JAVA AWT BASED – CRICKET WORLD CUP -SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

G.ANISH <1602-19-737-003>

M.RAKESH<1602-19-737-031>

N.ROHITH KUMAR <1602-19-737-033>

Under the guidance of Ms. S. Sreelakshmi

Department of Information Technology
Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31

BONAFIDE CERTIFICATE

This is to certify that this project report titled ‘Cricket World Cup’ is a project work by Mr. G.Anish and Mr. M.Rakesh and Mr.N.Rohith kumar bearing roll nos. 1602-19-737-003 and 1602-19-737-031 and 1602-19-737-033 who carried out the project under my supervision in the IV semester for the academic year 2020-2021.

Signature	Signature
Internal Examiner	External Examiner

ABSTRACT

This project on Cricket World Cup provides various information about the various information about the various teams participating in the various cricket tournaments,in which all the major countries participate. It also provides us with information about the various players participating in the tournament.The database contains details of players,coaches,and umpires among others.All the useful information about the entire world cup can be found here.

G.Anish <1602-19-737-003>

M.Rakesh <1602-19-737-031>

N.Rohith kumar<1602-19-737-033>

INTRODUCTION

REQUIREMENTS

TABLE	ATTRIBUTES
Teams	team_rank number(5), country_name varchar2(20), no_of_wins number(5), no_of_loses number(5);
Umpire	Umpire_id number(5), Umpire_name varchar2(20), No_of_matches number(5), Country varchar2(5),
Player	Player_name varchar2(20), No_of_matches number(5), Batting_average number(5,2), Runsscored number(5), Wicketstake number(5), Economy number(5,2);
Match	Match_number number(5), Stadium varchar2(20), Team_name1 varchar2(20), Team_name2 varchar2(20), Winner varchar2(20), Loser varchar2(20);

AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based on Cricket World Cup which shows us the information related to player's information, player's statistics, list of all teams etc. These values are to be updated in the database using JDBCconnectivity.

ARCHITECTURE AND TECHNOLOGY

Software Requirement Specifications:

Operating System Front End Back End Server Documentation: Windows 10

Frontend Software: Java NetBeans 12.0: JDK 14

Backend Software: SQL Developer

Java AWT:

Java AWT (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

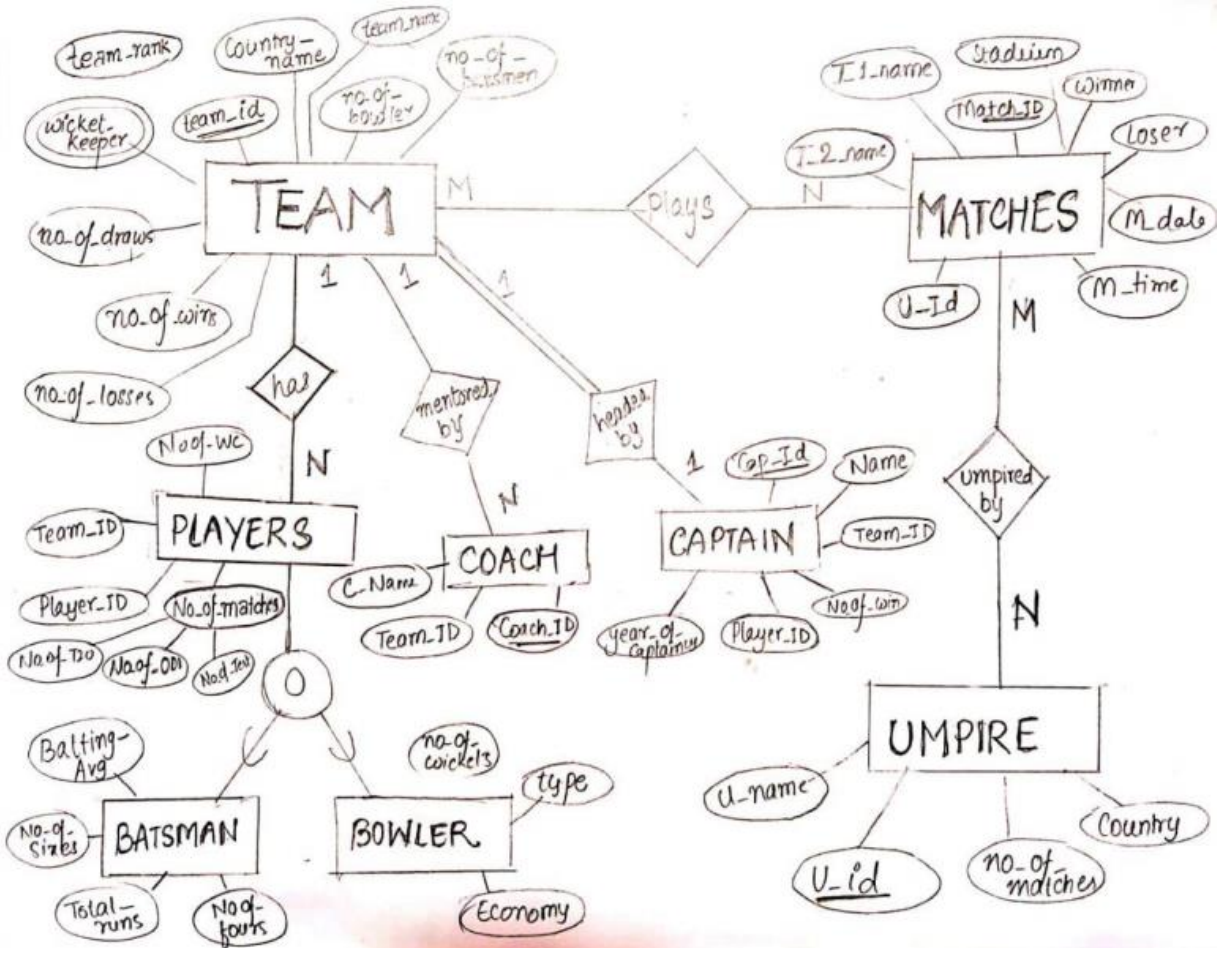
SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use SQL as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

DESIGN

Entity relationship diagram

Cricket World Cup



G.Anish <1602-19-737-003>

M.Rakesh <1602-19-737-031>

N.Rohith kumar<1602-19-737-033>

Front end programs and its connectivity

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
package airline.management.system;

import java.sql.*;

public class conn{
    Connection c;
    Statement s;
    public conn(){
        String dbURL = "jdbc:oracle:thin:@218.248.0.7:1521:RDBMS";
        String username = "it19737033";
        String password = "vasavi";
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            c = DriverManager.getConnection(dbURL, username, password);
            s =c.createStatement();
        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

Mainframe

```
package cricstats.management.system;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class MainFrame extends JFrame{ public static void main(String[] args) {
new MainFrame().setVisible(true);
}

public MainFrame() {
super("CRICKET STATISTICS MANAGEMENT SYSTEM");
initialize();
}

private void initialize() {

setForeground(Color.CYAN); setLayout(null);

JLabel cricketstatisticsManagementSystem = new JLabel("WELCOME!");
cricketstatisticsManagementSystem.setForeground(Color.BLUE);
cricketstatisticsManagementSystem.setFont(new Font("Tahoma", Font.PLAIN, 40));
cricketstatisticsManagementSystem.setBounds(240, 150, 1000, 55);
add(cricketstatisticsManagementSystem);


JMenuBar menuBar = new JMenuBar(); setJMenuBar(menuBar);

JMenu Teams = new JMenu("TEAMS"); Teams.setForeground(Color.BLUE);
menuBar.add(Teams);

JMenuItem InsertTeams = new JMenuItem("Insert"); Teams.add(InsertTeams);

JMenuItem updateTeams = new JMenuItem("Update"); Teams.add(updateTeams);

JMenuItem deleteTeams = new JMenuItem("Delete"); Teams.add(deleteTeams);

JMenu umpire = new JMenu("UMPIRE_DETAILS "); umpire.setForeground(Color.BLUE);
menuBar.add(umpire);

JMenuItem Insertumpire = new JMenuItem("Insert");  umpire.add(Insertumpire);

JMenuItem updateumpire = new JMenuItem("Update"); umpire.add(updateumpire);

JMenuItem deleteumpire = new JMenuItem("Delete"); umpire.add(deleteumpire);


JMenu player = new JMenu("PLAYER_DETAILS "); player.setForeground(Color.BLUE);
menuBar.add(player);

JMenuItem Insertplayer = new JMenuItem("Insert"); player.add(Insertplayer);

JMenuItem updateplayer = new JMenuItem("Update"); player.add(updateplayer);

JMenuItem deleteplayer = new JMenuItem("Delete"); player.add(deleteplayer);

JMenu match= new JMenu("MATCH_DETAILS"); match.setForeground(Color.BLUE);
menuBar.add(match);

JMenuItem Insertmatch = new JMenuItem("Insert");  match.add(Insertmatch);
JMenuItem updatematch = new JMenuItem("Update");match.add(updatematch);

JMenuItem deletematch = new JMenuItem("Delete"); match.add(deletematch);


InsertTeams.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
ae){
try {
    InsertTeams i = new InsertTeams();
    i.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
    });
} catch (Exception e) {

e.printStackTrace();
}
}
});

updateTeams.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
ae){
try {
    updateTeams ups = new updateTeams();
    ups.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
    });
} catch (Exception e) { e.printStackTrace();
}
}
});

deleteTeams.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
ae){
    try {
        deleteTeams del = new deleteTeams();
        del.addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
    }catch (Exception e) {e.printStackTrace();
}
}
});

Insertumpire.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
ae){
try {
    Insertumpire i = new Insertumpire();
    i.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
				}
			});
		} catch (Exception e) { e.printStackTrace();
		}
		}
		});

		updateumpire.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
		ae){
		try {
			updateumpire ups = new updateumpire();
			ups.addWindowListener(new WindowAdapter(){
				public void windowClosing(WindowEvent e)
				{
					System.exit(0);
				}
			});
		} catch (Exception e) { e.printStackTrace();
		}
		}
		});


		deleteumpire.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
		ae){
			try {
			deleteumpire del = new deleteumpire();
			del.addWindowListener(new WindowAdapter(){
				public void windowClosing(WindowEvent e)
				{
					System.exit(0);
				}
			});
		}catch (Exception e) {e.printStackTrace();
		}
		}
		});


		Insertplayer.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
		ae){
		try {
			Insertplayer i = new Insertplayer();
			i.addWindowListener(new WindowAdapter(){
				public void windowClosing(WindowEvent e)
				{
					System.exit(0);
				}
			});
		} catch (Exception e) { e.printStackTrace();
		}
		}
		});

		updateplayer.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
		ae){
		try {
			updateplayer ups = new updateplayer();
			ups.addWindowListener(new WindowAdapter(){
				public void windowClosing(WindowEvent e)
				{
```

```
                        System.exit(0);
                }
        });
} catch (Exception e) { e.printStackTrace();
}
}
});


deleteplayer.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
ae){
        try {
                deleteplayer del = new deleteplayer();
                del.addWindowListener(new WindowAdapter(){
                        public void windowClosing(WindowEvent e)
                        {
                                System.exit(0);
                        }
                });
        }catch (Exception e) { e.printStackTrace();
                }
                }
});
Insertmatch.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
ae){
try {
        Insertmatch i = new Insertmatch();
        i.addWindowListener(new WindowAdapter(){
                public void windowClosing(WindowEvent e)
                {
                        System.exit(0);
                }
        });
} catch (Exception e) { e.printStackTrace();
}
}
});

updatematch.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
ae){
try {
        updatematch ups = new updatematch();
        ups.addWindowListener(new WindowAdapter(){
                public void windowClosing(WindowEvent e)
                {
                        System.exit(0);
                }
        });
} catch (Exception e) { e.printStackTrace();
}

}
});

deletematch.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent
ae){
        try {
                deletematch del = new deletematch();
                del.addWindowListener(new WindowAdapter(){
                        public void windowClosing(WindowEvent e)
                        {
```

```
                System.exit(0);
            }
        });
}catch (Exception e) { e.printStackTrace();
}
}
});
setSize(700,500); setLocation(285,100);
setVisible(true);
}
}
```

Insert Teams Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
public class InsertTeams extends Frame
{
    Button insertTeamsButton;
    TextField team_rank,country_name,no_of_wins,no_of_loses;
    TextArea errorText;
    Connection connection;
    Statement statement;
    public InsertTeams()
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
        buildGUI();
    }
    public void connectToDB()
    {
        try
        {
            Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");

            statement = con.createStatement();
            System.out.println("connected");
        }
        catch (SQLException e)
        {
            System.out.println(e);
        }
    }
    public void buildGUI()
    {

        insertTeamsButton = new Button("Insert");
        insertTeamsButton.addActionListener(new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        try
                        {

