OBJECTIVE OF THE PROJECT:

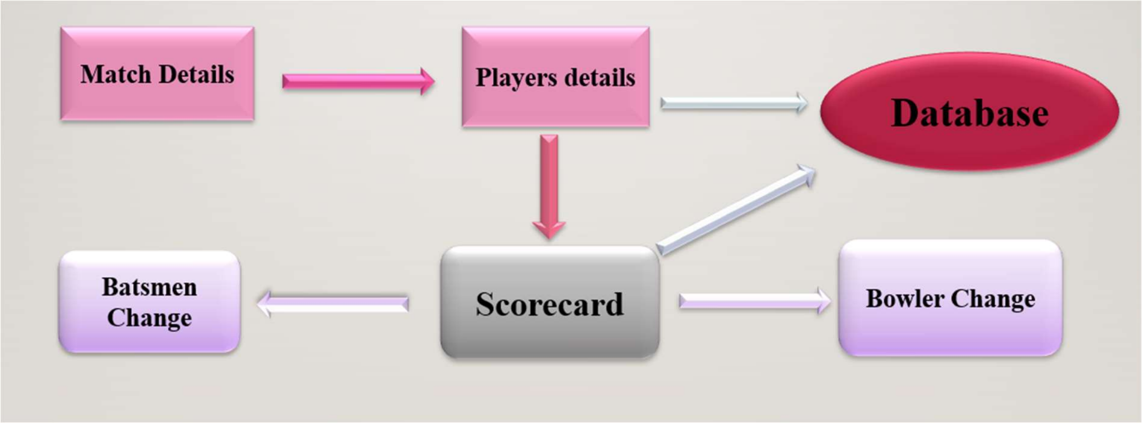
The proposed system "Cricket Score Card " is utilized to get the live cricket Score update. Main objective of the project is to develop the software for the event requirement. In this project used to maintain the details in database so easily retrieve the details from the database. This system also having the details of player and match are maintained in the repository management system.

As per the current work in our project we have created an interface for an admin/ match referee or the person who update the score while having a cricket match. We made a user-friendly Graphical User Interface which makes the user or the viewers can easily understand regarding the scores of the match. The Data Generated during the match will be collected and saved as a record within the MYSQL Database which may help in knowing the data of the respective match in the future. There will be different tables that may combine the both players and the match data.

Key Features in the Project:

- List of the Players
- Striking Batsman
- Bowler to the Batsman
- Ball by Ball record

Data Flow Diagram of the Project:



DEFINITIONS OF THE ELEMENTS USED IN THE PROJECT:

1. SWINGS:

- **Java Swings** is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*.
- It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.
- Unlike AWT, Java Swing provides platform-independent and lightweight components.
- The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

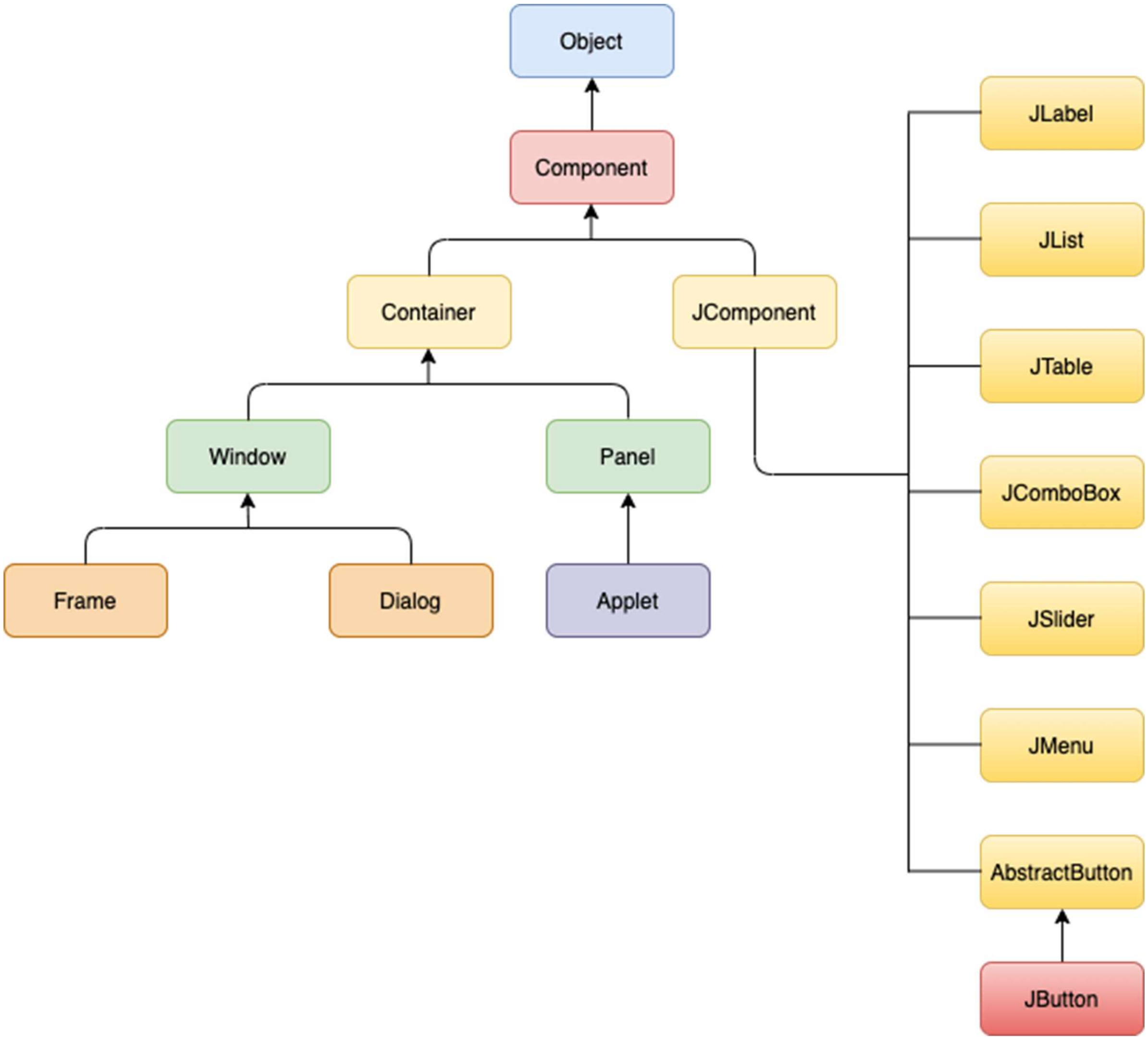
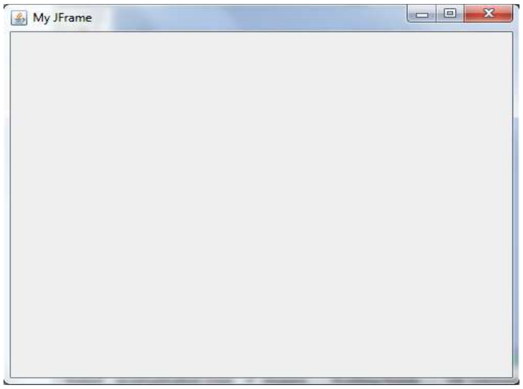


Fig 2.1 - Hierarchy of Java Swing classes

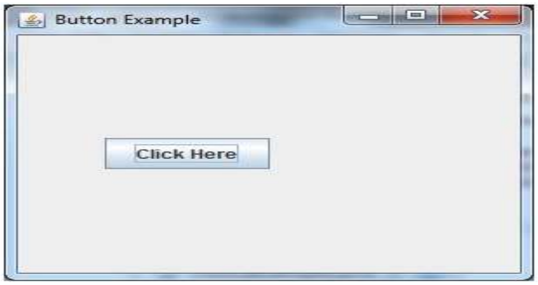
These are the Java Swing API's that we used in our project:

- **Java JFrame**



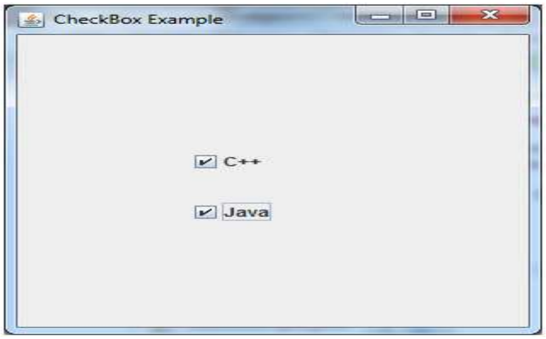
The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.Unlike Frame, JFrame has the option to hide or close the window with the help of setDefaultCloseOperation(int) method.

- **Java JButton**



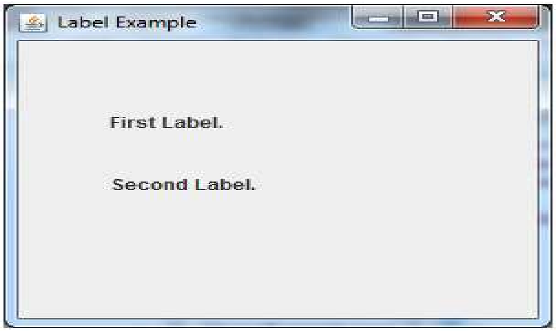
The JButton class is used to create a labelled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits Abstract Button class.

