

VIDYAGYAN SCHOOL
BULANDSHAHAR
A PROJECT REPORT ON
Indian Premier League

For

AISSCE 2017-18 Examination

[As a part of the Informatics Practices Course (065)]

SUBMITTED BY

Kunwar Bahadur [Roll No:-5850521]

Shivam Kumar [Roll No:-5850550]

Under the Guidance of:
Mr. Sanjay Saxena
Dept of (Comp.Sci)

ACKNOWLEDGEMENT

I undertook this Project work, as the part of my XII-Informatics

Practices course. I had tried to apply my best of knowledge and experience, gained during the study and class work experience. However, developing software system is generally a quite complex and time-consuming process.

It requires a systematic study, insight vision and professional approach during the design and development. Moreover, the developer always feels the need, the help and good wishes of the people near you, who have considerable experience and idea.

I would like to extend my sincere thanks and gratitude to my teacher [Mr. Sanjay Saxena](#). I am very much thankful to our Principal [Mr. B. Banerjee](#) for giving valuable time and moral support to develop this software.

I also feel indebted to my friends for the valuable suggestions during the project work.

Shivam Kumar

Class XII

CONTENTS

1. Introduction	5
2. Objective & Scope of the Project	6
3. Theoretical Background	7
4. Problem Definition & Analysis	10
5. System Implementation	12
5.1 The Hardware used:	12
5.2 The Softwares used:	12
6. System Design & Development	13
6.1 Database Design:	13
6.2 I/O Forms Design & Event Coding:	14
7. User Manual	45
8.1 How to install:	45
8.2 Working with Software:	46
8. References	47

INTRODUCTION

This software project is developed to automate the functionalities of **Indian Premier League**.

The purpose of the software project is to develop a program which provides an easily accessible interface to the authorities and for the General Peoples visiting to application to keep his records up to date. This program mainly brings for the usage of GUI programming in the daily usage.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve , update and manage all information in proper way.

This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for managing the records of a Indian Premier League.

During coding and design of the software Project, Java NetBeans IDE, a powerful front-end tool is used for getting Graphical User Interface (GUI) based integrated platform and coding simplicity. As a back-end a powerful, open source RDBMS, My SQL is used as per requirement of the CBSE curriculum of Informatics Practices Course.

Objective & Scope of the Project

The objective of the software project is to develop a computerized MIS to automate the functions of **Indian Premier League**. This software project is also aimed to enhance the current record keeping system, which will help managers to retrieve the up-to-date information at right time in right shape.

The proposed software system is expected to do the following functionality-

- ✓ To provide user a friendly and an easily accessible Graphical User Interface (GUI) based integrated and centralized environment for MIS activities.
- ✓ The proposed system should maintain all the records and should generate the required reports and information when required.
- ✓ To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
- ✓ To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

In its current scope, the software enables user to retrieve and update the information from centralized database designed with MySQL. This software does not require much training time of the users due to limited functionality and simplicity.

During the development of **Indian Premier League** project, Java NetBeans IDE, a powerful, open source event-driven form-based development environment is used for modular design and future expandability of the system.

Despite of the best effort of the developer, the following limitations and functional boundaries are visible, which limits the scope of this application software.

1. This software can store records and produce reports in pre-designed format in soft copy. There is no facility yet to produce customized reports. Only specified reports are covered.
2. There is no provision to calculate fine or penalty etc. for defaulter members; however it can be developed easily with the help of adding modules.

3. Some application areas like book the seats for the audiences , Update their seats facilities,The Rating Table of the league etc. are implemented in the project. It facilitates the user to keep the record of the peoples entering and leaving the IPL.
4. So far as future scope of the project is concerned, firstly it is open to any modular expansion i.e. other modules or functions can be designed and embedded to handle the user need in future. Any part of the software and reports can be modified independently without much effort.

Theoretical Background

What is Database?

Introduction and Concepts:

A database is a collection of information related to a particular subject or purpose, such as tracking customer orders or maintaining a product collection. Using any RDBMS application software like MS SQL Server, MySQL, Oracle, Sybase etc., you can manage all your information from a single database file. Within the file, divide your data into separate storage containers called tables. You may add and retrieve the data using queries.

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means you can store that data only once, which makes your database more efficient and reduces data-entry errors. Table organises data into columns (called fields) and rows (called records).

A Primary key is one or more fields whose value or values uniquely identify each record in a table. In a relationship, a primary key is used to refer to specific record in one table from another table. A primary key is called foreign key when it is referred to from another table.

To find and retrieve just the data that meets conditions you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform built-in or custom calculations on your data.



Role of RDBMS Application Program:

A computer database works as a electronic filing system, which has a large number of ways of cross-referencing, and this allows the user many different ways in which to re-organize and retrieve data. A database can handle business inventory, accounting and filing and use the information in its files to prepare summaries, estimates and other reports. The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available DBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase. A database management system, therefore, is a combination of hardware and software that can be used to set up and monitor a database, and can manage the updating and retrieval of database that has been stored in it. Most of the database management systems have the following capabilities:

- ◆ Creating of a table, addition, deletion, modification of records.
- ◆ Retrieving data collectively or selectively.
- ◆ The data stored can be sorted or indexed at the user's discretion and direction.
- ◆ Various reports can be produced from the system. These may be either standardized report or that may be specifically generated according to specific user definition.
- ◆ Mathematical functions can be performed and the data stored in the database can be manipulated with these functions to perform the desired calculations.
- ◆ To maintain data integrity and database use.

The DBMS interprets and processes users' requests to retrieve information from a database. In most cases, a query request will have to penetrate several layers of software in the DBMS and operating system before the physical database can be accessed. The DBMS responds to a query by invoking the appropriate subprograms, each of which performs its special function to interpret the query, or to locate the desired data in the database and present it in the desired order.

What is My SQL?



The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available RDBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. MySQL is named after co-founder Monty Wideness's daughter, My. The name of the MySQL Dolphin (our logo) is "Sakila,".

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL is based on SQL.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License),

- **The MySQL Database Server is very fast, reliable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server also has a practical set of features developed in close cooperation with our users. You can find a performance comparison of MySQL Server with other database managers on our benchmark page. MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

The Main Features of MySQL

- Written in C and C++.
- Works on many different platforms.
- Uses multi-layered server design with independent modules.
- Provides transactional and non-transactional storage engines.
- Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
- Uses a very fast thread-based memory allocation system.
- Executes very fast joins using an optimized nested-loop join.
- Implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization.
- Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.
- Password security by encryption of all password traffic when you connect to a server.
- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- MySQL client programs can be written in many languages. A client library written in C is available for clients written in C or C++, or for any language that provides C bindings.
- APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available, enabling MySQL clients to be written in many languages.
- The Connector/ODBC (MyODBC) interface provides MySQL support for client programs that use ODBC (Open Database Connectivity) connections.
- The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix. Connector/J source is available.

What is NetBeans IDE ?

NetBeans started as a student project (originally called Xelfi) in the Czech Republic in 1996. The goal was to write a Delphi-like Java IDE in Java. Xelfi was the first Java IDE (Integrated Development Environment) written in Java, with its first pre-releases in 1997. Xelfi was a fun project to work on, especially since Java IDE space was uncharted territory at that time. The project attracted enough interest that these students, once they graduated, decided that they could market it as a commercial product. Soliciting resources from friends and relatives for a web space, they formed a company around it.

