

Java Code:

Classes:

- 1.afghanistanPlayerProfile
- 2.australiaPlayerProfile
- 3.bangladeshPlayerProfile
- 4.indiaPlayerProfile
- 5.newZealandPlayerProfile
- 6.pakistanPlayerProfile
- 7.sriLankaPlayerProfile
- 8.englandPlayerProfile
- 9.best5BowlersAndBatsman
- 10.Controller
- 11.dictionariesForEachTeamProfiles
- 12.editDeleteAndDisplayPlayerProfiles
- 13.editDeleteAndDisplayTeamProfile
- 14.listOfAllThePlayers
- 15.listOfAllTheTeams
- 16.Main
- 17.overviewOfMatchSummery
- 18.playerProfileDictionaryStructure
- 19.positionOfTeams
- 20.teamWonLooseDecisionInOneMatch
- 21.tournamentMatches
- 22.WeightedRandom

FXML files:

- 1.guiAskingForEditAndDeleteTeamProfile
- 2.guiAskingForEditDeletePlayerProfile
- 3.guiForDeletePlayerProfile
- 4.guiForDeleteTeamProfile
- 5.guiForEditPlayerProfile
- 6.guiForEditTeamProfile
- 7.guiToDisplayMatchSummery
- 8.guiToDisplayPlayerProfile
- 9.guiToDisplayTeamProfile

CSS File:

- 1.style.css

Code of Classes:

afghanistanPlayerProfile Class:

```
package sample;

import java.util.ArrayList;
import java.util.Hashtable;

public class afghanistanPlayerProfile extends
playerProfileDictionaryStructure {
    private Hashtable RashidKhan =new Hashtable();
    private Hashtable MohommadNabi =new
Hashtable();
    private Hashtable AsgharAfghan =new
Hashtable();
    private Hashtable HashmatullahShahidi =new
Hashtable();
    private Hashtable MujeebRahman =new
Hashtable();
    private Hashtable HamidHasan =new Hashtable();
    private Hashtable RahmatShah =new Hashtable();
    private Hashtable NoorAli =new Hashtable();
    private Hashtable GulbadinNaib =new
Hashtable();
    private Hashtable SayedShirzad =new
Hashtable();
    private Hashtable IkramAlikhil =new
Hashtable();
    private Hashtable SamiullahShinwari =new
Hashtable();

    //Creating an arrayList consisting of all the
Afghanistan players dictionaries
    private ArrayList AfganistanTeam = new
ArrayList();
    private ArrayList AfganistanBowling = new
ArrayList();
```

```
public ArrayList getAfganistanTeam() {  
    return AfganistanTeam;  
}  
  
public ArrayList getAfganistanBowling() {  
    return AfganistanBowling;  
}  
  
public afghanPlayerProfile() {  
    //When user create an object with this  
    class, automatically dictionaries get updates  
  
super.addCommonValuesToTheDictionary(RashidKhan, "Ra  
shid Khan", "Bowler", "Afganistan");  
  