- **Java JCheckBox**



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

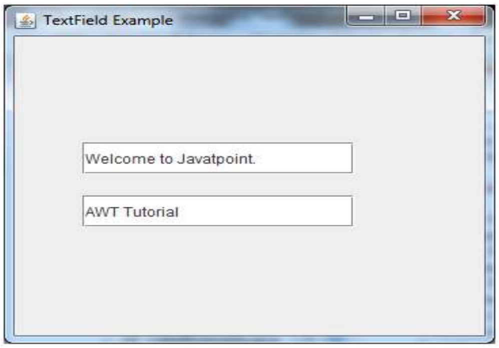
- **Java JLabel**



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

- **Java JTextField**

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



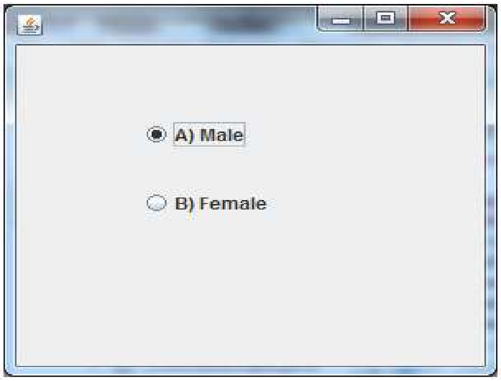
Welcome to Javatpoint.

AWT Tutorial

- **Java JRadioButton**

The JRadioButton class is used to create a radio button. It is used to choose one option from multiple options. It is widely used in exam systems or quiz.

It should be added in ButtonGroup to select one radio button only.



● A) Male

○ B) Female

2. JDBC Driver:

To connect Java application with the MySQL database, we need to follow 4 following steps.In this Project we used MySql as the database. So we need to know following informations for the mysql database:

- **Driver class:** The driver class for the mysql database is **com.mysql.jdbc.Driver**.
- **Connection URL:** The connection URL for the mysql database is **jdbc:mysql://localhost:3306/***database_name*

  where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and *CricketData* is the database name. We may use any database, in such case, we need to replace the *CricketData* with our database name.

- **Username:** The default username for the mysql database is **root**.
- **Password:** It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.

In our Project we have used the JDBC Driver to Connect the Java program to a Database

Packages:

A **java package** is a group of similar types of classes, interfaces and sub-packages.Package in java can be categorized in two form, built-in package and user-defined package.There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Advantage of Java Package

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.

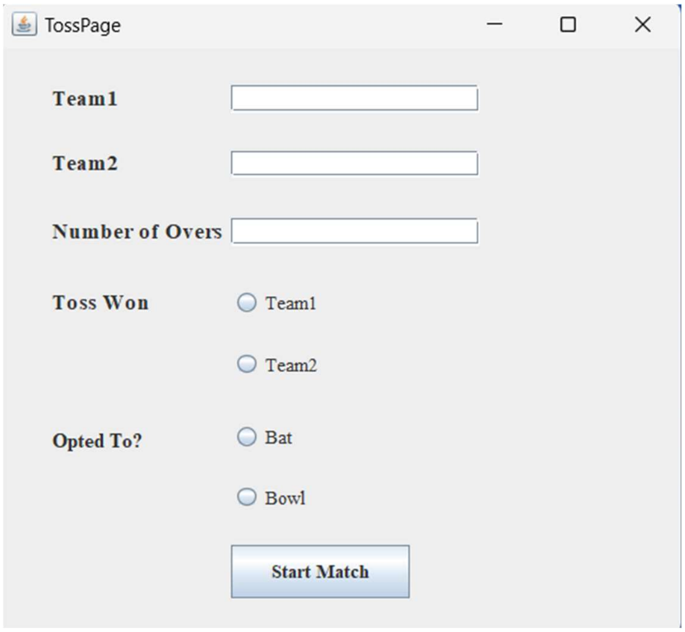
2) Java package provides access protection.

3) Java package removes naming collision.

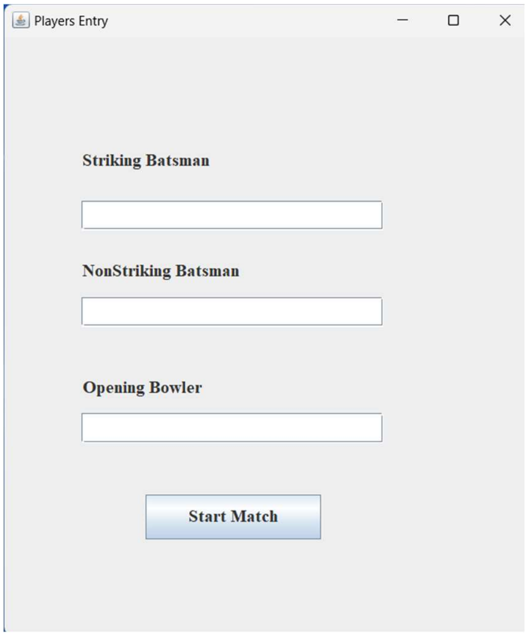
In our Project we have created a package called Cricket_Score and Saved all the .class , .java files into that package.

DESIGN

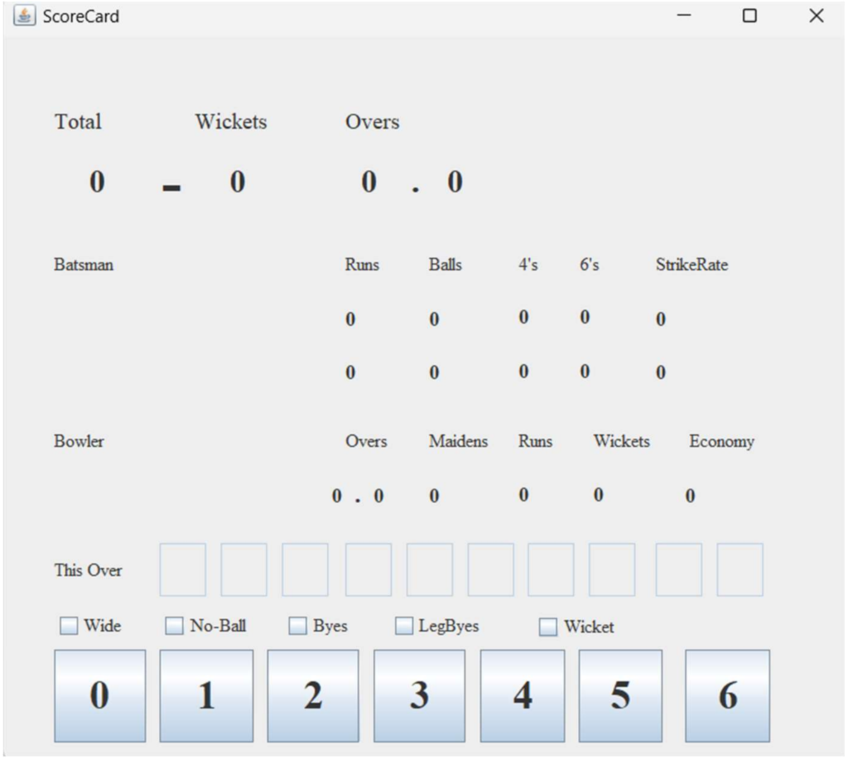
3.1 Screens



Toss Page



Players Entry



ScoreCard

IMPLEMENTATION

4.1 Code:

4.1.1 Tosspage Code:

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JRadioButton;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class tosspage extends JFrame {
    private JPanel contentPane;
    private JTextField team1;
    private JTextField team2;
    private JTextField Overs;
    private tosspage tossFrame;
    public static void main(String[] args) {

            public void run() {
                try {
                    tosspage frame = new tosspage();
                    frame.setVisible(true);
                    frame.tossFrame=frame;
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    public tosspage()
```

```
{

        setTitle("TossPage");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 444, 408);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Team1");
        lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 14));
        lblNewLabel.setBounds(33, 23, 46, 17);
        contentPane.add(lblNewLabel);

        team1 = new JTextField();
        team1.setColumns(10);
        team1.setBounds(146, 24, 157, 17);
        contentPane.add(team1);

        JLabel lblTeam = new JLabel("Team2");
        lblTeam.setFont(new Font("Times New Roman", Font.BOLD, 14));
        lblTeam.setBounds(33, 64, 46, 17);
        contentPane.add(lblTeam);

        team2 = new JTextField();
        team2.setColumns(10);
        team2.setBounds(146, 65, 157, 17);
        contentPane.add(team2);

        JLabel overs = new JLabel("Number of Overs");
        overs.setFont(new Font("Times New Roman", Font.BOLD, 14));
        overs.setBounds(33, 103, 116, 25);
```

```
contentPane.add(overs);

Overs = new JTextField();



Overs.setColumns(10);
Overs.setBounds(146, 108, 157, 17);
contentPane.add(Overs);

JLabel lblTossWon = new JLabel("Toss Won");
lblTossWon.setFont(new Font("Times New Roman", Font.BOLD, 14));
lblTossWon.setBounds(33, 148, 80, 25);
contentPane.add(lblTossWon);

JRadioButton team1toss = new JRadioButton("Team1");
team1toss.setFont(new Font("Times New Roman", Font.PLAIN, 12));
team1toss.setBounds(146, 151, 113, 21);
contentPane.add(team1toss);

JRadioButton team2toss = new JRadioButton("Team2");
team2toss.setFont(new Font("Times New Roman", Font.PLAIN, 12));
team2toss.setBounds(146, 190, 113, 21);
contentPane.add(team2toss);

JLabel lblOptedTo = new JLabel("Opted To?");
lblOptedTo.setFont(new Font("Times New Roman", Font.BOLD, 13));
lblOptedTo.setBounds(33, 236, 80, 25);
contentPane.add(lblOptedTo);

JRadioButton batopt = new JRadioButton("Bat");
batopt.setFont(new Font("Times New Roman", Font.PLAIN, 12));
batopt.setBounds(146, 236, 113, 21);
contentPane.add(batopt);
```

