



INFORMATICS PRACTICES PROJECT REPORT ON

NBA TICKET AND AIR RESERVATION SYSTEM



Done by :

Name : Chirag Radhakrishna

Class : XII

School Name : National Hill View Public School, RR Nagar

Year : 2018-2019

TABLE OF CONTENTS

Sl.no	Topic	Pg.no
1.	About E-Business & the NBA	1-5
2.	About Java, Netbeans and MySQL	6-8
3.	Introduction about the project	9
4.	Netbeans components used	10-11
5.	Flow Chart of the program	12-20
6.	MySQL Table Structure and data	21-23
7.	Output Screen Shot with Source Code	24-77
8.	Bibliography	78



E-BUSINESS

As our project is a seat cum destination booking system which allows booking of match and flight tickets online, it can be classified under the category of Electronic business.

We all know that ELECTRONIC COMMERCE describes the process of buying, selling, transferring, or exchanging products, services/information via computer networks including the Internet. **E-business** refers to a broader definition of EC, not just the buying and selling of goods and services, but also servicing customers, collaborating with business partners, conducting e-learning, & conducting electronic transactions within an organization.

Hence we can sum up E-BUSINESS as:

"E-business refers to any form of transaction (exchange) that uses an electronic medium to facilitate the transaction."



➤ HISTORY

In 1994, IBM, with its agency Ogilvy & Mather, began to use its foundation in IT solutions and expertise to market itself as a leader of conducting business on the Internet through the term "e-business." Then CEO Louis V. Gerstner, Jr. was prepared to invest \$1 billion to market this new brand.

After conducting worldwide market research in October 1997, IBM began with an eight-page piece in the Wall Street Journal that would introduce the concept

of "e-business" and advertise IBM's expertise in the new field. IBM decided not to trademark the term "e-business" in the hopes that other companies would use the term and create an entire new industry. However, this proved to be too successful and by 2000, to differentiate it, IBM launched a \$300 million campaign about its "e-business infrastructure" capabilities. Since that time, the terms, "e-business" and "e-commerce" have been loosely interchangeable and have become a part of the common vernacular.

➤ BUSINESS MODEL: (Usage)

When organizations go online, they have to decide which e-business models best suit their goals. A business model is defined as the organization of product, service and information flows, and the source of revenues and benefits for suppliers and customers. The concept of e-business model is the same but used in the online presence. It can also be done on value-added networks (private networks).

➤ MAJOR E-BUSINESS PORTALS:



ADVANTAGES OF E-BUSINESS:

1.

- Offers new opportunities to access markets across the globe.

2.

- Provides chance to target market segments more effectively.

3.

- Improved efficiency and productivity.

4.

- Reduces inventory.

5.

- Improves communications, information and knowledge sharing.

6.

- Improves speed of response.



THE NBA IN A NUTSHELL

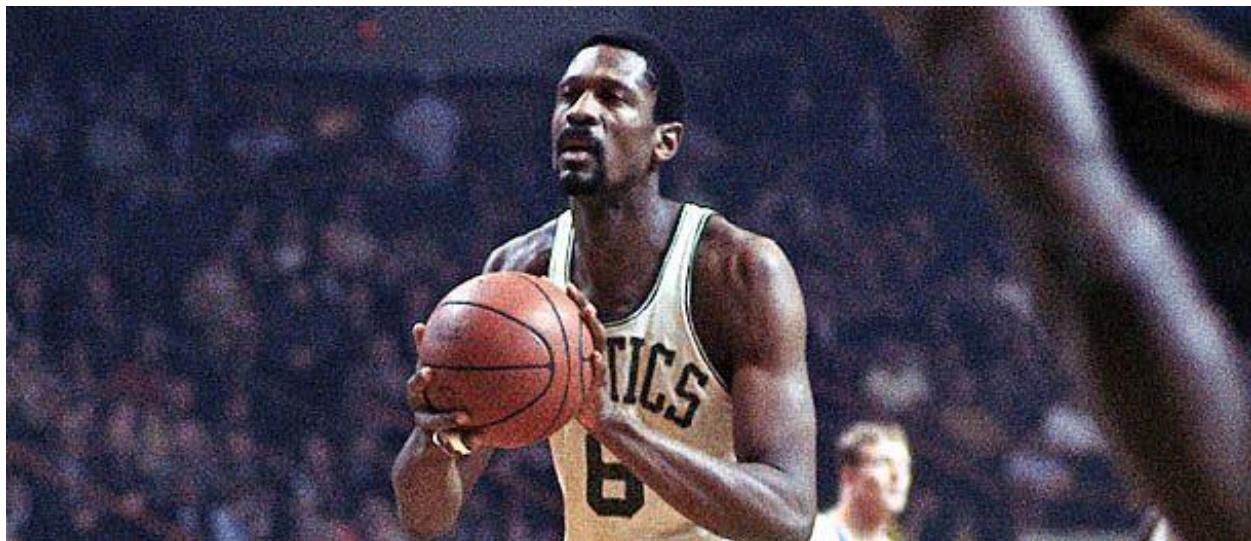
The National Basketball Association (NBA) is a men's professional basketball league in North America; composed of 30 teams (29 in the United States and 1 in Canada). It is widely considered to be the premier men's professional basketball league in the world. The NBA is an active member of USA Basketball (USAB), which is recognized by FIBA (also known as the International Basketball Federation) as the national governing body for basketball in the United States.

The NBA is one of the four major professional sports leagues in the United States and Canada. NBA players are the world's best paid athletes by average annual salary per player.

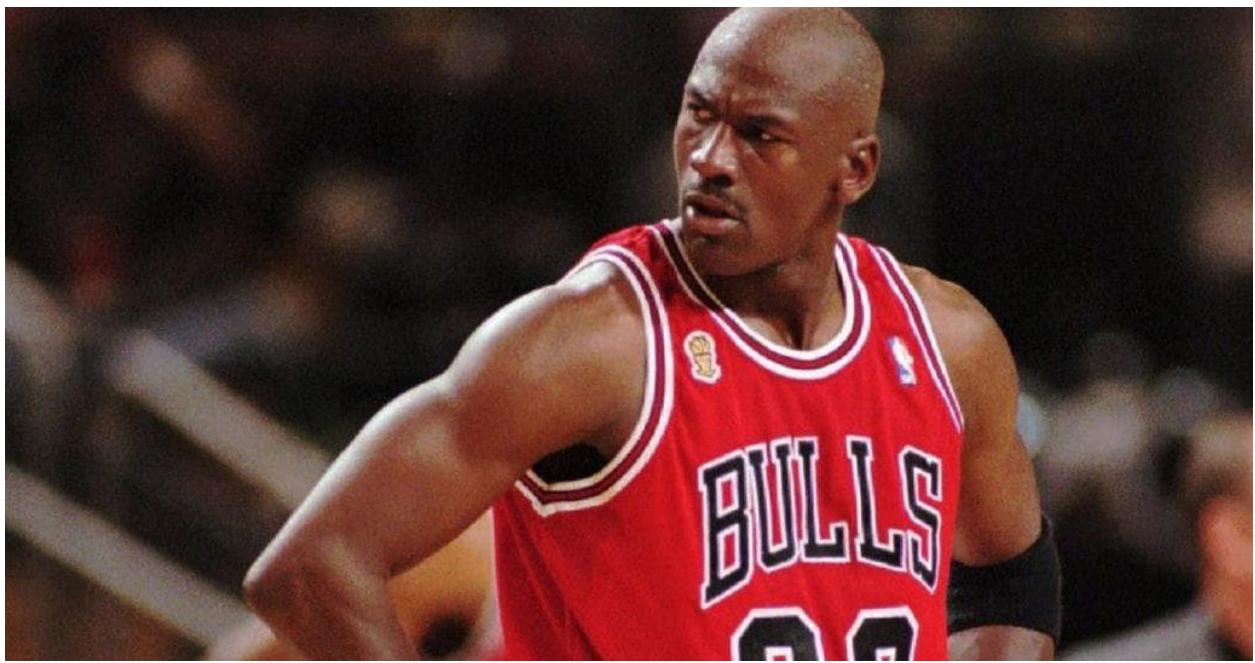
The NBA is the most popular basketball league in the world and has a record viewership. It also has several academies across the globe with the aim to promote the game and develop talents at the grassroots levels. NBA players like LeBron James, Kobe Bryant, Stephen Curry and Kevin Durant are household names.

The NBA has seen many great players rule the court for the past half a century. These players are an inspiration to viewers and young ballers across the globe.

[**Bill Russell, the most decorated player of all time with a record 11 NBA championships and rings.**](#)



Michael Jordan, considered by many the ‘greatest player of all time’. Jordan won 6 NBA titles with the Chicago Bulls.



The Golden State Warriors, the team with the most wins in a regular season with 73, an NBA Franchise record.



ABOUT JAVA, NETBEANS IDE AND MYSQL

JAVA PROGRAMMING LANGUAGE:

Java was originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Java is a general-purpose computer programming language that is object-oriented and concurrent, and specifically designed to have few dependencies as possible. Java has the “Write Once Run Anywhere (WORA)” characteristic. This implies that java is platform independent and can run on any platform provided there is a java compiler and a few native libraries ported. Java applications are typically compiled to a byte code that can run on any Java virtual machine (JVM) regardless of the computer architecture. In the present world, it is the most popular programming language.



NETBEANS IDE:

NetBeans is an open-source project dedicated to providing rock solid software development products (the NetBeans IDE and the NetBeans Platform) that address the needs of developers, users and the businesses who rely on NetBeans as a basis for their products; particularly, to enable them to develop these products quickly, efficiently and easily by leveraging the strengths of the Java platform and other relevant industry standards. In June 2000, NetBeans was made open source by Sun Microsystems, which remained the project sponsor until January 2010 when Sun Microsystems became a subsidiary of Oracle.

The two base products, the NetBeans IDE and NetBeans Platform, are free for commercial and non-commercial use. The source code to both is available to anyone to reuse as they see fit, within the terms of use.



MySQL:

MySQL is a freely available open source *Relational Database Management System* (RDBMS) that uses Structured Query Language (SQL). A single MySQL database can contain many tables at once and store thousands of individual records. MySQL provides you with a rich set of features that support a secure environment for storing, maintaining, and accessing data. MySQL is a fast, reliable, scalable alternative to many of the commercial RDBMS available today.

JDBC (Java Database Connectivity):

Helps to connect to a relational database (Ex: MySQL) from within a Java Application. We have done it using the JDBC API of Java.

It is part of the Java Standard Edition platform, from Oracle Corporation.

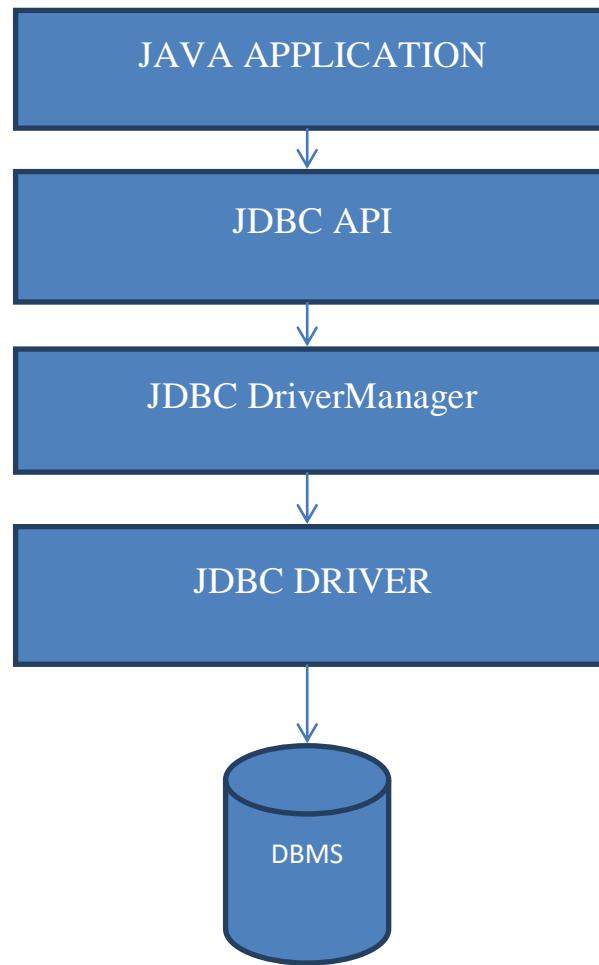
It mainly does the following tasks from within a Java application code:

1.
 - Establish connection with the database.
2.
 - Send SQL statements to the database server.
3.
 - Process the results obtained.