                            String query= "INSERT INTO Teams VALUES("+"'"
+team_rank.getText() + "','" + country_name.getText() + "','" + no_of_wins.getText() + "','"
+no_of_loses.getText() + "'" +")";
                            int i = statement.executeUpdate(query);
                            errorText.append("\nInserted " + i + " rows
successfully");
                        }
                        catch (SQLException insertException)
                        {
                            displaySQLErrors(insertException);
                        }
                    }
                });
        team_rank = new TextField(15);
        country_name = new TextField(15);
        no_of_wins = new TextField(15);
        no_of_loses = new TextField(15);
        errorText = new TextArea(10, 40);
        errorText.setEditable(false);
        Panel first = new Panel();
        first.setLayout(new GridLayout(4, 2));
        first.add(new Label("Team Rank : "));
        first.add(team_rank);
        first.add(new Label("Team Name : "));
        first.add(country_name);
        first.add(new Label("NO OF WINS : "));
        first.add(no_of_wins);
        first.add(new Label("NO OF LOSES : "));
        first.add(no_of_loses);
        first.setBounds(125,90,200,100);
        Panel second = new Panel(new GridLayout(4, 1));
        second.add(insertTeamsButton);
        second.setBounds(125,220,150,100);
        Panel third = new Panel();
        third.add(errorText);
        third.setBounds(125,320,300,200);
        setLayout(null);
        add(first);
        add(second);
        add(third);
        setTitle("Insert Gets");
        setSize(500, 600);
        setVisible(true);
    }
    private void displaySQLErrors(SQLException e)
    {
        errorText.append("\nSQLException: " + e.getMessage() + "\n");
        errorText.append("SQLState: " + e.getSQLState() + "\n");
        errorText.append("VendorError: " + e.getErrorCode() + "\n");
    }
}
```

Delete Teams Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class deleteTeams extends Frame
{
Button deleteTeamsButton;
List team_rankList;
TextField team_rankText,country_nameText,no_of_winsText,no_of_losesText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public deleteTeams()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
buildGUI();
}
public void connectToDB()
		{
			try
			{
				Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
				statement = con.createStatement();
				System.out.println("connected");
			}
			catch (SQLException e)
			{
				System.out.println(e);
			}
		}
private void loadTeams()
{
try
{
rs = statement.executeQuery("SELECT * FROM Teams");
while (rs.next())
{
team_rankList.add(rs.getString("Team_rank"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
team_rankList = new List(10);
loadTeams();
add(team_rankList);
```

```
team_rankList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM Teams where team_rank ='"+team_rankList.getSelectedItem()+"'");
while (rs.next())
{
if
(rs.getString("Team_rank").equals(team_rankList.getSelectedItem()))
break;
}
if (!rs.isAfterLast())
{
team_rankText.setText(rs.getString("team_rank"));
country_nameText.setText(rs.getString("country_name"));
no_of_winsText.setText(rs.getString("no_of_wins"));
no_of_losesText.setText(rs.getString("no_of_loses"));
}
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
deleteTeamsButton = new Button("Delete Team");
deleteTeamsButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Connection con= DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("DELETE FROM Teams WHERE team_rank = '"+team_rankList.getSelectedItem()+"' and
country_name='"+country_nameText.getText()+"' and  no_of_wins='"+no_of_winsText.getText()+"' and no_of_loses
='"+no_of_losesText.getText()+"'");
errorText.append("\nDeleted " + i + " rows successfully");
team_rankText.setText(null);
country_nameText.setText(null);
no_of_winsText.setText(null);
no_of_losesText.setText(null);
team_rankList.removeAll();
loadTeams();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
team_rankText = new TextField(15);
country_nameText = new TextField(15);
no_of_winsText = new TextField(15);
no_of_losesText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Team Rank : "));
first.add(team_rankText);
first.add(new Label("Team Name : "));
first.add(country_nameText);
first.add(new Label("NO OF WINS : "));
first.add(no_of_winsText);
first.add(new Label("NO OF LOSES : "));
first.add(no_of_losesText);
Panel second = new Panel(new GridLayout(4, 1));
second.add(deleteTeamsButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Remove  Team ");
setSize(450, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
```

Modify Teams Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class updateTeams extends Frame
{
        Button updateTeamsButton;
        List team_rankList;
        TextField team_rankText,country_nameText,no_of_winsText,no_of_losesText;
        TextArea errorText;
        Connection connection;
        Statement statement;
        ResultSet rs;
        public updateTeams()
        {
                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
                        System.err.println("Unable to find and load driver");
                        System.exit(1);
                }
                connectToDB();
                buildGUI();
        }
        public void connectToDB()
        {
            try
            {
                Connection con =

        DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
                statement = con.createStatement();
                System.out.println("connected");
            }
            catch (SQLException e)
            {
                System.out.println(e);
            }
        }
        private void loadTeams()
        {
            try
            {
                rs = statement.executeQuery("SELECT team_rank FROM Teams");
                while (rs.next())
                {
                    team_rankList.add(rs.getString("team_rank"));
                }
            }
            catch (SQLException e)
            {
                displaySQLErrors(e);
            }
        }
        public void buildGUI()
        {
            team_rankList = new List(10);
            loadTeams();
            add(team_rankList);
            team_rankList.addItemListener(new ItemListener()
                    {
                        public void itemStateChanged(ItemEvent e)
                        {
                            try
                            {
                                rs = statement.executeQuery("SELECT * FROM Teams
    where team_rank ='"+team_rankList.getSelectedItem()+"'");
                                rs.next();
                                team_rankText.setText(rs.getString("team_rank"));
                                country_nameText.setText(rs.getString("country_name"));
                                no_of_winsText.setText(rs.getString("no_of_wins"));
                                no_of_losesText.setText(rs.getString("no_of_loses"));
                            }
                            catch (SQLException selectException)
                            {
                                displaySQLErrors(selectException);
                            }
                        }
                    });
            updateTeamsButton = new Button("Update");
            updateTeamsButton.addActionListener(new ActionListener()
                                {
                                        public void actionPerformed(ActionEvent e)
                                        {
                                                try
                                                {

                                                        Connection con =

        DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
                                                        Statement statement1 = con.createStatement();
                                                        int i = statement1.executeUpdate("UPDATE Teams "+ "SET
country_name='" + country_nameText.getText() + "', "+ "no_of_wins=" + no_of_winsText.getText() + ", "+
"no_of_loses ="+ no_of_losesText.getText() + " WHERE team_rank= "+ team_rankList.getSelectedItem());
                                                        errorText.append("\nUpdated " + i + " rows successfully");
                                                        team_rankList.removeAll();
                                                        loadTeams();
                                                }
                                                catch (SQLException insertException)
                                                {
                                                        displaySQLErrors(insertException);
                                                }
                                        }
                                });
                team_rankText = new TextField(15);
                team_rankText.setEditable(false);
                country_nameText = new TextField(15);
                no_of_winsText = new TextField(15);
                no_of_losesText = new TextField(15);
                errorText = new TextArea(10, 40);
                errorText.setEditable(false);
                Panel first = new Panel();
                first.setLayout(new GridLayout(4, 2));
                first.add(new Label("Team Rank : "));
                first.add(team_rankText);
                first.add(new Label("Team Name : "));
                first.add(country_nameText);
                first.add(new Label("NO OF WINS : "));
                first.add(no_of_winsText);
                first.add(new Label("NO OF LOSES : "));
                first.add(no_of_losesText);
                Panel second = new Panel(new GridLayout(4, 1));
                second.add(updateTeamsButton);
                Panel third = new Panel();
                third.add(errorText);
                add(first);
                add(second);
                add(third);
                setTitle("Update Element ");
                setSize(500, 600);
                setLayout(new FlowLayout());
                setVisible(true);
        }
        private void displaySQLErrors(SQLException e)
        {
                errorText.append("\nSQLException: " + e.getMessage() + "\n");
```

```
                        errorText.append("SQLState: " + e.getSQLState() + "\n");
                        errorText.append("VendorError: " + e.getErrorCode() + "\n");
                }
```

Insert Umpire Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Insertumpire extends Frame
{
	Button insertumpireButton;
	TextField umpire_id,umpire_name,no_of_matches,country;
	TextArea errorText;
	Connection connection;
	Statement statement;
	public Insertumpire()
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
		buildGUI();
	}
	public void connectToDB()
	{
		try
		{
			Connection con = 
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");

			statement = con.createStatement();
			System.out.println("connected");
		}
		catch (SQLException e)
		{
			System.out.println(e);
		}
	}
	public void buildGUI()
	{

		insertumpireButton = new Button("Insert");
		insertumpireButton.addActionListener(new ActionListener()
				{
					public void actionPerformed(ActionEvent e)
					{
						try
						{

							String query= "INSERT INTO umpire VALUES("+"'"
+umpire_id.getText() + "','" + umpire_name.getText() + "','" + no_of_matches.getText() + "','"
+country.getText() + "'" +")";
							int i = statement.executeUpdate(query);
```

```
                            errorText.append("\nInserted " + i + " rows
successfully");
                        }
                        catch (SQLException insertException)
                        {
                            displaySQLErrors(insertException);
                        }
                    }
                });
        umpire_id = new TextField(15);
        umpire_name = new TextField(15);
        no_of_matches = new TextField(15);
        country = new TextField(15);
        errorText = new TextArea(10, 40);
        errorText.setEditable(false);
        Panel first = new Panel();
        first.setLayout(new GridLayout(4, 2));
        first.add(new Label("UMPIRE ID : "));
        first.add(umpire_id);
        first.add(new Label("UMPIRE NAME : "));
        first.add(umpire_name);
        first.add(new Label("NO OF MATCHES : "));
        first.add(no_of_matches);
        first.add(new Label("COUNTRY : "));
        first.add(country);
        first.setBounds(125,90,200,100);
        Panel second = new Panel(new GridLayout(4, 1));
        second.add(insertumpireButton);
        second.setBounds(125,220,150,100);
        Panel third = new Panel();
        third.add(errorText);
        third.setBounds(125,320,300,200);
        setLayout(null);
        add(first);
        add(second);
        add(third);
        setTitle("Insert UMPIRE");
        setSize(500, 600);
        setVisible(true);
    }
    private void displaySQLErrors(SQLException e)
    {
        errorText.append("\nSQLException: " + e.getMessage() + "\n");
        errorText.append("SQLState: " + e.getSQLState() + "\n");
        errorText.append("VendorError: " + e.getErrorCode() + "\n");
    }
}
```

Delete Umpire Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class deleteumpire extends Frame
{
Button deleteumpireButton;
List umpire_idList;
TextField umpire_idText,umpire_nameText,no_of_matchesText,countryText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public deleteumpire()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
buildGUI();
}
public void connectToDB()
		{
			try
			{
				Connection con = 
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
				statement = con.createStatement();
				System.out.println("connected");
			}
			catch (SQLException e)
			{
				System.out.println(e);
			}
		}
private void loadumpire()
{
try
{
rs = statement.executeQuery("SELECT * FROM umpire");
while (rs.next())
{
umpire_idList.add(rs.getString("umpire_id"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
umpire_idList = new List(10);
loadumpire();
add(umpire_idList);
```

```
umpire_idList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM umpire where umpire_id ='"+umpire_idList.getSelectedItem()+"'");
while (rs.next())
{
if
(rs.getString("umpire_id").equals(umpire_idList.getSelectedItem()))
break;
}
if (!rs.isAfterLast())
{
umpire_idText.setText(rs.getString("umpire_id"));
umpire_nameText.setText(rs.getString("umpire_name"));
no_of_matchesText.setText(rs.getString("no_of_matches"));
countryText.setText(rs.getString("country"));
}
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
deleteumpireButton = new Button("Delete umpire");
deleteumpireButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Connection con= DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("DELETE FROM umpire WHERE umpire_id = '"+umpire_idList.getSelectedItem()+"' and
umpire_name='"+umpire_nameText.getText()+"' and  no_of_matches='"+no_of_matchesText.getText()+"' and country
='"+countryText.getText()+"'");
errorText.append("\nDeleted " + i + " rows successfully");
umpire_idText.setText(null);
umpire_nameText.setText(null);
no_of_matchesText.setText(null);
countryText.setText(null);
umpire_idList.removeAll();
loadumpire();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
umpire_idText = new TextField(15);
umpire_nameText = new TextField(15);
no_of_matchesText = new TextField(15);
countryText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("UMPIRE ID : "));
first.add(umpire_idText);
first.add(new Label("UMPIRE NAME : "));
first.add(umpire_nameText);
first.add(new Label("NO OF MATCHES : "));
first.add(no_of_matchesText);
first.add(new Label("COUNTRY : "));
first.add(countryText);
Panel second = new Panel(new GridLayout(4, 1));
second.add(deleteumpireButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Remove UMPIRE ");
setSize(450, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
```

Modify Umpire Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class updateumpire extends Frame
{
        Button updateumpireButton;
        List umpire_idList;
        TextField umpire_idText,umpire_nameText,no_of_matchesText,countryText;
        TextArea errorText;
        Connection connection;
        Statement statement;
        ResultSet rs;
        public updateumpire()
        {
                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
                        System.err.println("Unable to find and load driver");
                        System.exit(1);
                }
```

```
		connectToDB();
		buildGUI();
	}
	public void connectToDB()
	{
		try
		{
			Connection con =
	DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
			statement = con.createStatement();
			System.out.println("connected");
		}
		catch (SQLException e)
		{
			System.out.println(e);
		}
	}
	private void loadumpire()
	{
		try
		{
			rs = statement.executeQuery("SELECT umpire_id FROM UMPIRE");
			while (rs.next())
			{
				umpire_idList.add(rs.getString("umpire_id"));
			}
		}
		catch (SQLException e)
		{
			displaySQLErrors(e);
		}
	}
	public void buildGUI()
	{
		umpire_idList = new List(10);
		loadumpire();
		add(umpire_idList);
		umpire_idList.addItemListener(new ItemListener()
				{
					public void itemStateChanged(ItemEvent e)
					{
						try
						{
							rs = statement.executeQuery("SELECT * FROM Umpire where
umpire_id ='"+umpire_idList.getSelectedItem()+"'");
                            rs.next();
                            umpire_idText.setText(rs.getString("umpire_id"));
                            umpire_nameText.setText(rs.getString("umpire_name"));
                            no_of_matchesText.setText(rs.getString("no_of_matches"));
                            countryText.setText(rs.getString("country"));
                        }
                        catch (SQLException selectException)
                        {
                            displaySQLErrors(selectException);
                        }
                    }
                });
        updateumpireButton = new Button("Update");
        updateumpireButton.addActionListener(new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        try
                        {

                            Connection con =

    DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
                            Statement statement1 = con.createStatement();
                            int i = statement1.executeUpdate("UPDATE Umpire "+ "SET
umpire_name='" + umpire_nameText.getText() + "', "+ "no_of_matches='" + no_of_matchesText.getText() + "', "+ "country ='"+
countryText.getText() + "' WHERE umpire_id='"+ umpire_idList.getSelectedItem()+"'");
                            errorText.append("\nUpdated " + i + " rows successfully");
                            umpire_idList.removeAll();
                            loadumpire();
                        }
                        catch (SQLException insertException)
                        {
                            displaySQLErrors(insertException);
                        }
                    }
                });
        umpire_idText = new TextField(15);
        umpire_idText.setEditable(false);
        umpire_nameText = new TextField(15);
        no_of_matchesText = new TextField(15);
        countryText = new TextField(15);
        errorText = new TextArea(10, 40);
        errorText.setEditable(false);
```

```
		Panel first = new Panel();
		first.setLayout(new GridLayout(4, 2));
		first.add(new Label("UMPIRE ID : "));
		first.add(umpire_idText);
		first.add(new Label("UMPIRE NAME : "));
		first.add(umpire_nameText);
		first.add(new Label("NO OF MATCHES : "));
		first.add(no_of_matchesText);
		first.add(new Label("COUNTRY : "));
		first.add(countryText);
		Panel second = new Panel(new GridLayout(4, 1));
		second.add(updateumpireButton);
		Panel third = new Panel();
		third.add(errorText);
		add(first);
		add(second);
		add(third);
		setTitle("Update UMPIRE ");
		setSize(500, 600);
		setLayout(new FlowLayout());
		setVisible(true);
	}
	private void displaySQLErrors(SQLException e)
	{
		errorText.append("\nSQLException: " + e.getMessage() + "\n");
		errorText.append("SQLState: " + e.getSQLState() + "\n");
		errorText.append("VendorError: " + e.getErrorCode() + "\n");
	}
```