Soon after, they were contacted by Roman Stanek, an entrepreneur who had already been involved in several startups in the Czech Republic. He was looking for a good idea to invest in, and discovered Xelfi. He met with the founders; they hit it off, and a business was born.

In the spring of 1999, NetBeans DeveloperX2 was released, supporting Swing. The performance improvements that came in JDK 1.3, released in the fall of 1999, made NetBeans a viable choice

for development tools. By the summer of 1999, the team was hard at work re-architecting DeveloperX2 into the more modular NetBeans that forms the basis of the software today.

Something else was afoot in the summer of 1999: Sun Microsystems wanted better Java development tools, and had become interested in NetBeans. It was a dream come true for the NetBeans team: NetBeans would become the flagship tool set of the maker of Java itself! By the fall, with the next generation of NetBeans Developer in beta, a deal was struck. Sun Microsystems had also acquired another tools company, during the acquisition, the young developers who had been involved in open-source projects for most of their programming careers, mentioned the idea of open-sourcing NetBeans. Fast forward to less than six months later, the decision was made that NetBeans would be open sourced. While Sun had contributed considerable amounts of code to open source projects over the years, this was Sun's first *sponsored* open source project, one in which Sun would be paying for the site and handling the infrastructure.

Features of NetBeans

A free, open-source Integrated Development Environment for software developers. You get all the tools you need to create professional desktop, enterprise, web, and mobile applications with the Java platform, as well as C/C++, PHP, JavaScript, Groovy, and Ruby.

NetBeans IDE 6.9 introduces the JavaFX Composer, support for JavaFX SDK 1.3, OSGi interoperability, support for the PHP Zend framework and Ruby on Rails 3.0, and more.

Problem Definition & Analysis

The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavor has eluded many of us. Yet, we continue to make progress.

Problem definition describes the *What* of a system, not *How*. The quality of a software product is only as good as the process that creates it. Problem definition is one of the most crucial steps in this creation process. Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement.

Problem definition and Analysis is the activity that encompasses learning about the problem to be solved, understanding the needs of customer and users, trying to find out who the user really is, and understanding all the constraints on the solution. It includes all activities related to the following:

- ✓ Identification and documentation of customer's or user's needs.
- ✓ Creation of a document that describes the external behavior and the association constraints that will satisfies those needs.
- ✓ Analysis and validation of the requirements documents to ensure consistency, completeness, and feasibility
- ✓ Evolution of needs.

After the analysis of the functioning of a Clothing Store system, the proposed System is expected to do the following: -

- ✓ To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for computerized Purchasing of Products.
- ✓ The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
- ✓ To provide efficient and secured Information storage, flow and retrieval system, ensuring the integrity and validity of records.
- ✓ To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
- ✓ To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

System Implementation

The Hardware used:

While developing the system, the used hardware are:

PC with Pentium Dual Core processor having 2.00 GB RAM, SVGA and other required devices.

The Softwares used:

Microsoft Windows® 8 as Operating System.

Java NetBeans 7.2 as Front-end Development environment.

MySQL as Back-end Sever with Database for Testing.

MS-Word 2013 for documentation.

SYSTEM DESIGN AND DEVELOPMENT

An important aspect of system design is the design of data storage structure. To begin with a logical model of data structure is developed first. A database is a container object which contains tables, queries, reports and data validation policies enforcement rules or constraint's etc. A logical data often represented as a records are kept in different tables after reducing anomalies and redundancies. The goodness of data base design lies in the table structure and its relationship.

- *This software project maintains a database named Indian Premier League which contains the following tables.*

The database of Indian Premier League Delivery contains 3 tables. The tables are normalized to minimize the redundancies of data and enforcing the validation rules of the organization. Most of the tables are designed to store master records. The tables and their structure are given below.

Keeps the records of the all the audiences

```
mysql> desc stadium;
```

Field	Type	Null	Key	Default	Extra
Aadhar_card	varchar(23)	NO	PRI		
Name	varchar(54)	YES		NULL	
mob_no	varchar(23)	YES		NULL	
Seat_Type	varchar(12)	YES		NULL	
Date_Of_Match	date	YES		NULL	
Venue	varchar(23)	YES		NULL	
Team_1	varchar(54)	YES		NULL	
Team_2	varchar(43)	YES		NULL	
Extra_Facilites	varchar(34)	YES		NULL	
Seat_Fees	int(10)	YES		NULL	
Extra_Facilites_Bill	int(43)	YES		NULL	
Tatal_Fees	int(12)	YES		NULL	
Payment_type	varchar(34)	NO		NULL	
Other_offer	varchar(12)	YES		NULL	

14 rows in set (0.36 sec)

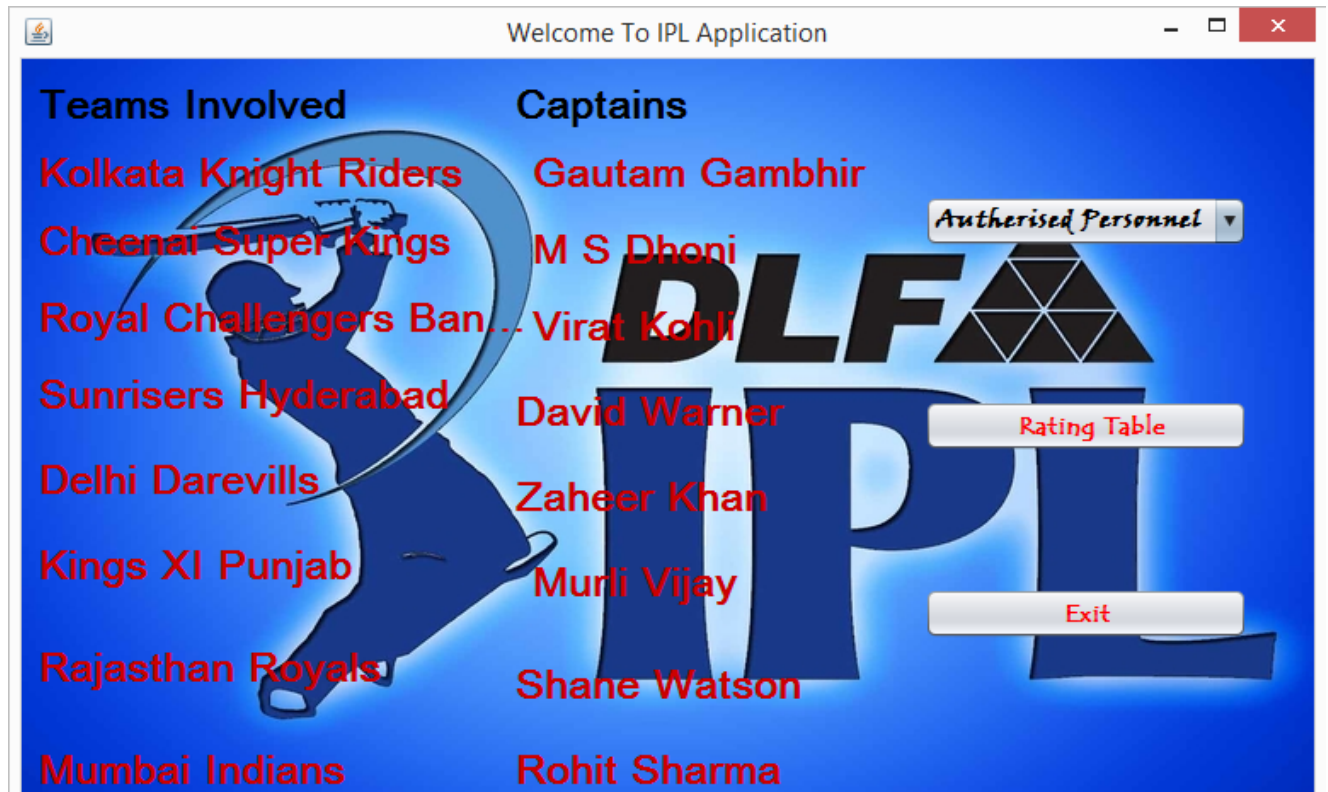
Keeps all the records of the followers

```
mysql> desc follower;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Team  | varchar(39)   | YES  |     | NULL    |       |
| Name  | varchar(23)   | YES  |     | NULL    |       |
| Contact | varchar(34)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.05 sec)
```