super.addCommonValuesToTheDictionary(MohommadNabi, "  
Mohommad Nabi", "All Rounder", "Afganistan");  
  
super.addCommonValuesToTheDictionary(AsgharAfghan, "  
Asghar Afghan", "Batsman", "Afganistan");  
  
super.addCommonValuesToTheDictionary(HashmatullahSh  
ahidi, "Hashmatullah  
Shahidi", "Batsman", "Afganistan");  
  
super.addCommonValuesToTheDictionary(MujeebRahman, "  
Mujeeb Rahman", "Bowler", "Afganistan");  
  
super.addCommonValuesToTheDictionary(HamidHasan, "Ha  
mid Hasan", "Bowler", "Afganistan");  
  
super.addCommonValuesToTheDictionary(RahmatShah, "Ra  
hmat Shah", "All Rounder", "Afganistan");  
  
super.addCommonValuesToTheDictionary(NoorAli, "Noor  
Ali", "Batsman", "Afganistan");
```

```
super.addCommonValuesToTheDictionary(GulbadinNaib, "Gulbadin Naib", "Bowler", "Afghanistan");

super.addCommonValuesToTheDictionary(SayedShirzad, "Sayed Shirzad", "Bowler", "Afghanistan");

super.addCommonValuesToTheDictionary(IkramAlikhil, "Ikram Alikhil", "Batsman", "Afghanistan");

super.addCommonValuesToTheDictionary(SamiullahShinwari, "Samiullah Shinwari", "All Rounder", "Afghanistan");

AfganistanTeam.add(0, RashidKhan);
AfganistanTeam.add(1, MohommadNabi);
AfganistanTeam.add(2, AsgharAfghan);
AfganistanTeam.add(3, HashmatullahShahidi);
AfganistanTeam.add(4, MujeebRahman);
AfganistanTeam.add(5, HamidHasan);
AfganistanTeam.add(6, RahmatShah);
AfganistanTeam.add(7, NoorAli);
AfganistanTeam.add(8, GulbadinNaib);
AfganistanTeam.add(9, SayedShirzad);
AfganistanTeam.add(10, IkramAlikhil);
AfganistanTeam.add(11, SamiullahShinwari);

}

public Hashtable getRashidKhan() {
    return RashidKhan;
}

public Hashtable getMohommadNabi() {
    return MohommadNabi;
}

public Hashtable getAsgharAfghan() {
    return AsgharAfghan;
}
```

```
public Hashtable getHashmatullahShahidi() {  
    return HashmatullahShahidi;  
}  
  
public Hashtable getMujeebRahman() {  
    return MujeebRahman;  
}  
  
public Hashtable getHamidHasan() {  
    return HamidHasan;  
}  
  
public Hashtable getRahmatShah() {  
    return RahmatShah;  
}  
  
public Hashtable getNoorAli() {  
    return NoorAli;  
}  
  
public Hashtable getGulbadinNaib() {  
    return GulbadinNaib;  
}  
  
public Hashtable getSayedShirzad() {  
    return SayedShirzad;  
}  
  
public Hashtable getIkramAlikhil() {  
    return IkramAlikhil;  
}  
  
public Hashtable getSamiullahShinwari() {  
    return SamiullahShinwari;  
}  
}
```