Insert Player Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Insertplayer extends Frame
{
	Button InsertplayerButton;
	TextField player_name,no_of_matches,batting_average,runsscored,wicketstaken,economy;
	TextArea errorText;
	Connection connection;
	Statement statement;
	public Insertplayer()
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
```

```
            System.exit(1);
        }
        connectToDB();
        buildGUI();
    }
    public void connectToDB()
    {
        try
        {
            Connection con = 
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");

            statement = con.createStatement();
            System.out.println("connected");
        }
        catch (SQLException e)
        {
            System.out.println(e);
        }
    }
    public void buildGUI()
    {

        InsertplayerButton = new Button("Insert");
        InsertplayerButton.addActionListener(new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        try
                        {

                            String query= "INSERT INTO player VALUES("+"'" 
+player_name.getText() + "','" + no_of_matches.getText() + "','" + batting_average.getText() + "','" +runsscored.getText() + "','" 
+wicketstaken.getText()+ "','" +economy.getText() + "'" +")";
                            int i = statement.executeUpdate(query);
                            errorText.append("\nInserted " + i + " rows successfully");
                        }
                        catch (SQLException insertException)
                        {
                            displaySQLErrors(insertException);
                        }
                    }
                });
        player_name = new TextField(15);
        no_of_matches = new TextField(15);
        batting_average = new TextField(15);
        runsscored = new TextField(15);
        wicketstaken = new TextField(15);
        economy = new TextField(15);
        errorText = new TextArea(10, 40);
        errorText.setEditable(false);
        Panel first = new Panel();
        first.setLayout(new GridLayout(6, 2));
        first.add(new Label("PLAYER NAME : "));
        first.add(player_name);
        first.add(new Label("NO OF MATCHES : "));
        first.add(no_of_matches);
        first.add(new Label("BATTING AVERAGE "));
        first.add(batting_average);
        first.add(new Label("RUNS SCORED : "));
        first.add(runsscored);
        first.add(new Label("WICKETS TAKEN: "));
        first.add(wicketstaken);
        first.add(new Label("ECONOMY : "));
        first.add(economy);

        first.setBounds(125,90,200,100);
        Panel second = new Panel(new GridLayout(6, 1));
        second.add(InsertplayerButton);
        second.setBounds(125,220,150,100);
        Panel third = new Panel();
        third.add(errorText);
        third.setBounds(125,320,300,200);
        setLayout(null);
        add(first);
        add(second);
        add(third);
        setTitle("Insert player");
        setSize(500, 600);
        setVisible(true);
    }
    private void displaySQLErrors(SQLException e)
    {
        errorText.append("\nSQLException: " + e.getMessage() + "\n");
        errorText.append("SQLState: " + e.getSQLState() + "\n");
        errorText.append("VendorError: " + e.getErrorCode() + "\n");
    }
```

Delete Player Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class deleteplayer extends Frame
{
Button deleteplayerButton;
List player_nameList;
TextField player_nameText,no_of_matchesText,batting_averageText,runsscoredText,wicketstakenText,economyText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public deleteplayer()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
buildGUI();
}
public void connectToDB()
    {
        try
        {
            Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
            statement = con.createStatement();
            System.out.println("connected");
        }
        catch (SQLException e)
        {
            System.out.println(e);
        }
    }
private void loadplayer()
{
try
{
rs = statement.executeQuery("SELECT * FROM player");
while (rs.next())
{
player_nameList.add(rs.getString("player_name"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
player_nameList = new List(10);
loadplayer();
add(player_nameList);
player_nameList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM player where player_name ='"+player_nameList.getSelectedItem()+"'");
while (rs.next())
{
if
(rs.getString("player_name").equals(player_nameList.getSelectedItem()))
break;
}
if (!rs.isAfterLast())
{
    player_nameText.setText(rs.getString("player_name"));
    no_of_matchesText.setText(rs.getString("no_of_matches"));
    batting_averageText.setText(rs.getString("batting_average"));
    runsscoredText.setText(rs.getString("runsscored"));
    wicketstakenText.setText(rs.getString("wicketstaken"));
    economyText.setText(rs.getString("economy"));
}
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
deleteplayerButton = new Button("Delete player");
```

```
deleteplayerButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Connection con= DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("DELETE FROM player WHERE player_name = '"+player_nameList.getSelectedItem()+"' and
no_of_matches='"+no_of_matchesText.getText()+"' and  batting_average='"+batting_averageText.getText()+"' and runsscored
='"+runsscoredText.getText()+"' and  wicketstaken='"+wicketstakenText.getText()+"' and
economy='"+economyText.getText()+"'");
errorText.append("\nDeleted " + i + " rows successfully");
no_of_matchesText.setText(null);
batting_averageText.setText(null);
runsscoredText.setText(null);
wicketstakenText.setText(null);
economyText.setText(null);
player_nameList.removeAll();
loadplayer();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
player_nameText = new TextField(15);
no_of_matchesText = new TextField(15);
batting_averageText = new TextField(15);
runsscoredText = new TextField(15);
wicketstakenText = new TextField(15);
economyText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(6, 2));
first.add(new Label("PLAYER NAME : "));
first.add(player_nameText);
first.add(new Label("NO OF MATCHES : "));
first.add(no_of_matchesText);
first.add(new Label("BATTING AVERAGE: "));
first.add(batting_averageText);
first.add(new Label("RUNS SCORED : "));
first.add(runsscoredText);
first.add(new Label("WICKETS TAKEN: "));
first.add(wicketstakenText);
first.add(new Label("ECONOMY: "));
first.add(economyText);
Panel second = new Panel(new GridLayout(6, 1));
second.add(deleteplayerButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Remove  player ");
setSize(450, 600);
setLayout(new FlowLayout());
setVisible(true);
```

```
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
```

Modify Player Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class updateplayer extends Frame
{
	Button updateplayerButton;
	List player_nameList;
	TextField player_nameText,no_of_matchesText,batting_averageText,runsscoredText,wicketstakenText,economyText;
	TextArea errorText;
	Connection connection;
	Statement statement;
	ResultSet rs;
	public updateplayer()
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
		buildGUI();
	}
	public void connectToDB()
	{
		try
		{
			Connection con =

	DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
			statement = con.createStatement();
			System.out.println("connected");
		}
		catch (SQLException e)
		{
			System.out.println(e);
		}
	}
	private void loadplayer()
	{
		try
		{
			rs = statement.executeQuery("SELECT player_name FROM player");
			while (rs.next())
			{
				player_nameList.add(rs.getString("player_name"));
			}
```

```
        }
        catch (SQLException e)
        {
            displaySQLErrors(e);
        }
    }
    public void buildGUI()
    {
        player_nameList = new List(10);
        loadplayer();
        add(player_nameList);
        player_nameList.addItemListener(new ItemListener()
                {
                    public void itemStateChanged(ItemEvent e)
                    {
                        try
                        {
                            rs = statement.executeQuery("SELECT * FROM player where 
player_name ='"+player_nameList.getSelectedItem()+"'");
                            rs.next();
                            player_nameText.setText(rs.getString("player_name"));
                            no_of_matchesText.setText(rs.getString("no_of_matches"));
                            batting_averageText.setText(rs.getString("batting_average"));
                            runsscoredText.setText(rs.getString("runsscored"));
                            wicketstakenText.setText(rs.getString("wicketstaken"));
                            economyText.setText(rs.getString("economy"));
                        }
                        catch (SQLException selectException)
                        {
                            displaySQLErrors(selectException);
                        }
                    }
                });
        updateplayerButton = new Button("Update");
        updateplayerButton.addActionListener(new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        try
                        {

                            Connection con = 

    DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
                            Statement statement1 = con.createStatement();
                            int i = statement1.executeUpdate("UPDATE player "+ "SET 
no_of_matches='" + no_of_matchesText.getText() + "', "+ "batting_average='" + batting_averageText.getText() + "', "+ 
"runsscored ='"+ runsscoredText.getText()+"',"+"wicketstaken='" +wicketstakenText.getText()+"',"+"economy='" 
+economyText.getText() + "' WHERE player_name= '"+ player_nameList.getSelectedItem()+"'");
                            errorText.append("\nUpdated " + i + " rows successfully");
                            player_nameList.removeAll();
                            loadplayer();
                        }
                        catch (SQLException insertException)
                        {
                            displaySQLErrors(insertException);
                        }
                    }
                });
        player_nameText = new TextField(15);
```

```
		player_nameText.setEditable(false);
		no_of_matchesText = new TextField(15);
		batting_averageText = new TextField(15);
		runsscoredText = new TextField(15);
		wicketstakenText = new TextField(15);
		economyText = new TextField(15);
		errorText = new TextArea(10, 40);
		errorText.setEditable(false);
		Panel first = new Panel();
		first.setLayout(new GridLayout(6, 2));
		first.add(new Label("PLAYER NAME : "));
		first.add(player_nameText);
		first.add(new Label("NO OF MATCHES : "));
		first.add(no_of_matchesText);
		first.add(new Label("BATTING AVERAGE: "));
		first.add(batting_averageText);
		first.add(new Label("RUNS SCORED : "));
		first.add(runsscoredText);
		first.add(new Label("WICKETS TAKEN: "));
		first.add(wicketstakenText);
		first.add(new Label("ECONOMY: "));
		first.add(economyText);
		Panel second = new Panel(new GridLayout(6, 1));
		second.add(updateplayerButton);
		Panel third = new Panel();
		third.add(errorText);
		add(first);
		add(second);
		add(third);
		setTitle("Update player ");
		setSize(500, 600);
		setLayout(new FlowLayout());
		setVisible(true);
	}
	private void displaySQLErrors(SQLException e)
	{
		errorText.append("\nSQLException: " + e.getMessage() + "\n");
		errorText.append("SQLState: " + e.getSQLState() + "\n");
		errorText.append("VendorError: " + e.getErrorCode() + "\n");
	}
```

Insert Match Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Insertmatch extends Frame
{
	Button InsertmatchButton;
	TextField match_date,stadium,team_name1,team_name2,winner,loser;
	TextArea errorText;
	Connection connection;
	Statement statement;
	public Insertmatch()
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
```

```
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
        buildGUI();
    }
    public void connectToDB()
    {
        try
        {
            Connection con = 
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");

            statement = con.createStatement();
            System.out.println("connected");
        }
        catch (SQLException e)
        {
            System.out.println(e);
        }
    }
    public void buildGUI()
    {

        InsertmatchButton = new Button("Insert");
        InsertmatchButton.addActionListener(new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        try
                        {

                            String query= "INSERT INTO match VALUES("+"'"
+match_date.getText() + "','" + stadium.getText() + "','" + team_name1.getText() + "','" +team_name2.getText() + "','"
+winner.getText()+ "','" +loser.getText() + "'" +")";
                            int i = statement.executeUpdate(query);
                            errorText.append("\nInserted " + i + " rows successfully");
                        }
                        catch (SQLException insertException)
                        {
                            displaySQLErrors(insertException);
                        }
                    }
                });
        match_date = new TextField(15);
        stadium = new TextField(15);
        team_name1 = new TextField(15);
        team_name2 = new TextField(15);
        winner = new TextField(15);
        loser = new TextField(15);
        errorText = new TextArea(10, 40);
        errorText.setEditable(false);
        Panel first = new Panel();
        first.setLayout(new GridLayout(6, 2));
        first.add(new Label("MATCH DATE : "));
        first.add(match_date);
        first.add(new Label("STADIUM : "));
        first.add(stadium);
        first.add(new Label("TEAM 1 "));
```

```
                first.add(team_name1);
                first.add(new Label("TEAM 2 : "));
                first.add(team_name2);
                first.add(new Label("WINNER: "));
                first.add(winner);
                first.add(new Label("LOSER : "));
                first.add(loser);

                first.setBounds(125,90,200,100);
                Panel second = new Panel(new GridLayout(6, 1));
                second.add(InsertmatchButton);
                second.setBounds(125,220,150,100);
                Panel third = new Panel();
                third.add(errorText);
                third.setBounds(125,320,300,200);
                setLayout(null);
                add(first);
                add(second);
                add(third);
                setTitle("Insert MATCH");
                setSize(500, 600);
                setVisible(true);
        }
        private void displaySQLErrors(SQLException e)
        {
                errorText.append("\nSQLException: " + e.getMessage() + "\n");
                errorText.append("SQLState: " + e.getSQLState() + "\n");
                errorText.append("VendorError: " + e.getErrorCode() + "\n");
        }
```

Delete Match Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class deletematch extends Frame
{
Button deletematchButton;
List match_numberList;
TextField match_numberText,stadiumText,team_name1Text,team_name2Text,winnerText,loserText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public deletematch()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
buildGUI();
}
public void connectToDB()
        {
                try
```

```
             {
                    Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
                    statement = con.createStatement();
                    System.out.println("connected");
             }
             catch (SQLException e)
             {
                    System.out.println(e);
             }
      }
private void loadmatch()
{
try
{
rs = statement.executeQuery("SELECT * FROM match");
while (rs.next())
{
match_numberList.add(rs.getString("match_number"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
match_numberList = new List(10);
loadmatch();
add(match_numberList);
match_numberList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM match where match_number
='"+match_numberList.getSelectedItem()+"'");
while (rs.next())
{
if
(rs.getString("match_number").equals(match_numberList.getSelectedItem()))
break;
}
if (!rs.isAfterLast())
{
       match_numberText.setText(rs.getString("match_number"));
       stadiumText.setText(rs.getString("stadium"));
       team_name1Text.setText(rs.getString("team_name1"));
       team_name2Text.setText(rs.getString("team_name2"));
       winnerText.setText(rs.getString("winner"));
       loserText.setText(rs.getString("loser"));
}
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
```