Keeps all the records of the schedule

```
mysql> desc schedule;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Match_No       | varchar(2)    | NO   | PRI |         |       |
| dateofmatch    | varchar(13)   | YES  |     | NULL    |       |
| Time           | varchar(13)   | YES  |     | NULL    |       |
| Team_1         | varchar(50)   | YES  |     | NULL    |       |
| Captain_Team1  | varchar(50)   | YES  |     | NULL    |       |
| Team_2         | varchar(50)   | YES  |     | NULL    |       |
| Captain_Team2  | varchar(49)   | YES  |     | NULL    |       |
| Venue          | varchar(14)   | YES  |     | NULL    |       |
| Stadium_Name   | varchar(34)   | YES  |     | NULL    |       |
| Chief_Guest    | varchar(23)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.03 sec)
```

Home Page(Home.java)



```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(jComboBox1.getSelectedIndex()==0)  
    {  
        new Security().setVisible(true);  
        dispose();  
    }  
    if(jComboBox1.getSelectedIndex()==1)  
    {  
        new GeneralPeoples().setVisible(true);  
        dispose();  
    }  
}
```


Coding For Rating Table Button:-

// TODO add your handling code here:

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

```
dispose();
```

Coding For Exit Button:-

// TODO add your handling code here:

```
System.exit(0);
```

Rating Table(RatingTable.java)



```
public RatingTable() {
```

```
    initComponents();
```

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

```
    int rows=model.getRowCount();
```

```
    if(rows>0)
```

```
{
```

```

        model.removeRow(0);
    }
    jTable1.getModel();
    try{
        Class.forName("java.sql.DriverManager");
        Connection con= (Connection)
        DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");
        Statement stmt= (Statement) con.createStatement();
        String query="Select Team,Count(Team) from follower group by team order by count(Team)
desc;";
        ResultSet rs=stmt.executeQuery(query);
        while(rs.next())
        { String team=rs.getString("Team");
            String name=rs.getString("Count(Team)");
            model.addRow(new Object[]
                {team,name,});    }
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,e.getMessage());
        }
    }

```

Coding For Back Button:-

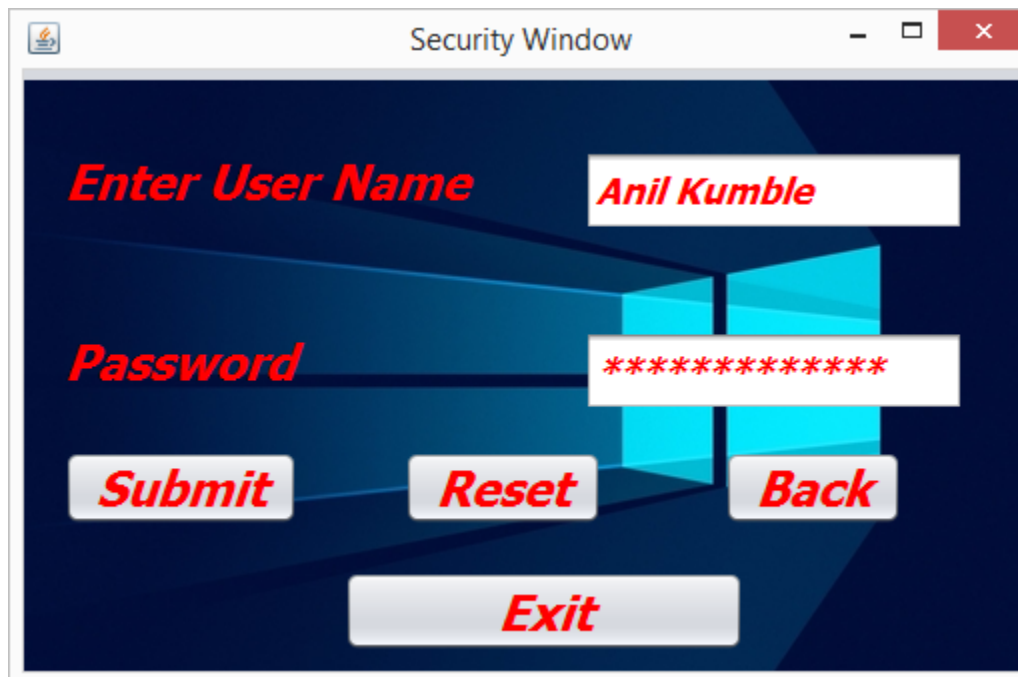
// TODO add your handling code here:

```

new Home().setVisible(true);
dispose();

```

Security Window(Security.java)



Coding For Submit Button:-

// TODO add your handling code here:

```
String name=jTextField2.getText();
String Password=jPasswordField1.getText();
if(name.isEmpty())
{
    JOptionPane.showMessageDialog(null,"Enter User Name");
}
else
    if(Password.isEmpty())
    {
        JOptionPane.showMessageDialog(null,"Enter Password");
    }
```

```
        else{  
            if(jPasswordField1.getText().equals("Vidyagyan@123"))  
            {  
                new Authentic().setVisible(true);  
                dispose();  
            }  
            else{  
                jTextField2.setText("");  
                jPasswordField1.setText("");  
                JOptionPane.showMessageDialog(null,"Wrong Password,Try Again");  
            }  
        }
```

Coding For Clear Button:-

// TODO add your handling code here:

```
jTextField2.setText("");  
jPasswordField1.setText("");
```

Coding For Exit Button:-

// TODO add your handling code here:

```
System.exit(0);
```

Coding For Back Button:-

// TODO add your handling code here:

```
new Home().setVisible(true);  
dispose();
```

Authorised Peoples (Authentic.java)



Coding For Edit Schedule:-

```
new schedule().setVisible(true);  
dispose();
```

Coding For Cancel Any Ticket:-

```
new delete().setVisible(true);  
dispose();
```

Coding For Advertisement Button:-

```
new Advertisement().setVisible(true);  
dispose();
```

Coding For Show All Booking Button:-

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

```
dispose();
```


Coding For Back Button:-

```
// TODO add your handling code here:
```

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

```
dispose();
```

All The Bookings Till Now (ShowAllBooking.java)