australiaPlayerProfile class:

```
package sample;

import java.util.ArrayList;
import java.util.Hashtable;

public class australiaPlayerProfile extends
playerProfileDictionaryStructure {
    private Hashtable SteveSmith =new Hashtable();
    private Hashtable DavidWarner =new Hashtable();
    private Hashtable AaronFinch =new Hashtable();
    private Hashtable MitchelStarc =new
Hashtable();
    private Hashtable AdamZampa =new Hashtable();
    private Hashtable GlennMaxwell =new
Hashtable();
    private Hashtable MatthewWade =new Hashtable();
    private Hashtable PatCummins =new Hashtable();
    private Hashtable MarcusStoinis =new
Hashtable();
    private Hashtable AlexCarey =new Hashtable();
    private Hashtable NathanLyon =new Hashtable();
    private Hashtable KaneRichardson =new
Hashtable();

    //Creating an arrayList consisting of all the
Afganistan players dictionaries
    private ArrayList AustraliaTeam = new
ArrayList();
    private ArrayList AustraliaBowling = new
ArrayList();

    public ArrayList getAustraliaTeam() {
        return AustraliaTeam;
    }

    public ArrayList getAustraliaBowling() {
```

```
        return AustraliaBowling;
    }

    public australiaPlayerProfile() {
        //When user create an object with this
        class, automatically dictionaries get updates

        super.addCommonValuesToTheDictionary(SteveSmith,"St
        eve Smith","Batsman","Australia");

        super.addCommonValuesToTheDictionary(DavidWarner,"D
        avid Warner","Batsman","Australia");

        super.addCommonValuesToTheDictionary(AaronFinch,"Aa
        ron Finch","Batsman","Australia");

        super.addCommonValuesToTheDictionary(MitchelStarc,"
        Mitchel Starc","Bowler","Australia");

        super.addCommonValuesToTheDictionary(AdamZampa,"Ada
        m Zampa","Bowler","Australia");

        super.addCommonValuesToTheDictionary(GlennMaxwell,"
        Glenn Maxwell","All Rounder","Australia");

        super.addCommonValuesToTheDictionary(MatthewWade,"M
        atthew Wade","Batsman","Australia");

        super.addCommonValuesToTheDictionary(PatCummins,"Pa
        t Cummins","Bowler","Australia");

        super.addCommonValuesToTheDictionary(MarcusStoinis,
        "Marcus Stoinis","All Rounder","Australia");

        super.addCommonValuesToTheDictionary(AlexCarey,"Ale
        x Carey","Batsman","Australia");

        super.addCommonValuesToTheDictionary(NathanLyon,"Na
```

```
than Lyon", "Batsman", "Australia") ;  
  
super.addCommonValuesToTheDictionary(KaneRichardson  
, "Kane Richardson", "Bowler", "Australia") ;  
  
AustraliaTeam.add(0, SteveSmith) ;  
AustraliaTeam.add(1, DavidWarner) ;  
AustraliaTeam.add(2, AaronFinch) ;  
AustraliaTeam.add(3, MitchelStarc) ;  
AustraliaTeam.add(4, AdamZampa) ;  
AustraliaTeam.add(5, GlennMaxwell) ;  
AustraliaTeam.add(6, MatthewWade) ;  
AustraliaTeam.add(7, PatCummins) ;  
AustraliaTeam.add(8, MarcusStoinis) ;  
AustraliaTeam.add(9, AlexCarey) ;  
AustraliaTeam.add(10, NathanLyon) ;  
AustraliaTeam.add(11, KaneRichardson) ;  
}  
public Hashtable getSteveSmith() {  
    return SteveSmith;  
}  
  
public Hashtable getDavidWarner() {  
    return DavidWarner;  
}  
  
public Hashtable getAaronFinch() {  
    return AaronFinch;  
}  
  
public Hashtable getMitchelStarc() {  
    return MitchelStarc;  
}  
  
public Hashtable getAdamZampa() {  
    return AdamZampa;  
}
```

```
public Hashtable getGlennMaxwell() {
    return GlennMaxwell;
}

public Hashtable getMatthewWade() {
    return MatthewWade;
}

public Hashtable getPatCummins() {
    return PatCummins;
}

public Hashtable getMarcusStoinis() {
    return MarcusStoinis;
}

public Hashtable getAlexCarey() {
    return AlexCarey;
}

public Hashtable getNathanLyon() {
    return NathanLyon;
}

public Hashtable getKaneRichardson() {
    return KaneRichardson;
}
```