```
deletematchButton = new Button("Delete match");
deletematchButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Connection con=
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("DELETE FROM match WHERE match_number =
'"+match_numberList.getSelectedItem()+"' and stadium='"+stadiumText.getText()+"' and
team_name1='"+team_name1Text.getText()+"' and team_name2 ='"+team_name2Text.getText()+"' and
winner='"+winnerText.getText()+"' and  loser='"+loserText.getText()+"'");
errorText.append("\nDeleted " + i + " rows successfully");
stadiumText.setText(null);
team_name1Text.setText(null);
team_name2Text.setText(null);
winnerText.setText(null);
loserText.setText(null);
match_numberList.removeAll();
loadmatch();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
match_numberText = new TextField(15);
stadiumText = new TextField(15);
team_name1Text = new TextField(15);
team_name2Text = new TextField(15);
winnerText = new TextField(15);
loserText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(6, 2));
first.add(new Label("MATCH DATE : "));
first.add(match_numberText);
first.add(new Label("STADIUM : "));
first.add(stadiumText);
first.add(new Label("TEAM 1: "));
first.add(team_name1Text);
first.add(new Label("TEAM 2 : "));
first.add(team_name2Text);
first.add(new Label("WINNER : "));
first.add(winnerText);
first.add(new Label("LOSER: "));
first.add(loserText);
Panel second = new Panel(new GridLayout(6, 1));
second.add(deletematchButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Remove  match ");
setSize(450, 600);
setLayout(new FlowLayout());
```

```
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
}
```

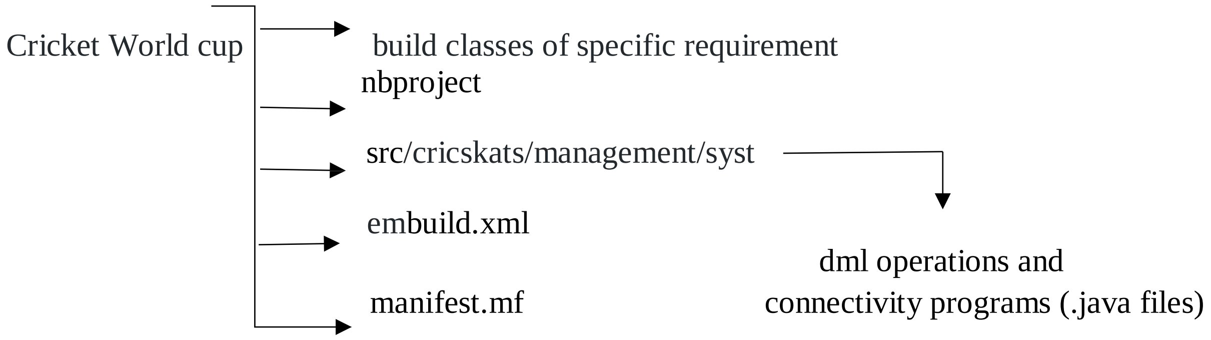
Modify Match Details

```
package cricstats.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class updatematch extends Frame
{
	Button updatematchButton;
	List winnerList;
	TextField match_numberText,stadiumText,team_name1Text,team_name2Text,winnerText,loserText;
	TextArea errorText;
	Connection connection;
	Statement statement;
	ResultSet rs;
	public updatematch()
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
		buildGUI();
	}
	public void connectToDB()
	{
		try
		{
			Connection con =

	DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
			statement = con.createStatement();
			System.out.println("connected");
		}
		catch (SQLException e)
		{
			System.out.println(e);
		}
	}
	private void loadmatch()
	{
		try
		{
			rs = statement.executeQuery("SELECT winner FROM match");
			while (rs.next())
			{
				winnerList.add(rs.getString("winner"));
			}
		}
		catch (SQLException e)
		{
			displaySQLErrors(e);
		}
	}
	public void buildGUI()
	{
		winnerList = new List(10);
		loadmatch();
		add(winnerList);
		winnerList.addItemListener(new ItemListener()
				{
					public void itemStateChanged(ItemEvent e)
					{
						try
						{
							rs = statement.executeQuery("SELECT * FROM match where 
winner ='"+winnerList.getSelectedItem()+"'");
							rs.next();
							winnerText.setText(rs.getString("winner"));
							match_numberText.setText(rs.getString("match_number"));
							stadiumText.setText(rs.getString("stadium"));
							team_name1Text.setText(rs.getString("team_name1"));
							team_name2Text.setText(rs.getString("team_name2"));
							loserText.setText(rs.getString("loser"));
						}
						catch (SQLException selectException)
						{
							displaySQLErrors(selectException);
						}
					}
				});
		updatematchButton = new Button("Update");
		updatematchButton.addActionListener(new ActionListener()
				{
					public void actionPerformed(ActionEvent e)
					{
						try
						{

							Connection con = 

	DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737033","vasavi");
							Statement statement1 = con.createStatement();
							int i= statement1.executeUpdate("UPDATE match "+ "SET 
match_number='" + match_numberText.getText() + "', "+ "stadium='" + stadiumText.getText() + "', "+ "team_name1 ='"+ 
team_name1Text.getText()+"',"+"team_name2='" +team_name2Text.getText()+"',"+"loser='" +loserText.getText() + "' WHERE 
winner= '"+ winnerList.getSelectedItem()+"'");
							errorText.append("\nUpdated " + i + " rows successfully");
							winnerList.removeAll();
							loadmatch();
						}
						catch (SQLException insertException)
						{
							displaySQLErrors(insertException);
						}
```