There are four main classes in the JDBC API hierarchy that are generally used for database connectivity. These are:

- ✓ **DriverManager Class:** The JDBC DriverManager class loads the JDBC driver needed to access a particular data source, locates and logs on to the database and returns a *Connection Object*.
- ✓ **Connection Class:** The JDBC Connection class manages the communication between a Java client application and a specific database, including passing SQL statements to the DBMS and managing transactions.
- ✓ **Statement Class:** The JDBC Statement Class contains SQL strings that are submitted to the DBMS. An SQL *Select statement* returns a *ResultSet object* that contains the data retrieved as the result of an SQL statement.
- ✓ **ResultSet Class:** The JDBC ResultSet class provides predefined methods to access analyze and convert data values returned by SQL *Select Statement*.

JDBC Architecture:



NBA SEATS AND AIR DESTINATION BOOKING SYSTEM: INTRODUCTION

THE NBA SEAT RESERVATION SYSTEM is a project designed to help fans and users book match tickets and destination tickets to watch the NBA All Star Weekend Charlotte 2019. The user can book a match ticket and is provided an option of booking a flight to the stadium destination-**Charlotte**.

THE project is a robust, user-friendly application, yet easy and simple to use. It is based on the concept of database connectivity i.e., connecting the java application to MySQL.

PROJECT PROCESS OVERVIEW:

- The user can choose any of the 4 teams and can learn about the players, the matches.
- The matches are part of a round –robin format wherein the teams face off against each other. Details of the matches are stored in MySQL table.
- If the user is interested in booking a seat for the All-Star Weekend, he/she can do so by selecting the desired seat through the interactive seat booking frame model.
- Once the match and the seats have been selected, the user is taken to a payment form, where he/she can pay via credit card or debit card by giving valid details.
- Once the formalities are finished, a final seat frame consisting of all the match and ticket details will be displayed automatically. The user then can either logout or book an NBA destination flight to the All Star destination through our “BOOK YOUR DESTINATION FLIGHT” option.
- If the user is interested he she can choose a suitable flight and book the ticket and seats. After these procedures, details of the user are stored in SQL tables, and the final frame consists of offers that the user is entitled to.
- All the information and details are stored in MySQL table and details from java application are erased for the next user.
- Our MySQL database has 4 tables. The ‘matches’ table displays details of the matches whereas the ‘flight’ table displays the list of flights available and journey details. Personal Information of the user along with the selected match and seats are stored in the ‘user’ table whereas flight details are stored in the ‘air’ table.

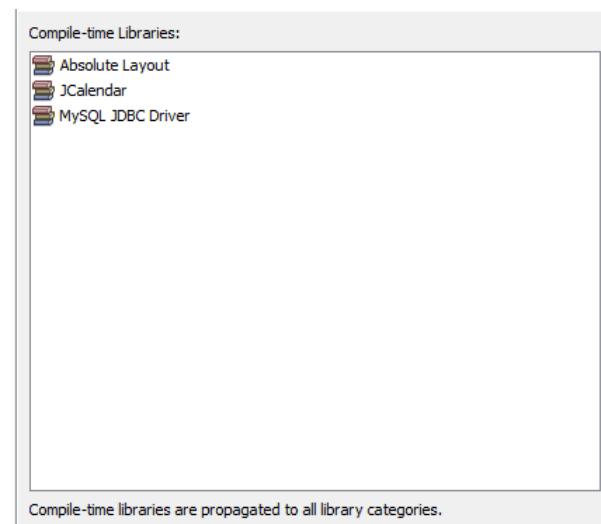
NETBEANS COMPONENTS USED

Front end application: User interface or that part of a software or a website that a user sees on the screen, and acts on to enter commands or to access other parts of the software or website.

The front-end tool in our project is the java application.

Libraries used in our project:

1. Absolute Layout
2. MySQL JDBC Driver
3. JCalendar



Containers used in our project are:

Top level containers:

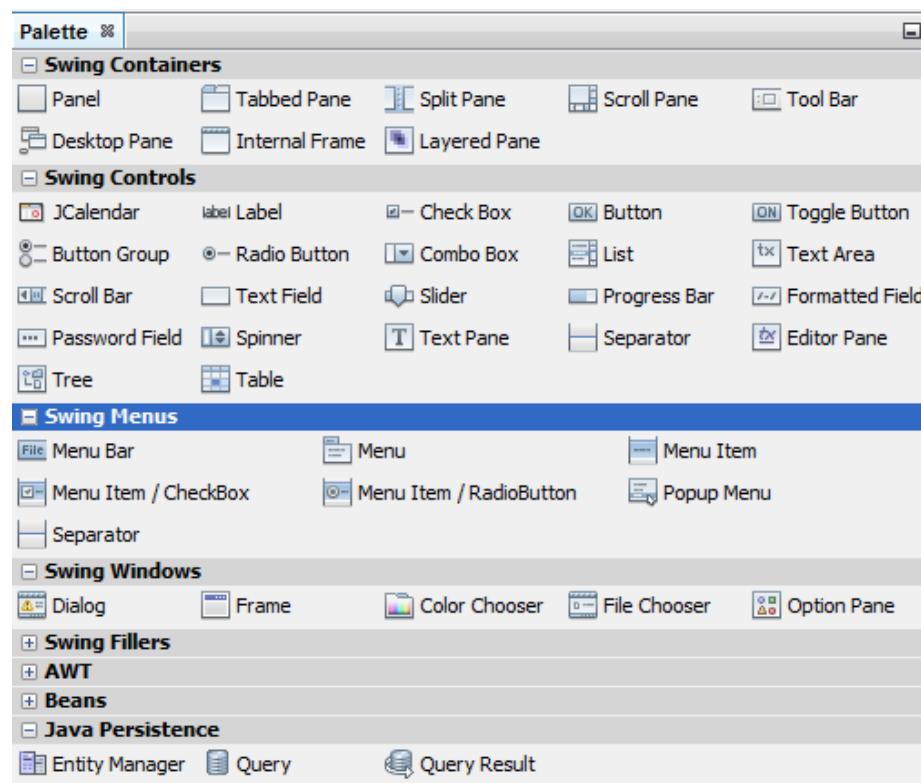
- ✓ JFrame
- ✓ JDialog

Medium Level Containers:

- ✓ JPanel
- ✓ JTable

Low Level Containers:

- ✓ JButton
- ✓ JCheckBox
- ✓ JComboBox
- ✓ JRadioButton
- ✓ JTextField
- ✓ JLabel
- ✓ JOptionPane
- ✓ JTextArea
- ✓ JButtonGroup
- ✓ JCalendar
- ✓ JPasswordField



FLOW OF CONTROL OF THE PROGRAM

The flow of our program is given as follows with brief explanation of all the frames.

NBA Frame: This frame consists of the NBA logo and guides the user to the teams frame.



Teams Frame: This frame displays the logos of the four teams which are taking part in the All-Star Weekend.



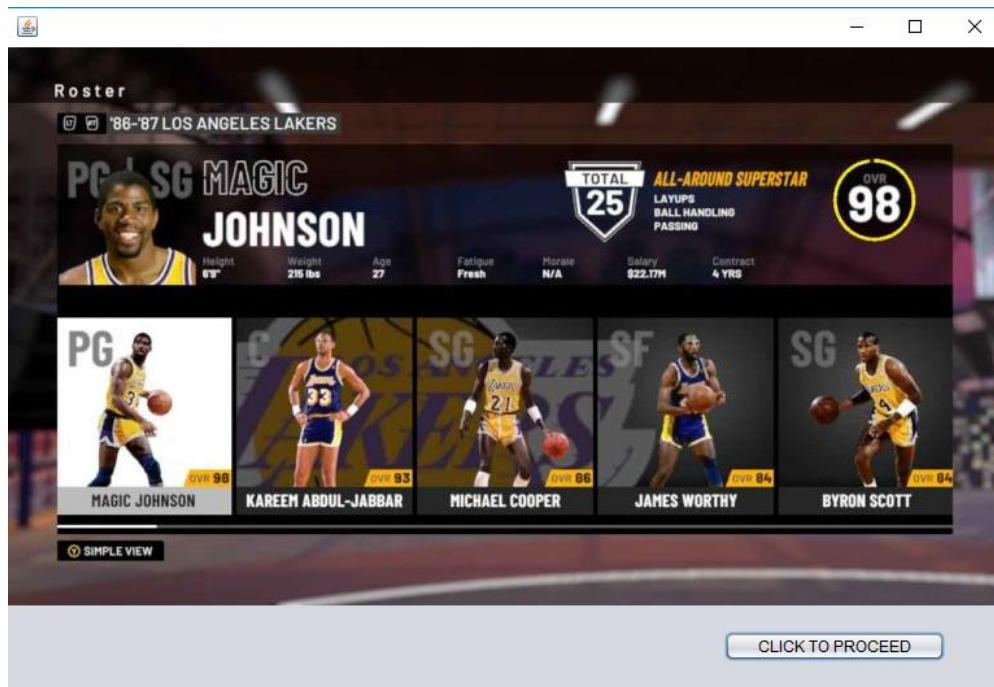
Lineups' Frame: These frames display the Starting-line ups of the teams.

// the starting line-up of the Golden State Warriors.



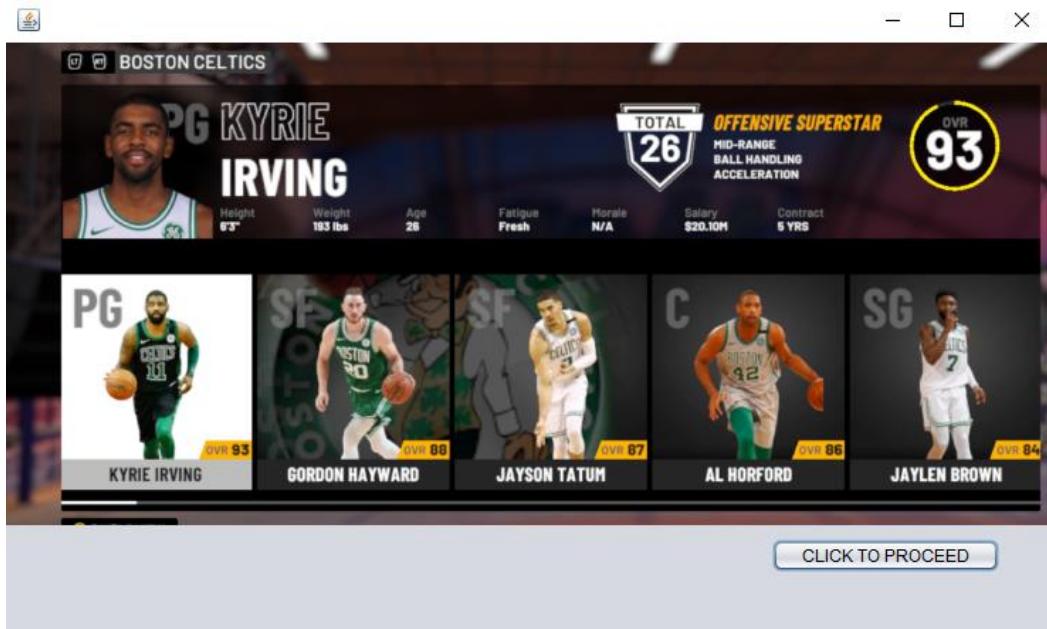
Or

// the starting line-up of the Los Angeles Lakers.



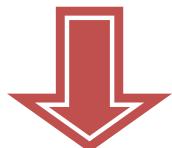
Or

// the starting line-up of the Boston Celtics.

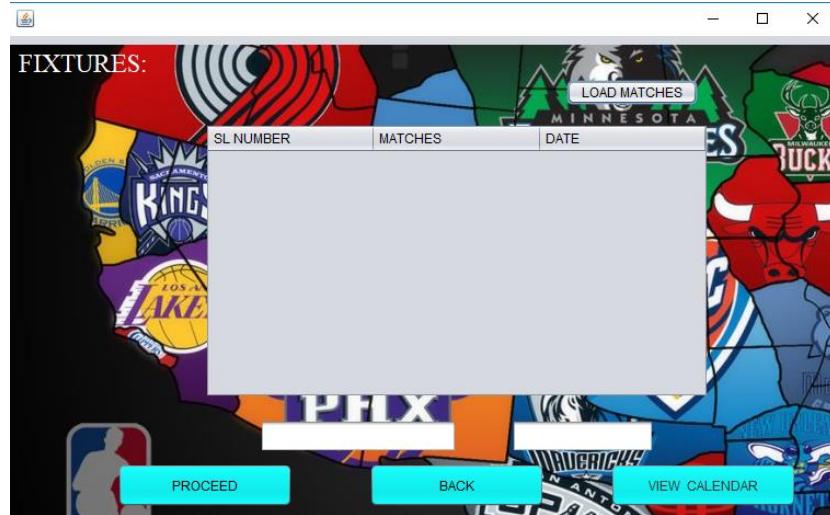


Or

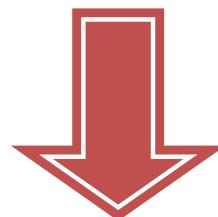
// the starting line-up of the Dallas Mavericks.



