

The object of a JTextArea class is a multi line region that displays text. It allows the editing of multiple line text.

➤ **JButton:**

The JButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed.

➤ **JCheckBox:**

The JCheckBox class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a CheckBox changes its state from "on" to "off" or from "off" to "on ".

➤ **JPasswordField:**

The object of a JPasswordField class is a text component specialized for password entry. It allows the editing of a single line of text.

➤ **JOptionPane:**

The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user.

• **ArrayList:**

The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user. The important points about Java ArrayList class are:

- Java ArrayList class can contain duplicate elements.
- Java ArrayList class maintains insertion order.
- Java ArrayList class is non synchronized.
- Java ArrayList allows random access because array works at the index basis.

- In Java ArrayList class, manipulation is slow because a lot of shifting needs to occur if any element is removed from the array list.

- **Serialization:**

Serialization in Java is a mechanism of *writing state of object into byte stream*. The reverse operation of serialization is called *deserialization* where byte-stream is converted into an object. The serialization and deserialization process is platform-independent, it means you can serialize an object in a platform and deserialize in different platform.

- **Iterators:**

Iterators are used in Collection framework in Java to retrieve elements one by one.

- **Calendar:**

Java Calendar class is an abstract class that provides methods for converting date between a specific instant in time and a set of calendar fields such as MONTH, YEAR, HOUR, etc.

- **I/O Streams:**

Mainly we have used Byte Streams. The reason to use byte streams is it hides the data i.e data is unreadable to the people.

- **Exception Handling:**

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions. Here have try and catch blocks to handle exceptions.

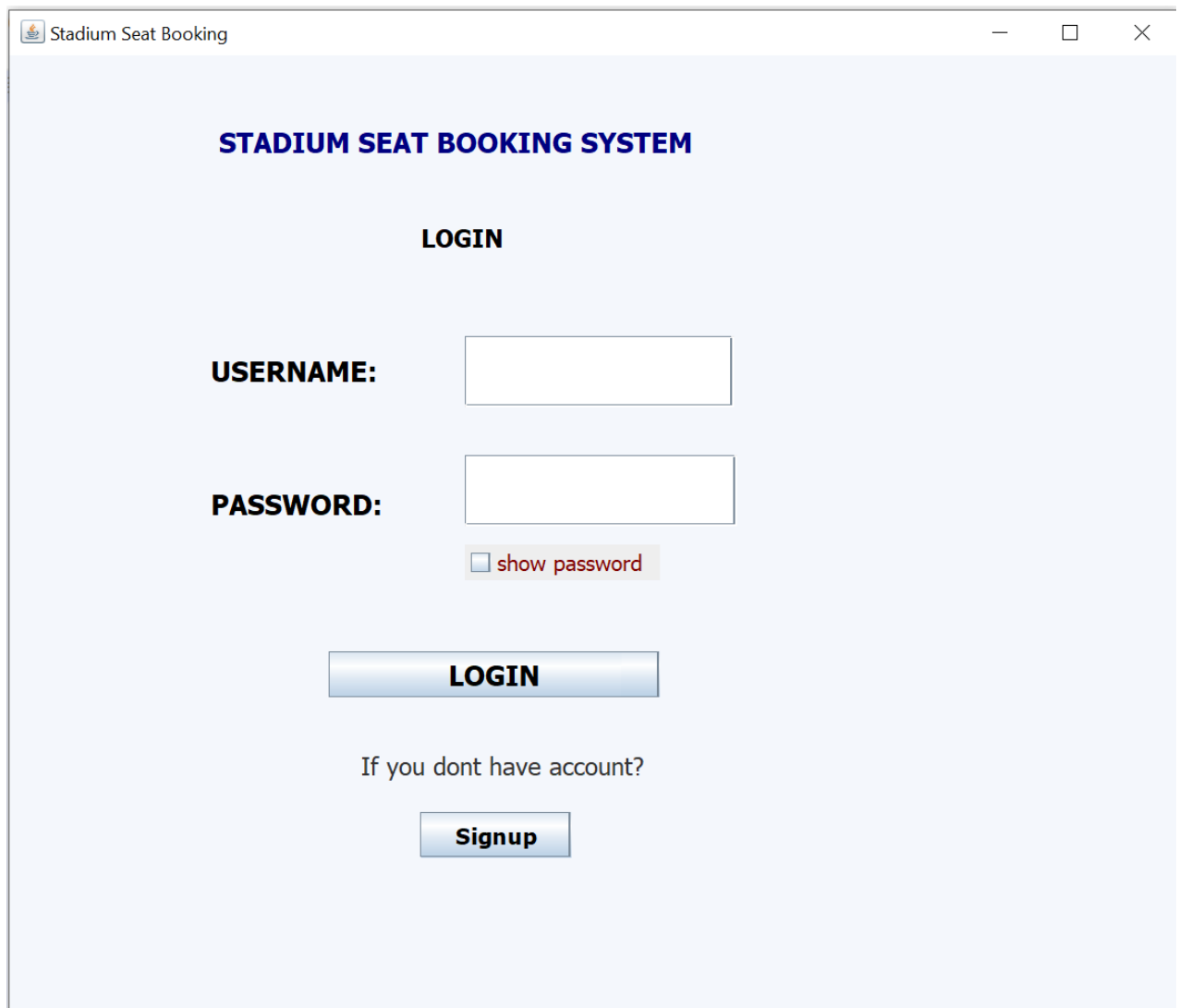
- **Java Event Handling:**

Changing the state of an object is known as an event. For example, click on button, dragging mouse etc. The java.awt.event package provides many event classes and Listener interfaces for event handling.

3.DESIGN

3.1 SCREENS


- **Login Frame:**



The image shows a screenshot of a web application window titled "Stadium Seat Booking". The window has a light blue background and a white title bar with standard window controls (minimize, maximize, close). The main content area is centered and contains the following elements:

- STADIUM SEAT BOOKING SYSTEM**: A bold, dark blue header.
- LOGIN**: A bold, black sub-header.
- USERNAME:** A label followed by a white text input field.
- PASSWORD:** A label followed by a white text input field.
- ☐ **show password**: A checkbox with a red label, positioned below the password field.
- LOGIN**: A blue button with a gradient and a shadow, positioned below the password field.
- If you dont have account?**: A text link positioned below the login button.
- Signup**: A blue button with a gradient and a shadow, positioned below the text link.

- **Registration Frame:**

 Stadium Seat Booking - SignUp—□×

SIGN UP

First Name : **Last Name** :

Age :

Gender : ☐ **Male** ☐ **Female**

E-Mail :

Username ...

Password :

- **Third Frame:**

Stadium seat booking

— □ ×

Wel-Come :

Log Out

Stadium Name

Select Stadium ▼

Choose Date

28-Mar-2020 ▼

Choose Time

11 : 00 AM to 11 : 00 PM ▼


No. of Seats

1 ▼

Select Seats

Show Last Boo...Show All Booki...