```
                                        }
                                });
                winnerText = new TextField(15);
                winnerText.setEditable(false);
                match_numberText = new TextField(15);
                stadiumText = new TextField(15);
                team_name1Text = new TextField(15);
                team_name2Text = new TextField(15);
                loserText = new TextField(15);
                errorText = new TextArea(10, 40);
                errorText.setEditable(false);
                Panel first = new Panel();
                first.setLayout(new GridLayout(6, 2));
                first.add(new Label("MATCH NUMBER : "));
                first.add(match_numberText);
                first.add(new Label("STADIUM : "));
                first.add(stadiumText);
                first.add(new Label("TEAM 1: "));
                first.add(team_name1Text);
                first.add(new Label("TEAM 2 : "));
                first.add(team_name2Text);
                first.add(new Label("WINNER: "));
                first.add(winnerText);
                first.add(new Label("LOSER: "));
                first.add(loserText);
                Panel second = new Panel(new GridLayout(6, 1));
                second.add(updatematchButton);
                Panel third = new Panel();
                third.add(errorText);
                add(first);
                add(second);
                add(third);
                setTitle("Update match ");
                setSize(500, 600);
                setLayout(new FlowLayout());
                setVisible(true);
        }
        private void displaySQLErrors(SQLException e)
        {
                errorText.append("\nSQLException: " + e.getMessage() + "\n");
                errorText.append("SQLState: " + e.getSQLState() + "\n");
                errorText.append("VendorError: " + e.getErrorCode() + "\n");
        }
```

Github links and folder structure:

Link: https://github.com/Rohith2609/cricket-world-cup


Cricket World cup
build classes of specific requirement
nbproject
src/cricskats/management/syst
embuild.xml
manifest.mf
dml operations and
connectivity programs (.java files)

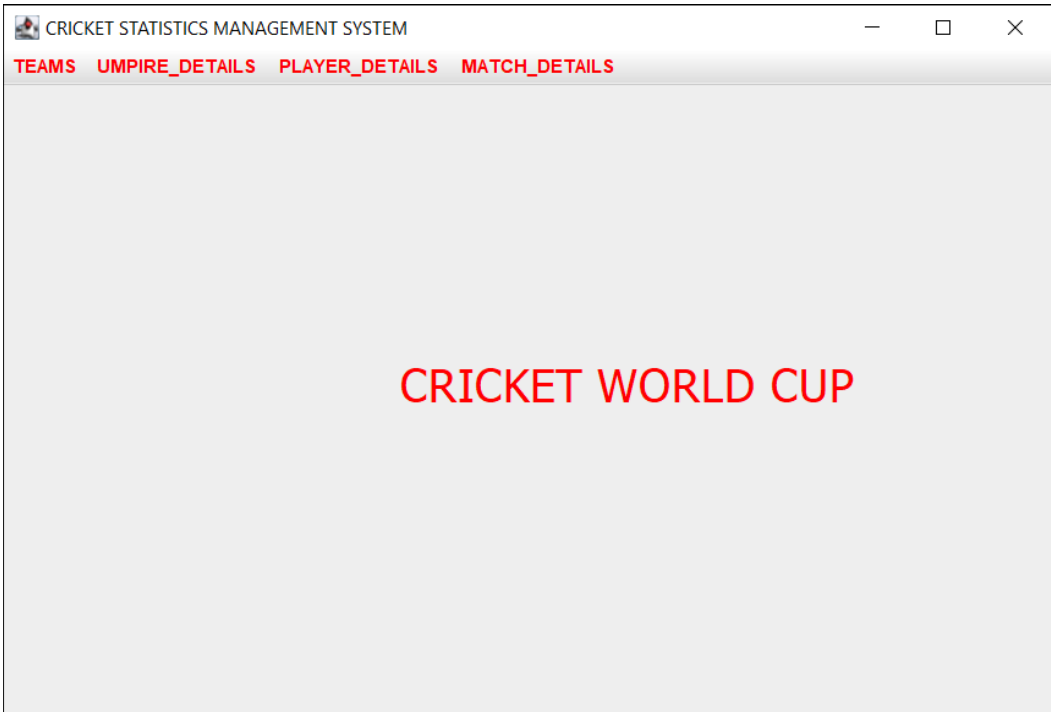
G.Anish <1602-19-737-003>

M.Rakesh <1602-19-737-03>

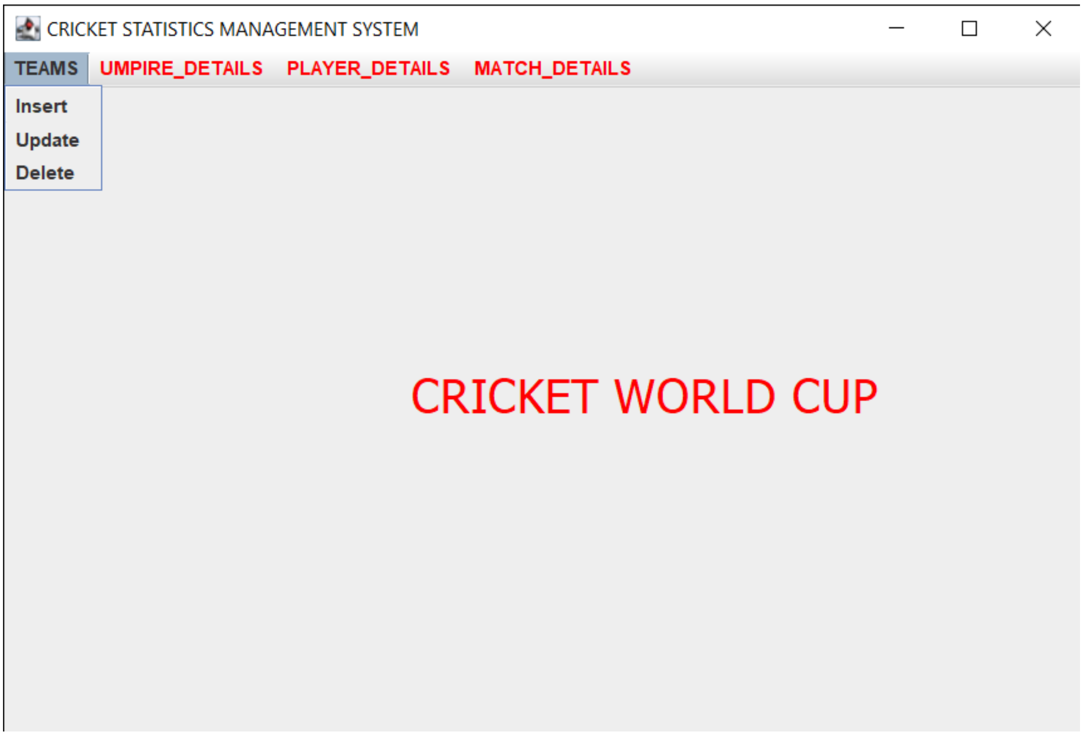
N.Rohith kumar <1602-19-737-033>

TESTING AND RESULT

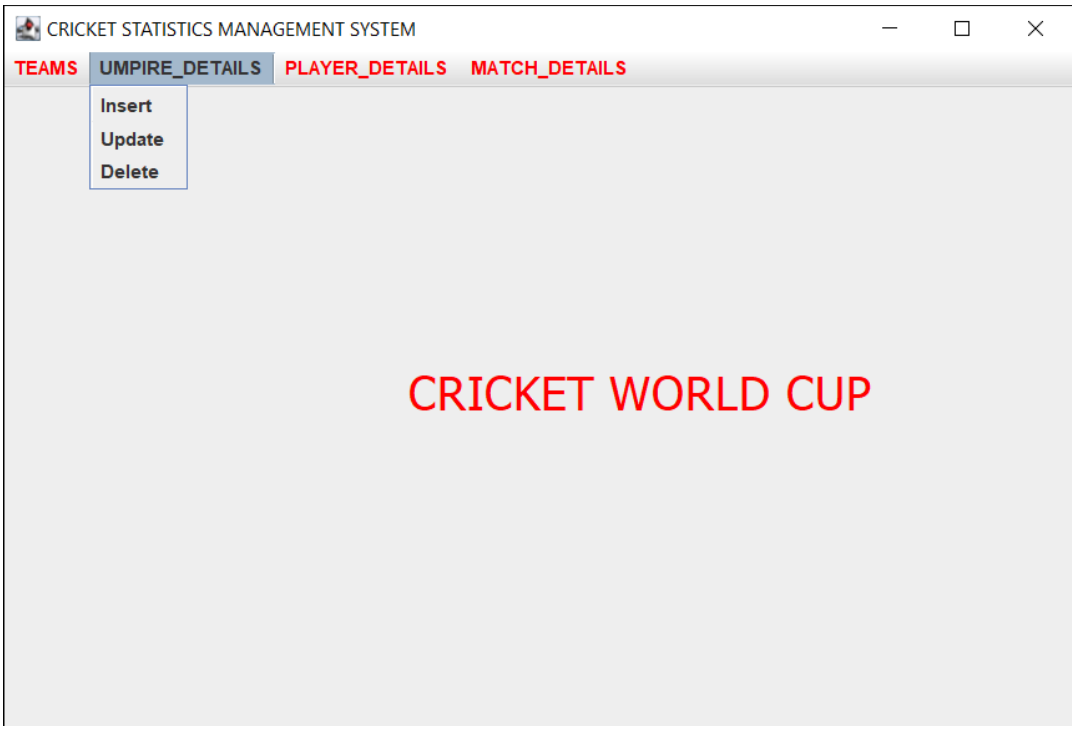
MAIN MENU ITEMS



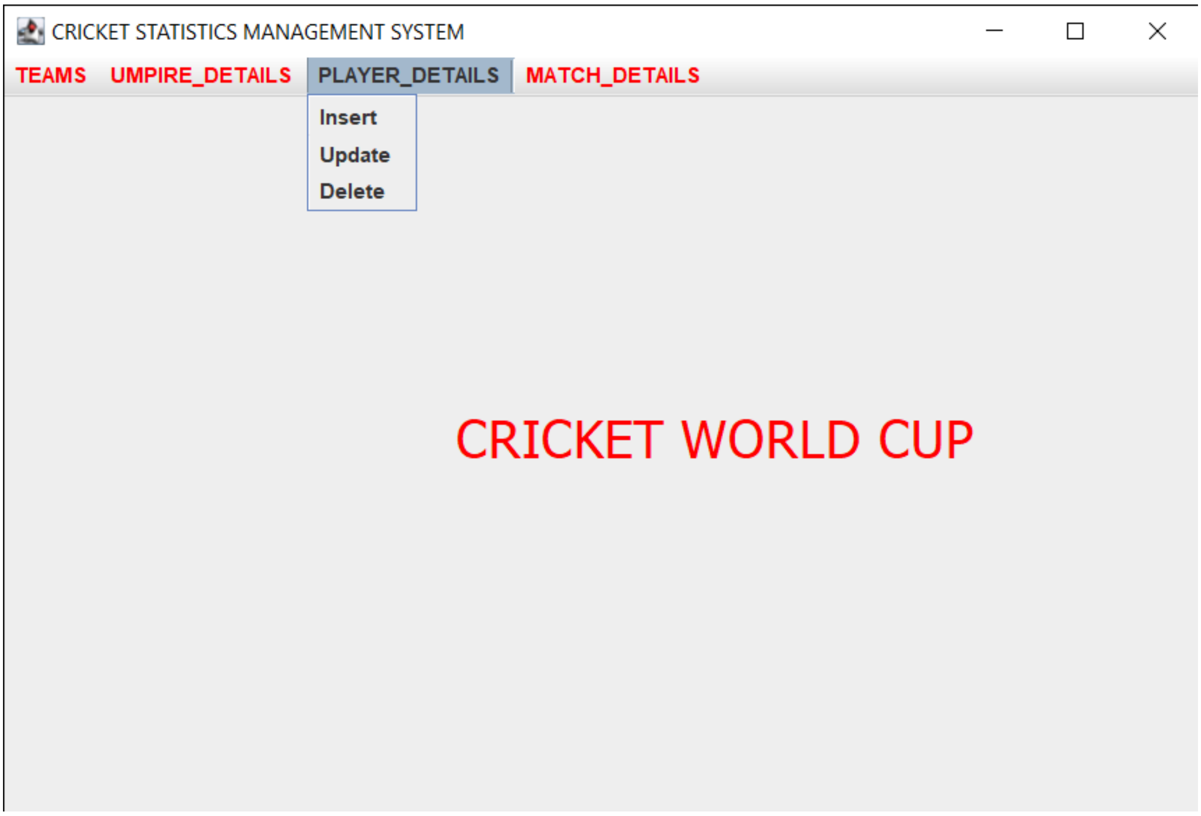
TEAMS DETAILS MENU ITEMS



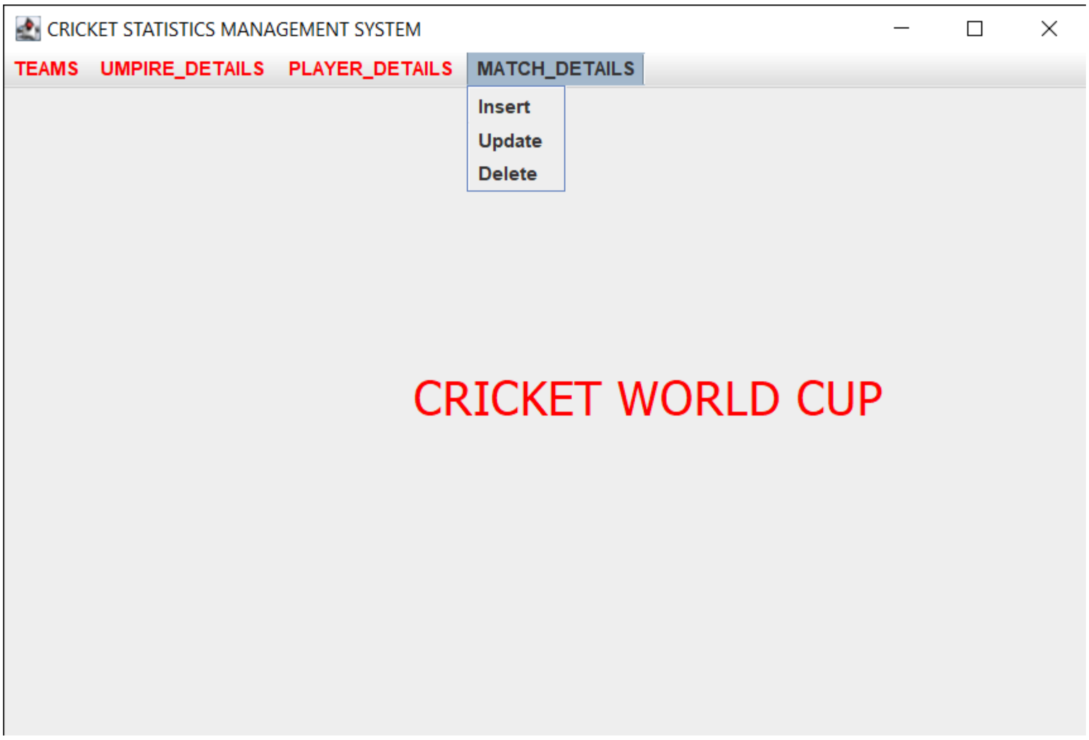
UMPIRE DETAILS MENU ITEMS



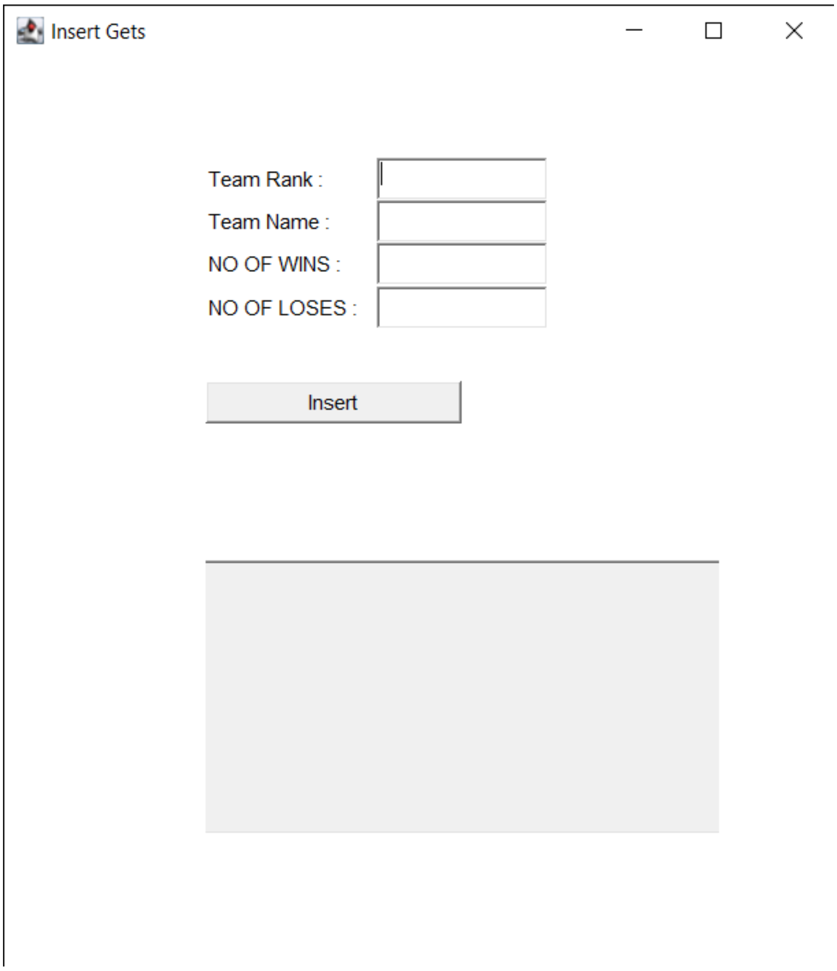
PLAYER DETAILS MENU ITEMS



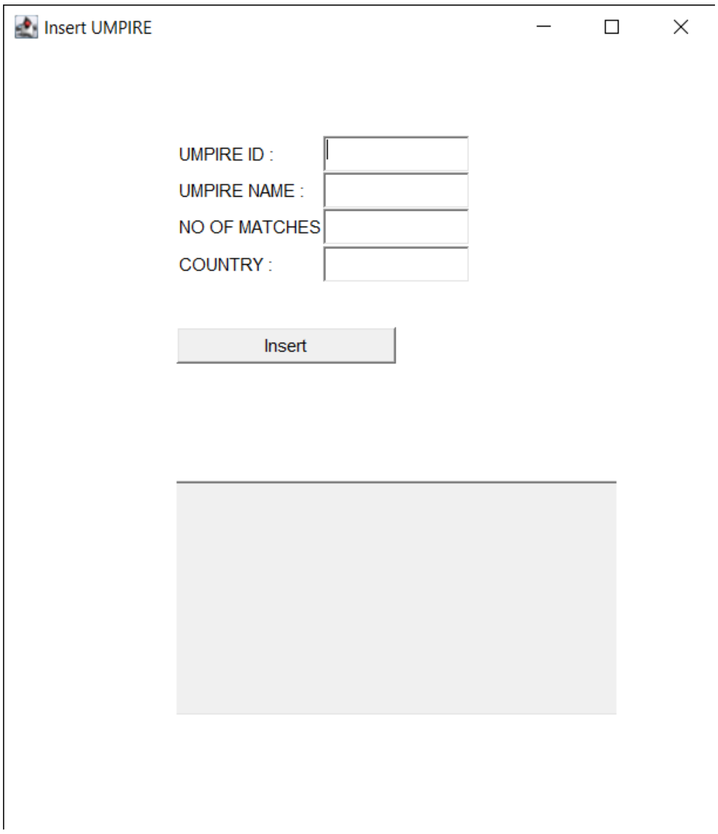
MATCH DETAILS MENU ITEMS



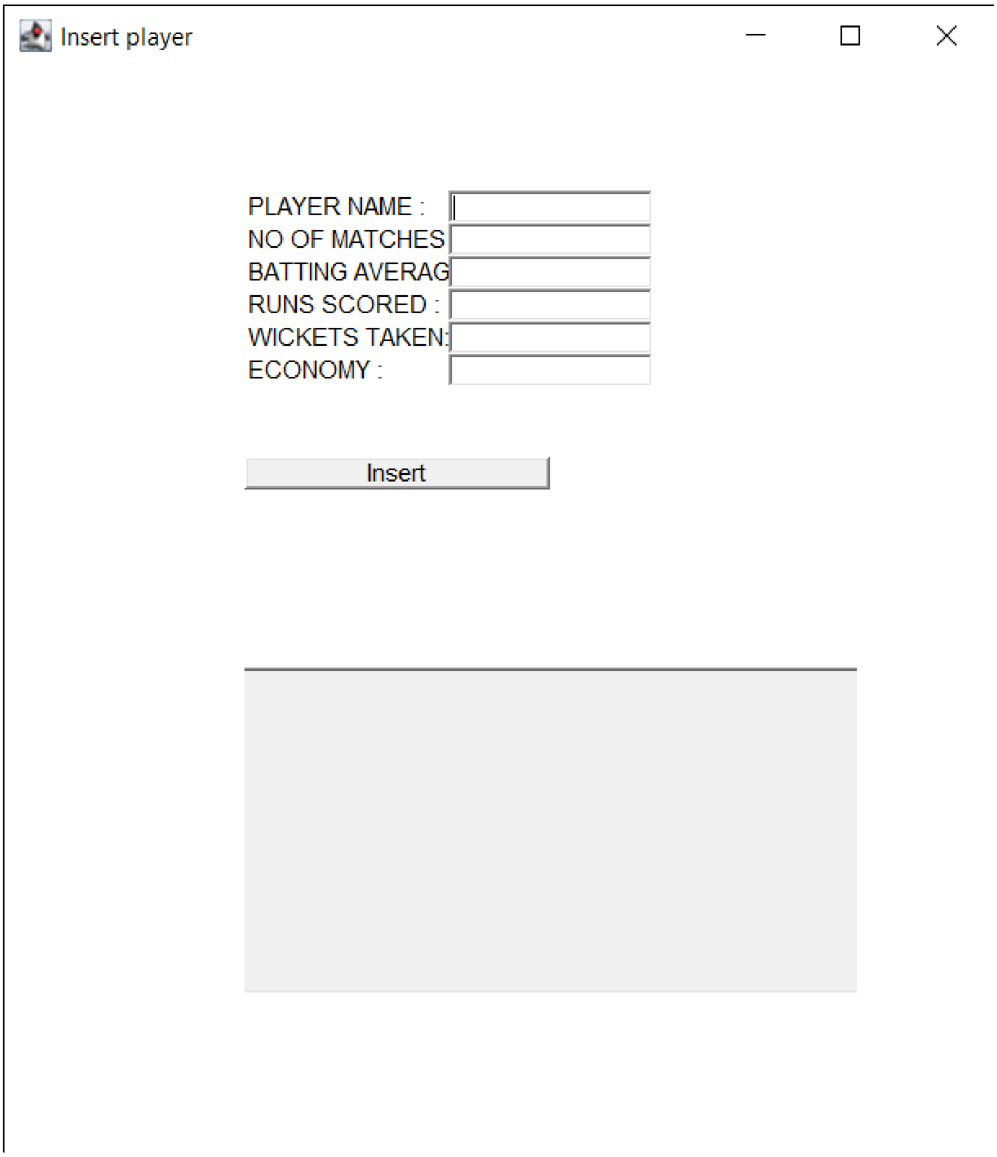
INSERTING TEAMS DETAILS



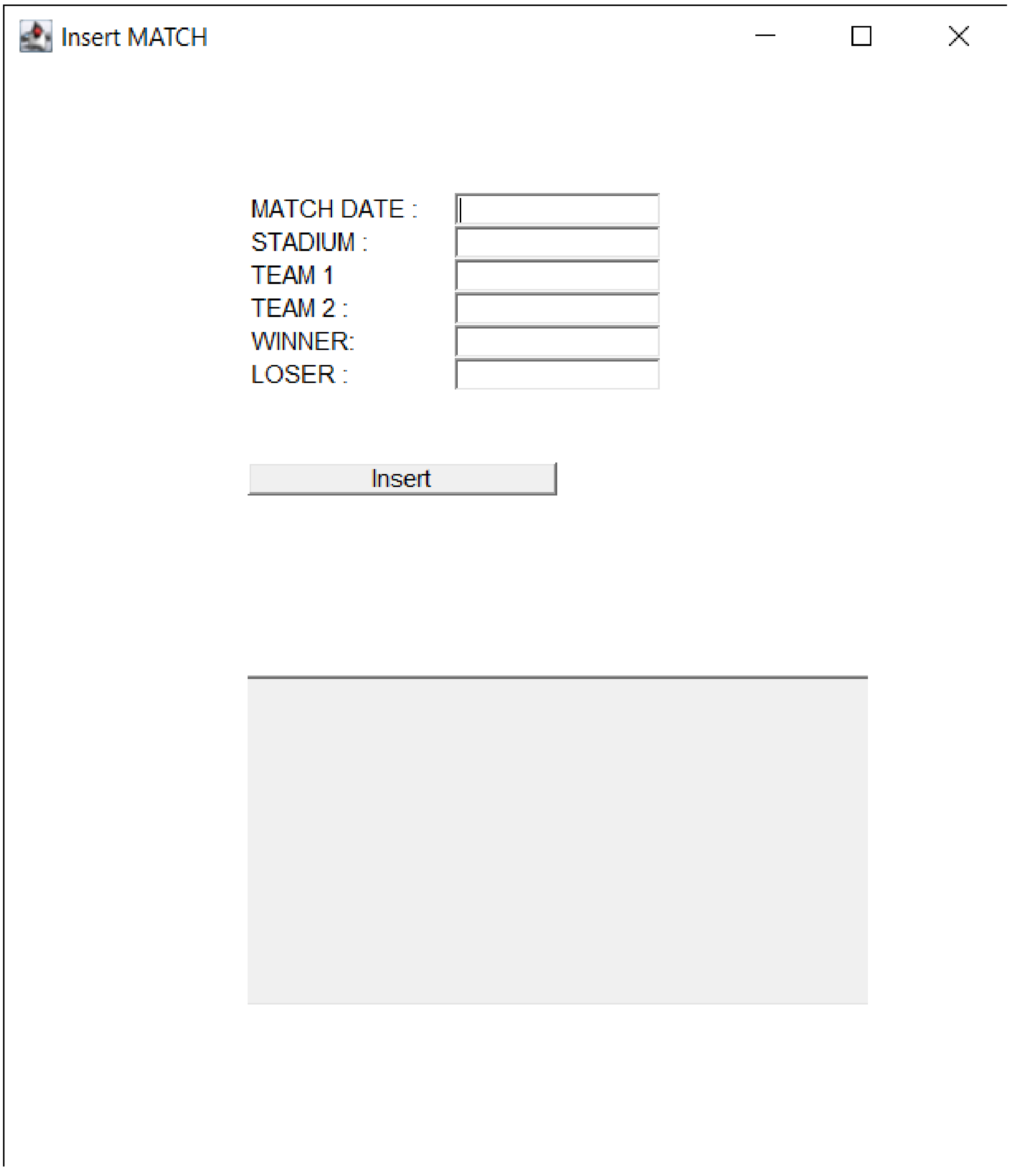
INSERTING UMPIRE DETAILS



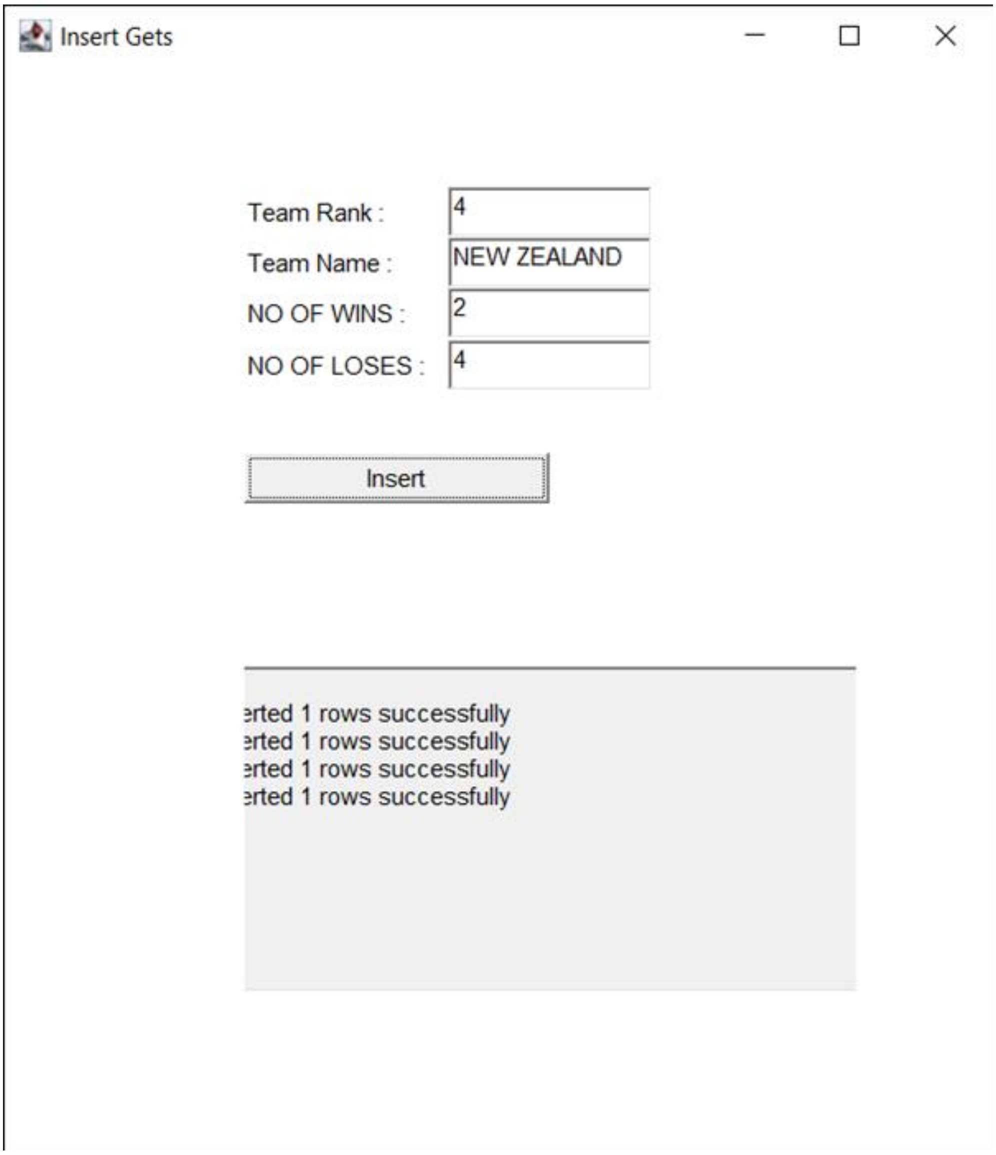
INSERTING PLAYER DETAILS



INSERTING MATCH DETAILS



OUTPUTS FOR INSERTING DETAILS



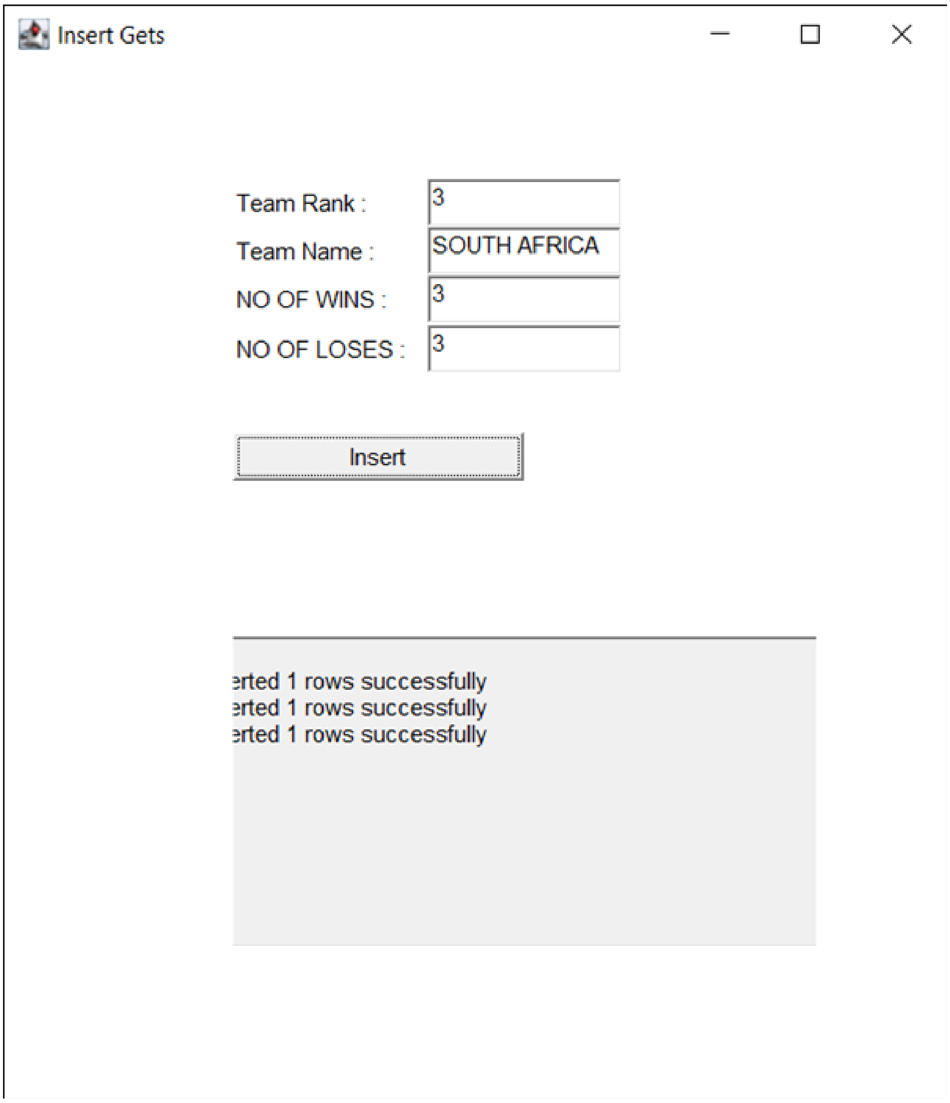
Team Rank : 4

Team Name : NEW ZEALAND

NO OF WINS : 2

NO OF LOSES : 4