The fixtures frame displays the match details by loading the match content.



The seating frame is designed for the user to select the seats based on the price.



Ticket frame: Here the user enters his/her personal details and method of payment after choosing the match and seats.



The screenshot shows a 'TICKET CONFIRMATION' screen. On the left, there's a 'PERSONAL DETAILS' section with fields for First Name, Last Name, E-Mail ID, and Mobile Number. Below it is a 'PAYMENT OPTIONS' section with radio buttons for 'CREDIT CARD' (showing logos for VISA, MasterCard, American Express, and Discover) and 'DEBIT CARD' (with a note: '* Options Available for Debit Payment' and a dropdown menu showing 'American Express'). On the right, there's a 'LOAD MATCH INFORMATION' section with fields for No. of Seats Booked, Seat Numbers, Date of Game, Match, Stadium (set to 'Spectrum Center'), and Grand Total (In Dollars). At the bottom are buttons for 'PROCEED FOR FURTHER PAYMENT', 'BACK', and 'CLEAR'.

The user can then make payments using credit card or debit card.

The screenshot shows a 'PAYMENT THROUGH CREDIT CARD' screen. It has three input fields: 'ENTER CARD NAME:' with a placeholder ' ', 'ENTER CARD NUMBER:' with a placeholder ' ', and 'ENTER EXPIRY DATE:' with a placeholder ' '. At the bottom are buttons for 'CONFIRM PAYMENT', 'BACK', and 'CLEAR'. The background features a stylized basketball court floor.

Or



PAYMENT THROUGH DEBIT CARD

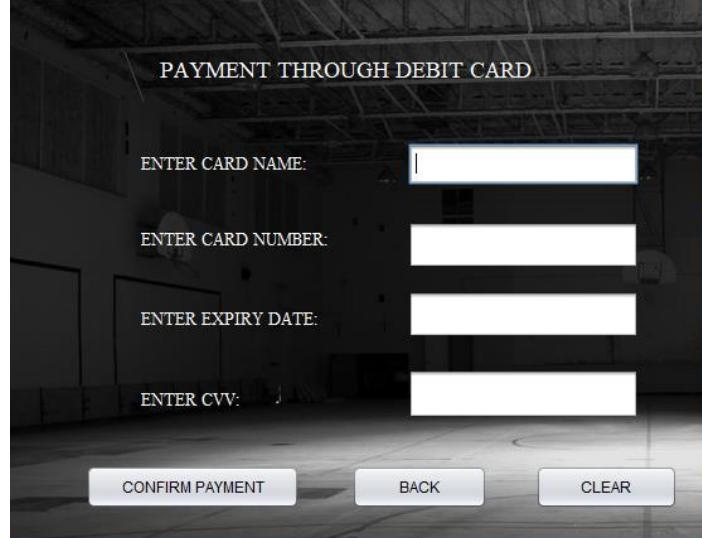
ENTER CARD NAME:

ENTER CARD NUMBER:

ENTER EXPIRY DATE:

ENTER CVV:

CONFIRM PAYMENT BACK CLEAR



The confirmation frame shows the final NBA ticket of the user. The user can then exit or book an NBA Destination. Details are saved and the ticket can be printed.

CLICK THE BACKGROUND

FINAL NBA TICKET :

NBA

NAME:

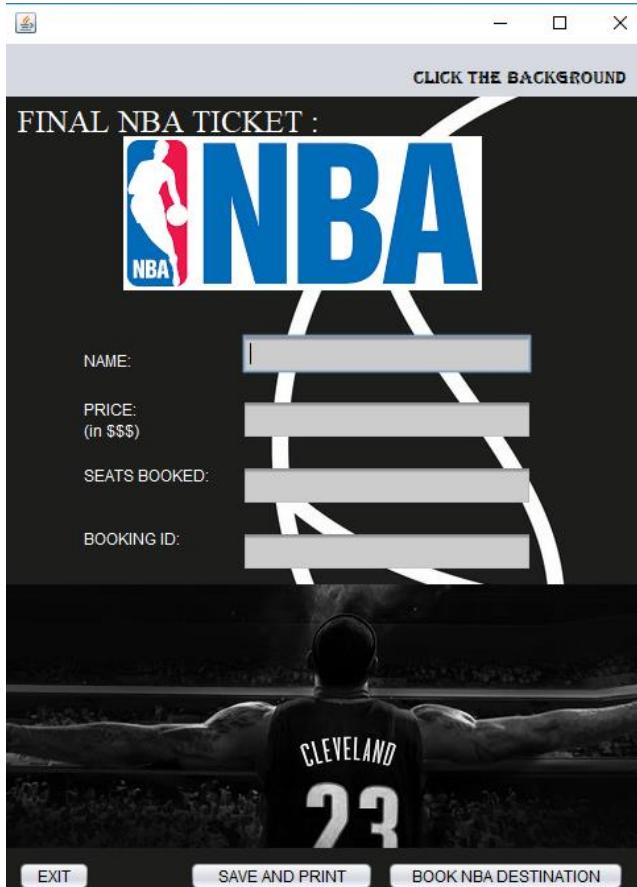
PRICE:
(in \$\$\$)

SEATS BOOKED:

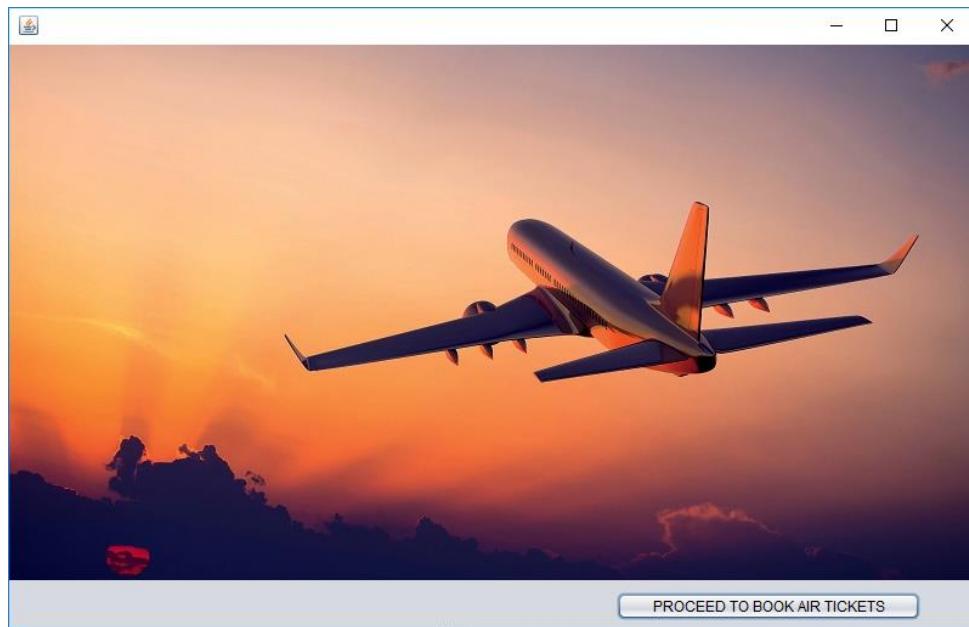
BOOKING ID:

CLEVELAND
23

EXIT SAVE AND PRINT BOOK NBA DESTINATION



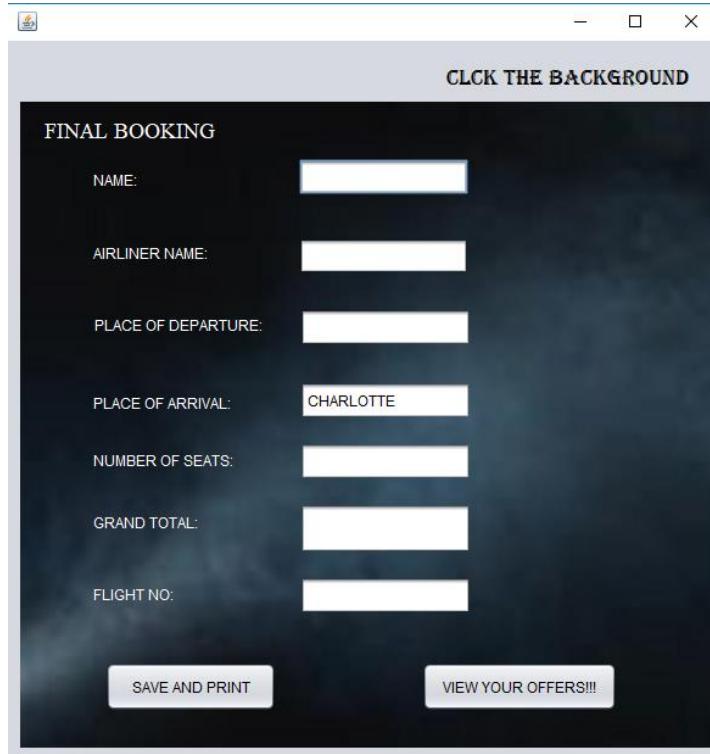
If the user decides to book an NBA destination ticket to Charlotte, an introductory frame is shown which guides them to book an air ticket.



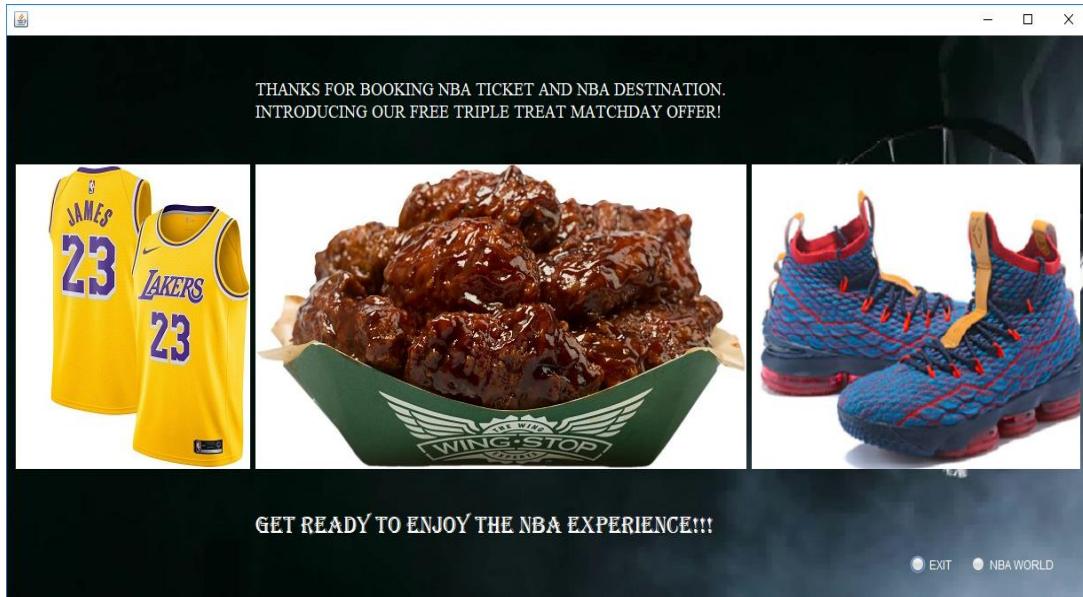
The air ticket reservation frame displays flight details and timings, and helps the user book the air ticket using NBA credits.

A screenshot of the "AIR TICKET RESERVATION" form. The background shows a large airplane on a runway at sunset. The form includes fields for "ENTER NBA CREDITS:", "Departure Airport" (with a dropdown menu), "Arrival Airport" (set to "CHARLOTTE"), a "SEARCH FLIGHTS" button, and a table for flight details with columns: FLIGHT NO, AIRLINES, DEPARTURE, ARRIVAL, DEPARTURE TIME, ARRIVAL TIME, DURATION, and SEATS AVAILABLE. Below the table are fields for "Airlines Name:", "Place of departure:", "Enter number of seats:", "Flight NO.", and "CONFIRM BOOKING THROUGH POINTS". At the bottom are buttons for "CALCULATE PRICE(IN DOLLARS)" and "CONFIRM BOOKING THROUGH POINTS".

The flight confirmatory frame confirms flight details of the user and guides them to the offers frame. The details are saved and the ticket can be printed.