- **Fourth Frame:**

 Stadium seat booking

Stadium Na... Rajiv Gandhi International Cricket Stadium, Hyd...

Dat... 28-Mar-2020

Timminqs ... 11 : 00 AM to 11 : 00 PM

Number Of Se... 1

	<input type="checkbox"/> C1	<input type="checkbox"/> C2	<input type="checkbox"/> C3	<input type="checkbox"/> C4	<input type="checkbox"/> C5	<input type="checkbox"/> C6			
	<input type="checkbox"/> B1	<input type="checkbox"/> B2	<input type="checkbox"/> B3	<input type="checkbox"/> B4	<input type="checkbox"/> B5	<input type="checkbox"/> B6	<input type="checkbox"/> B7	<input type="checkbox"/> B8	
<input type="checkbox"/> C28	<input type="checkbox"/> B32	<input type="checkbox"/> A1	<input type="checkbox"/> A2	<input type="checkbox"/> A3	<input type="checkbox"/> A4	<input type="checkbox"/> A5	<input type="checkbox"/> A6	<input type="checkbox"/> B9	<input type="checkbox"/> C7
<input type="checkbox"/> C27	<input type="checkbox"/> B31	<input type="checkbox"/> A24					<input type="checkbox"/> A7	<input type="checkbox"/> B10	<input type="checkbox"/> C8
<input type="checkbox"/> C26	<input type="checkbox"/> B30	<input type="checkbox"/> A23					<input type="checkbox"/> A8	<input type="checkbox"/> B11	<input type="checkbox"/> C9
<input type="checkbox"/> C25	<input type="checkbox"/> B29	<input type="checkbox"/> A22					<input type="checkbox"/> A9	<input type="checkbox"/> B12	<input type="checkbox"/> C10
<input type="checkbox"/> C24	<input type="checkbox"/> B28	<input type="checkbox"/> A21					<input type="checkbox"/> A10	<input type="checkbox"/> B13	<input type="checkbox"/> C11
<input type="checkbox"/> C23	<input type="checkbox"/> B27	<input type="checkbox"/> A20					<input type="checkbox"/> A11	<input type="checkbox"/> B14	<input type="checkbox"/> C12
<input type="checkbox"/> C22	<input type="checkbox"/> B26	<input type="checkbox"/> A19					<input type="checkbox"/> A12	<input type="checkbox"/> B15	<input type="checkbox"/> C13
<input type="checkbox"/> C21	<input type="checkbox"/> B25	<input type="checkbox"/> A18	<input type="checkbox"/> A17	<input type="checkbox"/> A16	<input type="checkbox"/> A15	<input type="checkbox"/> A14	<input type="checkbox"/> A13	<input type="checkbox"/> B16	<input type="checkbox"/> C14
	<input type="checkbox"/> B24	<input type="checkbox"/> B23	<input type="checkbox"/> B22	<input type="checkbox"/> B21	<input type="checkbox"/> B20	<input type="checkbox"/> B19	<input type="checkbox"/> B18	<input type="checkbox"/> B17	
		<input type="checkbox"/> C20	<input type="checkbox"/> C19	<input type="checkbox"/> C18	<input type="checkbox"/> C17	<input type="checkbox"/> C16	<input type="checkbox"/> C15		

Select Seats...

Amount :

A1 - A24 : UPPERCLASS | 15...
B1 - B32 : MIDDLECLASS | 10...

BACK

PROCCED

- **Fifth Frame:**

Wel-Come :

Log Out

SUCCESSFULLY BOOKED !

Name

Stadium Name

Timing

Date

No Of Seats <dynamic> Cost <dynamic>

Seat Numb...

THANK YOU :-)

4.IMPLEMENTATION

4.1CODE

- **Login Page Code:**

```
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.awt.*;
import java.util.*;

class User implements Serializable {
    String password;
    String uname;
    User(String uname, String password) {
        this.uname = uname;
        this.password = password;
    }
    public String getPassword(){
        return password;
    }
    public String getUname(){
        return uname;
    }
}

public class Screen1MainPage{
    private static JTextFieldusernamevalue;
    private static JPasswordFieldpasswordvalue;
    public
    boolean nsa1=true, nsa2=true, nsa3=true, nsa4=true, nsa5=true, nsa6=true, nsa7=true, nsa8=true, nsa9
    =true, nsa10=true;
```

```

        public boolean
nsa11=true,nsa12=true,nsa13=true,nsa14=true,nsa15=true,nsa16=true,nsa17=true,nsa18=true,nsa
19=true,nsa20=true;

        public boolean nsa21=true,nsa22=true,nsa23=true,nsa24=true;

        public boolean
nsb1=true,nsb2=true,nsb3=true,nsb4=true,nsb5=true,nsb6=true,nsb7=true,nsb8=true,nsb9=true,n
sb10=true;

        public boolean
nsb11=true,nsb12=true,nsb13=true,nsb14=true,nsb15=true,nsb16=true,nsb17=true,nsb18=true,ns
b19=true,nsb20=true;

        public boolean
nsb21=true,nsb22=true,nsb23=true,nsb24=true,nsb25=true,nsb26=true,nsb27=true,nsb28=true,ns
b29=true,nsb30=true,nsb31=true,nsb32=true;

        public boolean
nsc1=true,nsc2=true,nsc3=true,nsc4=true,nsc5=true,nsc6=true,nsc7=true,nsc8=true,nsc9=true,ns
c10=true;

        public boolean
nsc11=true,nsc12=true,nsc13=true,nsc14=true,nsc15=true,nsc16=true,nsc17=true,nsc18=true,ns
c19=true,nsc20=true;

        public boolean
nsc21=true,nsc22=true,nsc23=true,nsc24=true,nsc25=true,nsc26=true,nsc27=true,nsc28=true;

        FileInputStream fin;
        ObjectInputStreamois;
        public static void main(String[] args)
        {
                new Screen1MainPage();
        }
        Screen1MainPage(){
                Values();
        }

```



```

void Values(){
    ArrayList<User> al=new ArrayList<User>();
    JFramejf= new JFrame("Stadium Seat Booking");
    jf.getContentPane().setForeground(SystemColor.activeCaption);
    jf.setForeground(Color.WHITE);
    jf.getContentPane().setBackground(SystemColor.inactiveCaptionBorder);
    jf.setSize(705,600);
    jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    jf.getContentPane().setLayout(null);

```

```

ShowPassword.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(ShowPassword.isSelected())
            passwordvalue.setEchoChar((char)0);
        else
            passwordvalue.setEchoChar('*');
    }
});