```
erted 1 rows successfully
erted 1 rows successfully
erted 1 rows successfully
erted 1 rows successfully
```



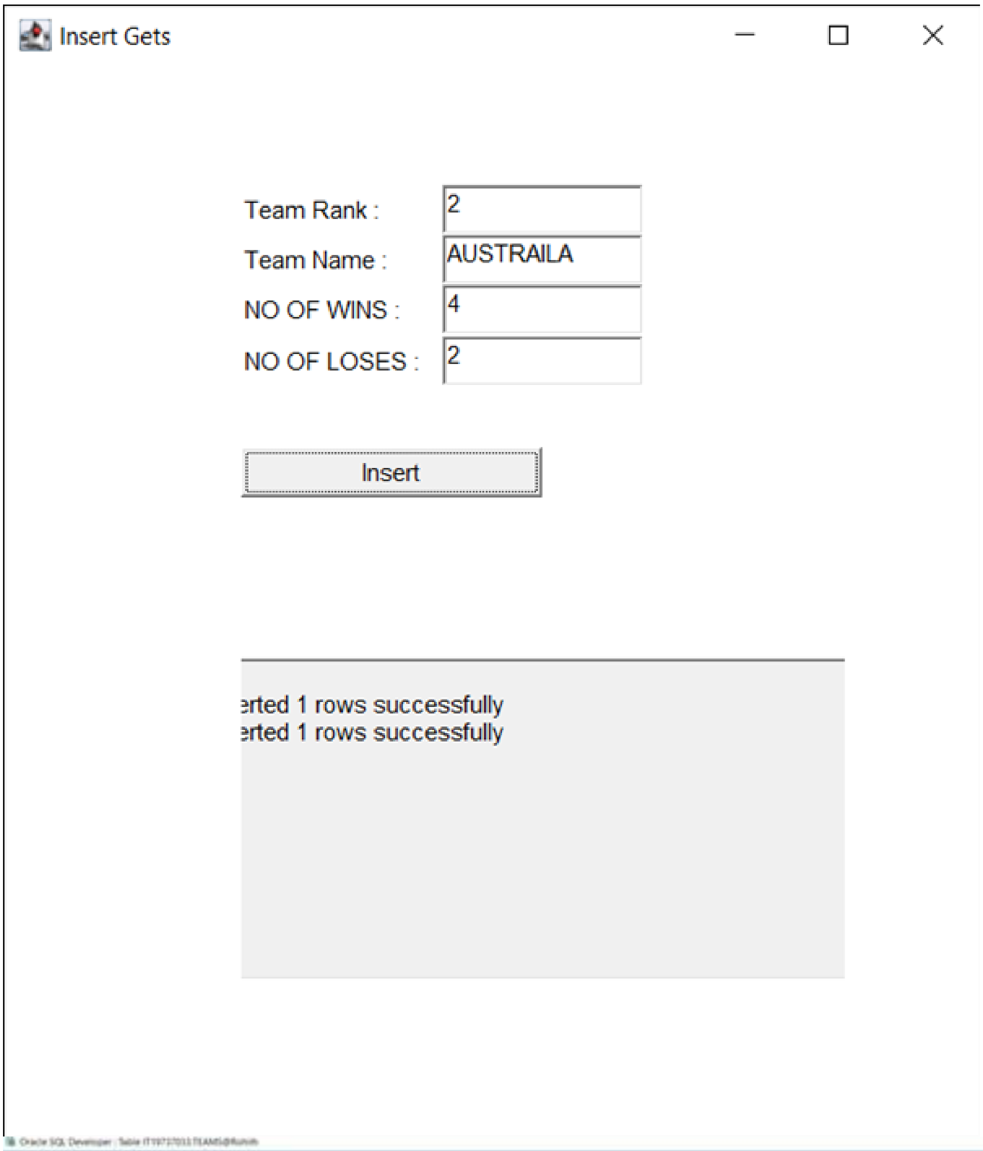
Team Rank : 3

Team Name : SOUTH AFRICA

NO OF WINS : 3

NO OF LOSES : 3