Aadhar_No.	Name	Mobile	Seat_Type	Date_of_Mat...	Venue	Team_1	Team_2	Extra_Faciliti...	Seat_Fees	Extra_Faciliti...	Total_Fees	Payment_Ty...	Other_Offer
255645823...	Krishna Yad...	8596956458	VIP Box	2018-04-28	Mumbai	Mumbai Indi...	Delhi Darevi...	Coldrink	10000	100	10100	Cash	No
654782569...	Ankit Singh	8565748532	Front	2018-04-20	Kolkata	Kolkata Knig...	Royal Chall...	Lunch	5000	600	5600	PayTM	No
759865324...	Bhaumik Ch...	9568324857	Middle	2018-04-21	Mohali	Kings XI Pu...	Sunrisers H...	Lunch	4000	600	4600	Rupaay	Coldrink
856324598...	Arvind Chau...	7895632456	Front	2018-04-23	Cheenal	Cheenal Su...	Kolkata Knig...	Popcorn	5000	50	5050	ATM	Coldrink
8569548756	Surdeep Ku...	7895683265	Middle	2018-04-20	Mohali	Kolkata Knig...	Delhi Darevi...	Snacks	4000	200	4200	PayTM	No
862820316...	Kunwar Bah...	9568327048	VIP Box	2018-04-18	Banglore	Royal Chall...	Sunrisers H...	Lunch	10000	600	10600	NetBanking	No Offer
869532457...	Aakash Ku...	9856327458	Ground	2018-04-21	Hyderabad	Royal Chall...	Kolkata Knig...	Lunch	2000	600	2600	Cash	No
986574521...	Amandeep	8569324586	Ground	2018-04-17	Banglore	Mumbai Indi...	Royal Chall...	Lunch	2000	600	2600	PayTM	No

Back

```
public ShowAllBooking() {
```

```
    initComponents();
```

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

```
    int rows=model.getRowCount();
```

```
    while(rows>0)
```

```
    {
```

```
        model.removeRow(0);
```

```
        rows--; }  
jTable1.getModel();
```

```
try{
```

```
    Class.forName("java.sql.DriverManager");
```

```
    Connection con= (Connection)
```

```
    DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");
```

```

Statement stmt= (Statement) con.createStatement();

String query="Select * from Stadium;";

ResultSet rs=stmt.executeQuery(query);

while(rs.next())
{
String aadharno=rs.getString("Aadhar_card");

    String Name=rs.getString("Name");

    String mobno=rs.getString("Mob_no");

    String st=rs.getString("Seat_type");

    String dateofmatch=rs.getString("Date_of_match");

    String venue=rs.getString("Venue");

    String team1=rs.getString("Team_1");

    String team2=rs.getString("Team_2");

    String extrafacilities=rs.getString("Extra_Facilites");

    int seatfees=Integer.parseInt(rs.getString("Seat_Fees"));

    int Extb=Integer.parseInt(rs.getString("Extra_Facilites_Bill"));

    int tf=Integer.parseInt(rs.getString("Tatal_Fees"));

    String pt=rs.getString("Payment_Type");

    String otheroffer=rs.getString("Other_Offer");

    model.addRow(new Object[]

        {aadharno,Name,mobno,st,dateofmatch,venue,team1,team2,extrafacilities,seatfees,
Extb,tf,pt,otheroffer});

}}

catch(Exception e)

{

    JOptionPane.showMessageDialog(null,e.getMessage()); }

```

Coding For Back Button:-

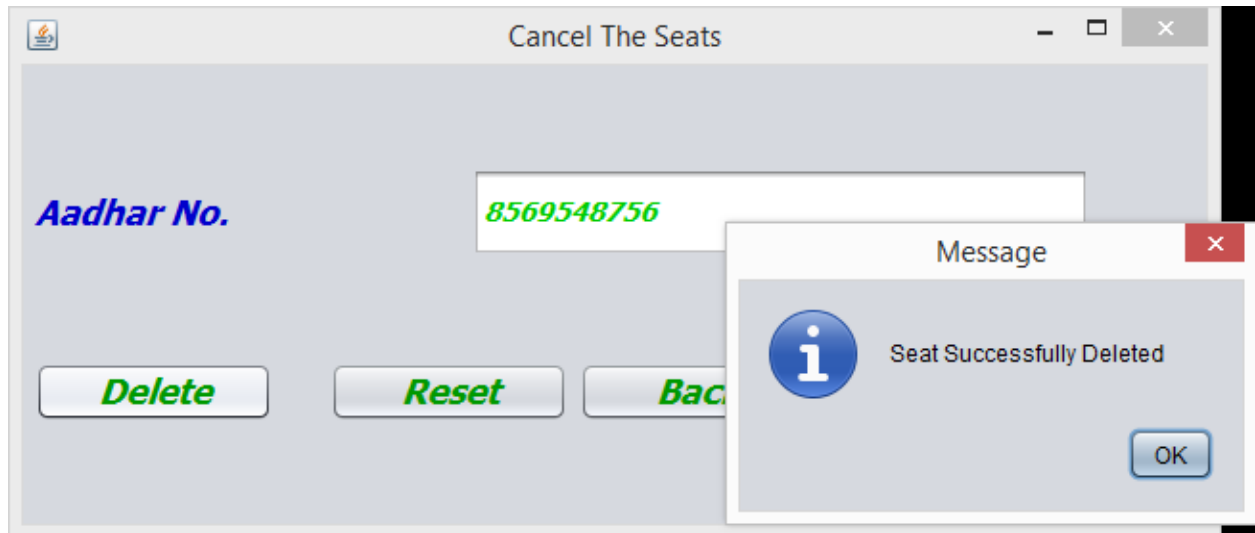
```

new Authentic().setVisible(true);

dispose();

```

Cancel The Seats(delete.java)



Coding For Delete Button:-

// TODO add your handling code here:

```
String aadhar=jTextField1.getText();
```

```
if(aadhar.isEmpty())
```

```
{
```

```
    JOptionPane.showMessageDialog(null,"Enter Aadhar Number");
```

```
}
```

```
else{
```

```
try{
```

```
    Class.forName("java.sql.DriverManager");
```

```
    Connection con= (Connection)
```

```
    DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");
```

```
    Statement stmt= (Statement) con.createStatement();
```

```
    String query="DELETE FROM STADIUM WHERE Aadhar_Card='"+aadhar+"'";
```

```
    stmt.executeUpdate(query);
```

```
    JOptionPane.showMessageDialog(null,"Seat Successfully Deleted");
```



```
}  
  
catch(Exception e)  
{  
    JOptionPane.showMessageDialog(null,e.getMessage());  
}  
}
```

Coding For Reset Button:-

```
// TODO add your handling code here:  
    jTextField1.setText("");  
}
```

Coding For Back Button:-

```
// TODO add your handling code here:  
    new Authentic().setVisible(true);  
    dispose();  
}
```

Coding For Exit Button:-

```
// TODO add your handling code here:  
    System.exit(0);  
}
```

Add New Schedule(Schedule.java)

The screenshot shows a Java Swing window titled "Schedule" with a light gray background. It contains several text input fields and two dropdown menus. The fields are arranged in two columns. The left column contains: "Match_no" (09), "Date" (26-04-2018), "Team1" (Delhi Darevills), "Team 2" (Royal Challengers Bangalore), and "Venue" (Delhi). The right column contains: "Cheif Guest" (Ricki Ponting), "Time" (8:00), "Captain" (Zaheer Khan), "Captain" (Virat Kohli), and "Stadium Name" (Firoj Shah Kotla). At the bottom of the window are three buttons: "Add", "Reset", and "Exit". Overlaid on the bottom center of the window is a "Message" dialog box with a blue information icon and the text "Match Schedule Successfully added". It has an "OK" button at the bottom right.