```

```

LoginButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        boolean flag = false;
        try {
            fin = new FileInputStream("LoginDetails.dat");
            ois = new ObjectInputStream(fin);
            ArrayList<User> al1;
            String name = usernamevalue.getText();
            String pass = passwordvalue.getText()getText();
            while((al1=(ArrayList)ois.readObject())!=null){
                User u1 = al1.get(0);
                if(pass.equals(u1.getPassword())&&name.equals(u1.getUname())){
                    flag = true;

```

```

        break;
    }
}
if(flag==true){
    JOptionPane.showMessageDialog(jf,"Login Successfull !"); new
MainFunction(name,nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa
14,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,ns
b6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb2
1,nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,n
sc5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,n
sc21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);
    jf.setVisible(false);
}
else{
JOptionPane.showMessageDialog(jf,"Please Register then Login or Password may Incorrect ");
}
} catch(Exception er){}
if(flag == false) {
if(usernamevalue.getText().equals("") || passwordvalue.getText().equals("")) {
JOptionPane.showMessageDialog(jf, "Please enter Fields !");
}
else {
JOptionPane.showMessageDialog(jf,"Please Register then or Password may Incorrect !");
}}}
});
SignUpButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new
Register(nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa14,nsa15,ns
a16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb6,nsb7,nsb
8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,nsb22,nsb

```

```

23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc5,nsc6,nsc
7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc21,nsc22,n
sc23,nsc24,nsc25,nsc26,nsc27,nsc28);

        jf.setVisible(false);
    }

});

}

}

```

- **Registration Frame code:**

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import java.io.*;
import java.util.ArrayList;
class User implements Serializable {
    String password;
    String uname;
    User(String uname, String password) {
        this.uname = uname;
        this.password = password;
    }
    public String getPassword(){
        return password;
    }
    public String getUname(){
        return uname;
    }
}

class Register extends JFrame{

```

```

        FileOutputStreamfout;
        ObjectOutputStreammoos;
        FileInputStream fin;
        ObjectInputStreammois;
        boolean
        nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa14,nsa15,nsa16,nsa1
        7,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb6,nsb7,nsb8,nsb9,ns
        b10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,nsb22,nsb23,nsb24,
        nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc5,nsc6,nsc7,nsc8,ns
        c9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc21,nsc22,nsc23,nsc
        24,nsc25,nsc26,nsc27,nsc28;

        public static void main(String[] args) throws Exception
        {
            new Register();
        }

        Register(){
            Values();
        }

        void Values() {
            ArrayList<User> al=new ArrayList<User>();
            JFramejf=new JFrame("Stadium Seat Booking - SignUp");
            jf.getContentPane().setBackground(SystemColor.inactiveCaptionBorder);
            jf.setSize(500,500);
            jf.getContentPane().setLayout(null);
            jf.setLocation(0, 0);
            SignUpbutton.addActionListener(new ActionListener(){
                public void actionPerformed(ActionEvent e){
                    String u = usernamevalue.getText();
                    String p = passwordvalue.getText();
                    if(usernamevalue.getText().equals("") || passwordvalue.getText().equals("")) {

```

```

JOptionPane.showMessageDialog(jf, "Please enter Fields !");
    }
    else {
    try {
        al.add(new User(u,p));
        File file = new File("LoginDetails.dat");
        if(file.exists()) {
oos = new ObjectOutputStream(new FileOutputStream(file,true)) {
            @Override
            protected void writeStreamHeader() throws IOException {
                reset();
            }
        };
        }
        else
oos = new ObjectOutputStream(new FileOutputStream(file));
        oos.writeObject(al);
        usernamevalue.setText("");
        passwordvalue.setText("");

        JOptionPane.showMessageDialog(jf,"Signup Successfull ! You can Login now !!");
new
Screen1MainPage(nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa1
4,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb
6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,
nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc
5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc
21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);

        jf.setVisible(false);
        al.clear();
        oos.close();
    }catch(Exception ae){

```

```

        ae.printStackTrace();
    }
}

});

btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new
Screen1MainPage(nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa1
4,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb
6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,
nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc
5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc
21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);

        jf.setVisible(false);
    }
});

JButtonAlredyRegisteredbutton = new JButton("Already Registered ? Login Now");
AlredyRegisteredbutton.setFont(new Font("Tw Cen MT", Font.PLAIN, 15));
AlredyRegisteredbutton.setBounds(105, 345, 267, 35);
jf.getContentPane().add(AlredyRegisteredbutton);
AlredyRegisteredbutton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new
Screen1MainPage(nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa1
4,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb
6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,
nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc
5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc
21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);

        jf.setVisible(false);
    }
});

```

```

        }
    });
    jf.setVisible(true);
}
}

```

- **Third Frame Code:**

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.FileInputStream;
import java.io.ObjectInputStream;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Iterator;

class Bookings implements Serializable {
    String username1,sname1,date1,time1,seats1,bookingdate,bookingtime;
    int n1,cost1;
    Bookings(String username1,String sname1,String date1,String time1,int n1,int
cost1,String seats1,String bookingdate,Stringbookingtime) {
        this.username1 = username1;
        this.sname1 = sname1;
        this.date1 = date1;
        this.time1= time1;
        this.n1=n1;
        this.cost1=cost1;
        this.seats1=seats1;
        this.bookingdate=bookingdate;
        this.bookingtime=bookingtime;
    }

    public class MainFunction extends JFrame {

```

```

    public String sname,username;
    public int n;
    String sname2,date2,time2,seats2;
    int n2,cost2;
    boolean
    nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa14,nsa15,nsa16,nsa1
    7,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb6,nsb7,nsb8,nsb9,ns
    b10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,nsb22,nsb23,nsb24,
    nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc5,nsc6,nsc7,nsc8,ns
    c9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc21,nsc22,nsc23,nsc
    24,nsc25,nsc26,nsc27,nsc28;
    FileInputStream fin;
    ObjectInputStreamois;
    private JTextFieldbookings1;
    public static void main(String[] args) {
        new MainFunction();
    }
    MainFunction(){
        Values();
    }
    void Values() {
        ArrayList<Bookings>al=new ArrayList<Bookings>();
        JFramejf=new JFrame("Stadium seat booking");
        jf.setSize(700,600);
        jf.getContentPane().setBackground(SystemColor.inactiveCaptionBorder);
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jf.getContentPane().setLayout(null);
        JLabelWelCome = new JLabel("Wel-Come :");
        WelCome.setFont(new Font("Tw Cen MT", Font.BOLD, 18));
        WelCome.setBounds(10, 10, 261, 29);
        jf.getContentPane().add(WelCome);