```
erted 1 rows successfully
erted 1 rows successfully
erted 1 rows successfully
```



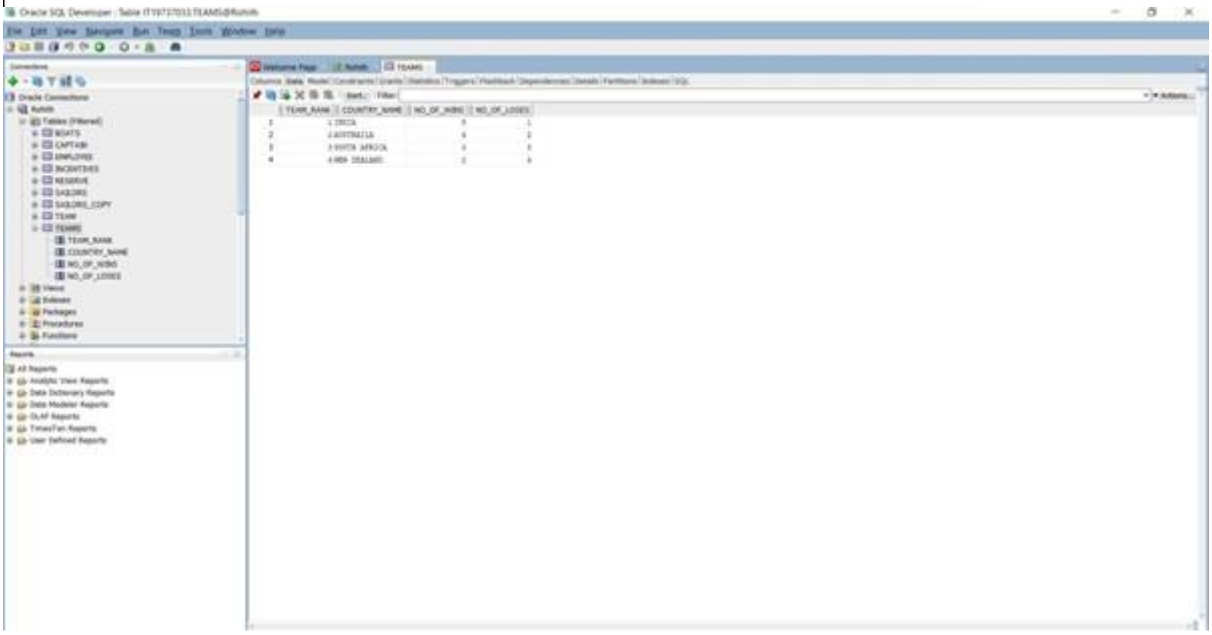
Team Rank : 2

Team Name : AUSTRAILA

NO OF WINS : 4

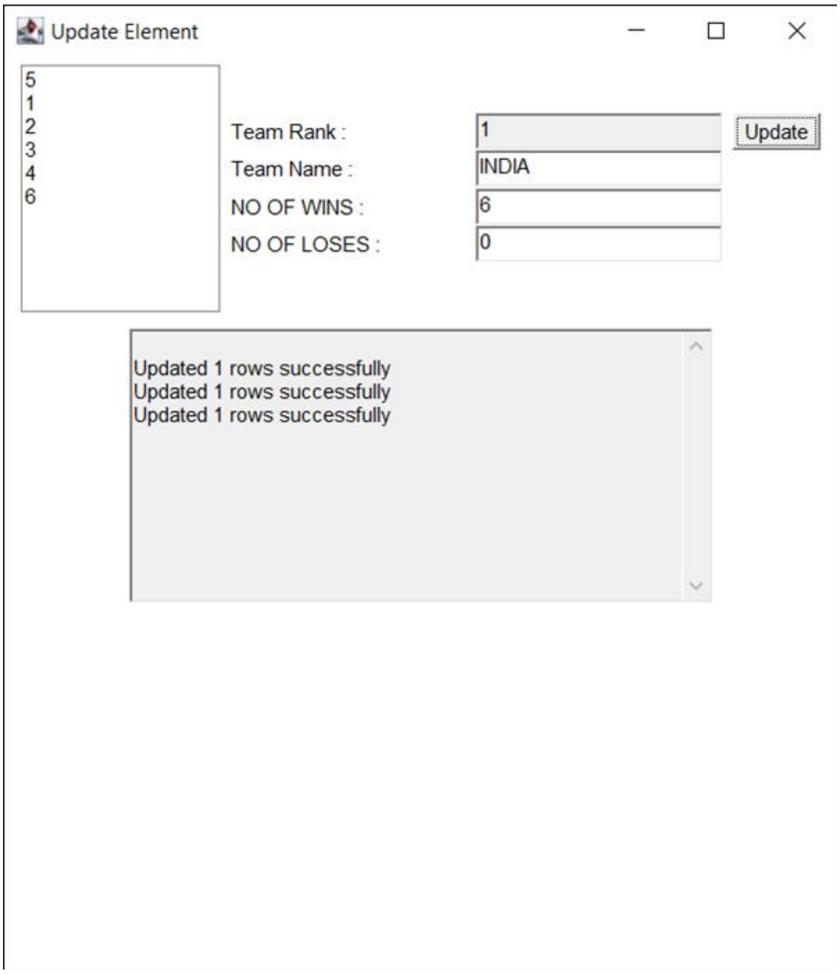
NO OF LOSES : 2

```
erted 1 rows successfully
erted 1 rows successfully
```



	TEAM_RANK	COUNTRY_NAME	NO_OF_WINS	NO_OF_LOSES
1	1	INDIA	5	1
2	2	AUSTRAILA	4	2
3	3	SOUTH AFRICA	3	3
4	4	NEW ZEALAND	2	4

OUTPUTS FOR UPDATING PLAYER DETAILS



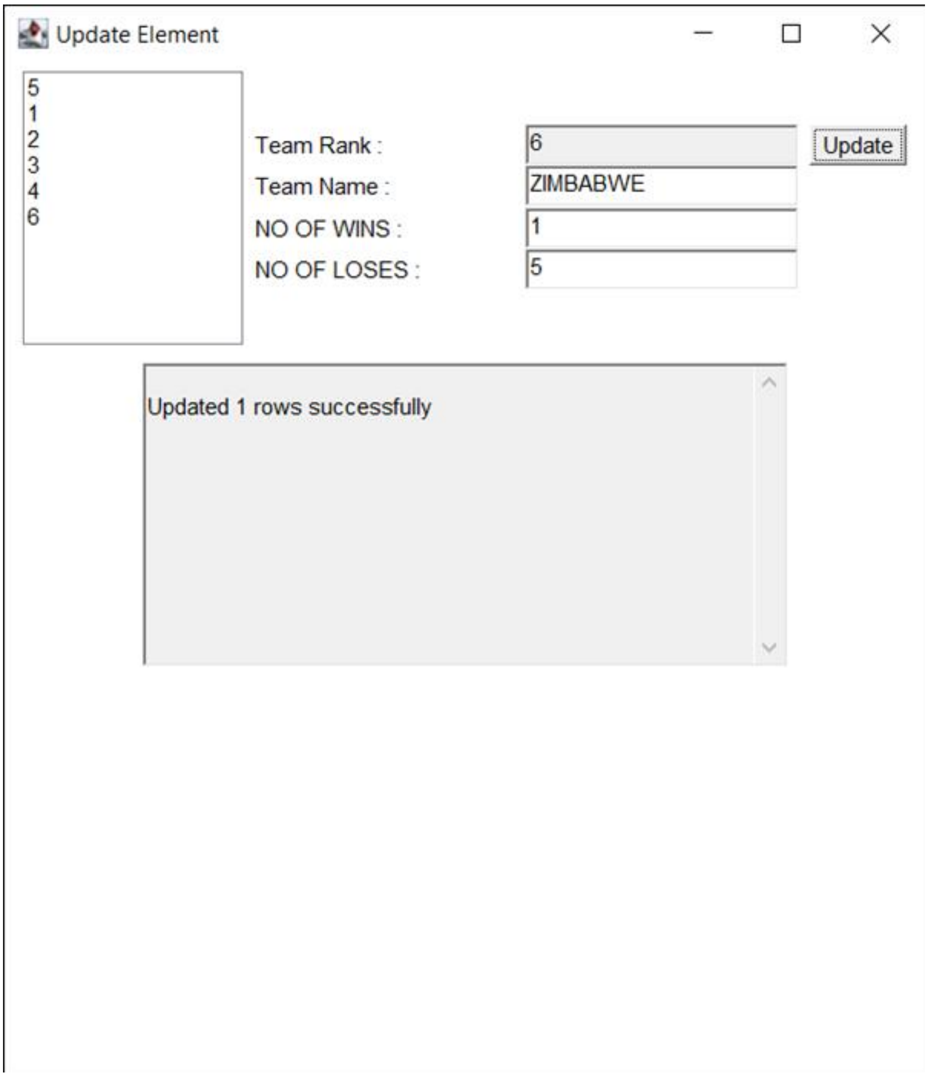
Team Rank : 1

Team Name : INDIA

NO OF WINS : 6

NO OF LOSES : 0