Match_no	09	Cheif Guest	Ricki Ponting
Date	26-04-2018	Time	8:00
Team1	Delhi Darevills	Captain	Zaheer Khan
Team 2	Royal Challengers Bangalore	Captain	Virat Kohli
Venue	Delhi	Stadium Name	Firoj Shah Kotla

Buttons: Add, Reset, Exit

Message: Match Schedule Successfully added

Coding For Add Button:-

```
String Match_no=jTextField1.getText();  
    String date=jTextField2.getText();  
    String team_1=(String) jComboBox1.getSelectedItem();  
    String team_2=(String) jComboBox2.getSelectedItem();  
    String venue=jTextField5.getText();  
    String stadiumname=jTextField6.getText();  
    String captain_1=jTextField7.getText();  
    String captain_2=jTextField8.getText();  
    String time=jTextField10.getText();  
    String cheifguest=jTextField9.getText();  
    try{
```

```

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

Connection con= (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");

Statement stmt= (Statement) con.createStatement();

String query="Insert into schedule
values('"+Match_no+"','"+date+"','"+time+"','"+team_1+"','"+captain_1+"','"+team_2+"','"+captain_2+"','"+venue+"','"+stadiumname+"','"+cheifguest+"');";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,"Match Schedule Successfully added"); }

catch(Exception e){

JOptionPane.showMessageDialog(null,e.getMessage()); }

```

Coding For Reset Button:-

```

jTextField1.setText("");

jTextField2.setText("");

jTextField5.setText("");

jTextField6.setText("");

jTextField7.setText("");

jTextField8.setText("");

jTextField9.setText("");

jTextField10.setText("");

jComboBox1.setSelectedIndex(0);

jComboBox2.setSelectedIndex(0);

```

Coding For Back Button:-

```

new Authentic().setVisible(true);

dispose();

```

Coding For Exit Button:-

```

System.exit(0);

```

Advertisements(Advertisement.java)

The screenshot shows a Java Swing window titled "Advertisements". Inside the window, there are four text input fields with labels in red italics: "Company Code" (value: 2654), "Company Name" (value: Nokia), "Advertisement Logo" (value: NOKIA), and "Area (cm)" (value: 5694). Below the fields are four yellow buttons with green text: "Submit", "Clear", "Back", and "Exit". In the foreground, a "Message" dialog box is open, displaying an information icon and the text "Request Submitted" with an "OK" button.

Coding For Submit Button:-

```
// TODO add your handling code here:  
  
int code=0,area=0;  
  
code=Integer.parseInt(jTextField2.getText());  
  
String name=jTextField3.getText();  
  
String clogo=jTextField4.getText();  
  
area=Integer.parseInt(jTextField1.getText());  
  
int price=area*10000;
```

```

if(code==0)
{
    JOptionPane.showMessageDialog(null,"Enter Company Code");
}
else
if(name.isEmpty())
{
    JOptionPane.showMessageDialog(null,"Enter Company Name");
}
else
if(clogo.isEmpty())
{
    JOptionPane.showMessageDialog(null,"Enter Company Logo");
}
else
if(area==0)
{
    JOptionPane.showMessageDialog(null,"Enter Area to Occupy");
}
else
{
    try{
        Class.forName("java.sql.DriverManager");
        java.sql.Connection con=(java.sql.Connection)
            DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");
        java.sql.Statement stmt=(java.sql.Statement)con.createStatement();

        String query="INSERT INTO company
VALUES("+code+", '"+name+"', '"+clogo+"', '"+area+"', '"+price+"');";

        stmt.executeUpdate(query);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

```
        JOptionPane.showMessageDialog(null,"Request Submitted");
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e.getMessage());
    }
}
```

Coding For Clear Button:-

// TODO add your handling code here:

```
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
```

Coding For Back Button:-

// TODO add your handling code here:

```
new Authentic().setVisible(true);
dispose();
```

Coding For Exit Button:-

// TODO add your handling code here:

```
System.exit(0);
```

General Peoples(GeneralPeoples.java)



Coding For Book Your Seat Button:-

```
// TODO add your handling code here:  
  
    new Bookyourseat().setVisible(true);  
  
    dispose();
```

Coding For Follow Any Team Button:-

```
// TODO add your handling code here:  
  
    new Follower().setVisible(true);  
  
    dispose();
```

Coding Modify Your Seat Type Back Button:-

```
// TODO add your handling code here:  
  
    new Modify().setVisible(true);  
  
    dispose();
```

Coding For Show Schedule Button:-

// TODO add your handling code here:

```
new ShowSchedule().setVisible(true);  
  
dispose();
```

Coding For Back Button:-

// TODO add your handling code here:

```
new ShowFollowers().setVisible(true);  
  
dispose();
```

Coding For Seach Your Seat Button:-

// TODO add your handling code here:

```
new Searchyourseat().setVisible(true);  
  
dispose();
```

Coding For Back Button:-

// TODO add your handling code here:

```
new FirstPage().setVisible(true);  
  
dispose();
```


Book Your Seat(BookYourSeat.java)

The screenshot shows a Java Swing window titled "Book Your Seats". It contains the following fields and controls:

- Aadhar No.:** Text field with value "986532748596".
- Name:** Text field with value "Shubham Chauhan".
- Mob No.:** Text field with value "9758969654".
- Seat Type:** Dropdown menu with "Middle" selected.
- Date Of Match:** Dropdown menu with "2018-04-29" selected.
- Venue:** Dropdown menu with "Bangalore" selected.
- Team 1:** Dropdown menu with "Cheenai Super King" selected.
- Team 2:** Dropdown menu with "Royal Challengers Bangalore" selected.
- Extra Facilities:** Dropdown menu with "Lunch" selected.
- Payment Type:** Dropdown menu with "NetBanking" selected.
- Other Offer:** Text field with value "No".
- Buttons:** "Submit", "Reset", "Back", and "Exit" (all in yellow boxes with green text).

A "Message" dialog box is open in the bottom right corner, displaying an information icon and the text "Seat Successfully registered", with an "OK" button.

Coding For Submit Button:-

// TODO add your handling code here:\

```
String Aadhar_no=jTextField1.getText();

String Name=jTextField2.getText();

String Mobile=jTextField3.getText();

String seattype=(String) jComboBox1.getSelectedItem();

String Dateofmatch=(String) jComboBox2.getSelectedItem();

String Venue=(String) jComboBox3.getSelectedItem();

String team1=(String) jComboBox4.getSelectedItem();

String team2=(String) jComboBox5.getSelectedItem();

String extrafacilities=(String) jComboBox6.getSelectedItem();

String PaymentType=(String) jComboBox7.getSelectedItem();

int seatfees=0;