```



```

JLabelusernameText = new JLabel("");
usernameText.setFont(new Font("Tw Cen MT", Font.BOLD, 18));
usernameText.setBounds(103, 10, 209, 29);
jf.getContentPane().add(usernameText);
usernameText.setText(username);
JButtonLogOut = new JButton("Log Out");
LogOut.setForeground(Color.BLACK);
LogOut.setFont(new Font("Tw Cen MT", Font.BOLD, 18));
LogOut.setBounds(555, 9, 121, 30);
jf.getContentPane().add(LogOut);
LogOut.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int input =JOptionPane.showConfirmDialog(jf, "Do you want LogOut ?");
        if(input==0) {
            new
Screen1MainPage(nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa1
4,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb
6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,
nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc
5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc
21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);
            jf.setVisible(false);
            JOptionPane.showMessageDialog(jf,"Logout Success !");
        }
    }
});
JComboBoxSelectStadium = new JComboBox();
SelectStadium.setFont(new Font("Tw Cen MT", Font.BOLD, 17));
SelectStadium.setModel(new DefaultComboBoxModel(new String[] {"Select
Stadium","Rajiv Gandhi International Cricket Stadium, Hyderabad."}));
SelectStadium.setBounds(174, 82, 471, 39);

```

```

jf.getContentPane().add(SelectStadium);
JComboBoxSelectTime = new JComboBox();
SelectTime.setModel(new DefaultComboBoxModel(new String[] { "11 : 00 AM to 11 :
00 PM" }));
SelectTime.setFont(new Font("Tw Cen MT", Font.BOLD, 17));
SelectTime.setBounds(174, 206, 237, 30);
jf.getContentPane().add(SelectTime);
JLabelStadiumName = new JLabel("Stadium Name ");
StadiumName.setFont(new Font("Tw Cen MT", Font.PLAIN, 17));
StadiumName.setBounds(43, 86, 121, 30);
jf.getContentPane().add(StadiumName);
JLabelChooseDate = new JLabel("Choose Date");
ChooseDate.setFont(new Font("Tw Cen MT", Font.PLAIN, 17));
ChooseDate.setBounds(43, 146, 121, 30);
jf.getContentPane().add(ChooseDate);
JLabelChooseTime = new JLabel("Choose Time");
ChooseTime.setFont(new Font("Tw Cen MT", Font.PLAIN, 17));
ChooseTime.setBounds(43, 206, 121, 30);
jf.getContentPane().add(ChooseTime);
JComboBoxnos = new JComboBox();
nos.setModel(new DefaultComboBoxModel(new String[] { "1", "2", "3", "4", "5", "6" }));
nos.setFont(new Font("Tw Cen MT", Font.BOLD, 17));
nos.setBounds(174, 269, 152, 30);
jf.getContentPane().add(nos);
JLabelNoOfSeats = new JLabel("No. of Seats");
NoOfSeats.setFont(new Font("Tw Cen MT", Font.PLAIN, 17));
NoOfSeats.setBounds(43, 269, 111, 30);
jf.getContentPane().add(NoOfSeats);
JButtonSelectSeats = new JButton("Select Seats");
SelectSeats.setForeground(Color.BLACK);
SelectSeats.setFont(new Font("Tw Cen MT", Font.BOLD, 25));

```

```

SelectSeats.setBounds(207, 365, 261, 48);
jf.getContentPane().add(SelectSeats);
JComboBoxSelectDate = new JComboBox();
SelectDate.setModel(new DefaultComboBoxModel(new String[] { "28-Mar-2020" }));
SelectDate.setFont(new Font("Tw Cen MT", Font.BOLD, 17));
SelectDate.setBounds(174, 146, 237, 30);
jf.getContentPane().add(SelectDate);
JLabelWarningS = new JLabel("");
WarningS.setForeground(Color.RED);
WarningS.setFont(new Font("Tw Cen MT", Font.PLAIN, 15));
WarningS.setBounds(174, 123, 236, 21);
jf.getContentPane().add(WarningS);
JTextAreaYourBookings = new JTextArea();
JScrollPane sp = new JScrollPane(YourBookings);
sp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);
sp.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
sp.setBounds(10, 439, 666, 114);
jf.getContentPane().add(sp, BorderLayout.CENTER);
JButtonShowBookings = new JButton("Show All Bookings");
ShowBookings.setForeground(Color.BLACK);
ShowBookings.setFont(new Font("Tw Cen MT", Font.BOLD, 15));
ShowBookings.setBounds(427, 418, 539, 21);
jf.getContentPane().add(ShowBookings);
JButtonLastBooking = new JButton("Show Last Booking");
LastBooking.setForeground(Color.BLACK);
LastBooking.setFont(new Font("Tw Cen MT", Font.BOLD, 15));
LastBooking.setBounds(153, 419, 521, 21);
jf.getContentPane().add(LastBooking);
SelectSeats.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String sname = (String) SelectStadium.getSelectedItemAt();
    }
});

```

```

String date = (String) SelectDate.getSelectedItemAt();
String time = (String) SelectTime.getSelectedItemAt();
int n = Integer.parseInt((String) nos.getSelectedItemAt());
if(sname == (String)SelectStadium.getItemAt(0) ) {
    WarningS.setText("Please Select Stadium Name !");
}
else
    WarningS.setText("");
if(sname != (String)SelectStadium.getItemAt(0) && time ==
(String)SelectTime.getItemAt(0) && date ==(String)SelectDate.getItemAt(0) ) {
    new
Seats(username,sname,date,time,n,nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa14,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);
    jf.setVisible(false);
}
}
});
ShowBookings.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int c=1;
        String[] PastBookings=new String[10];
        String Incr=new String("  ALL BOOKINGS \n");
        try {
            fin = new FileInputStream("Bookings.dat");
            ois = new ObjectInputStream(fin);
            ArrayList<Bookings> al2;
            while((al2=(ArrayList)ois.readObject())!=null){

```

```

        Iterator<Bookings>i=a1.iterator();
        while(i.hasNext()){
            Bookings b1=i.next();
            if(username.equals(b1.getUsername1())){
PastBookings[c] ="-----
-----\n" + c + ". Tickets Booked on Day :
"+b1.getBookingdate()+"\t Time : "+b1.getBookingtime()+"\n  Stadium Name :
"+b1.getSname1()+"\t"+"Date :"+b1.getDate1()+"\t"+"Cost :"+b1.getCost1()+"\n"+"
Timmings" + b1.getTime1()+"\tNumber of seats :"+b1.getN1()+"\t"+"Seats :"+
+b1.getSeats1()+"\n";
            c++;
        }
    }
}
} catch(Exception er){ }
if(c>1) {
    int c1;
    for(c1=c-1;c1>0;c1--) {
        Incr +=PastBookings[c1];
        YourBookings.setText(Incr);
    }
}
if(c==1) {
    YourBookings.setText("No Bookings Till Now !");
}
}
});
LastBooking.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int c=1;
        String lastbooking;