```
Updated 1 rows successfully
Updated 1 rows successfully
Updated 1 rows successfully
```



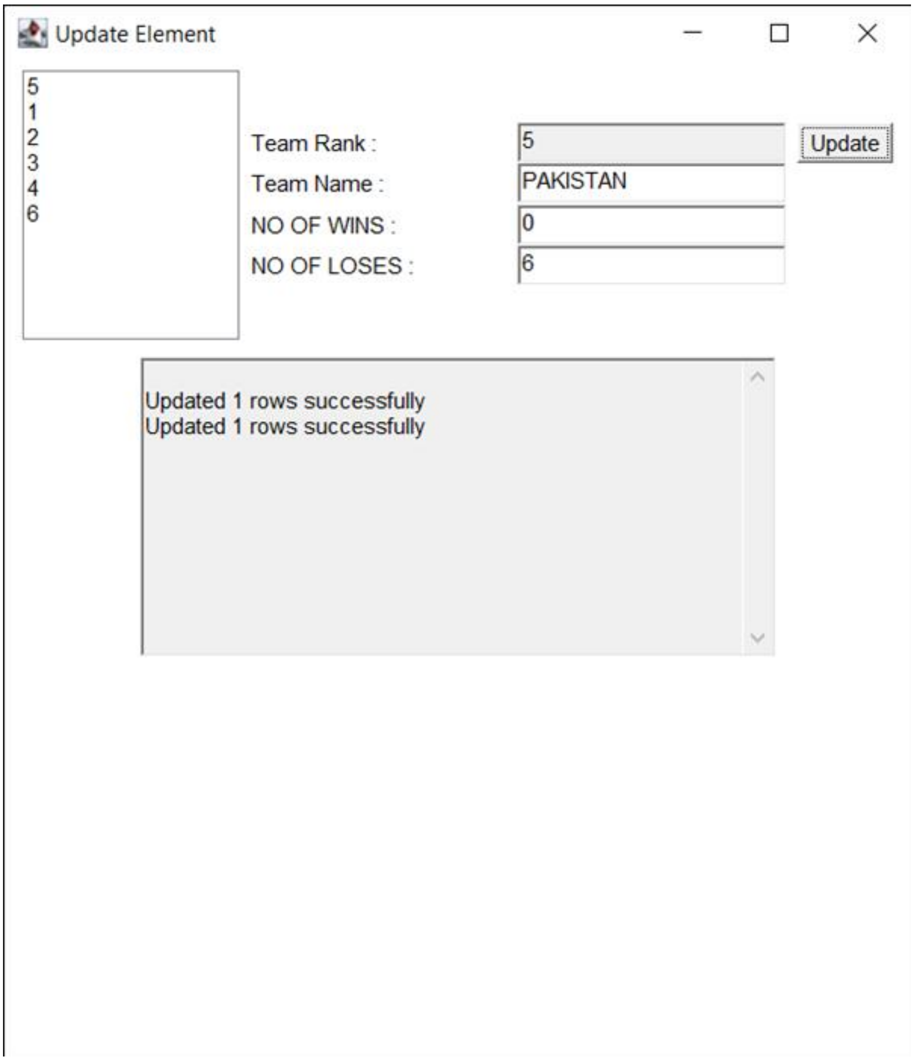
Team Rank : 6

Team Name : ZIMBABWE

NO OF WINS : 1

NO OF LOSES : 5

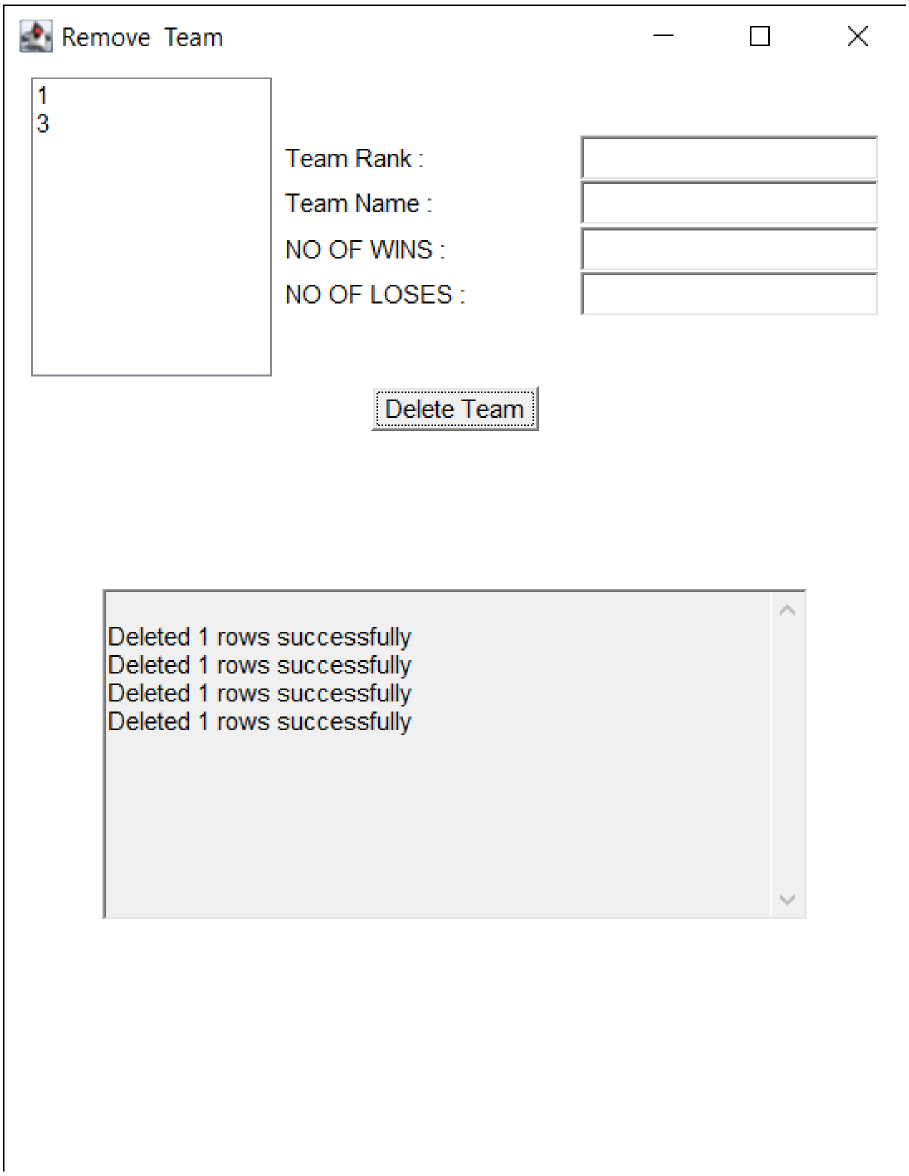
Updated 1 rows successfully



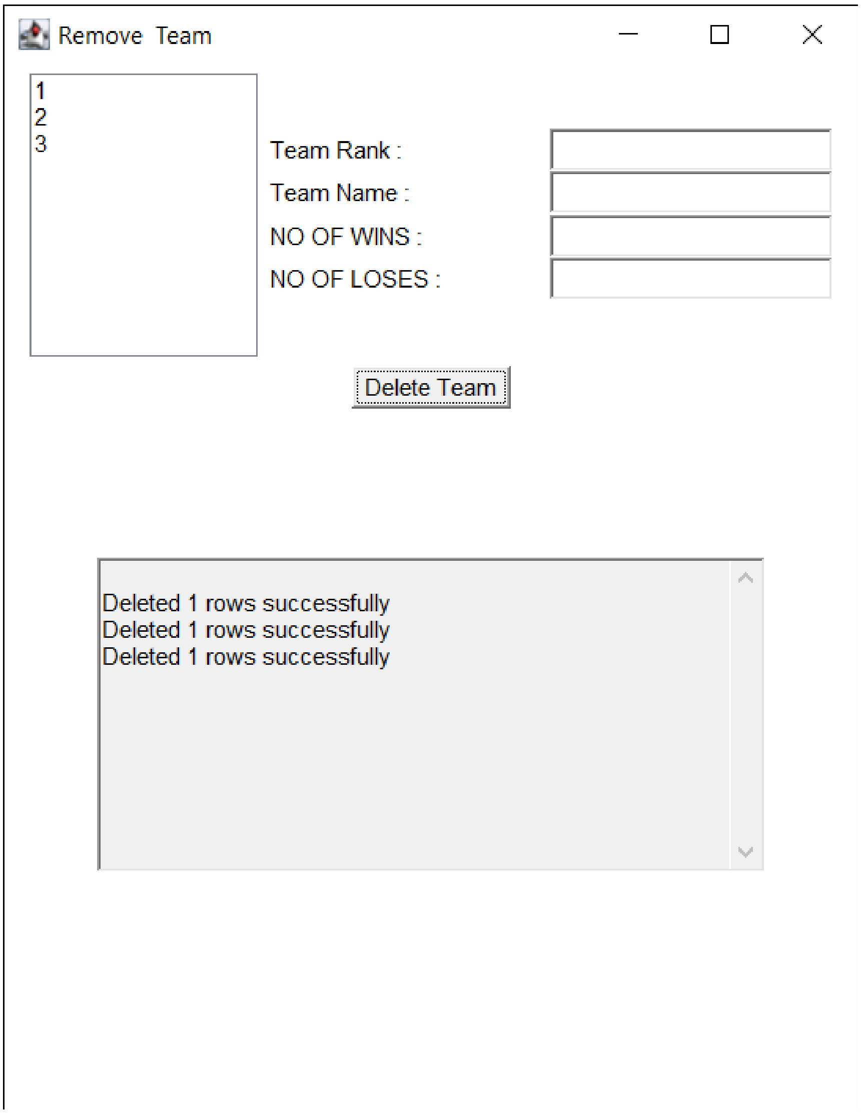
Team Rank : 5
Team Name : PAKISTAN
NO OF WINS : 0
NO OF LOSES : 6

```
Updated 1 rows successfully
Updated 1 rows successfully
```

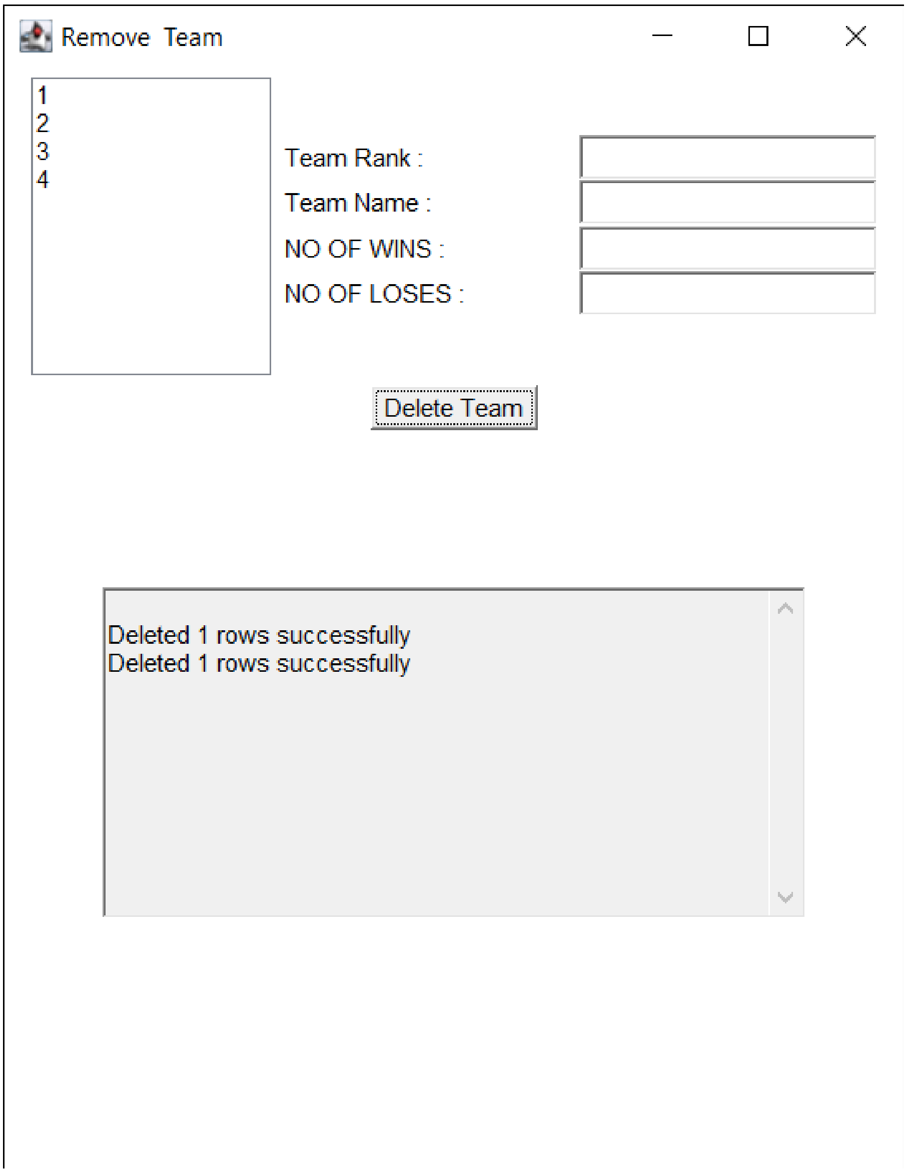
OUTPUTS FOR DELETING PLAYER DETAILS



```
Deleted 1 rows successfully
Deleted 1 rows successfully
Deleted 1 rows successfully
Deleted 1 rows successfully
```



```
Deleted 1 rows successfully
Deleted 1 rows successfully
Deleted 1 rows successfully
```



Remove Team

1
2
3
4

Team Rank :

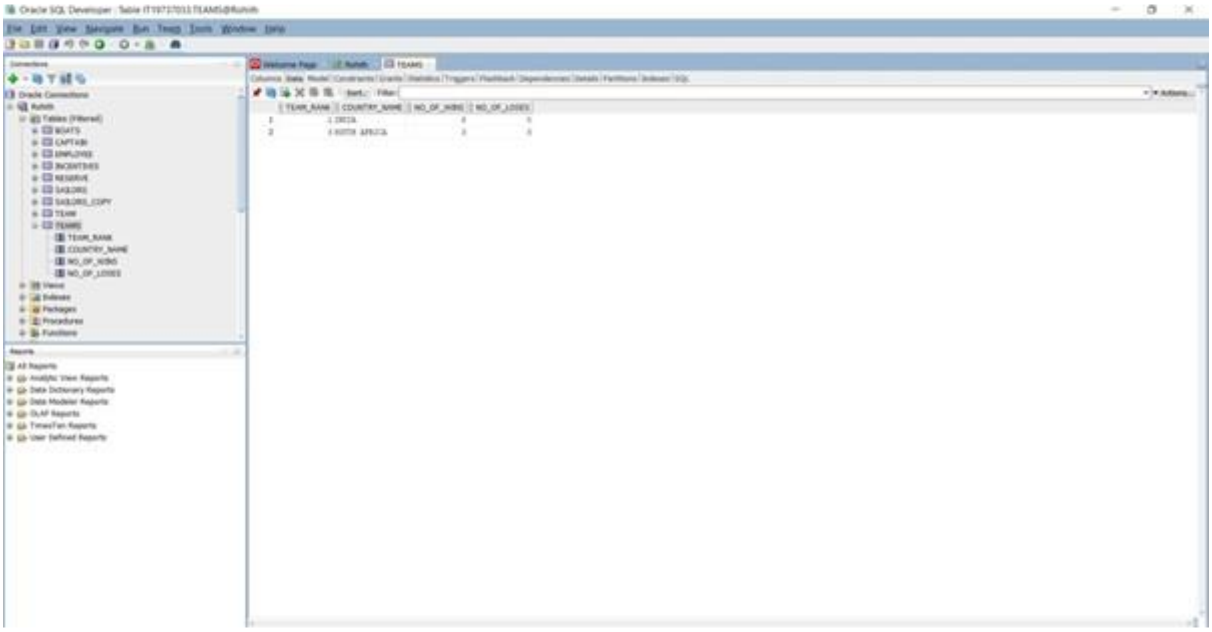
Team Name :

NO OF WINS :

NO OF LOSES :

Delete Team

```
Deleted 1 rows successfully
Deleted 1 rows successfully
```



	TEAM_RANK	COUNTRY_NAME	NO_OF_WINS	NO_OF_LOSES
1	1	INDIA	[illegible]	[illegible]
2	4	SOUTH AFRICA	[illegible]	[illegible]

DISCUSSION AND FUTURE WORK:-

This project on Cricket World Cup is the automation of getting the details of statistics of cricket. The system is able to provide much information like player's details, match details and the team details. The system allows us to add records when a there is a match between two countries. It also allows to delete and update the records based on player's gameplay . This project has guided our path through various aspects of computer science where developing online application plays a major role.

In future the most probable aspects would be including more tables like the coach details, partnering details etc. Lastly, there could be a room for including advanced software and other technologies that could make the project more purposeful and better for future use.

REFERENCES

https://www.oracle.com/in/database/technologies/jdbc-migration.html

https://www.youtube.com/watch?v=ea0HhETxOIs&t=1359s

G.Anish <1602-19-737-003>

M. Rakesh <1602-19-737-031>

N.Rohith kumar <1602-19-737-033>