String OtherOffer=jTextField4.getText();
```

```
if(Aadhar_no.isEmpty())
{
    JOptionPane.showMessageDialog(null,"Enter Aadhar Card Number");
}
else
    if(Name.isEmpty())
    {
        JOptionPane.showMessageDialog(null,"Enter Your Name");
    }
if(Mobile.isEmpty())
{
    JOptionPane.showMessageDialog(null,"Enter Mobile Number");
}
else
    if(OtherOffer.isEmpty())
    {
        JOptionPane.showMessageDialog(null,"Enter Other Offer");
    }
System.out.print(seattype);

if(seattype.equals("Front"))
{
    seatfees=5000;

}
else
if(seattype.equals("Middle"))
{
    seatfees=4000;
```

```
}  
  
else  
  
if(seattype.equals("VIP Box"))  
{  
    seatfees=10000;  
}  
  
else  
  
{  
    seatfees=2000;  
}  
  
int extrafacilitiesbill=0;  
if(extrafacilities.equals("Snacks"))  
{  
    extrafacilitiesbill=200;  
}  
else  
if(extrafacilities.equals("Lunch"))  
{  
    extrafacilitiesbill=600;  
}  
else  
if(extrafacilities.equals("Popcorn"))  
{  
    extrafacilitiesbill=50;  
}
```

```

else

{
    extrafacilitiesbill=100;
}

int Totalbill=seatfees+extrafacilitiesbill;

try
{
    Class.forName("java.sql.DriverManager");
    Connection con= (Connection)
    DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");
    Statement stmt= (Statement) con.createStatement();

    String query="INSERT INTO stadium
VALUES('"+Aadhar_no+"','"+Name+"','"+Mobile+"','"+seattype+"','"+Dateofmatch+"','"+Venue+"','"+tea
m1+"','"+team2+"','"+extrafacilities+"','"+seatfees+"','"+extrafacilitiesbill+"','"+Totalbill+"','"+PaymentType+"
','"+OtherOffer+"')";

    stmt.executeUpdate(query);

    JOptionPane.showMessageDialog(null,"Seat Successfully registered");

}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e.getMessage());
    JOptionPane.showMessageDialog(null,"Seat is not Successfully registered");
}

}

```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    jTextField1.setText("");  
    jTextField2.setText("");  
    jTextField3.setText("");  
    jTextField4.setText("");  
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

Coding Back Button:-

```
    // TODO add your handling code here:  
    new GeneralPeoples().setVisible(true);  
    dispose();  
}
```

Coding For Exit Button:-

```
// TODO add your handling code here:  
    System.exit(0);  
}
```

Search your Seat(SearchYourSeat.java)

Aadhar_No.	Name	Mobile	Seat_Type	Date_of_Mat...	Venue	Team_1	Team_2	Extra_Faciliti...	Seat_Fees	Extra_Faciliti...	Total_Fees	Payment_Ty...	Other_Offer
862820316097	Kunwar Baha...	9568327048	VIP Box	2018-04-18	Banglore	Royal Challen...	Sunrisers Hyd...	Lunch	10000	600	10600	NetBanking	No Offer

Coding For Search Button:-

// TODO add your handling code here:

```
String aadhar=jTextField1.getText();
```

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

```
int rows=model.getRowCount();
```

```
while(rows>0)
```

```
{
```

Coding For Reset Button:-

// TODO add your handling code here:

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

```
int rows=model.getRowCount();
```

```
if(rows>0)
```

```
{
```

```
model.removeRow(0);
```

```
}
```

```

    }

    model.removeRow(0);

    rows--;

    }

    jTable1.getModel();

    if(aadhar.isEmpty())

    {

        JOptionPane.showMessageDialog(null,"Enter Aadhar Number");

    }

    else

    {

try{

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

    Connection con= (Connection)

    DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");

    Statement stmt= (Statement) con.createStatement();

    String query="Select
Aadhar_card,Name,Mob_no,Seat_type,date_of_match,Venue,team_1,Team_2,extra_facilites,seat_fees,e
xtra_facilites_bill,Tatal_fees,Payment_Type,other_offer from stadium where Aadhar_card="+aadhar+"";

    ResultSet rs=stmt.executeQuery(query);

    while(rs.next())

    {

        String aadharno=rs.getString("Aadhar_card");

        String Name=rs.getString("Name");

        String mobno=rs.getString("Mob_no");

        String st=rs.getString("Seat_type");

        String dateofmatch=rs.getString("Date_of_match");

        String venue=rs.getString("Venue");

```

```

String team1=rs.getString("Team_1");
String team2=rs.getString("Team_2");
String extrafacilities=rs.getString("Extra_Facilites");
int seatfees=Integer.parseInt(rs.getString("Seat_Fees"));
int Extb=Integer.parseInt(rs.getString("Extra_Facilites_Bill"));
int tf=Integer.parseInt(rs.getString("Tatal_Fees"));
String pt=rs.getString("Payment_Type");
String otheroffer=rs.getString("Other_Offer");
model.addRow(new Object[]
                {aadhar,Name,mobno,st,dateofmatch,venue,team1,team2,extrafacilities,seatfees,
Extb,tf,pt,otheroffer});

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

```

Coding For Back Button:-

```

// TODO add your handling code here:
new GeneralPeoples().setVisible(true);
dispose();

```

Coding For Exit Button:-

```

// TODO add your handling code here:
System.exit(0);

```


Schedule(View Schedule.java)

Match No.	Date Of Match	Time	Team 1	Captain	Team 2	Captain	Venue	Stadium Name	Chief Guest
01	16/04/2017	8:00	Royal Challengers ...	Virat Kohli	Kolkata Knight Ride...	Gautam Gambhir	Kolkata	Eden Garden	V V S Lakshman
02	17/04/2017	8:00	Mumbai Indians	Rohit Sharma	Kolkata Knight Ride...	Gautam Gambhir	Kolkata	Eden Garden	V V S Lakshman
03	18/04/2017	8:00 pm	Cheennai Super King	Ms Dhoni	Kings XI Punjab	Murli Vijay	Mohali	DGT	Shane Warne
04	19/04/2017	8:00 pm	Sunrisers Hyderabad	David Warner	Royal Challengers ...	Virat Kohli	Hyderabad	DGT	Ricki Ponting
05	19/04/2017	8:00 pm	Rajasthan Royals	Shane Watson	Cheennai Super King	M S Dhoni	Ahmedabad	Sardar Patel Stadium	Ricki Ponting
06	15/04/2017	8:00	Cheennai Super King	Dhoni	Kings XI Punjab	Vijay	Cheennai	RK Stadium	Aadarsh
07	20/04/2017	8:00 pm	Delhi Darevills	Zaheer Khan	Kolkata Knight Ride...	Gautam Gambhir	Delhi	Feroz Shah Kotla	Arvind Kejriwal
09	26-04-2018	8:00	Delhi Darevills	Zaheer Khan	Royal Challengers ...	Virat Kohli	Delhi	Firoz Shah Kotla	Ricki Ponting
8	15/04/2018	8:00	Cheennai Super King	Virat	Royal Challengers ...	Dhoni	Ban	Chinnaswami	ST

[Back](#)

Coding For Show Schedule Button:-

```
public ShowSchedule() {
```

```
    initComponents();
```

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