```

```

try {
    fin = new FileInputStream("Bookings.dat");
    ois = new ObjectInputStream(fin);
    ArrayList<Bookings> al3;
    while((al3=(ArrayList)ois.readObject())!=null){
        Iterator<Bookings>i=al3.iterator();
        while(i.hasNext()){
            Bookings b2=i.next();
            if(username.equals(b2.getUsername1())){
                lastbooking ="
LAST BOOKING \n"+"-----
-----\n"+ c + ". Tickets Booked on Day :
"+b2.getBookingdate()+"\t Time : "+b2.getBookingtime()+"\n  Stadium Name :
"+b2.getSname1()+"\t"+"Date : " +b2.getDate1()+"\t"+"Cost : " +b2.getCost1()+"\n"+"
Timmings" + b2.getTime1()+"\tNumber of seats : " +b2.getN1()+"\t"+"Seats : "
+b2.getSeats1()+"\n";
                YourBookings.setText(lastbooking);
                c++;
            }
        }
    }
} catch(Exception er){
    if(c==1) {
        YourBookings.setText(" No Bookings Till Now !");
    }
}
j1.setVisible(true);
}
}

```

- **Fourth Frame:**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Seats extends JFrame {
    public String
    a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a20,a21,a22,a23,a24;
    public String
    b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17,b18,b19,b20,b21,b22,b23,b24,b2;
    public String
    b26,b27,b28,b29,b30,b31,b32,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16,c17,c18,c
    19,c20,c21,c22,c23,c24,c25,c26,c27,c28;
    public String sname,date,time,seats,username;
    public boolean
    nsa1=true,nsa2=true,nsa3=true,nsa4=true,nsa5=true,nsa6=true,nsa7=true,nsa8=true,nsa9=true,ns
    a10=true;
    public boolean
    nsa11=true,nsa12=true,nsa13=true,nsa14=true,nsa15=true,nsa16=true,nsa17=true,nsa18=true,nsa
    19=true,nsa20=true;
    public boolean nsa21=true,nsa22=true,nsa23=true,nsa24=true;
    public boolean
    nsb1=true,nsb2=true,nsb3=true,nsb4=true,nsb5=true,nsb6=true,nsb7=true,nsb8=true,nsb9=true,n
    sb10=true;
    public boolean
    nsb11=true,nsb12=true,nsb13=true,nsb14=true,nsb15=true,nsb16=true,nsb17=true,nsb18=true,ns
    b19=true,nsb20=true;
    public boolean
    nsb21=true,nsb22=true,nsb23=true,nsb24=true,nsb25=true,nsb26=true,nsb27=true,nsb28=true,ns
    b29=true,nsb30=true,nsb31=true,nsb32=true;
```

```

public boolean
nsc1=true,nsc2=true,nsc3=true,nsc4=true,nsc5=true,nsc6=true,nsc7=true,nsc8=true,nsc9=true,ns
c10=true;
public boolean
nsc11=true,nsc12=true,nsc13=true,nsc14=true,nsc15=true,nsc16=true,nsc17=true,nsc18=true,nsc
19=true,nsc20=true;
public boolean
nsc21=true,nsc22=true,nsc23=true,nsc24=true,nsc25=true,nsc26=true,nsc27=true,nsc28=true;
public int n,upperclass=1500,middleclass=1000,lowerclass=500,seatcost=0,cost=0;
public static void main(String[] args) {
new Seats();
}
Seats(){
public void Values(){
JFramejf=new JFrame("Stadium seat booking");
jf.setSize(770,750);
jf.getContentPane().setBackground(SystemColor.inactiveCaptionBorder);
jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
jf.getContentPane().setLayout(null);
JLabel Amount = new JLabel("Amount :");
Amount.setFont(new Font("Tw Cen MT", Font.BOLD, 18));
Amount.setBounds(395, 595, 84, 30);
jf.getContentPane().add(Amount);
JLabel amount = new JLabel("");
amount.setFont(new Font("Tw Cen MT", Font.BOLD, 18));
amount.setBounds(497, 590, 84, 30);
jf.getContentPane().add(amount);
JLabeldisplayseats = new JLabel("");
displayseats.setFont(new Font("Tw Cen MT", Font.BOLD, 18));
displayseats.setBounds(181, 590, 229, 30);
jf.getContentPane().add(displayseats);

```



```

JButtonDisplaySeats = new JButton("Select Seats :");
DisplaySeats.setFont(new Font("Tw Cen MT", Font.BOLD, 18));
DisplaySeats.setBounds(24, 590, 156, 30);
jf.getContentPane().add(DisplaySeats);
JLabel displayseats1 = new JLabel("");
displayseats1.setForeground(Color.RED);
displayseats1.setFont(new Font("Tw Cen MT", Font.BOLD, 18));
displayseats1.setBounds(181, 590, 229, 30);
jf.getContentPane().add(displayseats1);
DisplaySeats.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int i=0;
        if(Seata1.isSelected())
            i++;
        if(Seata2.isSelected())
            i++;
        if(Seata3.isSelected())
            i++;

                //same for all the seats!

                if(i==n) {
                    if(upperclass == 1500) {
                        if(Seata1.isSelected()) {
                            a1="A1 ";
                            seatcost+=upperclass;
                            Seata1.setEnabled(false);
                        }
                        else
                            a1="";
                        if(Seata2.isSelected()) {
                            a2="A2 ";

```

```
seatcost+=upperclass;
Seata2.setEnabled(false);
}
```

```
if( middleclass == 1000) {
if(Seatb1.isSelected()) {
b1="B1 ";
seatcost+=middleclass;
Seatb1.setEnabled(false);
}
else
b1="";
if(Seatb2.isSelected()) {
b2="B2 ";
seatcost+=middleclass;
Seatb2.setEnabled(false);
}
else
b2="";
if(Seatb3.isSelected()) {
b3="B3 ";
Seatb3.setEnabled(false);
seatcost+=middleclass;
}
else
b3="";
if(Seatb4.isSelected()) {
b4="B4 ";
seatcost+=middleclass;
Seatb4.setEnabled(false);
}
```

```

        if(lowerclass == 500) {
            if(Seatc1.isSelected()) {
                c1="C1 ";
                seatcost+=lowerclass;
                Seatc1.setEnabled(false);
            }
            else
                c1="";
            if(Seatc2.isSelected()) {
                c2="C2 ";
                seatcost+=lowerclass;
                Seatc2.setEnabled(false);
            }
            else
                c2="";
            if(Seatc3.isSelected()) {
                c3="C3 ";
                Seatc3.setEnabled(false);
                seatcost+=lowerclass;
            }
            else
                c3="";
            displayseats1.setText("");
            seats=a1+a2+a3+a4+a5+a6+a7+a8+a9+a10+a11+a12+a13+a14+a15+a16+a17+a18+a19+a20+a
                21+a22+a23+a24+b1+b2+b3+b4+b5+b6+b7+b8+b9+b10+b11+b12+b13+b14+b15+b16+b17+b
                18+b19+b20+b21+b22+b23+b24+b25+b26+b27+b28+b29+b30+b31+b32+c1+c2+c3+c4+c5+c6
                +c7+c8+c9+c10+c11+c12+c13+c14+c15+c16+c17+c18+c19+c20+c21+c22+c23+c24+c25+c26
                +c27+c28;
            displayseats.setText(seats);
            amount.setText(Integer.toString(seatcost));
            cost=seatcost;

```