The offers page shows the free NBA merchandise and snacks the user is entitled to receive on Match day!!! The user can then either exit or visit NBA world.



Ending frame: This frame marks the end of our project. This frame helps the user access the official website of the NBA, the NBA store and the league's YouTube channel.



// on clicking the Close button, the application terminates.

PROGRAM TERMINATION

MYSQL TABLE STRUCTURES & DATA

Back End Tool: In our project, the back end tool is the MySQL database. Here the database is accessed indirectly through an external application (Here Java).

Explanation of elements used in the back end tool:

Database Name: airline

Table Structures:

- 1) Flight Table: This table contains details regarding flights.

```
mysql> desc flight;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| flight_no | varchar(30) | NO   | PRI  | NULL    |       |
| airliner_name | varchar(30) | YES  |       | NULL    |       |
| departure | varchar(30) | YES  |       | NULL    |       |
| arrival  | varchar(30) | YES  |       | NULL    |       |
| departure_time | time   | YES  |       | NULL    |       |
| arrival_time | time   | YES  |       | NULL    |       |
| duration   | time   | YES  |       | NULL    |       |
| seats      | int(5) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

- 2) Matches Table: This table contains details of the matches that will be played during the All-Star Weekend.

```
mysql> desc matches;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| slno  | int(5) | NO   | PRI  | NULL    |       |
| matches | varchar(23) | YES  |       | NULL    |       |
| date   | date   | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

- 3) User: This table contains match and ticket details of the user.

```

mysql> desc user;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| slno  | int(5) | YES  |      | NULL    |       |
| name   | varchar(30) | YES  |      | NULL    |       |
| emailid | varchar(30) | YES  |      | NULL    |       |
| matches | varchar(30) | YES  |      | NULL    |       |
| date   | date   | YES  |      | NULL    |       |
| seat    | varchar(30) | YES  |      | NULL    |       |
| noofseats | int(5) | YES  |      | NULL    |       |
| seats   | varchar(30) | YES  |      | NULL    |       |
| price   | int(10) | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.10 sec)

```

4) Air: Contains flight details of the user.

```

mysql> desc air;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| slno  | int(5) | YES  |      | NULL    |       |
| name   | varchar(30) | YES  |      | NULL    |       |
| airliner | varchar(30) | YES  |      | NULL    |       |
| departure | varchar(30) | YES  |      | NULL    |       |
| seats   | int(10) | YES  |      | NULL    |       |
| flightno | varchar(30) | YES  |      | NULL    |       |
| price   | varchar(30) | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

Table Data:

1) Flight table:

```

mysql> select * from flight;
+-----+-----+-----+-----+-----+-----+-----+-----+
| flight_no | airliner_name | departure | arrival | departure_time | arrival_time | duration | seats |
+-----+-----+-----+-----+-----+-----+-----+-----+
| AC999  | Emirates  | Auckland | Charlotte | 05:45:00  | 07:15:00  | 01:30:00 | 18  |
| BC345  | Etihad    | Bangalore | Charlotte | 08:30:00  | 08:45:00  | 12:15:00 | 23  |
| JC567  | Jet Airways | Johannesburg | Charlotte | 12:15:00  | 03:45:00  | 03:30:00 | 24  |
| LC234  | Indigo    | London    | Charlotte | 06:30:00  | 09:45:00  | 03:15:00 | 30  |
| SC789  | Indigo    | Sydney    | Charlotte | 09:45:00  | 04:00:00  | 06:15:00 | 35  |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

2) Matches table:

```
mysql> select * from matches;
+-----+-----+-----+
| slno | matches           | date   |
+-----+-----+-----+
| 100  | Warriors vs Lakers | 2019-01-01 |
| 101  | Celtics vs Lakers  | 2019-01-04 |
| 102  | Celtics vs Mavericks | 2019-01-07 |
| 103  | Mavericks vs Lakers | 2019-01-10 |
| 104  | Warriors vs Celtics | 2019-01-13 |
| 105  | Warriors vs Mavericks | 2019-01-15 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

3) User table:

```
mysql> select * from user;
+-----+-----+-----+-----+-----+-----+-----+
| slno | name    | emailid      | matches           | date     | seat | noofseats | seats          | price |
+-----+-----+-----+-----+-----+-----+-----+
| 100  | Jeevan  | curry30@gmail.com | Warriors vs Lakers | 2019-01-13 | NULL | 8        | T3,T4,B3,B4,R3,R4,L3,L4, | 1200 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

4) Air table:

```
mysql> select * from air;
+-----+-----+-----+-----+-----+-----+
| slno | name      | airliner | departure | seats | flightno | price |
+-----+-----+-----+-----+-----+-----+
| 100  | Jeevan Chirag | Etihad   | Bangalore | 3     | BC345   | 225   |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

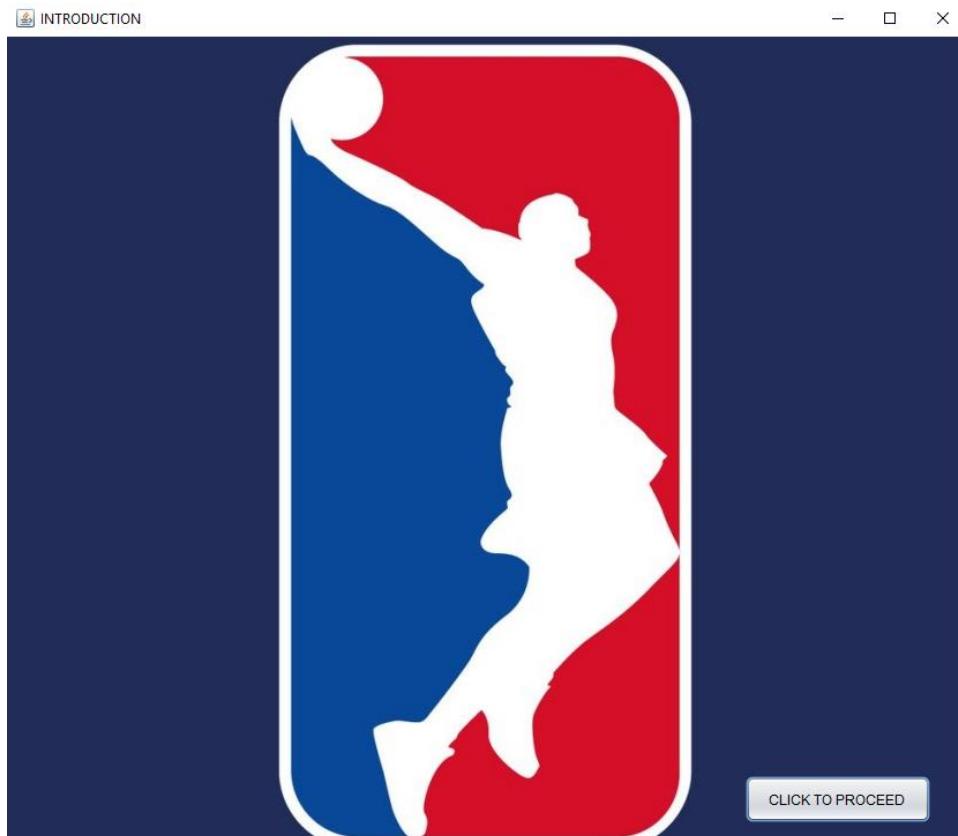
SOURCE CODE

NBA FRAME:-

```
public class abc extends javax.swing.JFrame
{
    public abc () {
        initComponents();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        new teamsFrame().setVisible(true);
        dispose();
    }
}
```

OUTPUT:-



TEAMS FRAME:-

```
public class teamsFrame extends javax.swing.JFrame {  
    public teamsFrame() {  
        initComponents();  
    }  
  
    private void jLabel1MouseClicked (java.awt.event.MouseEvent evt) {  
        new warrFrame().setVisible(true);  
        dispose();  
    }  
  
    private void jLabel2MouseClicked (java.awt.event.MouseEvent evt) {  
        new LakersFrame().setVisible(true);  
        dispose();  
    }  
  
    private void jLabel3MouseClicked (java.awt.event.MouseEvent evt) {  
        new celticsFrame().setVisible(true);  
        dispose();  
    }  
  
    private void jLabel4MouseClicked (java.awt.event.MouseEvent evt) {  
        new mavericksFrame().setVisible(true);  
        dispose();  
    }  
}
```

OUTPUT:-



LINEUPS' FRAME:

- 1) Warriors frame:-

```
public class warrFrame extends javax.swing.JFrame {
    public warrFrame() {
        initComponents();
    }
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    new fixturesFrame().setVisible(true);
    dispose();
}
}
```

OUTPUT:-



2) Lakers Frame:

```
public class lakersFrame extends javax.swing.JFrame {  
    public lakersFrame() {  
        initComponents();  
    }  
  
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
        new fixturesFrame().setVisible(true);  
        dispose();  
    }  
}
```

OUTPUT:-



3) Celtics Frame:

```
public class celticsFrame extends javax.swing.JFrame {  
    public celticsFrame() {  
        initComponents();  
    }  
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    new fixturesFrame().setVisible(true);  
    dispose();  
} }
```

OUTPUT:-



4) Mavericks Frame:

```
public class mavericksFrame extends javax.swing.JFrame {  
    public mavericksFrame() {  
        initComponents();  
    }
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    new fixturesFrame().setVisible(true);  
    dispose();  
}
```

OUTPUT:-



FIXTURES FRAME:

Import Statements:

```
import java.sql.*;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;
```

```
public class fixturesFrame extends javax.swing.JFrame {  
  
    public fixturesFrame() {  
        initComponents();  
    }  
  
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    DefaultTableModel mod = (DefaultTableModel)jTable1.getModel();  
    if(mod.getRowCount()==0)  
    {  
        try  
        {  
            Class.forName("java.sql.Driver");  
            Connection con =  
                DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");  
            Statement stmt = con.createStatement();  
            String query = "SELECT * FROM matches;";  
            ResultSet rs = stmt.executeQuery(query);  
            while(rs.next())  
            {  
                String n1 = rs.getString("slno");
```

```

String n2 = rs.getString("matches");
String n3 = rs.getString("date");
mod.addRow(new Object[]{n1,n2,n3});
}
rs.close();
con.close();
stmt.close();
}
catch(Exception e )
{
JOptionPane.showMessageDialog(null, "error");
}
}}
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
DefaultTableModel mod = (DefaultTableModel)jTable1.getModel();
int n = jTable1.getSelectedRow();
jTextField1.setText(mod.getValueAt(n,1).toString());
jTextField2.setText(mod.getValueAt(n,2).toString());
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
try
{
String n1=jTextField1.getText();
String n2=jTextField2.getText();

```

```

Class.forName("java.sql.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");

Statement stmt=con.createStatement();

String query="update user set matches='"+n1+"', date='"+n2+"' where slno=100;";

stmt.executeUpdate(query);

con.close();

stmt.close();

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,"Error");

}

new seatingFrame().setVisible(true);

dispose();

}

private void jButton2ActionPerformed(java.awt.event.ActionEventevt) {

new teamsFrame().setVisible(true);

dispose();

}

private void jButton3ActionPerformed(java.awt.event.ActionEventevt) {

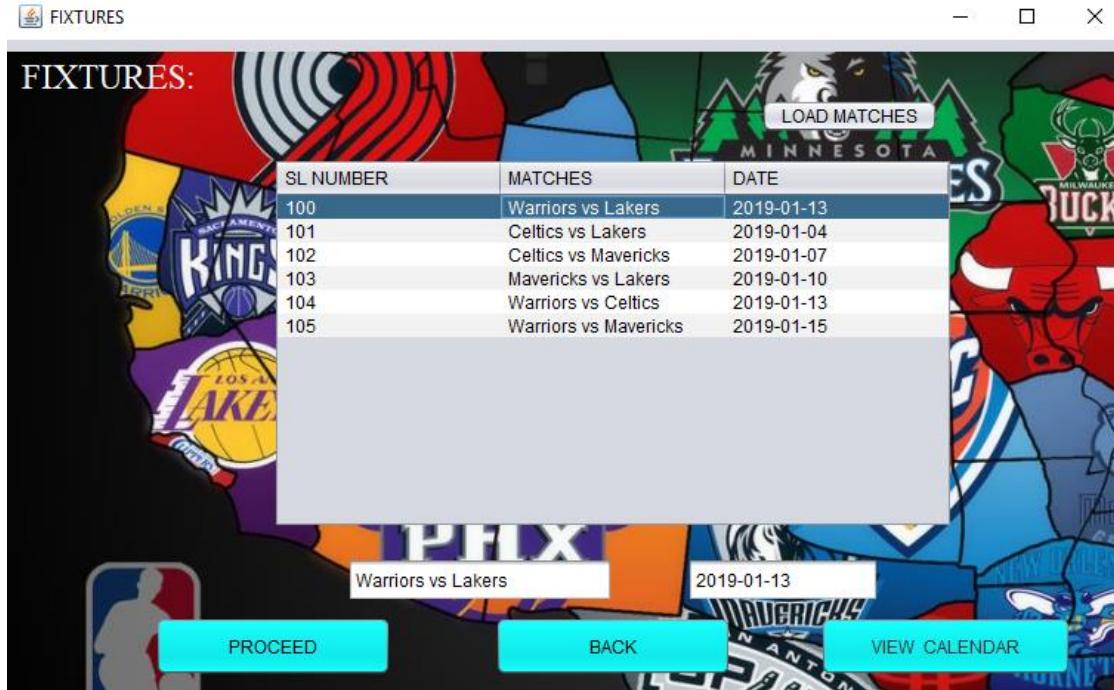
new calendar().setVisible(true);

dispose();