```
    int rows=model.getRowCount();
```

```
    if(rows>0)
```

```
    {
```

```
        model.removeRow(0);
```

```
    }
```

```
    jTable1.getModel();
```

```
    try{
```

```
        Class.forName("java.sql.DriverManager");
```

```
        Connection con= (Connection)
```

```
        DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");
```

```

Statement stmt= (Statement) con.createStatement();

String query="Select * From schedule;";

ResultSet rs=stmt.executeQuery(query);

while(rs.next())
{

    String a=rs.getString("Match_No");

    String b=rs.getString("dateofmatch");

    String c=rs.getString("Time");

    String d=rs.getString("Team_1");

    String e=rs.getString("Captain_Team1");

    String f=rs.getString("Team_2");

    String g=rs.getString("Captain_Team2");

    String h=rs.getString("Venue");

    String i=rs.getString("Stadium_Name");

    String j=rs.getString("Chief_Guest");

    model.addRow(new Object[]

        {a,b,c,d,e,f,g,h,i,j});

    }

}

catch(Exception e)

{

    JOptionPane.showMessageDialog(null,e.getMessage());

}

}

```

Coding For Back Button:-

```

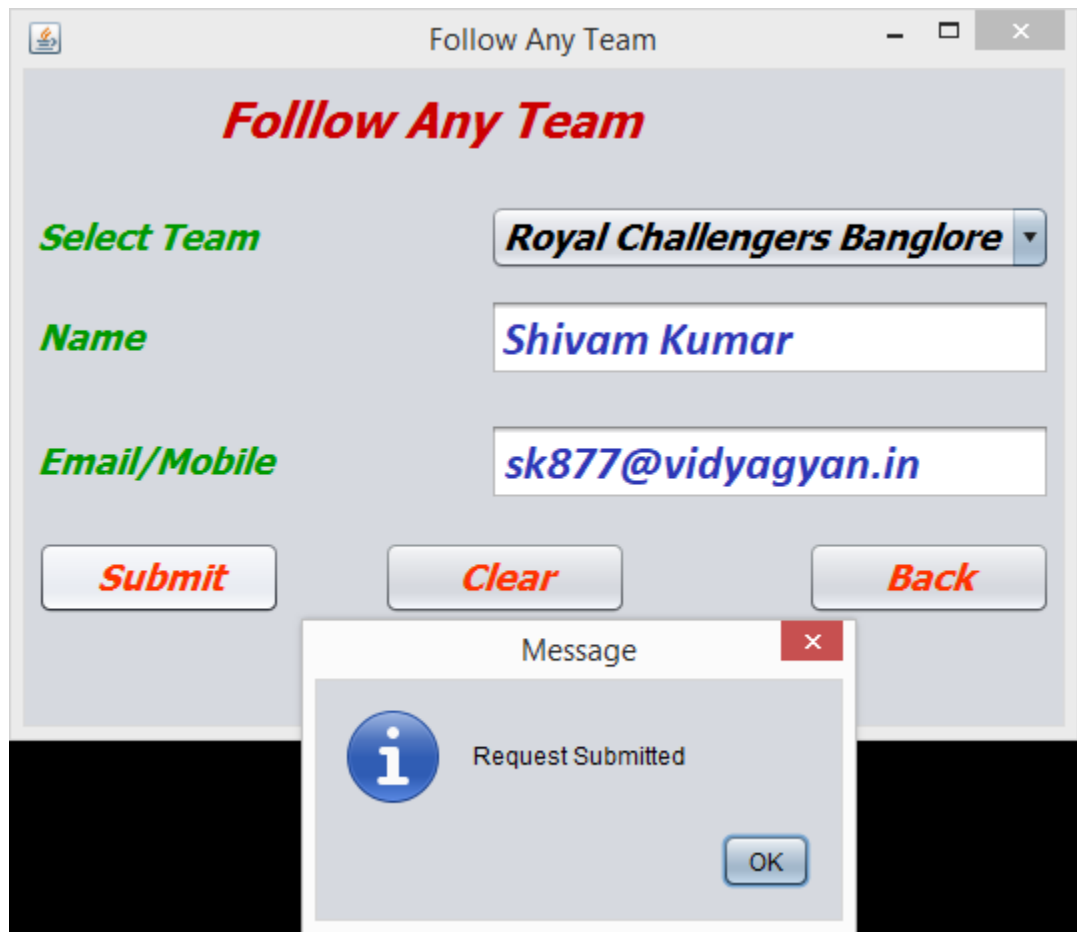
// TODO add your handling code here:

new GeneralPeoples().setVisible(true);

```

dispose();

Follow Any Team(Follower.java)



Coding For Submit Button:-

// TODO add your handling code here:

```
String team=(String) jComboBox1.getSelectedItem();  
String name=jTextField1.getText();  
String contact=jTextField2.getText();  
if(name.isEmpty())  
{  
    JOptionPane.showMessageDialog(null,"Enter Name");  
}
```

```

else
    if(contact.isEmpty())
{
    JOptionPane.showMessageDialog(null,"Enter Any Contact");
}
else
{
try{
    Class.forName("java.sql.DriverManager");
    java.sql.Connection con=(java.sql.Connection)
        DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");
    java.sql.Statement stmt=(java.sql.Statement)con.createStatement();
    String query="INSERT INTO FOLLOWER VALUES('"+team+"','"+name+"','"+contact+"');";
    stmt.executeUpdate(query);
    JOptionPane.showMessageDialog(null,"Request Submitted");
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e.getMessage());
}
}

```

Coding For Clear Button:-

// TODO add your handling code here:

```
jTextField1.setText("");
```

```
jTextField2.setText("");
```

Coding For Back Button:-

// TODO add your handling code here:

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

```
dispose();
```

Coding For Exit Button:-

// TODO add your handling code here:

System.exit(0);

Show Followers(Follower.java)

Followers		
Team	Name	Contact
Cheennai Super King	Vishal	9865327456
Cheennai Super King	Surdeep	986532278
Cheennai Super King	Surdeep Kumar	surdeep988@gmail.com
Cheennai Super King	Deepak	dy76786@gmail.com
Delhi Darevills	Bhaumik	9568655545
Kings XI Punjab	Sunny	sunnykumargill@gmail.com
Kings XI Punjab	saurabh	9568476775
Kolkata Knight Riders	Kunwar Bahadur	kb3349@vidyagyan.in
Kolkata Knight Riders	Kunwar Bahadur	kb3349@vidyagyan.in
Mumbai Indians	Deepak	956832748
Mumbai Indians	Vishal	986532145
Rajasthan Royals	Sachin	sachin234@gmail.com
Royal Challengers Bangalore	Sara	sa123@gmail.com
Royal Challengers Bangalore	Kunwar Bahadur	kb3349@vidyagyan.in
Royal Challengers Bangalore	Kunwar	95683217042
Sunrisers Hyderabad	Ankit	an755@gmail.com

[Back](#)

```
public ShowFollowers() {
```

```
initComponents();
```

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

```
int rows=model.getRowCount();
```

```
if(rows>0)
```

```
{
```

```
model.removeRow(0);
```

```
}
```

```
jTable1.getModel();
```

```

try{
    Class.forName("java.sql.DriverManager");
    Connection con= (Connection)
    DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");
    Statement stmt= (Statement) con.createStatement();
    String query="Select * From Follower order by team asc;";
    ResultSet rs=stmt.executeQuery(query);
    while(rs.next())
    {