```

seatcost=0;

}
if(i<n) {
OptionPane.showMessageDialog(jf, "Selected Less number of seats "+i+"<"+"n);
}
if(i>n) {
OptionPane.showMessageDialog(jf, "Selected More number of seats "+i+">"+"n);
}
}
});
JButton Back = new JButton("BACK");
Back.setFont(new Font("Tw Cen MT", Font.BOLD, 20));
Back.setBounds(151, 636, 100, 30);
jf.getContentPane().add(Back);
Back.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
new
MainFunction(username,nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13
,nsa14,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb
5,nsb6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,n
sb21,nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc
4,nsc5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc2
0,nsc21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);
jf.setVisible(false);
}
});
JButtonProcced = new JButton("PROCCED");
Procced.setFont(new Font("Tw Cen MT", Font.BOLD, 20));
Procced.setBounds(395, 636, 156, 30);
jf.getContentPane().add(Procced);

```

```

Procced.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
if(seats != null) {
if(Seata1.isSelected())
nsa1=false;

if(Seata2.isSelected())
nsa2=false;

if(Seata3.isSelected())
nsa3=false;
//Similar for all!
// c series lower class

if(Seatc1.isSelected())
nsc1=false;

if(Seatc2.isSelected())
nsc2=false;
new
Screen5(username,sname,date,time,n,cost,seats,nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa14,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);
jf.setVisible(false);
}
else
displayseats1.setText("Please Select Seats !");
}
}

```

```
});
```

```
jf.setVisible(true);
```

```
}
```

```
}
```

- **Fifth Frame:**

```
import javax.swing.*;
```

```
import java.io.*;
```

```
import java.text.DateFormat;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.time.*;
```

```
import java.time.format.*;
```

```
import java.text.*;
```

```
import java.util.*;
```

```
class Bookings implements Serializable {
```

```
String username1,sname1,date1,time1,seats1,bookingdate,bookingtime;
```

```
int n1,cost1;
```

```
Bookings(String username1,String sname1,String date1,String time1,int n1,int cost1,String  
seats1,String bookingdate,Stringbookingtime) {
```

```
this.username1 = username1;
```

```
this.sname1 = sname1;
```

```
this.date1 = date1;
```

```

this.time1= time1;

this.n1=n1;

this.cost1=cost1;

this.seats1=seats1;

this.bookingdate=bookingdate;

this.bookingtime=bookingtime;

}

public class Screen5 extends JFrame{

public String name,email,sname,timing,date,time,seats,username;

public int n,cost;

public boolean
nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa14,nsa15,nsa16,nsa1
7,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb6,nsb7,nsb8,nsb9,ns
b10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,nsb22,nsb23,nsb24,
nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc5,nsc6,nsc7,nsc8,ns
c9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc21,nsc22,nsc23,nsc
24,nsc25,nsc26,nsc27,nsc28;

FileOutputStreamfout;

ObjectOutputStreammoos;

FileInputStream fin;

ObjectInputStreammois;

public static void main(String[] args) {

new Screen5();

}

```

```

Screen5(){
Values();
}

public void Values()
{
ArrayList<Bookings>alb=new ArrayList<Bookings>();

JFramejf=new JFrame("Stadium Seat Booking");

jf.setSize(903,665);

jf.getContentPane().setBackground(SystemColor.inactiveCaptionBorder);

jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

jf.getContentPane().setLayout(null);

JLabellblNewLabel = new JLabel("SUCCESSFULLY BOOKED !");

lblNewLabel.setFont(new Font("Tw Cen MT", Font.BOLD | Font.ITALIC, 32));

lblNewLabel.setBounds(278, 40, 428, 53);

jf.getContentPane().add(lblNewLabel);

        Logout.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int input =JOptionPane.showConfirmDialog(jf, "Do you want Logout ?");

if(input==0) {

new

Screen1MainPage(nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa1
4,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb
6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,
nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc

```



```

5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc
21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);

jf.setVisible(false);

JOptionPane.showMessageDialog(jf,"Logout Success !");

}

}

});

LocalDate localdate = LocalDate.now();

String bookingdate =
localdate.format(DateTimeFormatter.ofLocalizedDate(FormatStyle.FULL));

DateFormat localtime = new SimpleDateFormat("hh:mm:ss aa");

String bookingtime = localtime.format(new Date()).toString();

JOptionPane.showMessageDialog(jf,"Successfully Booked !");

File file1 = new File("Bookings.txt");

try {

if(!file1.exists()){

file1.createNewFile();

}

FileWriter fw = new FileWriter(file1,true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n\n");

bw.write("=====
=====");

```

```

bw.write("\n");

bw.write("Tickets Booked on Day : "+bookingdate + "    Time : "+bookingtime);

bw.write("\n");

alb.add(new Bookings(u,s,d,t,n_1,c,s1,bd,bt));

try {

File file = new File("Bookings.dat");

if(file.exists()) {

oos = new ObjectOutputStream(new FileOutputStream(file,true))
    protected void writeStreamHeader() throws IOException {

reset();

}

};}

else

oos = new ObjectOutputStream(new FileOutputStream(file));

oos.writeObject(alb);

alb.clear();

oos.close();

} catch (Exception ae){

ae.printStackTrace();

}

mainmenu.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

```

new

```
MainFunction(username,nsa1,nsa2,nsa3,nsa4,nsa5,nsa6,nsa7,nsa8,nsa9,nsa10,nsa11,nsa12,nsa13,nsa14,nsa15,nsa16,nsa17,nsa18,nsa19,nsa20,nsa21,nsa22,nsa23,nsa24,nsb1,nsb2,nsb3,nsb4,nsb5,nsb6,nsb7,nsb8,nsb9,nsb10,nsb11,nsb12,nsb13,nsb14,nsb15,nsb16,nsb17,nsb18,nsb19,nsb20,nsb21,nsb22,nsb23,nsb24,nsb25,nsb26,nsb27,nsb28,nsb29,nsb30,nsb31,nsb32,nsc1,nsc2,nsc3,nsc4,nsc5,nsc6,nsc7,nsc8,nsc9,nsc10,nsc11,nsc12,nsc13,nsc14,nsc15,nsc16,nsc17,nsc18,nsc19,nsc20,nsc21,nsc22,nsc23,nsc24,nsc25,nsc26,nsc27,nsc28);
```