```
		JRadioButton bowlopt = new JRadioButton("Bowl");
		bowlopt.setFont(new Font("Times New Roman", Font.PLAIN, 12));
		bowlopt.setBounds(146, 274, 113, 21);
		contentPane.add(bowlopt);
		JButton start = new JButton("Start Match");
		start.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				matchDetails md=new matchDetails();
				try
				{
					boolean z=team1toss.isSelected();
					boolean y=batopt.isSelected();

	md.setData(team1.getText(),team2.getText(),Overs.getText(),z,y);
					System.out.println();
				}
				catch (Exception e1)
				{
					e1.printStackTrace();
				}



				EventQueue.invokeLater(new Runnable() {
					public void run() {
						try {
							Player_Entry frame = new
Player_Entry(team1.getText(),team2.getText());
							frame.setVisible(true);
						} catch (Exception e) {
							e.printStackTrace();
						}
					}
				});
				tossFrame.setVisible(false);
```

```
            }
        });
        start.setFont(new Font("Times New Roman", Font.BOLD, 12));
        start.setBounds(146, 314, 113, 34);
        contentPane.add(start);
    }
}
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
public class matchDetails{
    public void setData(String Team1,String Team2,String overs,boolean t1toss,boolean optbat)
throws Exception{
        try{
            int k=Integer.parseInt(overs);
            Class.forName("com.mysql.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/cricket","root","Abhilash@213");
            Statement stmt=con.createStatement();
            String toss;
            toss=Team1;
            if(!t1toss)
                toss=Team2;
            String opt="bat";
            if(!optbat)
                opt="bowl";
            stmt.execute("insert into cricket.match(team1,team2,toss,opt,overs)
values('"+Team1+"','"+Team2+"','"+toss+"','"+opt+"',"+k+");");
            con.close();
        }
        catch(Exception e){
            System.out.println(e);}
    }
}
```

4.1.2 PlayerEntry Code:

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class Player_Entry extends JFrame {
    private JPanel contentPane;
    private JTextField txtIn;
    private JTextField strike;
    private JLabel lblNonstrikingBatsman;
    private JTextField nonstrike;
    private JLabel lblOpeningBowler;
    private JTextField open;
    private Player_Entry temp;
    public static void main(String[] args) {
            public void run() {
                try {
                    Player_Entry frame = new Player_Entry(null,null);
                    frame.setVisible(true);
                    frame.temp=frame;
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    public Player_Entry(String t1,String t2) {
        setTitle("Players Entry");
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(100, 100, 710, 546);
contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("Striking Batsman");
lblNewLabel.setBounds(71, 100, 179, 27);
lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 16));
contentPane.add(lblNewLabel);

txtIn = new JTextField();
txtIn.setBounds(23, 28, 446, 48);
txtIn.setFont(new Font("Times New Roman", Font.BOLD, 18));
txtIn.setEditable(false);
contentPane.add(txtIn);
txtIn.setColumns(10);
txtIn.setBorder(null);

strike = new JTextField();
strike.setBounds(71, 150, 275, 27);
strike.setFont(new Font("Times New Roman", Font.PLAIN, 16));
contentPane.add(strike);
strike.setColumns(10);

lblNonstrikingBatsman = new JLabel("NonStriking Batsman");
lblNonstrikingBatsman.setBounds(71, 201, 179, 27);
lblNonstrikingBatsman.setFont(new Font("Times New Roman", Font.BOLD, 16));
contentPane.add(lblNonstrikingBatsman);

nonstrike = new JTextField();
nonstrike.setBounds(71, 238, 275, 27);
```

```
        nonstrike.setFont(new Font("Times New Roman", Font.PLAIN, 16));
        nonstrike.setColumns(10);
        contentPane.add(nonstrike);

        lblOpeningBowler = new JLabel("Opening Bowler");
        lblOpeningBowler.setBounds(71, 307, 179, 27);
        lblOpeningBowler.setFont(new Font("Times New Roman", Font.BOLD, 16));
        contentPane.add(lblOpeningBowler);

        open = new JTextField();
        open.setBounds(71, 344, 275, 27);
        open.setFont(new Font("Times New Roman", Font.PLAIN, 16));
        open.setColumns(10);
        contentPane.add(open);

        JButton start = new JButton("Start Match");
        start.setBounds(129, 418, 160, 41);
        start.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e)
            {
                getting obj=new getting();
                obj.SqlGet(strike.getText());
                obj.SqlGet(nonstrike.getText());
                obj.SqlGet(open.getText());

                EventQueue.invokeLater(new Runnable() {
                    public void run() {
                        try {
    ScoreCard window = new
ScoreCard(strike.getText(),nonstrike.getText(),open.getText(),t1,t2);
                            window.Frame.setVisible(true);

                        } catch (Exception e) {
                            e.printStackTrace();
                        }
                    }
                });
                temp.setVisible(false);
            }
        });
        start.setFont(new Font("Times New Roman", Font.BOLD, 16));
        contentPane.add(start);
    }
}
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
    public class database{
        public void setData(String pname,String runs,String balls,String four,String six) throws
Exception
        {
            try
            {
                Class.forName("com.mysql.cj.jdbc.Driver");
                Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/cricket","root","Abhilash@213");
                Statement stmt=con.createStatement();
                stmt.execute("insert into cricket.bat(batname,runs,balls,four,six)
values('"+pname+"','"+runs+"','"+balls+"','"+four+"','"+six+"');");
                con.close();
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
```

4.1.3 ScoreCard Code:

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JCheckBox;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.SwingConstants;
public class ScoreCard
{
	public JFrame Frame;
	private JTextField runs;
	private JTextField wickets;
	private JTextField over;
	private JTextField ball;
	private JTextField batsman1;
	private JTextField runs1;
	private JTextField balls1;
	private JTextField four1;
	private JTextField six1;
	private JTextField strikerate1;
	private JTextField batsman2;
	private JTextField runs2;
	private JTextField balls2;
	private JTextField four2;
	private JTextField six2;
	private JTextField strikerate2;
	private JTextField strike;
	private JTextField b1;
	private JTextField b2;
	private JTextField b3;
private JTextField b4;
private JTextField b5;
private JTextField b6;
private JTextField b7;
private JTextField b8;
private JTextField b9;
private JTextField b10;
private JLabel lblNewLabel_1_1_3;
private JCheckBox wide;
private JCheckBox noball;
private JCheckBox byes;
private JCheckBox legbyes;
private JCheckBox wicket;
private JButton one;
private JButton two;
private JButton three;
private JButton four;
private JButton five;
private JButton six;
private int ballCount=0;
private String k1,k2,k3;
private JTextField bowler;
private JTextField overs;
private JTextField maiden;
private String t1=null;
private String t2=null;
private JTextField runsb;
private JTextField wicketsb;
private JTextField economy;
private JLabel lblNewLabel_1_1_5;
private JTextField textField;
public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
                public void run() {
```

```
                try {
                        ScoreCard window = new ScoreCard(null,null,null,null,null);
                        window.Frame.setVisible(true);
                    }
                    catch (Exception e)
                    {
                        e.printStackTrace();
                    }
                }
        });
}
public ScoreCard(String k1,String k2,String k3,String t1,String t2){
        this.k1=k1;
        this.k2=k2;
        this.k3=k3;
        this.t1=t1;
        this.t2=t2;
        initialize();
}
private void initialize() {
        Frame = new JFrame();
        Frame.setBounds(100, 100, 613, 549);
        Frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Frame.getContentPane().setLayout(null);
        Frame.setTitle("ScoreCard");
        if(t1!=null) {
                JLabel match = new JLabel(t1+" vs "+t2);
                match.setHorizontalAlignment(SwingConstants.CENTER);
                match.setBounds(36, 10, 519, 38);
                match.setFont(new Font("Times New Roman", Font.BOLD, 22));
                Frame.getContentPane().add(match);
        }
        JLabel lblNewLabel_1 = new JLabel("Batsman");
        lblNewLabel_1.setBounds(36, 147, 94, 28);
```

```
lblNewLabel_1.setFont(new Font("Times New Roman", Font.PLAIN, 13));
Frame.getContentPane().add(lblNewLabel_1);
JLabel lblNewLabel_1_1 = new JLabel("Runs");
lblNewLabel_1_1.setBounds(242, 147, 49, 28);
lblNewLabel_1_1.setFont(new Font("Times New Roman", Font.PLAIN, 13));
Frame.getContentPane().add(lblNewLabel_1_1);

JLabel lblNewLabel_1_2 = new JLabel("Balls");
lblNewLabel_1_2.setBounds(301, 147, 49, 28);
lblNewLabel_1_2.setFont(new Font("Times New Roman", Font.PLAIN, 13));
Frame.getContentPane().add(lblNewLabel_1_2);

JLabel lblNewLabel_1_3 = new JLabel("4's");
lblNewLabel_1_3.setBounds(364, 147, 33, 28);
lblNewLabel_1_3.setFont(new Font("Times New Roman", Font.PLAIN, 13));
Frame.getContentPane().add(lblNewLabel_1_3);

JLabel lblNewLabel_1_4 = new JLabel("6's");
lblNewLabel_1_4.setBounds(407, 147, 33, 28);
lblNewLabel_1_4.setFont(new Font("Times New Roman", Font.PLAIN, 13));
Frame.getContentPane().add(lblNewLabel_1_4);