        String team=rs.getString("Team");
        String name=rs.getString("Name");
        String contact=rs.getString("Contact");
        model.addRow(new Object[]
            {team,name,contact});
    }
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e.getMessage());
}
}

```

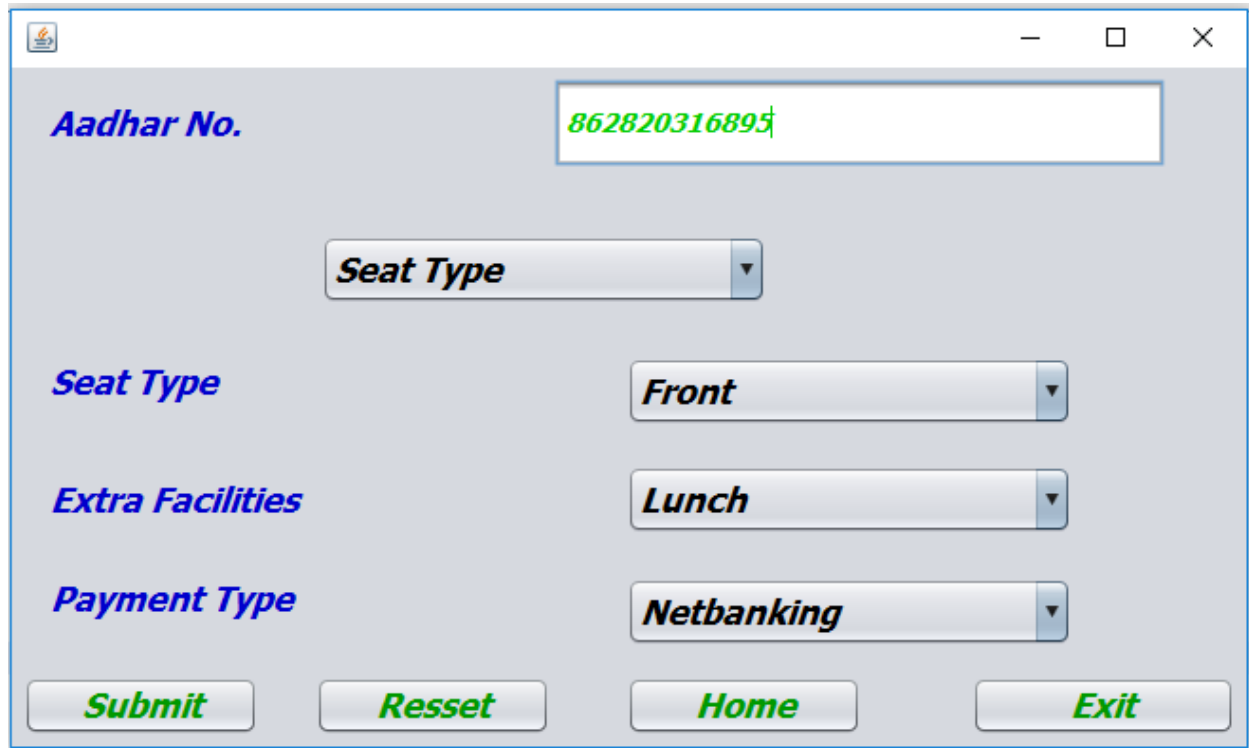
Coding For BackButton:-

```

// TODO add your handling code here:
    new GeneralPeoples().setVisible(true);
    dispose();
}
}

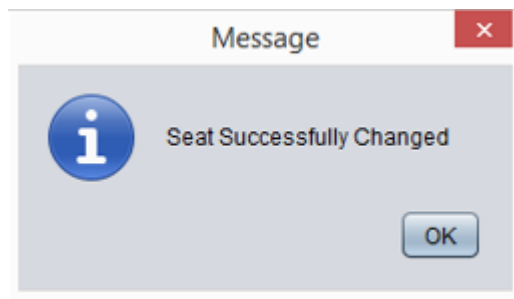
```

Modify Your Details(Modify.java)



The screenshot shows a Java application window titled "Modify Your Details(Modify.java)". The window has a light gray background and standard Windows window controls (minimize, maximize, close) in the top right corner. The main content area contains the following elements:

- Aadhar No.:** A text field containing the value "862820316895" in green text.
- Seat Type:** A dropdown menu with "Seat Type" as the label and "Front" as the selected option.
- Extra Facilities:** A dropdown menu with "Lunch" as the selected option.
- Payment Type:** A dropdown menu with "Netbanking" as the selected option.
- Buttons:** Four buttons at the bottom: "Submit", "Reset", "Home", and "Exit", all with green text.



Coding For Submit Button:-

```
// TODO add your handling code here:  
String adhar=jTextField1.getText();  
if(jComboBox1.getSelectedIndex()==0)  
{  
    jLabel3.setVisible(false);
```

```

jLabel4.setVisible(false);

jComboBox4.setVisible(false);

jComboBox3.setVisible(false);

String st=(String) jComboBox2.getSelectedItem();

try{

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

    Connection con= (Connection)

    DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");

    Statement stmt= (Statement) con.createStatement();

    String query="Update stadium set Seat_Type='"+st+"' where Aadhar_Card="+adhar+"";

    stmt.executeUpdate(query);

    JOptionPane.showMessageDialog(null,"Seat Successfully Changed");

}

catch(Exception e)

{

    JOptionPane.showMessageDialog(this,e.getMessage());

}

}

if(jComboBox1.getSelectedIndex()==1)

{
jLabel2.setVisible(false);

jLabel4.setVisible(false);

jComboBox4.setVisible(false);

jComboBox2.setVisible(false);


String ef=(String) jComboBox3.getSelectedItem();

try{

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

    Connection con= (Connection)

    DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");

```



```

Statement stmt= (Statement) con.createStatement();

String query="Update stadium set Extra_Facilites='"+ef+"' where Aadhar_Card="+adhar+"";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,"Seat Successfully Changed");
}

catch(Exception e)

{

    JOptionPane.showMessageDialog(this,e.getMessage());

}

}

if(jComboBox1.getSelectedIndex()==2)

{

String pt=(String) jComboBox4.getSelectedItem();

try{

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

    Connection con= (Connection)

    DriverManager.getConnection("jdbc:mysql://localhost:3306/kunwar","root","1234");

    Statement stmt= (Statement) con.createStatement();

    String query="Update stadium set Payment_type='"+pt+"' where Aadhar_Card="+adhar+"";

    stmt.executeUpdate(query);

    JOptionPane.showMessageDialog(null,"Seat Successfully Changed");

}

catch(Exception e)

{

    JOptionPane.showMessageDialog(this,e.getMessage());

}

}

```

Coding For Reset Button:-

// TODO add your handling code here:

```
jComboBox1.setSelectedIndex(0);
```

```
jComboBox2.setSelectedIndex(1);
```

```
jComboBox3.setSelectedIndex(2);
```

```
jComboBox4.setSelectedIndex(3);
```

Coding For Exit Button:-

// TODO add your handling code here:

```
System.exit(0);
```

Coding For BackButton:-

// TODO add your handling code here:

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

```
dispose();
```