}

```

OUTPUT:-



SEATING FRAME:

Import Statements:

```
import java.sql.*;
```

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

```
public class seatingFrame extends javax.swing.JFrame {
```

```
public seatingFrame() {
```

initComponents();

}

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
new fixturesFrame().setVisible(true);
```

dispose();

1

```
private void jCheckBox1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
if(jCheckBox1.isSelected()==true){
```

```
res.append("T1,");

}

if(jCheckBox1.isSelected()==false){

res.setText("");

} }

private void jCheckBox2ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox2.isSelected()==true){

res.append("T2,");

}

if(jCheckBox2.isSelected()==false){

res.setText("");

} }

private void jCheckBox3ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox3.isSelected()==true){

res.append("T3,");

}

if(jCheckBox3.isSelected()==false){

res.setText("");

} }

private void jCheckBox4ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox4.isSelected()==true){

res.append("T4,");

}

if(jCheckBox4.isSelected()==false){

res.setText("");

}
```

```
}

private void jCheckBox5ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox5.isSelected() == true){
        res.append("T5,");
    }

    if(jCheckBox5.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox6ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox6.isSelected() == true){
        res.append("T6,");
    }

    if(jCheckBox6.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox7ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox7.isSelected() == true){
        res.append("T7,");
    }

    if(jCheckBox7.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox8ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox8.isSelected() == true){
        res.append("T8,");
    }
}
```

```
}

if(jCheckBox8.isSelected() == false){
    res.setText("");
} }

private void jCheckBox9ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox9.isSelected() == true){
        res.append("T9,");
    }
    if(jCheckBox9.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox10ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox10.isSelected() == true){
        res.append("T10,");
    }
    if(jCheckBox10.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox11ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox11.isSelected() == true){
        res.append("T11,");
    }
    if(jCheckBox11.isSelected() == false){
        res.setText("");
    }
}
```

```
private void jCheckBox12ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox12.isSelected() == true){  
        res.append("T12,");  
    }  
    if(jCheckBox12.isSelected() == false){  
        res.setText("");  
    } }  
  
private void jCheckBox13ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox13.isSelected() == true){  
        res.append("T13,");  
    }  
    if(jCheckBox13.isSelected() == false){  
        res.setText("");  
    } }  
  
private void jCheckBox14ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox14.isSelected() == true){  
        res.append("T14,");  
    }  
    if(jCheckBox14.isSelected() == false){  
        res.setText("");  
    } }  
  
private void jCheckBox15ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox15.isSelected() == true){  
        res.append("T15,");  
    } }
```

```
if(jCheckBox15.isSelected()){

res.setText("");    }}

private void jCheckBox16ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox16.isSelected()==true){

res.append("B1,");

}

if(jCheckBox16.isSelected()==false){

res.setText("");

} }

private void jCheckBox17ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox17.isSelected()==true){

res.append("B2,");

}

if(jCheckBox17.isSelected()==false){

res.setText("");

} }

private void jCheckBox18ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox18.isSelected()==true){

res.append("B3,");

}

if(jCheckBox18.isSelected()==false){

res.setText("");

} }

private void jCheckBox19ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox19.isSelected()==true){
```

```
res.append("B4,");

}

if(jCheckBox19.isSelected()==false){

res.setText("");

} }

private void jCheckBox20ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox20.isSelected()==true){

res.append("B5,");

}

if(jCheckBox20.isSelected()==false){

res.setText("");

} }

private void jCheckBox21ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox21.isSelected()==true){

res.append("B6,");

}

if(jCheckBox21.isSelected()==false){

res.setText("");

}}}

private void jCheckBox22ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox22.isSelected()==true){

res.append("B7,");

}

if(jCheckBox22.isSelected()==false){

res.setText("");

}}
```

```
}

private void jCheckBox23ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox23.isSelected() == true){
        res.append("B8,");
    }

    if(jCheckBox23.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox24ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox24.isSelected() == true){
        res.append("B9,");
    }

    if(jCheckBox24.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox25ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox25.isSelected() == true){
        res.append("B10,");
    }

    if(jCheckBox25.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox26ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox26.isSelected() == true){
        res.append("B11,");
    }
}
```

```
}

if(jCheckBox26.isSelected()==false){

res.setText("");

} }

private void jCheckBox27ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox27.isSelected()==true){

res.append("B12,");

}

if(jCheckBox27.isSelected()==false){

res.setText("");

} }

private void jCheckBox28ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox28.isSelected()==true){

res.append("B13,");

}

if(jCheckBox28.isSelected()==false){

res.setText("");

} }

private void jCheckBox29ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox29.isSelected()==true){

res.append("B14,");

}

if(jCheckBox29.isSelected()==false){

res.setText("");

} }
```

```
private void jCheckBox30ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox30.isSelected()==true){  
        res.append("B15,");  
    }  
  
    if(jCheckBox30.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox34ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox34.isSelected()==true){  
        res.append("L1,");  
    }  
  
    if(jCheckBox34.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox31ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox31.isSelected()==true){  
        res.append("L2,");  
    }  
  
    if(jCheckBox31.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox32ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox32.isSelected()==true){  
        res.append("L3,");  
    } }
```

```
}

if(jCheckBox32.isSelected()==false){

res.setText("");
}

private void jCheckBox33ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox33.isSelected()==true){

res.append("L4,");

}

if(jCheckBox33.isSelected()==false){

res.setText("");
}

}

private void jCheckBox35ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox35.isSelected()==true){

res.append("L5,");

}

if(jCheckBox35.isSelected()==false){

res.setText("");
}

}

private void jCheckBox36ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox36.isSelected()==true){

res.append("L6,");

}

if(jCheckBox36.isSelected()==false){

res.setText("");
}

}
```

```
private void jCheckBox37ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox37.isSelected()==true){  
        res.append("L7,");  
    }  
    if(jCheckBox37.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox38ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox38.isSelected()==true){  
        res.append("L8,");  
    }  
    if(jCheckBox38.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox39ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox39.isSelected()==true){  
        res.append("L9,");  
    }  
    if(jCheckBox39.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox41ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox41.isSelected()==true){  
        res.append("L10,");  
    } }
```

```
if(jCheckBox41.isSelected()==false){  
    res.setText("");  
} }  
  
private void jCheckBox42ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox42.isSelected()==true){  
        res.append("L11,");  
    }  
  
    if(jCheckBox42.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox43ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox43.isSelected()==true){  
        res.append("L12,");  
    }  
  
    if(jCheckBox43.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox44ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox44.isSelected()==true){  
        res.append("L13,");  
    }  
  
    if(jCheckBox44.isSelected()==false){  
        res.setText("");  
    } }  
  
private void jCheckBox45ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
if(jCheckBox45.isSelected() == true){  
    res.append("L14,");  
}  
  
if(jCheckBox45.isSelected() == false){  
    res.setText("");  
} }  
  
private void jCheckBox46ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox46.isSelected() == true){  
        res.append("L15,");  
    }  
  
    if(jCheckBox46.isSelected() == false){  
        res.setText("");  
    } }  
  
private void jCheckBox40ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox40.isSelected() == true){  
        res.append("R1,");  
    }  
  
    if(jCheckBox40.isSelected() == false){  
        res.setText("");  
    } }  
  
private void jCheckBox47ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(jCheckBox47.isSelected() == true){  
        res.append("R2,");  
    }  
  
    if(jCheckBox47.isSelected() == false){
```

```
res.setText("");
} }

private void jCheckBox48ActionPerformed(java.awt.event.ActionEvent evt) {
if(jCheckBox48.isSelected()==true){
res.append("R3,");
}

if(jCheckBox48.isSelected()==false){
res.setText("");
} }

private void jCheckBox49ActionPerformed(java.awt.event.ActionEvent evt) {
if(jCheckBox49.isSelected()==true){
res.append("R4,");
}

if(jCheckBox49.isSelected()==false){
res.setText("");
} }

private void jCheckBox50ActionPerformed(java.awt.event.ActionEvent evt) {
if(jCheckBox50.isSelected()==true){
res.append("R5,");
}

if(jCheckBox50.isSelected()==false){
res.setText("");
} }

private void jCheckBox51ActionPerformed(java.awt.event.ActionEvent evt) {
if(jCheckBox51.isSelected()==true){
```

```
res.append("R6,");

}

if(jCheckBox51.isSelected()==false){

res.setText("");

} }

private void jCheckBox52ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox52.isSelected()==true){

res.append("R7,");

}

if(jCheckBox52.isSelected()==false){

res.setText("");

} }

private void jCheckBox53ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox53.isSelected()==true){

res.append("R8,");

}

if(jCheckBox53.isSelected()==false){

res.setText("");

} }

private void jCheckBox54ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox54.isSelected()==true){

res.append("R9,");

}

if(jCheckBox54.isSelected()==false){

res.setText("");

} }
```

```
}

private void jCheckBox55ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox55.isSelected() == true){
        res.append("R10,");
    }

    if(jCheckBox55.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox56ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox56.isSelected() == true){
        res.append("R11,");
    }

    if(jCheckBox56.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox57ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox57.isSelected() == true){
        res.append("R12,");
    }

    if(jCheckBox57.isSelected() == false){
        res.setText("");
    }
}

private void jCheckBox58ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jCheckBox58.isSelected() == true){
        res.append("R13,");
    }
}
```

```
}

if(jCheckBox58.isSelected()==false){

res.setText("");

} }

private void jCheckBox59ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox59.isSelected()==true){

res.append("R14,");

}

if(jCheckBox59.isSelected()==false){

res.setText("");

} }

private void jCheckBox60ActionPerformed(java.awt.event.ActionEvent evt) {

if(jCheckBox60.isSelected()==true){

res.append("R15,");

}

if(jCheckBox60.isSelected()==false){

res.setText("");

} }

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String n = res.getText();

int a =0,b=0,c=0;

for(int i=0;i<n.length();i++){

if((n.charAt(i)=='T')||(n.charAt(i)=='B'))



{



a++;
```

```

}

else if((n.charAt(i)=='L'||(n.charAt(i)=='R')){

b++;

} }

jTextField1.setText((a*100+b*200)+"");

int x=0;

for(int i=0;i<n.length();i++)

{

if((n.charAt(i)=='T'||(n.charAt(i)=='B')||(n.charAt(i)=='R')||(n.charAt(i)=='L'))

{ x++;

}

jTextField2.setText(""+x);

} }

private void jButton3ActionPerformed(java.awt.event.ActionEventevt {

jTextField1.setText(" ");

jTextField2.setText(" ");

res.setText(" ");

jCheckBox1.setSelected(false);

jCheckBox2.setSelected(false);

jCheckBox3.setSelected(false);

jCheckBox4.setSelected(false);

jCheckBox5.setSelected(false);

jCheckBox6.setSelected(false);

jCheckBox7.setSelected(false);

jCheckBox8.setSelected(false);