JLabel lblNewLabel_1_5 = new JLabel("StrikeRate");
lblNewLabel_1_5.setBounds(461, 147, 94, 28);
lblNewLabel_1_5.setFont(new Font("Times New Roman", Font.PLAIN, 13));
Frame.getContentPane().add(lblNewLabel_1_5);

runs = new JTextField();
runs.setFont(new Font("Times New Roman", Font.BOLD, 23));
runs.setEditable(false);
runs.setHorizontalAlignment(SwingConstants.CENTER);
runs.setBounds(36, 84, 65, 38);
Frame.getContentPane().add(runs);
runs.setColumns(10);
```

```
wickets = new JTextField();
wickets.setFont(new Font("Times New Roman", Font.BOLD, 23));
wickets.setEditable(false);
wickets.setHorizontalAlignment(SwingConstants.CENTER);
wickets.setBounds(135, 84, 65, 38);
wickets.setColumns(10);
Frame.getContentPane().add(wickets);

JLabel lblNewLabel_1_3_1 = new JLabel("-");
lblNewLabel_1_3_1.setBounds(111, 84, 20, 38);
lblNewLabel_1_3_1.setFont(new Font("Times New Roman", Font.PLAIN, 48));
Frame.getContentPane().add(lblNewLabel_1_3_1);

JLabel lblNewLabel_1_1_1 = new JLabel("Total");
lblNewLabel_1_1_1.setBounds(36, 46, 65, 28);
lblNewLabel_1_1_1.setFont(new Font("Times New Roman", Font.PLAIN, 16));
Frame.getContentPane().add(lblNewLabel_1_1_1);

JLabel lblNewLabel_1_1_2 = new JLabel("Wickets");
lblNewLabel_1_1_2.setBounds(135, 46, 70, 28);
lblNewLabel_1_1_2.setFont(new Font("Times New Roman", Font.PLAIN, 16));
Frame.getContentPane().add(lblNewLabel_1_1_2);

over = new JTextField();
over.setFont(new Font("Times New Roman", Font.BOLD, 23));
over.setEditable(false);
over.setHorizontalAlignment(SwingConstants.CENTER);
over.setBounds(236, 84, 49, 38);
over.setColumns(10);
Frame.getContentPane().add(over);

ball = new JTextField();
ball.setFont(new Font("Times New Roman", Font.BOLD, 23));
ball.setEditable(false);
ball.setHorizontalAlignment(SwingConstants.CENTER);
ball.setBounds(307, 84, 29, 38);
ball.setColumns(10);
Frame.getContentPane().add(ball);

batsman1 = new JTextField();
batsman1.setHorizontalAlignment(SwingConstants.LEFT);
batsman1.setFont(new Font("Times New Roman", Font.BOLD, 16));
batsman1.setBounds(36, 184, 186, 28);
Frame.getContentPane().add(batsman1);
batsman1.setColumns(10);

runs1 = new JTextField();
runs1.setBorder(null);
runs1.setEditable(false);
runs1.setText("0");
runs1.setFont(new Font("Times New Roman", Font.BOLD, 15));
runs1.setBounds(242, 185, 49, 28);
runs1.setColumns(10);
Frame.getContentPane().add(runs1);

balls1 = new JTextField();
balls1.setBorder(null);
balls1.setEditable(false);
balls1.setText("0");
balls1.setFont(new Font("Times New Roman", Font.BOLD, 15));
balls1.setBounds(301, 185, 49, 28);
balls1.setColumns(10);
Frame.getContentPane().add(balls1);

four1 = new JTextField();
four1.setBorder(null);
four1.setEditable(false);
```

```
four1.setText("0");
four1.setFont(new Font("Times New Roman", Font.BOLD, 15));
four1.setBounds(364, 184, 29, 28);
four1.setColumns(10);
Frame.getContentPane().add(four1);
six1 = new JTextField();
six1.setBorder(null);
six1.setEditable(false);
six1.setText("0");
six1.setFont(new Font("Times New Roman", Font.BOLD, 15));
six1.setBounds(407, 184, 33, 28);
six1.setColumns(10);
Frame.getContentPane().add(six1);

strikerate1 = new JTextField();
strikerate1.setBorder(null);
strikerate1.setEditable(false);
strikerate1.setText("0");
strikerate1.setFont(new Font("Times New Roman", Font.BOLD, 15));
strikerate1.setBounds(461, 185, 70, 28);
strikerate1.setColumns(10);
Frame.getContentPane().add(strikerate1);

batsman2 = new JTextField();
batsman2.setFont(new Font("Times New Roman", Font.BOLD, 16));
batsman2.setBounds(36, 222, 186, 28);
batsman2.setColumns(10);
Frame.getContentPane().add(batsman2);

runs2 = new JTextField();
runs2.setBorder(null);
runs2.setEditable(false);
runs2.setText("0");
runs2.setFont(new Font("Times New Roman", Font.BOLD, 15));
```

```
runs2.setBounds(242, 223, 49, 28);
runs2.setColumns(10);
Frame.getContentPane().add(runs2);

balls2 = new JTextField();
balls2.setBorder(null);
balls2.setEditable(false);
balls2.setText("0");
balls2.setFont(new Font("Times New Roman", Font.BOLD, 15));
balls2.setBounds(301, 223, 49, 28);
balls2.setColumns(10);
Frame.getContentPane().add(balls2);

four2 = new JTextField();
four2.setBorder(null);
four2.setEditable(false);
four2.setText("0");
four2.setFont(new Font("Times New Roman", Font.BOLD, 15));
four2.setBounds(364, 222, 29, 28);
four2.setColumns(10);
Frame.getContentPane().add(four2);

six2 = new JTextField();
six2.setBorder(null);
six2.setEditable(false);
six2.setText("0");
six2.setFont(new Font("Times New Roman", Font.BOLD, 15));
six2.setBounds(407, 222, 33, 28);
six2.setColumns(10);
Frame.getContentPane().add(six2);

strikerate2 = new JTextField();
strikerate2.setBorder(null);
strikerate2.setEditable(false);
```

```
strikerate2.setText("0");
strikerate2.setFont(new Font("Times New Roman", Font.BOLD, 15));
strikerate2.setBounds(461, 223, 70, 28);
strikerate2.setColumns(10);
Frame.getContentPane().add(strikerate2);
strike = new JTextField();
strike.setHorizontalAlignment(SwingConstants.RIGHT);
strike.setFont(new Font("Times New Roman", Font.BOLD, 16));
strike.setBounds(6, 184, 20, 28);
strike.setColumns(10);
Frame.getContentPane().add(strike);

b1 = new JTextField();
b1.setHorizontalAlignment(SwingConstants.CENTER);
b1.setFont(new Font("Times New Roman", Font.BOLD, 15));
b1.setEditable(false);
b1.setBounds(111, 358, 33, 38);
b1.setColumns(10);
Frame.getContentPane().add(b1);

b2 = new JTextField();
b2.setHorizontalAlignment(SwingConstants.CENTER);
b2.setFont(new Font("Times New Roman", Font.BOLD, 15));
b2.setEditable(false);
b2.setBounds(154, 358, 33, 38);
b2.setColumns(10);
Frame.getContentPane().add(b2);

b3 = new JTextField();
b3.setHorizontalAlignment(SwingConstants.CENTER);
b3.setFont(new Font("Times New Roman", Font.BOLD, 15));
b3.setEditable(false);
b3.setBounds(197, 358, 33, 38);
b3.setColumns(10);
```

```
Frame.getContentPane().add(b3);
b4 = new JTextField();
b4.setHorizontalAlignment(SwingConstants.CENTER);
b4.setFont(new Font("Times New Roman", Font.BOLD, 15));
b4.setEditable(false);
b4.setBounds(242, 358, 33, 38);
b4.setColumns(10);
Frame.getContentPane().add(b4);

b5 = new JTextField();
b5.setHorizontalAlignment(SwingConstants.CENTER);
b5.setFont(new Font("Times New Roman", Font.BOLD, 15));
b5.setEditable(false);
b5.setBounds(285, 358, 33, 38);
b5.setColumns(10);
Frame.getContentPane().add(b5);

b6 = new JTextField();
b6.setHorizontalAlignment(SwingConstants.CENTER);
b6.setFont(new Font("Times New Roman", Font.BOLD, 15));
b6.setEditable(false);
b6.setBounds(328, 358, 33, 38);
b6.setColumns(10);
Frame.getContentPane().add(b6);

b7 = new JTextField();
b7.setHorizontalAlignment(SwingConstants.CENTER);
b7.setFont(new Font("Times New Roman", Font.BOLD, 15));
b7.setEditable(false);
b7.setBounds(371, 358, 33, 38);
b7.setColumns(10);
Frame.getContentPane().add(b7);

b8 = new JTextField();
```

```
b8.setHorizontalAlignment(SwingConstants.CENTER);
b8.setFont(new Font("Times New Roman", Font.BOLD, 15));
b8.setEditable(false);
b8.setBounds(414, 358, 33, 38);
b8.setColumns(10);
Frame.getContentPane().add(b8);
b9 = new JTextField();
b9.setHorizontalAlignment(SwingConstants.CENTER);
b9.setFont(new Font("Times New Roman", Font.BOLD, 15));
b9.setEditable(false);
b9.setBounds(461, 358, 33, 38);
b9.setColumns(10);
Frame.getContentPane().add(b9);

b10 = new JTextField();
b10.setHorizontalAlignment(SwingConstants.CENTER);
b10.setFont(new Font("Times New Roman", Font.BOLD, 15));
b10.setEditable(false);
b10.setBounds(504, 358, 33, 38);
b10.setColumns(10);
Frame.getContentPane().add(b10);

lblNewLabel_1_1_3 = new JLabel("This Over");
lblNewLabel_1_1_3.setBounds(36, 358, 65, 38);
lblNewLabel_1_1_3.setFont(new Font("Times New Roman", Font.PLAIN, 13));
Frame.getContentPane().add(lblNewLabel_1_1_3);

wide = new JCheckBox("Wide");
wide.setBounds(36, 406, 65, 21);
wide.setFont(new Font("Times New Roman", Font.PLAIN, 13));
Frame.getContentPane().add(wide);

noball = new JCheckBox("No-Ball");
noball.setFont(new Font("Times New Roman", Font.PLAIN, 13));
```