```
jf.setVisible(false);
```

```
}
```

```
});
```

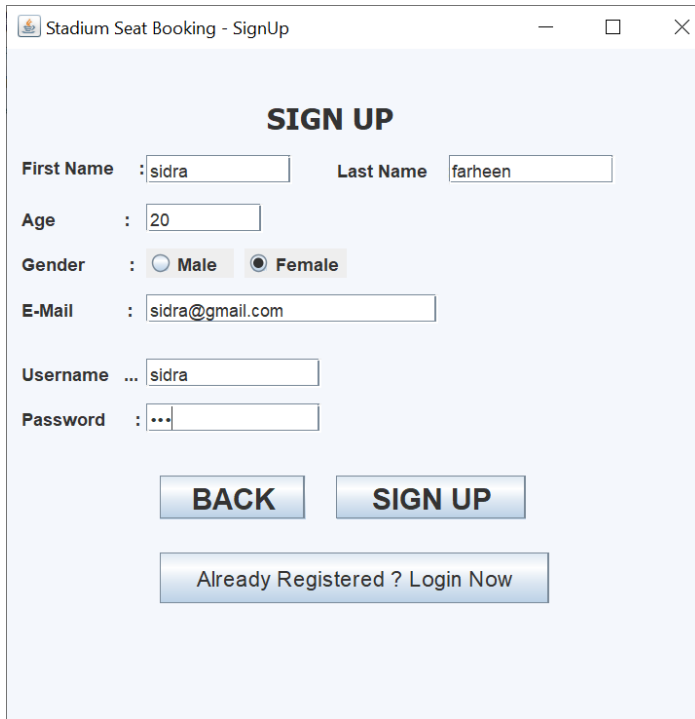
```
jf.setVisible(true);
```