```

```
jCheckBox9.setSelected(false);  
jCheckBox10.setSelected(false);  
jCheckBox11.setSelected(false);  
jCheckBox12.setSelected(false);  
jCheckBox13.setSelected(false);  
jCheckBox14.setSelected(false);  
jCheckBox15.setSelected(false);  
jCheckBox16.setSelected(false);  
jCheckBox17.setSelected(false);  
jCheckBox18.setSelected(false);  
jCheckBox19.setSelected(false);  
jCheckBox20.setSelected(false);  
jCheckBox21.setSelected(false);  
jCheckBox22.setSelected(false);  
jCheckBox23.setSelected(false);  
jCheckBox24.setSelected(false);  
jCheckBox25.setSelected(false);  
jCheckBox26.setSelected(false);  
jCheckBox27.setSelected(false);  
jCheckBox28.setSelected(false);  
jCheckBox29.setSelected(false);  
jCheckBox30.setSelected(false);  
jCheckBox31.setSelected(false);  
jCheckBox32.setSelected(false);  
jCheckBox33.setSelected(false);
```

```
jCheckBox34.setSelected(false);  
jCheckBox35.setSelected(false);  
jCheckBox36.setSelected(false);  
jCheckBox37.setSelected(false);  
jCheckBox38.setSelected(false);  
jCheckBox39.setSelected(false);  
jCheckBox40.setSelected(false);  
jCheckBox41.setSelected(false);  
jCheckBox42.setSelected(false);  
jCheckBox43.setSelected(false);  
jCheckBox44.setSelected(false);  
jCheckBox45.setSelected(false);  
jCheckBox46.setSelected(false);  
jCheckBox47.setSelected(false);  
jCheckBox48.setSelected(false);  
jCheckBox49.setSelected(false);  
jCheckBox50.setSelected(false);  
jCheckBox51.setSelected(false);  
jCheckBox52.setSelected(false);  
jCheckBox53.setSelected(false);  
jCheckBox54.setSelected(false);  
jCheckBox55.setSelected(false);  
jCheckBox56.setSelected(false);  
jCheckBox57.setSelected(false);  
jCheckBox58.setSelected(false);
```

```

jCheckBox59.setSelected(false);
jCheckBox60.setSelected(false); }

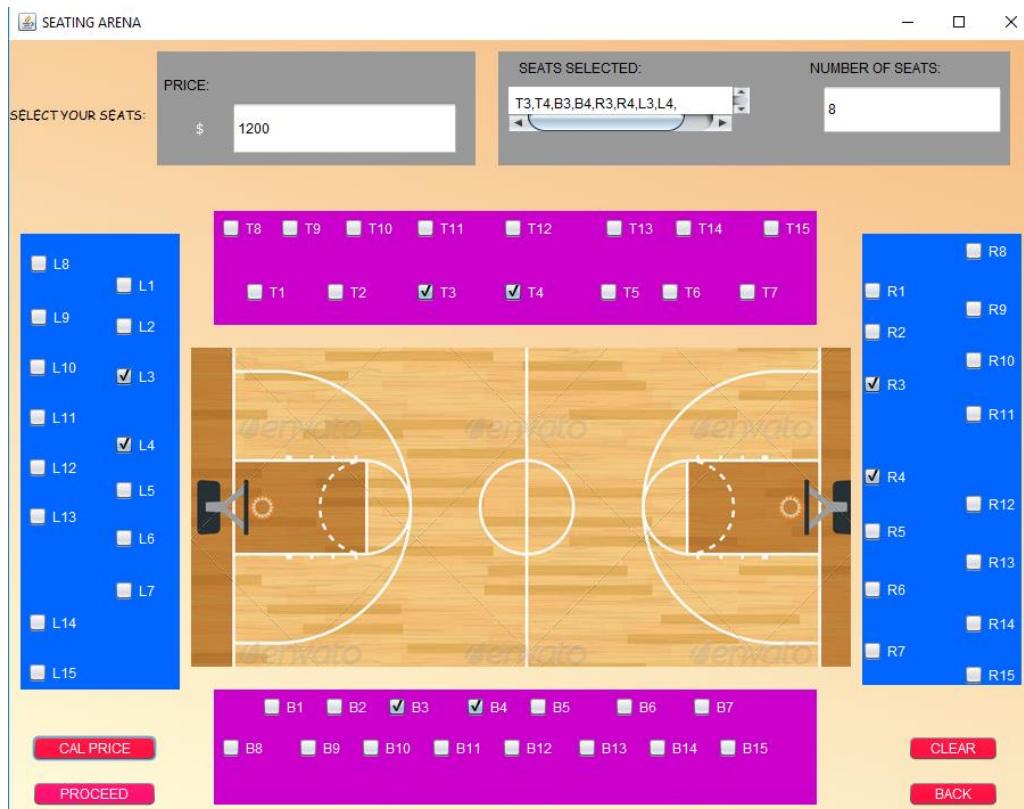
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    String n1=res.getText();
    String n2=jTextField1.getText();
    String n=jTextField2.getText();
    if(n>5){
        JOptionPane.showMessageDialog(null,"Limited to 5 seats"); }
    else {
        try {
            Class.forName("java.sql.Driver");
            Connection
            con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");
            Statement stmt=con.createStatement();
            String quer="update user set seats='"+n1+"', price='"+n2+"', noofseats='"+n+"' where
slno=100;";
            System.out.println(quer);
            stmt.executeUpdate(quer);
            con.close();
            stmt.close();
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,"Error");
        }
        new ticketFrame().setVisible(true);
    }
}

```

```
dispose();
```

```
} }
```

OUTPUT:-



TICKET FRAME:

Import Statements:-

```
import java.sql.*;
```

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

```
public class ticketFrame extends javax.swing.JFrame {
```

```
public ticketFrame() {
```

```
initComponents();
```

```
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
new seatingFrame().setVisible(true);

dispose();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
if(jRadioButton1.isSelected()==true)
{
new creditcardFrame().setVisible(true);
dispose();
}
if(jRadioButton2.isSelected()==true)
{
new debitcardFrame().setVisible(true);
dispose();
}
try
{
String n1 = jTextField1.getText();
String n2 = jTextField3.getText();
Class.forName("java.sql.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");
Statement stmt = con.createStatement();
String query="update user set name='"+n1+"', emailid='"+n2+"' where slno=100;";
stmt.executeUpdate(query);
con.close();
stmt.close();
}
```

```
}

catch(Exception e){

JOptionPane.showMessageDialog(null,"Error");

} }

private void jButton4ActionPerfromed(java.awt.event.ActionEvent evt) {

try

{

Class.forName("java.sql.Driver");

Connection

con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");

Statement stmt=con.createStatement();

String query="select matches,date,seats,price,noofseats from user where slno=100";

ResultSet rs=stmt.executeQuery(query);

if(rs.next())

{

String n1=rs.getString("matches");

String n2=rs.getString("date");

String n3=rs.getString("seats");

String n4=rs.getString("price");

String n5=rs.getString("noofseats");

jTextField5.setText(""+n5);

jTextField8.setText(""+n1);

jTextField7.setText(""+n2);

jTextField6.setText(""+n3);

jTextField10.setText(""+n4);

}

}
```

```

        rs.close();

        con.close();

        stmt.close();

    }

    catch(Exception e) {

        JOptionPane.showMessageDialog(null,"Error"); }

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    jTextField1.setText(" ");

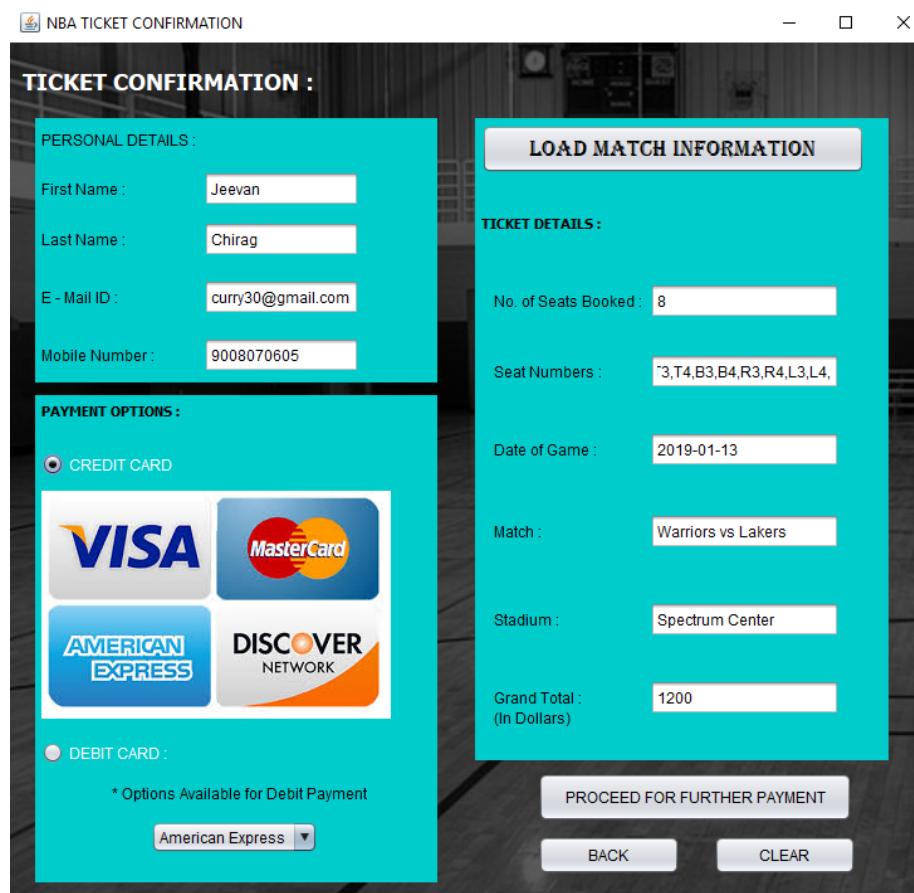
    jTextField2.setText(" ");

    jTextField3.setText(" ");

    jTextField4.setText(" "); }

```

OUTPUT:-



CREDIT CARD FRAME:

Import Statements:

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

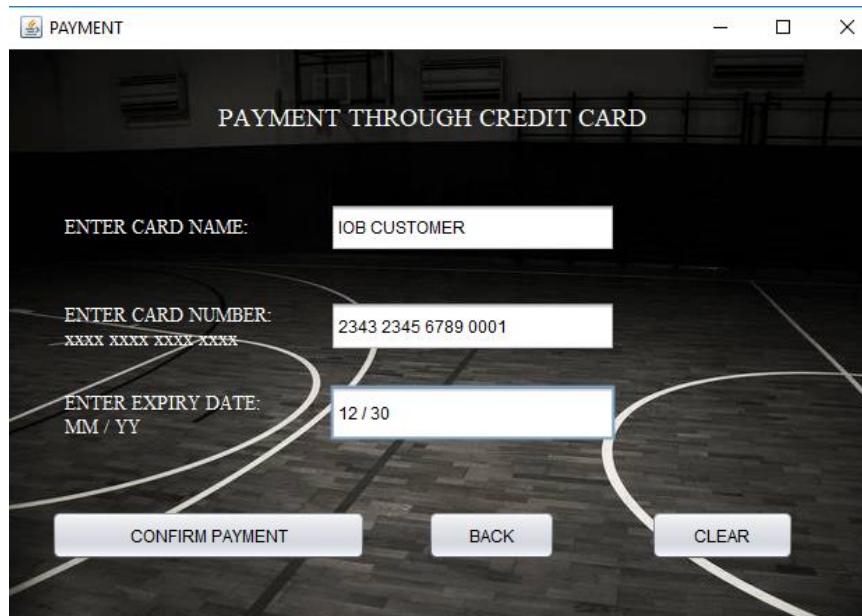
```
public class creditcardFrame extends javax.swing.JFrame {  
    public creditcardFrame() {  
        initComponents();  
    }  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String e = jTextField2.getText();  
    if(e.length()!=19)  
    {  
        JOptionPane.showMessageDialog(null,"Credit Card should be 16 digits");  
    }  
    else  
    {  
        JOptionPane.showMessageDialog(null,"Payment Confirmed");  
        new confirmationFrame().setVisible(true);  
        dispose();  
    }  
    if((jTextField1.getText().equals(""))||(jTextField2.getText().equals("")))  
    {  
        JOptionPane.showMessageDialog(null,"INVALID DETAILS");  
    }  
} }
```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    new ticketFrame().setVisible(true);
    dispose();
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("} 
```

OUTPUT:-



DEBIT CARD FRAME:

Import Statements:

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

```
public class debitcardFrame extends javax.swing.JFrame {
```

```
public debitcardFrame() {
```

```

initComponents();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    if((jTextField1.getText().equals(""))||(jPasswordField1.getText().equals(""))||(jTextField3.getText().equals("")))
    {
        JOptionPane.showMessageDialog(null,"INVALID DETAILS");
    }