```
    noball.setBounds(111, 406, 65, 21);
    Frame.getContentPane().add(noball);

    byes = new JCheckBox("Byes");
    byes.setFont(new Font("Times New Roman", Font.PLAIN, 13));
    byes.setBounds(198, 406, 65, 21);
    Frame.getContentPane().add(byes);

    legbyes = new JCheckBox("LegByes");
    legbyes.setFont(new Font("Times New Roman", Font.PLAIN, 13));
    legbyes.setBounds(273, 406, 77, 21);
    Frame.getContentPane().add(legbyes);

    wicket = new JCheckBox("Wicket");
    wicket.setFont(new Font("Times New Roman", Font.PLAIN, 13));
    wicket.setBounds(375, 405, 65, 25);
    Frame.getContentPane().add(wicket);

    JButton zero = new JButton("0");
    zero.addActionListener(new ActionListener() {

        public void actionPerformed(ActionEvent e)
        {
scoreConvert(0,wicket.isSelected(),wide.isSelected(),noball.isSelected());
            SetButton("0",wide.isSelected(),noball.isSelected());
        }
    });
    zero.setFont(new Font("Times New Roman", Font.BOLD, 28));
    zero.setBounds(36, 433, 65, 66);
    Frame.getContentPane().add(zero);

    one = new JButton("1");
    one.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e)
        {
        scoreConvert(1,wicket.isSelected(),wide.isSelected(),noball.isSelected());
            SetButton("1",wide.isSelected(),noball.isSelected());
        }
    });
    one.setFont(new Font("Times New Roman", Font.BOLD, 28));
    one.setBounds(111, 433, 66, 66);
    Frame.getContentPane().add(one);
    two = new JButton("2");
    two.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
scoreConvert(2,wicket.isSelected(),wide.isSelected(),noball.isSelected());
            SetButton("2",wide.isSelected(),noball.isSelected());
        }
    });
    two.setFont(new Font("Times New Roman", Font.BOLD, 28));
    two.setBounds(187, 433, 65, 66);
    Frame.getContentPane().add(two);

    three = new JButton("3");
    three.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
scoreConvert(3,wicket.isSelected(),wide.isSelected(),noball.isSelected());
            SetButton("3",wide.isSelected(),noball.isSelected());
        }
    });
    three.setFont(new Font("Times New Roman", Font.BOLD, 28));
    three.setBounds(262, 433, 65, 66);
    Frame.getContentPane().add(three);

    four = new JButton("4");
    four.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
scoreConvert(4,wicket.isSelected(),wide.isSelected(),noball.isSelected());
```

```
            SetButton("4",wide.isSelected(),noball.isSelected());
        }
    });
    four.setFont(new Font("Times New Roman", Font.BOLD, 28));
    four.setBounds(337, 433, 60, 66);
    Frame.getContentPane().add(four);

    five = new JButton("5");
    five.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            SetButton("5",wide.isSelected(),noball.isSelected());
    scoreConvert(5,wicket.isSelected(),wide.isSelected(),noball.isSelected());
        }
    });
    five.setFont(new Font("Times New Roman", Font.BOLD, 28));
    five.setBounds(407, 433, 59, 66);
    Frame.getContentPane().add(five);

    six = new JButton("6");
    six.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
scoreConvert(6,wicket.isSelected(),wide.isSelected(),noball.isSelected());
            SetButton("6",wide.isSelected(),noball.isSelected());
        }
    });
    six.setFont(new Font("Times New Roman", Font.BOLD, 28));
    six.setBounds(482, 433, 60, 66);
    Frame.getContentPane().add(six);

    JLabel lblNewLabel_1_6 = new JLabel("Bowler");
    lblNewLabel_1_6.setFont(new Font("Times New Roman", Font.PLAIN, 13));
    lblNewLabel_1_6.setBounds(36, 272, 94, 28);
    Frame.getContentPane().add(lblNewLabel_1_6);
```

```
JLabel lblNewLabel_1_1_4 = new JLabel("Overs");
lblNewLabel_1_1_4.setFont(new Font("Times New Roman", Font.PLAIN, 13));
lblNewLabel_1_1_4.setBounds(242, 272, 49, 28);
Frame.getContentPane().add(lblNewLabel_1_1_4);

JLabel lblNewLabel_1_2_1 = new JLabel("Maidens");
lblNewLabel_1_2_1.setFont(new Font("Times New Roman", Font.PLAIN, 13));
lblNewLabel_1_2_1.setBounds(301, 272, 49, 28);
Frame.getContentPane().add(lblNewLabel_1_2_1);

JLabel lblNewLabel_1_3_2 = new JLabel("Runs");
lblNewLabel_1_3_2.setFont(new Font("Times New Roman", Font.PLAIN, 13));
lblNewLabel_1_3_2.setBounds(364, 272, 33, 28);
Frame.getContentPane().add(lblNewLabel_1_3_2);

JLabel wkt = new JLabel("Wickets");
wkt.setFont(new Font("Times New Roman", Font.PLAIN, 13));
wkt.setBounds(417, 272, 49, 28);
Frame.getContentPane().add(wkt);

JLabel lblNewLabel_1_5_1 = new JLabel("Economy");
lblNewLabel_1_5_1.setFont(new Font("Times New Roman", Font.PLAIN, 13));
lblNewLabel_1_5_1.setBounds(485, 272, 70, 28);
Frame.getContentPane().add(lblNewLabel_1_5_1);

bowler = new JTextField();
bowler.setFont(new Font("Times New Roman", Font.BOLD, 16));
bowler.setColumns(10);
bowler.setBounds(36, 309, 186, 28);
Frame.getContentPane().add(bowler);

overs = new JTextField();
overs.setBorder(null);
overs.setEditable(false);
overs.setText("0");
overs.setFont(new Font("Times New Roman", Font.BOLD, 15));
overs.setColumns(10);
overs.setBounds(262, 310, 20, 28);
Frame.getContentPane().add(overs);

maiden = new JTextField();
maiden.setBorder(null);
maiden.setEditable(false);
maiden.setText("0");
maiden.setFont(new Font("Times New Roman", Font.BOLD, 15));
maiden.setColumns(10);
maiden.setBounds(301, 310, 49, 28);
Frame.getContentPane().add(maiden);

runsb = new JTextField();
runsb.setBorder(null);
runsb.setEditable(false);
runsb.setText("0");
runsb.setFont(new Font("Times New Roman", Font.BOLD, 15));
runsb.setColumns(10);
runsb.setBounds(364, 309, 40, 28);
Frame.getContentPane().add(runsb);

wicketsb = new JTextField();
wicketsb.setBorder(null);
wicketsb.setEditable(false);
wicketsb.setText("0");
wicketsb.setFont(new Font("Times New Roman", Font.BOLD, 15));
wicketsb.setColumns(10);
wicketsb.setBounds(417, 309, 49, 28);
Frame.getContentPane().add(wicketsb);
economy = new JTextField();
economy.setBorder(null);
economy.setEditable(false);
economy.setText("0");
economy.setFont(new Font("Times New Roman", Font.BOLD, 15));
economy.setColumns(10);
economy.setBounds(482, 310, 49, 28);
Frame.getContentPane().add(economy);

batsman1.setText(k1);
batsman2.setText(k2);
bowler.setText(k3);
batsman1.setEditable(false);
batsman1.setBorder(null);
batsman2.setEditable(false);
batsman2.setBorder(null);
bowler.setEditable(false);
bowler.setBorder(null);
runs.setText("0");
wickets.setText("0");
over.setText("0");
ball.setText("0");
strike.setEditable(false);
strike.setBorder(null);
runs.setBorder(null);
over.setBorder(null);
ball.setBorder(null);

wickets.setBorder(null);
lblNewLabel_1_1_5 = new JLabel("Overs");
lblNewLabel_1_1_5.setFont(new Font("Times New Roman", Font.PLAIN, 16));
lblNewLabel_1_1_5.setBounds(242, 46, 55, 28);
Frame.getContentPane().add(lblNewLabel_1_1_5);

textField = new JTextField();
textField.setText(".");
```

```
		textField.setHorizontalAlignment(SwingConstants.CENTER);
		textField.setFont(new Font("Times New Roman", Font.BOLD, 25));
		textField.setEditable(false);
		textField.setColumns(10);
		textField.setBounds(282, 83, 20, 38);
		textField.setBorder(null);
		Frame.getContentPane().add(textField);

		textField_1 = new JTextField();
		textField_1.setText("0");
		textField_1.setFont(new Font("Times New Roman", Font.BOLD, 15));
		textField_1.setEditable(false);
		textField_1.setColumns(10);
		textField_1.setBorder(null);
		textField_1.setBounds(232, 310, 17, 28);
		Frame.getContentPane().add(textField_1);
		textField_2 = new JTextField();
		textField_2.setText(".");
		textField_2.setHorizontalAlignment(SwingConstants.CENTER);
		textField_2.setFont(new Font("Times New Roman", Font.BOLD, 18));
		textField_2.setEditable(false);
		textField_2.setColumns(10);
		textField_2.setBorder(null);
		textField_2.setBounds(242, 310, 20, 27);
		Frame.getContentPane().add(textField_2);
	}
	public int m=0;
	public void SetButton(String bno,boolean wide,boolean noball)
	{
		if(ballCount==6)
		{
			database1 u =new database1();
			try {
```