```
}
```

```
}
```

5.RESULTS SCREENS:

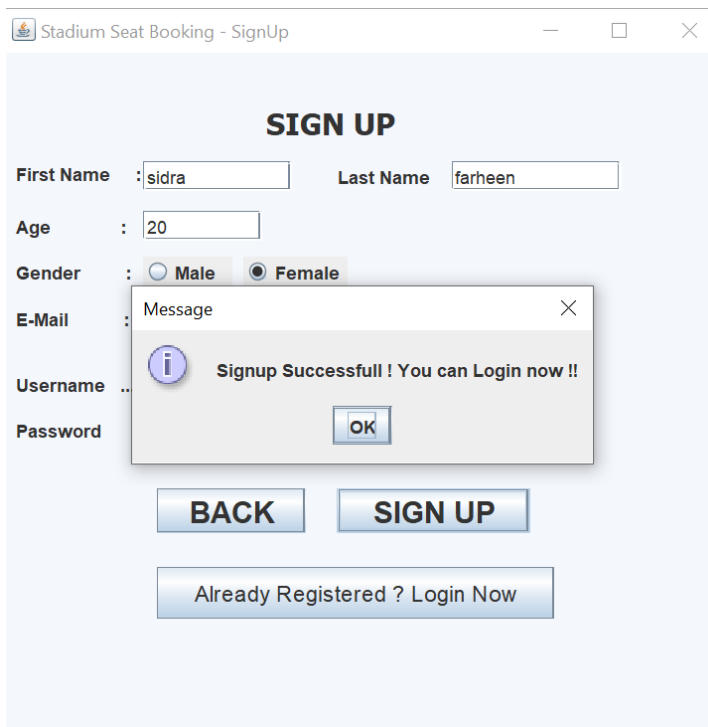
- If a customer doesn't have an account he/she needs to register:



The screenshot shows a web browser window titled "Stadium Seat Booking - SignUp". The page has a light blue background and a white form area. The form is titled "SIGN UP" in bold. It contains the following fields and controls:

- First Name**: Input field with "sidra".
- Last Name**: Input field with "farheen".
- Age**: Input field with "20".
- Gender**: Radio buttons for "Male" (unselected) and "Female" (selected).
- E-Mail**: Input field with "sidra@gmail.com".
- Username**: Input field with "sidra".
- Password**: Input field with masked characters "...".

Below the form are three buttons: "BACK", "SIGN UP", and "Already Registered ? Login Now".



This screenshot shows the same "Stadium Seat Booking - SignUp" window as the previous one, but with a success message dialog box overlaid. The dialog box is titled "Message" and contains the text "Signup Successfull ! You can Login now !!". It has an "OK" button. The form fields and buttons are still visible in the background.

- After Registering customer needs to login:

Stadium Seat Booking

STADIUM SEAT BOOKING SYSTEM

LOGIN

USERNAME:

PASSWORD:

☒ show password

LOGIN

If you dont have account?

Signup

Stadium Seat Booking

STADIUM SEAT BOOKING SYSTEM

LOGIN

USERNAME:

PASSWORD:

☐ show password

Message

Login Successfull !

OK

Signup

- Screen for customer to select Stadium name, time, date and number of seats:

The screenshot shows a web application window titled "Stadium seat booking". The header includes a welcome message "Wel-Com..sidra" and a "Log Out" button. The main form contains the following fields:

- Stadium Name:** A dropdown menu with "Rajiv Gandhi International Cricket Stadium, Hydera..." selected.
- Choose Date:** A date picker showing "28-Mar-2020".
- Choose Time:** A time range picker showing "11 : 00 AM to 11 : 00 PM".
- No. of Seats:** A numeric input field with the value "3".

Below the form, there is a large "Select Seats" button. At the bottom, there are two smaller buttons: "Show Last Boo..." and "Show All Booki...".

- Screen for selecting seats according to customers convenience:

The screenshot shows the seat selection interface. At the top, it displays the selected stadium, date, time, and number of seats: "Stadium Na... Rajiv Gandhi International Cricket Stadium, Hvd...", "Dat... 28-Mar-2020", "Timminas ... 11 : 00 AM to 11 : 00 PM", and "Number Of Se... 3".

The main area is a grid of seat selection buttons, each with a checkbox and a label (e.g., C1, B1, A1, etc.). The seats are arranged in a grid that is roughly 10 columns wide and 10 rows high. Some seats are already selected, indicated by checked checkboxes: A2, A3, and A4.

At the bottom, there is a "Select Seats..." button. To its right, the selected seats are listed: "A2 A3 A4". Further right, the total amount is displayed: "Amount #500". On the far right, there is a legend for the seat classes: "A1 - A24 : UPPERCLASS | 15..." and "B1 - B32 : MIDDLECLASS | 10...".

- Final screen displaying all customer details:

Stadium Seat Booking

Welcome sidra Log Out

SUCCESSFULLY BOOKED...

Name: sidra

Stadium Name: Rajiv Gandhi International Cricket Stadium, Hyderabad

Timing: 11 : 00 AM to 11 : 00 PM

Date: 28-Mar-2020

No Of Seats: 3 Cost: 4500

Seat Numbers: A2 A3 A4

THANK YOU :-)

Book More Tickets

- If same customer wants to book seats again:

Stadium seat booking

Welcome..sidra Log Out

Stadium Name: Rajiv Gandhi International Cricket Stadium, Hyderabad

Choose Date: 28-Mar-2020

Choose Time: 11 : 00 AM to 11 : 00 PM

No. of Seats: 2

Select Seats

Show Last Bookings Show All Bookings

LAST BOOKING

1. Tickets Booked on Day : Saturday, April 4, 2020 Time : 07:38:58 PM
 Stadium Name : Rajiv Gandhi International Cricket Stadium, Hyderabad. Date :28-Mar-2020 Cost :4500
 Timings:11 : 00 AM to 11 : 00 PM Number of seats :3 Seats :A2 A3 A4

Stadium seat booking

Stadium Na... Raiiv Gandhi International Cricket Stadium, Hyd...

Dat... 28-Mar-2020

Timminas ... 11 : 00 AM to 11 : 00 PM

Number Of Se... 2

☐ C1
☐ C2
☐ C3
☐ C4
☐ C5
☐ C6

☐ B1
☐ B2
☐ B3
☐ B4
☒ B5
☒ B6
☐ B7
☐ B8

☐ C28
☐ B32
☐ A1
☐ A5
☐ A6
☐ B9
☐ C7

☐ C27
☐ B31
☐ A24
☐ A7
☐ B10
☐ C8

☐ C26
☐ B30
☐ A23
☐ A8
☐ B11
☐ C9

☐ C25
☐ B29
☐ A22
☐ A9
☐ B12
☐ C10

☐ C24
☐ B28
☐ A21
☐ A10
☐ B13
☐ C11

☐ C23
☐ B27
☐ A20
☐ A11
☐ B14
☐ C12

☐ C22
☐ B26
☐ A19
☐ A12
☐ B15
☐ C13

☐ C21
☐ B25
☐ A18
☐ A17
☐ A16
☐ A15
☐ A14
☐ A13
☐ B16
☐ C14

☐ B24
☐ B23
☐ B22
☐ B21
☐ B20
☐ B19
☐ B18
☐ B17

☐ C20
☐ C19
☐ C18
☐ C17
☐ C16
☐ C15

Select Seats...

B5 B6

Amount 2000

A1 - A24 : UPPERCLASS | 15...
B1 - B32 : MIDDLECLASS | 10...

Stadium Seat Booking

Wel-Come sidra

Log Out

SUCCESSFULLY BOO...

Name

sidra

Stadium N...

Raiiv Gandhi International Cricket Stadium, Hv...

Timing

11 : 00 AM to 11 : 00 PM

Date

28-Mar-2020

No Of Seats

2

Cost

2000

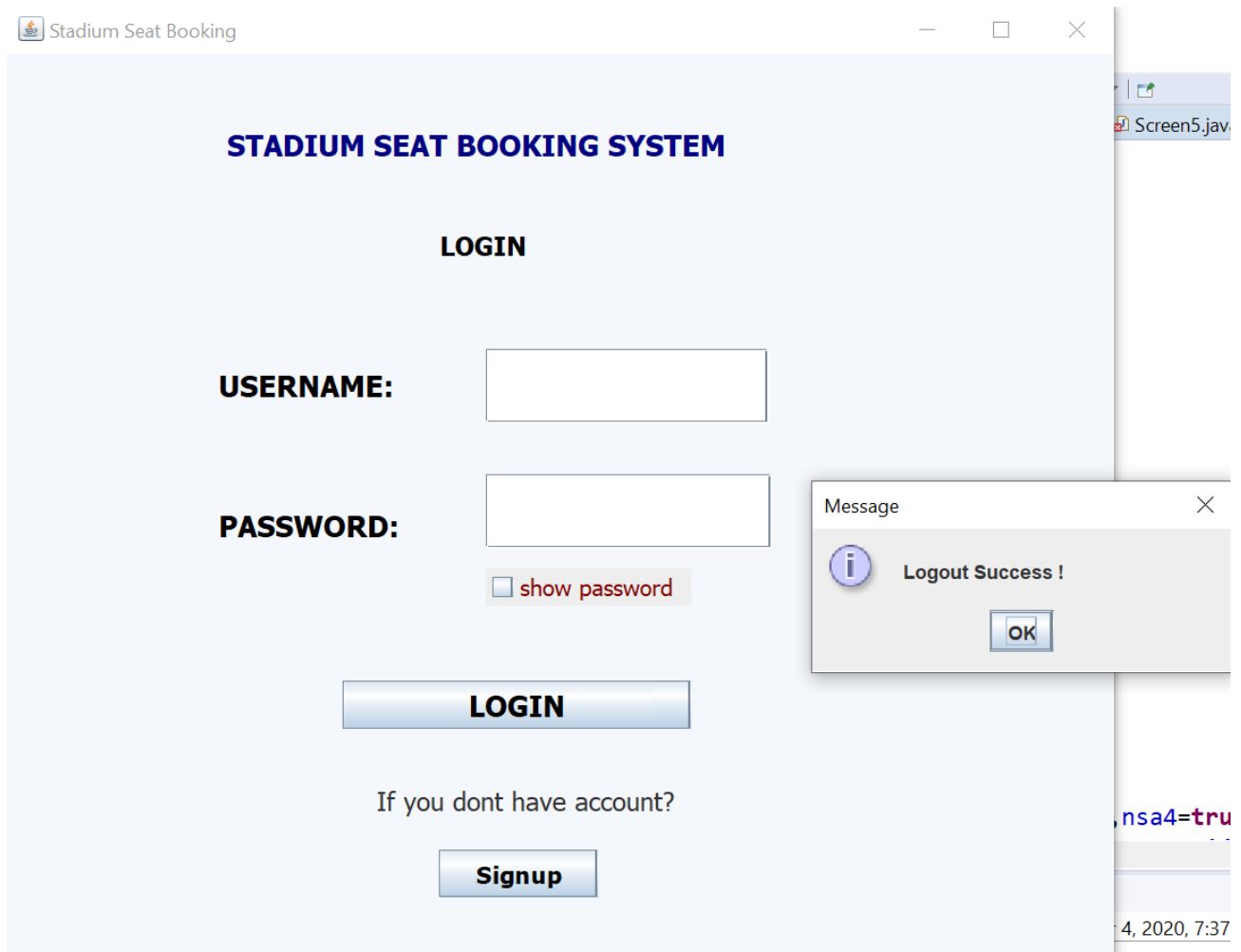
Seat Numb...

B5 B6

THANK YOU :-)

Book More Tic...

- After booking seats customer needs to logout using 5th or 3rd screen logout button :



CONCLUSION:

This makes our project clear to understand because of user friendly frames which we have implemented. Here data is also safe as it is stored in files. This project contains lot of scope of development. Never the less its overall look and feel provides rough image of an existing solution.