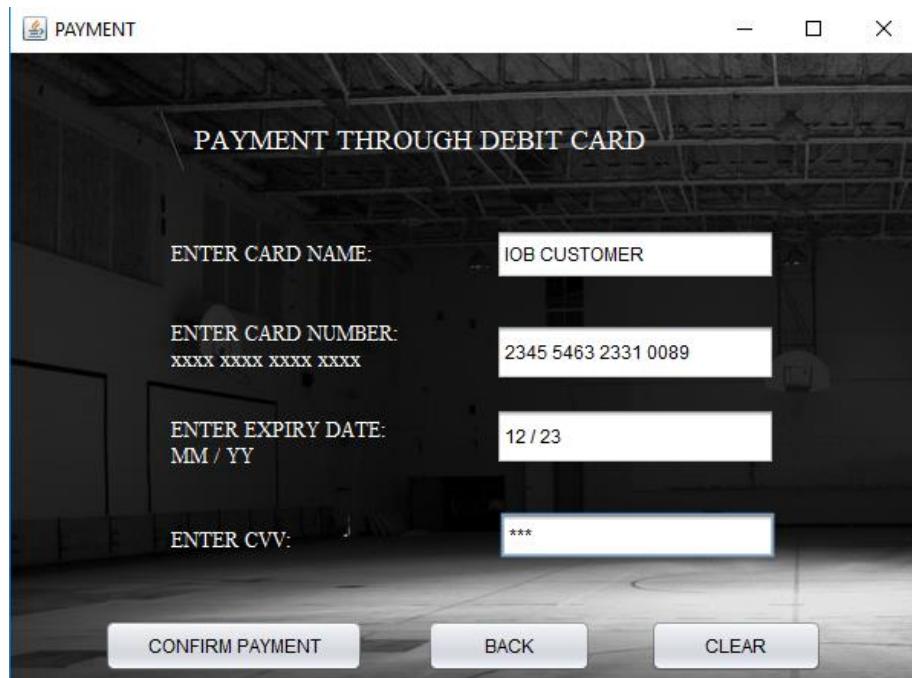
    String e = jTextField3.getText();
    String f = jPasswordField1.getText();
    if((e.length()!=19)|| (f.length()!=3))
    {
        JOptionPane.showMessageDialog(null,"Please enter correct card details");
    }
    else
    {
        JOptionPane.showMessageDialog(null,"Payment Confirmed");
        new confirmationFrame().setVisible(true);
        dispose();  } }

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    new ticketFrame().setVisible(true);
    dispose();      }

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setText("");
    jPasswordField1.setText("");
    jTextField3.setText("");
    jTextField4.setText(""); }


```

OUTPUT:-



NBA TICKET CONFIRMATION FRAME:

Import Statements:

```
import java.sql.*;  
import javax.swing.JOptionPane;
```

```
public class confirmationFrame extends javax.swing.JFrame {  
  
    public confirmationFrame() {  
        initComponents();  
    }  
  
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
        try {  
            String n1=jTextField5.getText();  
            Class.forName("java.sql.Driver");
```

```

Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");

Statement stmt=con.createStatement();

String query="update air set name='"+n1+"' where slno=100;";

stmt.executeUpdate(query);

con.close();

stmt.close();

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,"Error");

}

new airplaneIntroFrame().setVisible(true);

dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try

{

Class.forName("java.sql.Driver");

Connection

con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");

Statement stmt=con.createStatement();

String quer="update user set
matches=null,date=null,name=null,emailid=null,seat=null,noofseats=null,price=null,
seats=null;";
```

```
System.out.println(quer);
stmt.executeUpdate(quer);
con.close();
stmt.close();
}

catch(Exception e)
{
JOptionPane.showMessageDialog(null,"Error");
}

JOptionPane.showMessageDialog(null,"Details Saved");
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
JOptionPane.showMessageDialog(null,"THANKS FOR BOOKING NBA TICKETS");
System.exit(0);
}

private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
if(jTextField5.getText().equals(""))
{
try
{
Class.forName("java.sql.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");
Statement stmt=con.createStatement();
String quer="select name,price,seats from user where slno=100";
System.out.println(quer);
}
}
}
```

```

ResultSet rs=stmt.executeQuery(quer);
if(rs.next())
{
String n1=rs.getString("name");
String n2=rs.getString("price");
String n3=rs.getString("seats");
jTextField5.setText(""+n1);
jTextField6.setText(""+n2);
jTextField7.setText(""+n3);
}
rs.close();
con.close();
stmt.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(null,"Error");
}}
jTextField8.setText(""+Math.round(Math.random()*100000));
}

private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
int n = Integer.parseInt(jTextField6.getText());
if((n>600))
{
JOptionPane.showMessageDialog(null,"You have won 1500 NBA points. NBA points
can be used to book NBA flights, or to buy exclusive merchandise from NBystore.in");
}

```

```

}

if((n>300)&&(n<=500))

{

JOptionPane.showMessageDialog(null,"You have won 700 NBA points. NBA points can
be used to book NBA flights, or to buy exclusive merchandise from NBystore.in");

}

if((n>50)&&(n<=300))

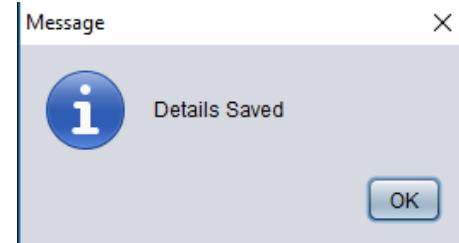
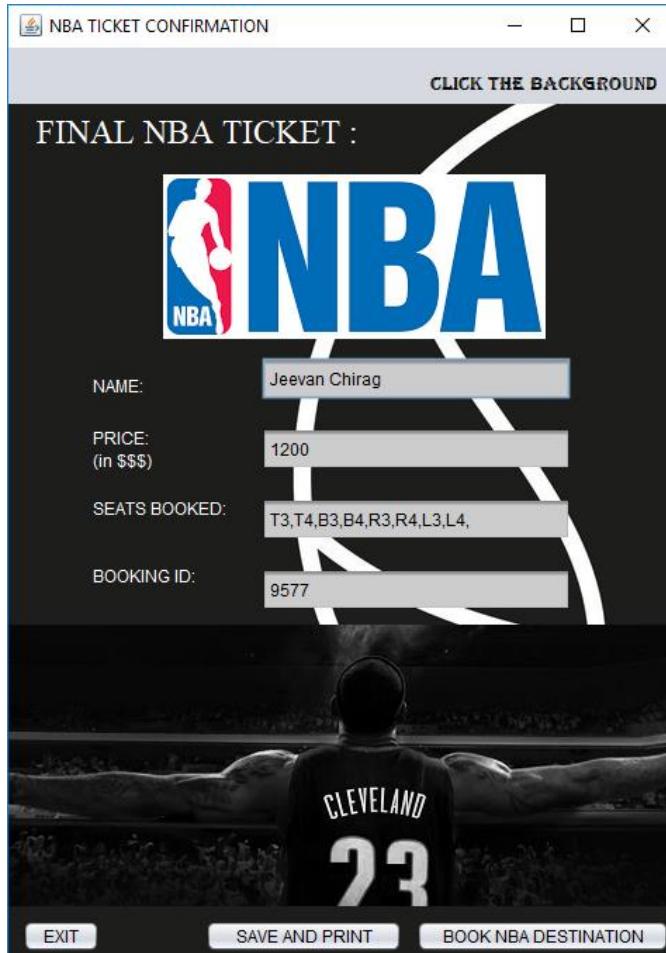
{

JOptionPane.showMessageDialog(null,"You have won 500 NBA points. NBA points can
be used to book NBA flights, or to buy exclusive merchandise from NBystore.in");

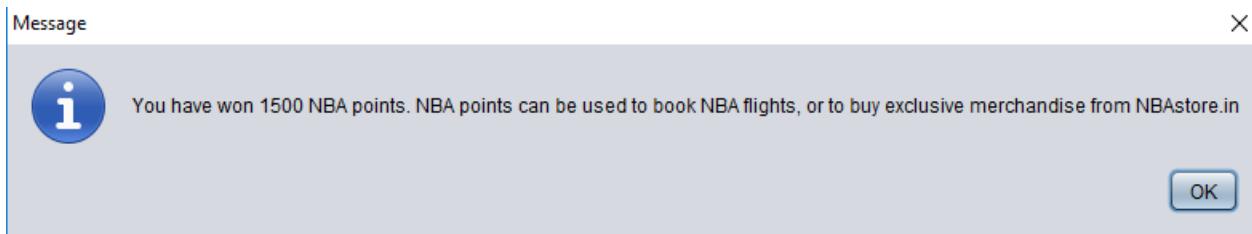
}
}

```

OUTPUT:-



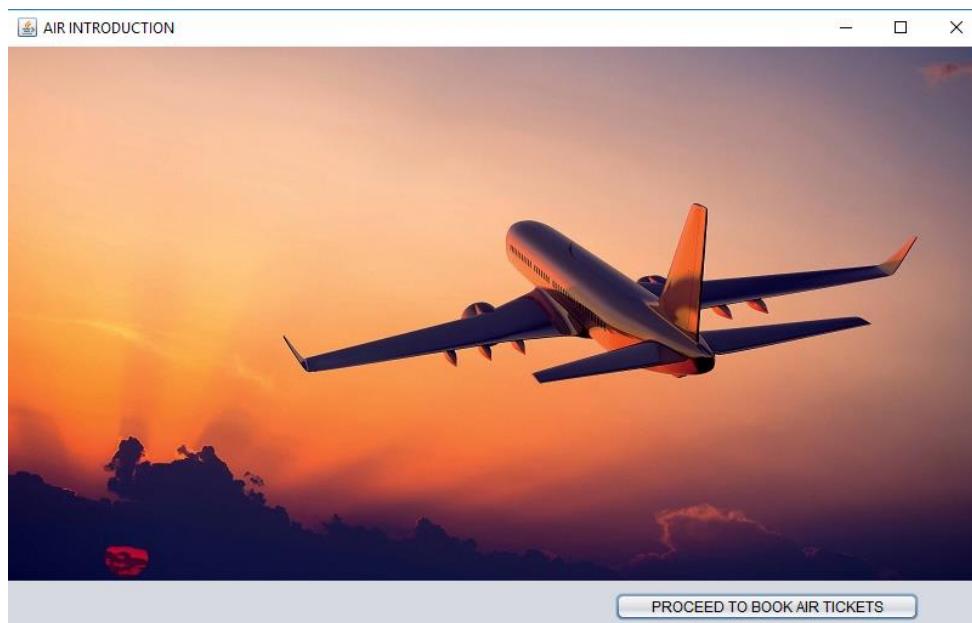
// message displaying NBA credits won by the user:



AIRPLANE INTRO FRAME:

```
public class airplaneIntroFrame extends javax.swing.JFrame {  
    public airplaneIntroFrame()  
    initComponents();  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    new flightFrame().setVisible(true);  
    dispose();  
}
```

OUTPUT:-



AIR TICKET RESERVATION:

Import Statements:

```
import java.sql.*;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;  
  
public class flightFrame extends javax.swing.JFrame {  
    public flightFrame() {  
        initComponents();  
    }  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    int res = JOptionPane.showConfirmDialog(null,"Confirm payment of $" + price + " for  
    flight ticket using NBA points");  
    if(res==JOptionPane.YES_OPTION)  
    {  
        try  
        {  
            String n1=jTextField6.getText();  
            String n2=jTextField5.getText();  
            int n3 = Integer.parseInt(jTextField3.getText());  
            String n4=jTextField7.getText();  
            int n5 = Integer.parseInt(jTextField8.getText());  
            Class.forName("java.sql.Driver");  
            Connection  
            con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");  
            Statement stmt=con.createStatement();
```

```

String query="update air set
airliner='"+n1+"',departure='"+n2+"',seats='"+n3+"',flightno='"+n4+"',price='"+n5+"'
where slno=100;";

stmt.executeUpdate(query);

con.close();

stmt.close();

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,"ERROR");

}

new flightConfirmation().setVisible(true);

dispose();

} }

```

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

```
DefaultTableModel mod = (DefaultTableModel)jTable1.getModel();
```

```
int n = jTable1.getSelectedRow();
```

```
jTextField6.setText(mod.getValueAt(n,1).toString());
```

```
jTextField5.setText(mod.getValueAt(n,2).toString());
```

```
jTextField7.setText(mod.getValueAt(n, 0).toString());
```

```
}
```

```
int price = 0 ; ——————> global variable
```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