```
                u.setData(bowler.getText(),overs.getText(), runsb.getText(),
maiden.getText(), wicketsb.getText());
            } catch (Exception e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            b1.setText("");
            b2.setText("");
            b3.setText("");
            b4.setText("");
            b5.setText("");
            b6.setText("");
            b7.setText("");


            b8.setText("");
            b9.setText("");
            b10.setText("");
            ballCount=0;
        }
        if(b1.getText()==null || b1.getText().trim().equals(""))
        {
            b1.setText(bno);
            if(wide==true || noball==true)
            {
                ballCount--;
                if(wide==true)
                    b1.setText("WD");
                else
                    b1.setText("NB");
            }
            if(wicket.isSelected()==true)
            {
                b1.setText("OUT");
```

```
    }
}
else if(b2.getText()==null || b2.getText().trim().equals(""))
{
    b2.setText(bno);
    if(wide==true || noball==true)
    {
        ballCount--;
        if(wide==true)
            b2.setText("WD");
        else
            b2.setText("NB");
    }
    if(wicket.isSelected()==true)
    {
        b2.setText("OUT");
    }

}
else if(b3.getText()==null || b3.getText().trim().equals(""))
{
    b3.setText(bno);
    if(wide==true || noball==true)
    {
        ballCount--;
        if(wide==true)
            b3.setText("WD");
        else
            b3.setText("NB");
    }
    if(wicket.isSelected()==true)
    {
        b3.setText("OUT");
    }
```

```
}
else if(b4.getText()==null || b4.getText().trim().equals(""))
{
        b4.setText(bno);
        if(wide==true || noball==true)
        {
                ballCount--;
                if(wide==true)
                        b4.setText("WD");
                else
                        b4.setText("NB");
        }
        if(wicket.isSelected()==true)
        {
                b4.setText("OUT");
        }
}
else if(b5.getText()==null || b5.getText().trim().equals(""))
{
        b5.setText(bno);
        if(wide==true || noball==true)
        {
                ballCount--;
                if(wide==true)
                        b5.setText("WD");
                else
                        b5.setText("NB");
        }
        if(wicket.isSelected()==true)
        {
                b5.setText("OUT");
        }
}
else if(b6.getText()==null || b6.getText().trim().equals(""))
```

```
{
        b6.setText(bno);
        if(wide==true || noball==true)
        {
                ballCount--;
                if(wide==true)
                        b6.setText("WD");
                else
                        b6.setText("NB");
        }
        if(wicket.isSelected()==true)
        {
                b6.setText("OUT");
        }
}
else if(b7.getText()==null || b7.getText().trim().equals(""))
{
        b7.setText(bno);
        if(wide==true || noball==true)
        {
                ballCount--;
                if(wide==true)
                        b7.setText("WD");
                else
                        b7.setText("NB");
        }
        if(wicket.isSelected()==true)
        {
                b7.setText("OUT");
        }
}
else if(b8.getText()==null || b8.getText().trim().equals(""))
{
        b8.setText(bno);
        if(wide==true || noball==true)
        {
                ballCount--;
                if(wide==true)
                        b8.setText("WD");
                else
                        b8.setText("NB");
        }
        if(wicket.isSelected()==true)
        {
                b8.setText("OUT");
        }
}
else if(b9.getText()==null || b9.getText().trim().equals(""))
{
        b9.setText(bno);
        if(wide==true || noball==true)
        {
                ballCount--;
                if(wide==true)
                        b9.setText("WD");
                else
                        b9.setText("NB");
        }
        if(wicket.isSelected()==true)
        {
                b9.setText("OUT");
        }
}
else if(b10.getText()==null || b10.getText().trim().equals(""))
{
        b10.setText(bno);
        if(wide==true || noball==true)
        {
            ballCount--;
            if(wide==true)
                b10.setText("WD");
            else
                b10.setText("NB");
        }
        if(wicket.isSelected()==true)
        {
            b10.setText("OUT");
        }
    }
    ballCount++;
}
public boolean flag1=true;
private JTextField textField_1;
private JTextField textField_2;
public void bowlerchange(String x)
{
    bowler.setText(x);
    initialize();
}
int change(String temp){
    int t;
    t=Integer.parseInt(temp);
    return t;
}
public void scoreConvert(int i,boolean wt,boolean wi,boolean no)
{
    int ss=change(overs.getText());
    ss=ss+1;
    overs.setText(String.valueOf(ss));

    int y=change(runsb.getText());
    y=y+i;
```

```
        runsb.setText(String.valueOf(y));
        if(y!=0)
        {
            float ii=((float)(ss/y))*6;
            economy.setText(String.valueOf(ii));
        }
        strike.setText("*");
        pop obj=new pop();
        boolean ext=false;
        if(wi==true || no==true)
            ext=true;

        obj.SqlGet(i, wt,ext,batsman1.getText(),bowler.getText());
        if(wt==true)
        {
            y=change(wicketsb.getText());
            y=y+1;
            wicketsb.setText(String.valueOf(y));
            database bb=new database();
            try
            {
                bb.setData(batsman1.getText(), runs1.getText(),
balls1.getText(),four1.getText() , six1.getText());


            }
            catch (Exception e)
            {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            int z=Integer.parseInt(wickets.getText());
            z++;
```

```
    String s=String.valueOf(z);
    wickets.setText(s);
    runs1.setText("0");
    four1.setText("0");
    six1.setText("0");
    strikerate1.setText("0");
    balls1.setText("-1");
    newbat obj1=new newbat();
    obj1.main(null);
            }
int z=Integer.parseInt(runs.getText());
int k=z+i;


if(wi==true || no==true)
    k++;
String s=String.valueOf(k);
runs.setText(s);


if(i==4)
{
    int l=Integer.parseInt(four1.getText());
    l=l+1;
    four1.setText(String.valueOf(l));



}
if(i==6){
    int a=Integer.parseInt(six1.getText());
    a=a+1;
    six1.setText(String.valueOf(a));
}
int a;
```

```
if(wi==false && no==false)
{
        a=Integer.parseInt(runs1.getText());
        a=a+i;
        runs1.setText(String.valueOf(a));
        int b=Integer.parseInt(balls1.getText());
        b=b+1;
        balls1.setText(String.valueOf(b));
        z=Integer.parseInt(ball.getText());
        z++;
        if(z==6)
        {
                z=0;
                int d=Integer.parseInt(over.getText());
                d++;
                String j=String.valueOf(d);
                over.setText(j);
        }
        ball.setText(String.valueOf(z));
}
if(i%2!=0)
{
        String tempp;
        String runs;
        String balls;
        String fours;
        String sixes;
        tempp=batsman1.getText();
        runs=runs1.getText();
        balls=balls1.getText();
        fours=four1.getText();
        sixes=six1.getText();
        batsman1.setText(batsman2.getText());
        runs1.setText(runs2.getText());
```

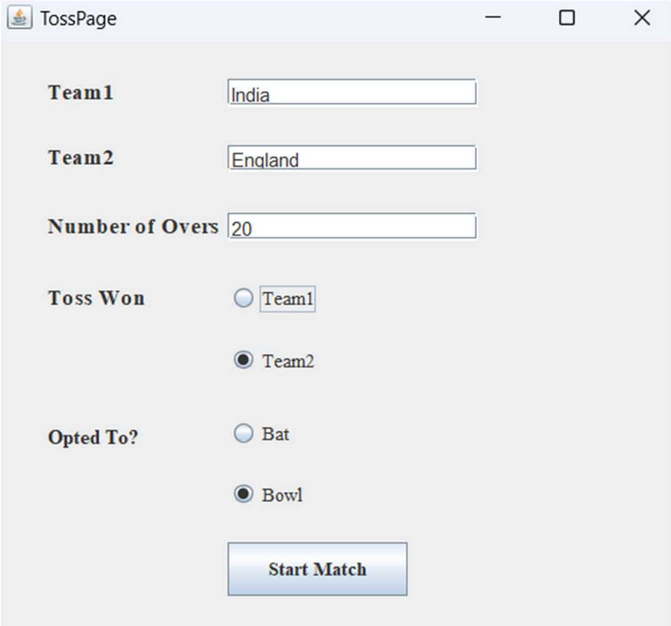
```
            balls1.setText(balls2.getText());
            four1.setText(four2.getText());
            six1.setText(six2.getText());

            batsman2.setText(tempp);
            runs2.setText(runs);
            balls2.setText(balls);
            four2.setText(fours);
            six2.setText(sixes);
        }
        int o=Integer.parseInt(runs1.getText());
        int p=Integer.parseInt(balls1.getText());
        if(p!=0)
            strikerate1.setText(String.valueOf((((float)o)/p)*100));
        o=Integer.parseInt(runs2.getText());
        p=Integer.parseInt(balls2.getText());
        if(p!=0)
            strikerate2.setText(String.valueOf((((float)o)/p)*100));
    }
}
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
public class pop
{
    void SqlGet(int score,boolean wkt,boolean extra,String bat1,String bowler){
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/cricket","root","Abhilash@213");
            int wicket=0;
            int ext=0;
            if(wkt==true)
```