```
int n = Integer.parseInt(jTextField3.getText());
```

```
if(n>4)
```

```
{
```

```
JOptionPane.showMessageDialog(null,"Maximum Seats per person is 4.");  
}  
else{  
    price = 75*n;  
    jTextField8.setText(""+price+"");  
}  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    DefaultTableModel mod = (DefaultTableModel)jTable1.getModel();  
    String a=jTextField1.getText();  
    if(a.equalsIgnoreCase("bangalore")||(a.equalsIgnoreCase("sydney")||(a.equalsIgnoreCase("auckland")||(a.equalsIgnoreCase("johannesburg")||a.equalsIgnoreCase("london")))) {  
        if(mod.getRowCount()==0) {  
            try {  
                Class.forName("java.sql.Driver");  
                Connection con =  
                    DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");  
                Statement stmt = con.createStatement();  
                String query = "SELECT * FROM flight;";  
                ResultSet rs = stmt.executeQuery(query);  
                while(rs.next())  
                {  
                    String n1 = rs.getString("flight_no");  
                    String n2 = rs.getString("airliner_name");  
                    String n3 = rs.getString("departure");  
                    String n4 = rs.getString("arrival");  
                }  
            }  
        }  
    }  
}
```

```

String n5 = rs.getString("departure_time");
String n6 = rs.getString("arrival_time");
String n7 = rs.getString("duration");
String n8 = rs.getString("seats");
mod.addRow(new Object[]{n1,n2,n3,n4,n5,n6,n7,n8}); }

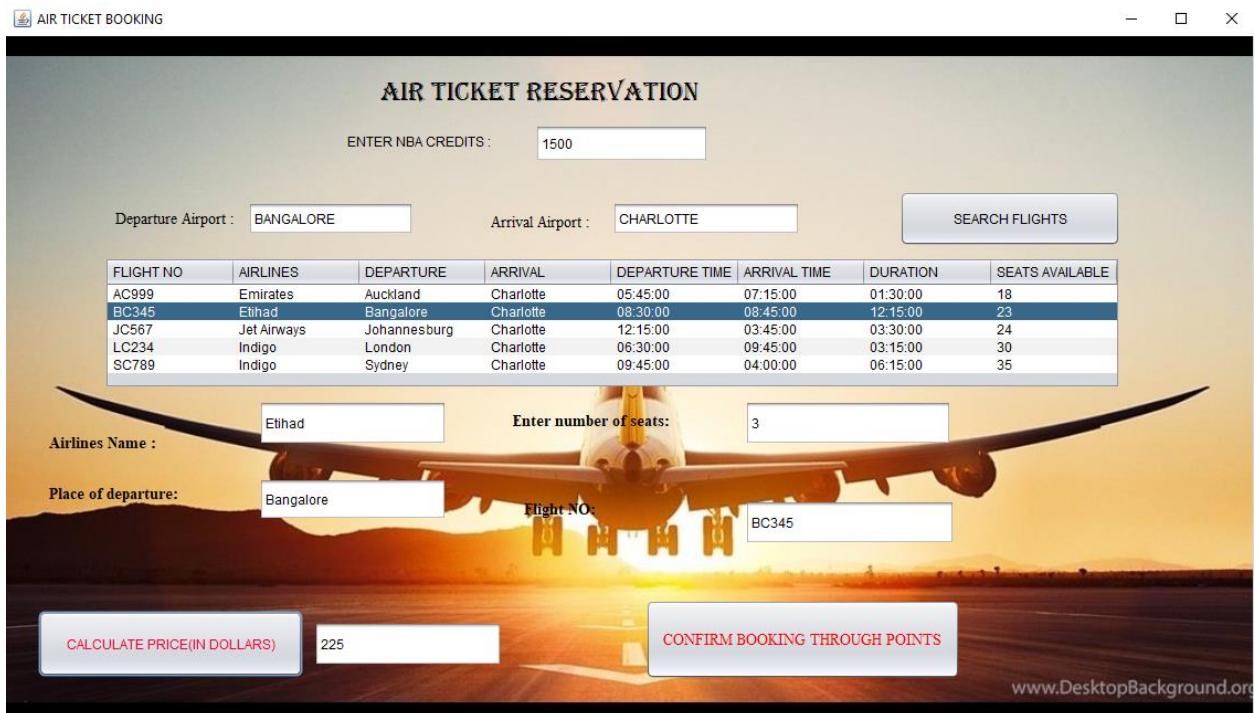
rs.close();
con.close();
stmt.close(); }

catch(Exception e )
{
JOptionPane.showMessageDialog(null, "error"); } }

else {
JOptionPane.showMessageDialog(this,"Flights are not available from this
destination"); }

```

OUTPUT:-



AIR TICKET CONFIRMATION FRAME:

Import Statements:

```
import java.sql.*;  
import javax.swing.JOptionPane;
```

```
public class flightConfirmation extends javax.swing.JFrame {  
  
    public flightConfirmation() {  
  
        initComponents(); }  
  
private void jLabel9MouseClicked(java.awt.event.MouseEvent evt) {  
  
    try  
  
    {  
  
        Class.forName("java.sql.Driver");  
  
        Connection  
        con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");  
  
        Statement stmt=con.createStatement();  
  
        String query="select name,airliner,departure,seats,flightno,price from air where  
        slno=100";  
  
        ResultSet rs=stmt.executeQuery(query);  
  
        if(rs.next())  
  
        {  
  
            String n1=rs.getString("name");  
  
            String n2=rs.getString("airliner");  
  
            String n3=rs.getString("departure");  
  
            String n4=rs.getString("seats");  
  
            String n5=rs.getString("flightno");  
  
            String n6=rs.getString("price");
```

```

jTextField1.setText(""+n1);
jTextField2.setText(""+n2);
jTextField3.setText(""+n3);
jTextField5.setText(""+n4);
jTextField6.setText(""+n5);
jTextField7.setText(""+n6); }

rs.close();
con.close();
stmt.close();
}

catch(Exception e)

{
JOptionPane.showMessageDialog(null,"ERROR"); } }

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
new offersFrame().setVisible(true);
dispose();      }

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
try
{
Class.forName("java.sql.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");
Statement stmt=con.createStatement();

String quer="update air set
airliner=null,departure=null,name=null,flightno=null,seats=null,price=null;";

System.out.println(quer);
}

```

```

stmt.executeUpdate(quer);

con.close();

stmt.close();

}

catch(Exception e) {

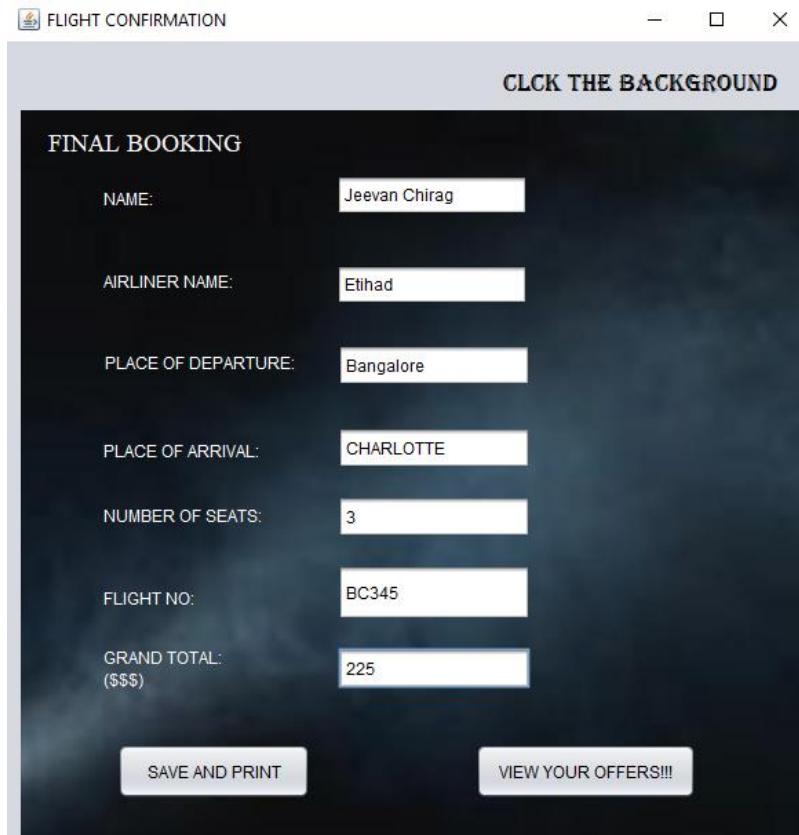
JOptionPane.showMessageDialog(null,"Error"); }

JOptionPane.showMessageDialog(null,"Details Saved");

}

```

OUTPUT:-



OFFERS FRAME:

Import Statements:

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

```

public class offersFrame extends javax.swing.JFrame {

    public offersFrame() {
        initComponents();
    }

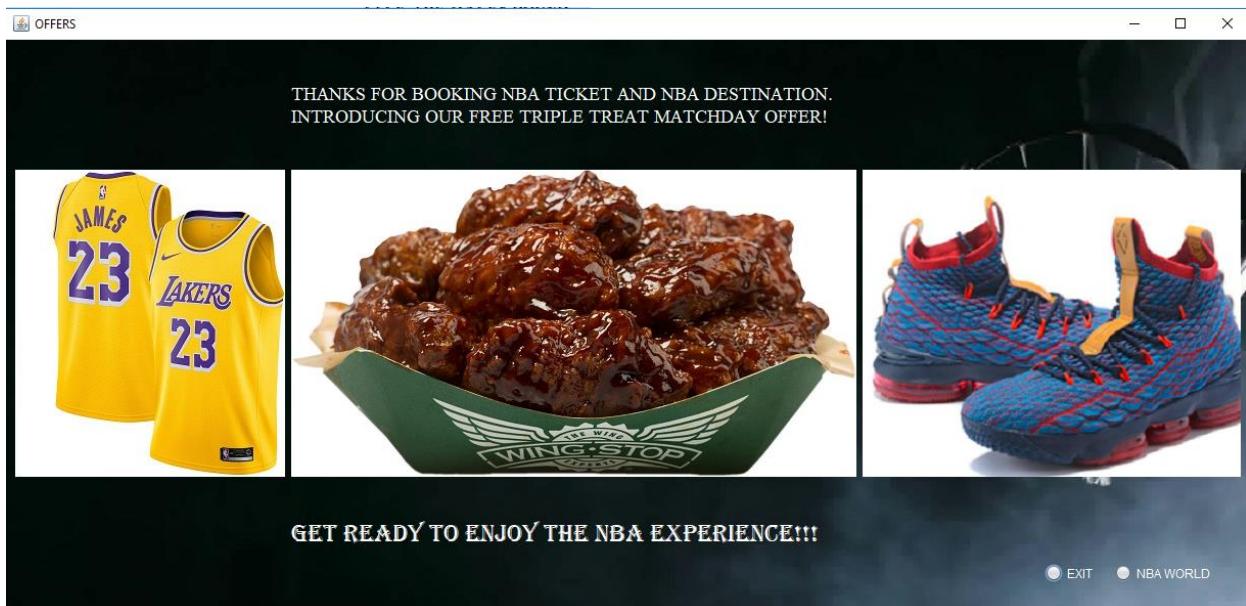
    private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        JOptionPane.showMessageDialog(null," THANKS FOR BOOKING NBA AND FLIGHT
TICKETS WITH US , HOPE TO SEE YOU SOON ");

        System.exit(0);
    }

    private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        new thankYouFrame().setVisible(true);
        dispose();
    }
}

```

OUTPUT:-



ENDING FRAME:

Import Statements:

```
import java.awt.Desktop;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
public class thankYouFrame extends javax.swing.JFrame {
    public thankYouFrame() {
        initComponents();
    }
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Desktop browser = Desktop.getDesktop();
    try {
        browserbrowse(new URI("https://in.nba.com/?gr=www"));
    } catch(IOException err) {
        }catch(URISyntaxException ex) {
            Logger.getLogger(calc.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    Desktop browser = Desktop.getDesktop();
    try {
```

```
        browser/browse(new URI("https://store.nba.com/")); }

    catch(IOException err)

    { }catch(URISyntaxException ex) {

        Logger.getLogger(calc.class.getName()).log(Level.SEVERE, null, ex);

    }

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    Desktop browser = Desktop.getDesktop();

    try

    { browser/browse(new URI("https://www.youtube.com/results?search_query=nba"));

    }

    catch(IOException err)

    {

    }catch(URISyntaxException ex) {

        Logger.getLogger(calc.class.getName()).log(Level.SEVERE, null, ex);

    }

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    JOptionPane.showMessageDialog(null,"HOPE YOU ENJOYED YOUR JOURNEY
WITH US. HOPE TO SEE YOU AGAIN");

    System.exit(0);

}

}
```

OUTPUT:-



// on clicking any of the buttons, the corresponding webpage is opened.

BIBLIOGRAPHY

- *INFORMATICS PRACTICES FOR CLASS 12 BY SUMITA ARORA*
- *GOOGLE IMAGES*
- *GOOGLE SEARCH*
- *WIKIPEDIA*
- *YOUTUBE*