```
                wicket=1;
            if(extra==true)
                ext=1;
            String query = "insert into
match_(matchid,score,isout,extra,batterid,bowlerid,innings)
values(1,"+score+","+wicket+","+ext+",'"+bat1+"','"+bowler+"',1)";
            Statement stk=con.createStatement();
            stk.execute(query);
            System.out.println("SuccessFully Entered into Database");
        }
        catch (Exception e1)
        {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
}
```

RESULT SCREENS



Team1 India

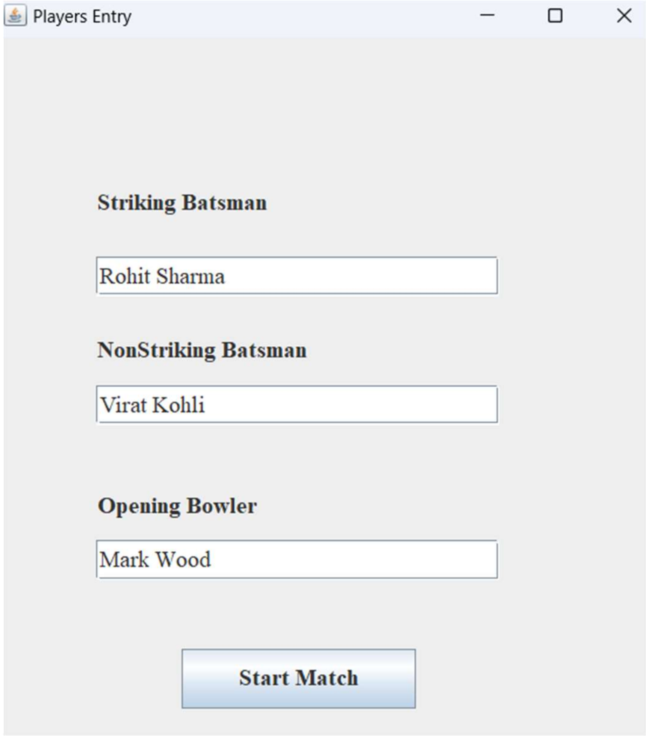
Team2 England

Number of Overs 20

Toss Won ○ Team1 ● Team2

Opted To? ○ Bat ● Bowl

Tosspage Screen



Striking Batsman

Rohit Sharma

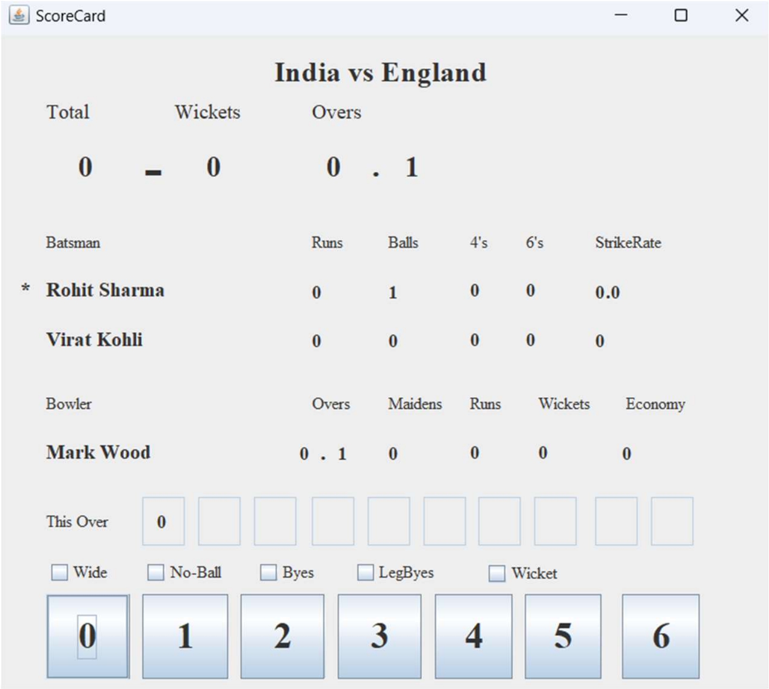
NonStriking Batsman

Virat Kohli

Opening Bowler

Mark Wood

Player Entry Page (Redirects to this page after submitting the above page)



India vs England

Total		Wickets	Overs
0	-	0	0 . 1

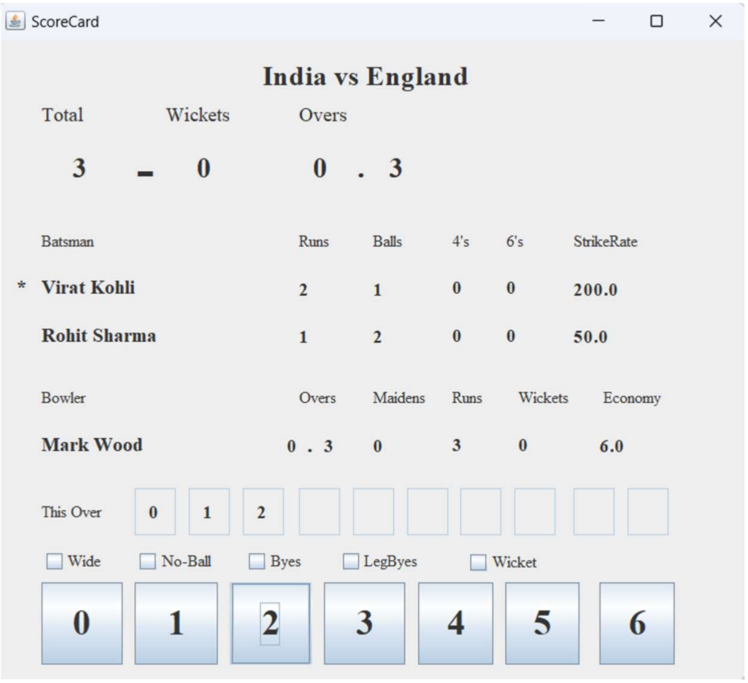
	Batsman	Runs	Balls	4's	6's	StrikeRate
*	Rohit Sharma	0	1	0	0	0.0
	Virat Kohli	0	0	0	0	0

Bowler	Overs	Maidens	Runs	Wickets	Economy
Mark Wood	0 . 1	0	0	0	0

This Over	0									

□ Wide □ No-Ball □ Byes □ LegByes □ Wicket

ScoreCard (After Clicking on Start Match Button ,it redirects to this interface)



India vs England

Total		Wickets	Overs
3	-	0	0 . 3

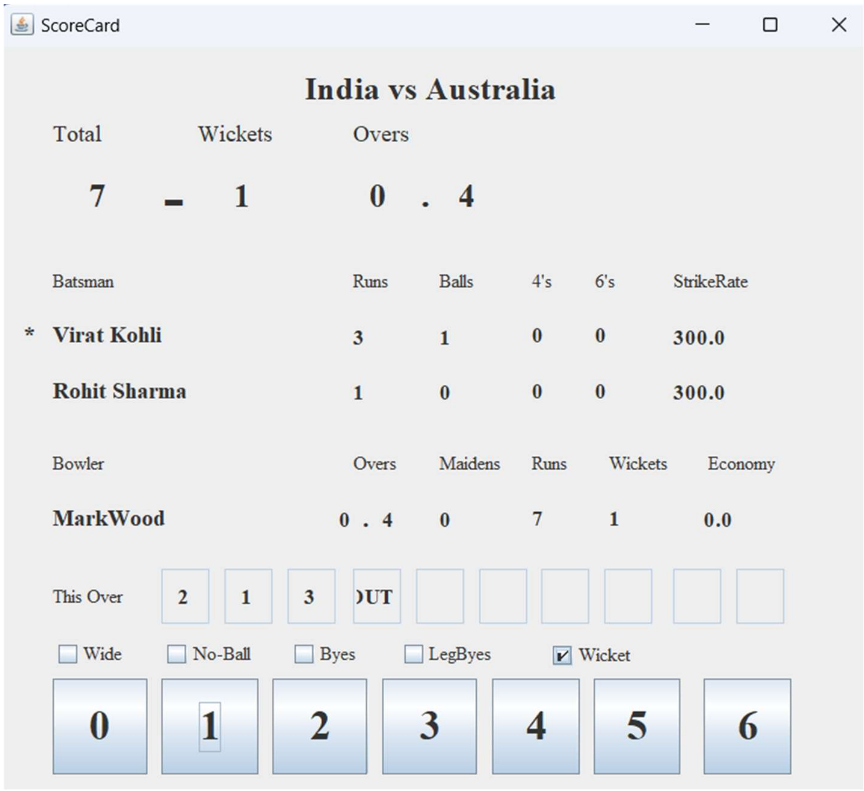
	Batsman	Runs	Balls	4's	6's	StrikeRate
*	Virat Kohli	2	1	0	0	200.0
	Rohit Sharma	1	2	0	0	50.0

Bowler	Overs	Maidens	Runs	Wickets	Economy
Mark Wood	0 . 3	0	3	0	6.0

This Over	0	1	2							

□ Wide □ No-Ball □ Byes □ LegByes □ Wicket

Dynamic Entry of the Interface(Will be updated as per the Button clicks)



ScoreCard

India vs Australia

Total		Wickets	Overs
7	-	1	0 . 4

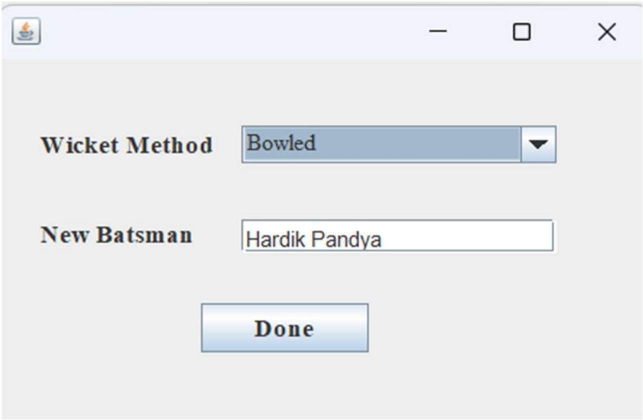
	Batsman	Runs	Balls	4's	6's	StrikeRate
*	Virat Kohli	3	1	0	0	300.0
	Rohit Sharma	1	0	0	0	300.0

Bowler	Overs	Maidens	Runs	Wickets	Economy
MarkWood	0 . 4	0	7	1	0.0

This Over: 2 | 1 | 3 |)UT

☐ Wide ☐ No-Ball ☐ Byes ☐ LegByes ☑ Wicket

Batsman Out Scenario (will get a new Interaface to provide the New Batsman name)

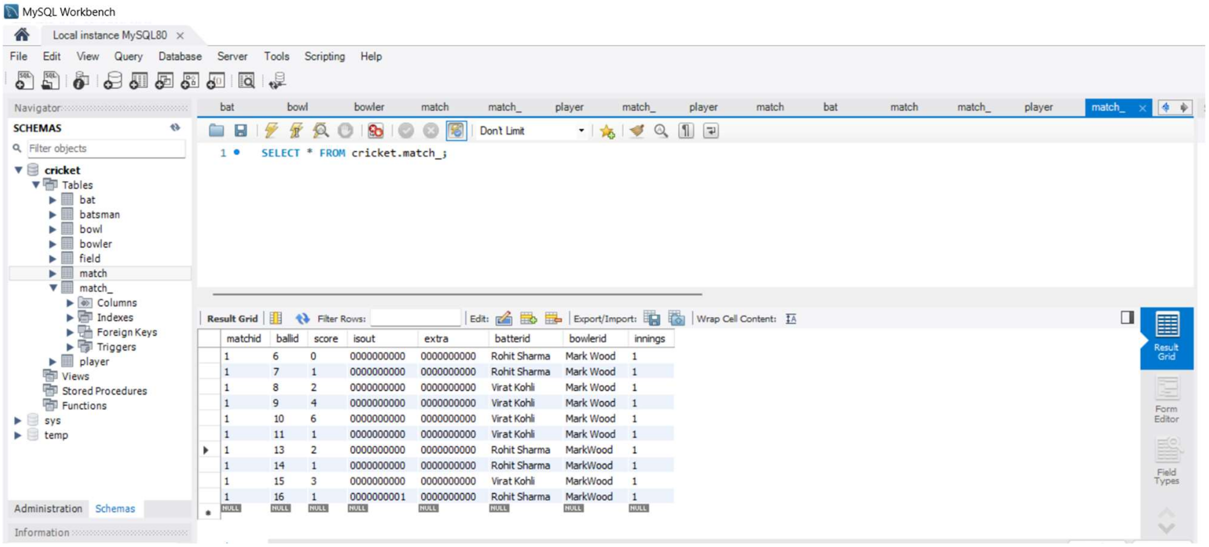


Wicket Method: Bowled

New Batsman: Hardik Pandya

Need to Enter the Name of the new Batsman

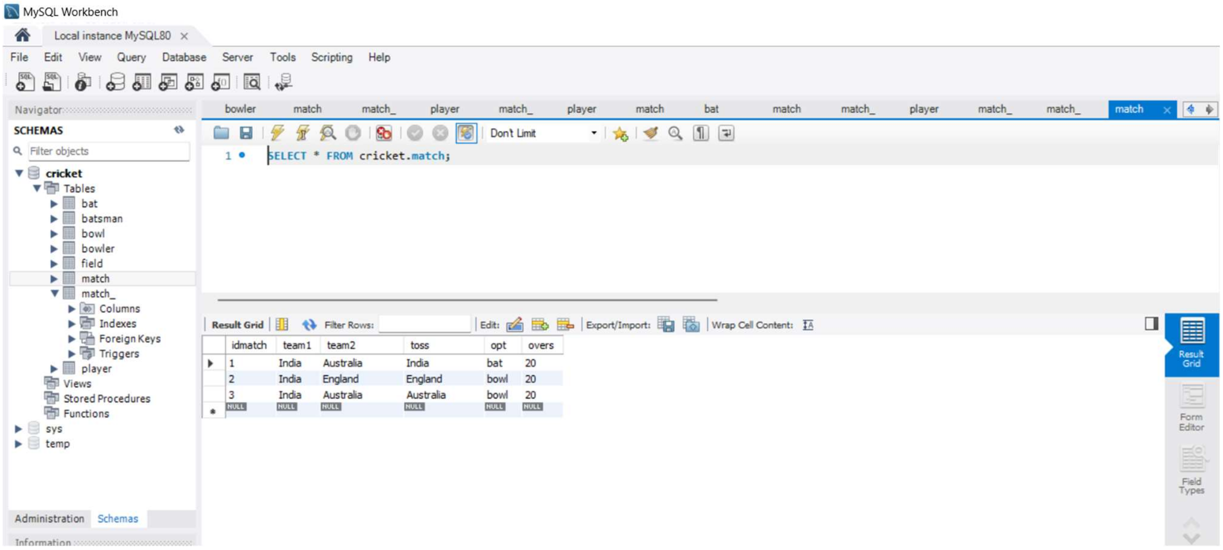
Data Base Screens:



```
SELECT * FROM cricket.match_;
```

	matchid	ballid	score	isout	extra	batterid	bowlerid	innings
	1	6	0	0000000000	0000000000	Rohit Sharma	Mark Wood	1
	1	7	1	0000000000	0000000000	Rohit Sharma	Mark Wood	1
	1	8	2	0000000000	0000000000	Virat Kohli	Mark Wood	1
	1	9	4	0000000000	0000000000	Virat Kohli	Mark Wood	1
	1	10	6	0000000000	0000000000	Virat Kohli	Mark Wood	1
	1	11	1	0000000000	0000000000	Virat Kohli	Mark Wood	1
▶	1	13	2	0000000000	0000000000	Rohit Sharma	MarkWood	1
	1	14	1	0000000000	0000000000	Rohit Sharma	MarkWood	1
	1	15	3	0000000000	0000000000	Virat Kohli	MarkWood	1
	1	16	1	0000000001	0000000000	Rohit Sharma	MarkWood	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

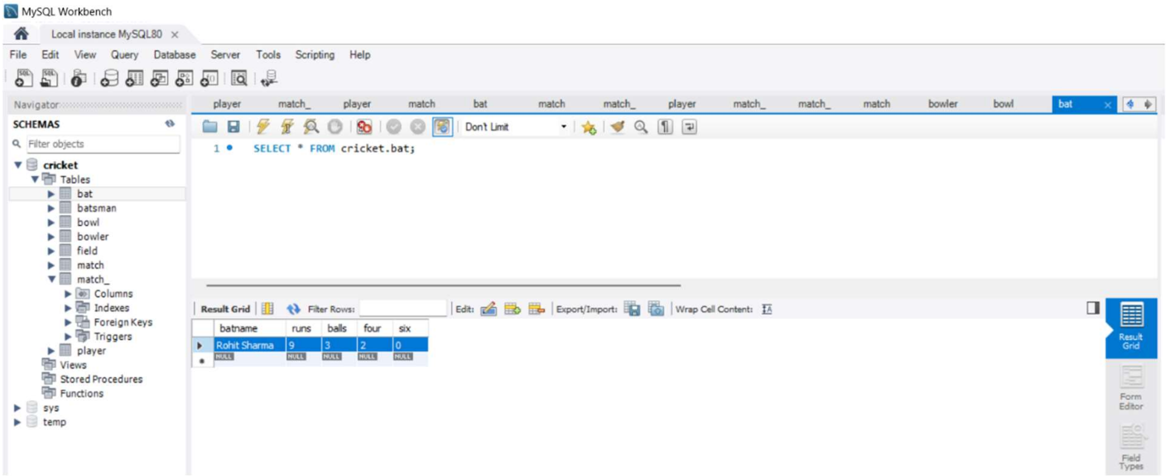
Ball By Ball Record of the match



```
SELECT * FROM cricket.match;
```

	idmatch	team1	team2	toss	opt	overs
▶	1	India	Australia	India	bat	20
	2	India	England	England	bowl	20
	3	India	Australia	Australia	bowl	20
*	NULL	NULL	NULL	NULL	NULL	NULL

MatchDetails Records will be saved as shown in the above screen



```
SELECT * FROM cricket.bat;
```

	batname	runs	balls	four	six
▶	Rohit Sharma	9	3	2	0
*	NULL	NULL	NULL	NULL	NULL

Player Records or stats will be saved as per the above Screen

CONCLUSION

It is concluded that the application works well and satisfy the end users. The application is tested very well and errors are properly debugged. We have checked all the possible test cases to avoid the errors or any exceptions. This system is user friendly so everyone can use easily. Proper documentation is provided. The end user can easily understand how the whole system is implemented by going through the documentation. The system is tested, implemented and the performance is found to be satisfactory. All necessary output is generated. Thus, the project is completed successfully. Further enhancements can be made to the application, so that the application functions very attractive and useful manner than the present one.

FUTURE SCOPE

There is scope for future development of this project. The world of computer fields is not static; it is always subject to be dynamic. The technology which is famous today becomes outdated the very next day. To keep abstract of technical improvements, the system may be further refined. So, it is not concluded. Yet it will improve with further enhancements. Enhancements can be done in an efficient manner. We can even update the same with further modification establishment and can be integrated with minimal modification. Thus the project is flexible and can be enhanced at any time with more advanced features.

Future Enhancements:

- Player wise Details
- Previous Match Details
- Head-to-Head Encounter(teams)
- Stats of (Player vs Player)
- Ball by Ball Commentary
- Necessary Graphics such as Graphs for run rate, Economy etc...
- Providing an end User interface