bangladeshPlayerProfile class:

```
package sample;

import java.util.ArrayList;
import java.util.Hashtable;

public class bangladeshPlayerProfile extends
playerProfileDictionaryStructure {
    private Hashtable ShakibAlHasan =new
Hashtable();
    private Hashtable Mahmudullah =new Hashtable();
    private Hashtable MushfiqurRahim =new
Hashtable();
    private Hashtable LitonDas =new Hashtable();
    private Hashtable MustafizurRahman =new
Hashtable();
    private Hashtable TamimIqbal =new Hashtable();
    private Hashtable SaumyaSarkar = new
Hashtable();
    private Hashtable MashrafeMortaza =new
Hashtable();
    private Hashtable MehidyHasan =new Hashtable();
    private Hashtable MohammadSaifuddin =new
Hashtable();
    private Hashtable SabbirRahman =new
Hashtable();
    private Hashtable RubelHossain =new
Hashtable();

    //Creating an arrayList consisting of all the
    Bangladesh players dictionaries
    private ArrayList BangladeshTeam = new
ArrayList();
    private ArrayList BangladeshBowling = new
ArrayList();

    public ArrayList getBangladeshTeam() {
```

```
        return BangladeshTeam;
    }

    public ArrayList getBangladeshBowling() {
        return BangladeshBowling;
    }

    public bangladeshPlayerProfile() {
        //When user create an object with this
        class, automatically dictionaries get updates
        super.addCommonValuesToTheDictionary(ShakibAlHasan,
        "Shakib Al Hasan", "All Rounder", "Bangladesh");

        super.addCommonValuesToTheDictionary(Mahmudullah, "M
        ahmudullah", "All Rounder", "Bangladesh");

        super.addCommonValuesToTheDictionary(MushfiqurRahim
        , "Mushfiqur Rahim", "Batsman", "Bangladesh");

        super.addCommonValuesToTheDictionary(LitonDas, "Lito
        n Das", "Batsman", "Bangladesh");

        super.addCommonValuesToTheDictionary(MustafizurRahm
        an, "Mustafizur Rahman", "Bowler", "Bangladesh");

        super.addCommonValuesToTheDictionary(TamimIqbal, "Ta
        mim Iqbal", "Batsman", "Bangladesh");

        super.addCommonValuesToTheDictionary(SaumyaSarkar,
        "Saumya Sarkar", "All Rounder", "Bangladesh");

        super.addCommonValuesToTheDictionary(MashrafeMortaza
        , "Mashrafe Mortaza", "Bowler", "Bangladesh");

        super.addCommonValuesToTheDictionary(MehidyHasan,
        "Mehidy Hasan", "All Rounder", "Bangladesh");
    }
}
```

```
super.addCommonValuesToTheDictionary(MohammadSaifudin,"Mohammad Saifuddin","All Rounder","Bangladesh");

super.addCommonValuesToTheDictionary(SabbirRahman,"Sabbir Rahman","All Rounder","Bangladesh");

super.addCommonValuesToTheDictionary(RubelHossain,"Rubel Hossain","Bowler","Bangladesh");

    BangladeshTeam.add(0,ShakibAlHasan);
    BangladeshTeam.add(1,Mahmudullah);
    BangladeshTeam.add(2,MushfiqurRahim);
    BangladeshTeam.add(3,LitonDas);
    BangladeshTeam.add(4,MustafizurRahman);
    BangladeshTeam.add(5,TamimIqbal);
    BangladeshTeam.add(6,SaumyaSarkar);
    BangladeshTeam.add(7,MashrafeMortaza);
    BangladeshTeam.add(8,MehidyHasan);
    BangladeshTeam.add(9,MohammadSaifuddin);
    BangladeshTeam.add(10,SabbirRahman);
    BangladeshTeam.add(11,RubelHossain);
}

public Hashtable getShakibAlHasan() {
    return ShakibAlHasan;
}

public Hashtable getMahmudullah() {
    return Mahmudullah;
}

public Hashtable getMushfiqurRahim() {
    return MushfiqurRahim;
}

public Hashtable getLitonDas() {
    return LitonDas;
}
```

```
public Hashtable getMustafizurRahman() {
    return MustafizurRahman;
}

public Hashtable getTamimIqbal() {
    return TamimIqbal;
}

public Hashtable getSaumyaSarkar() {
    return SaumyaSarkar;
}

public Hashtable getMashrafeMortaza() {
    return MashrafeMortaza;
}

public Hashtable getMehidyHasan() {
    return MehidyHasan;
}

public Hashtable getMohommadSaifuddin() {
    return MohommadSaifuddin;
}

public Hashtable getSabbirRahman() {
    return SabbirRahman;
}

public Hashtable getRubelHossain() {
    return RubelHossain;
}
```

englandPlayerProfile class:

```
package sample;

import java.util.ArrayList;
import java.util.Hashtable;

public class englandPlayerProfile extends
playerProfileDictionaryStructure {
    private Hashtable JoeRoot =new Hashtable();
    private Hashtable BenStokes =new Hashtable();
    private Hashtable JosButtler =new Hashtable();
    private Hashtable MoeenAli =new Hashtable();
    private Hashtable EoinMorgan =new Hashtable();
    private Hashtable JonnyBairstow =new
Hashtable();
    private Hashtable ChrisWoakes = new
Hashtable();
    private Hashtable JofraArcher =new Hashtable();
    private Hashtable JasonRoy =new Hashtable();
    private Hashtable AdilRashid =new Hashtable();
    private Hashtable DomSibley =new Hashtable();
    private Hashtable MarkWood =new Hashtable();

    //Creating an arrayList consisting of all the
    England players dictionaries
    private ArrayList EnglandTeam = new
ArrayList();
    private ArrayList EnglandBowling = new
ArrayList();

    public ArrayList getEnglandTeam() {
        return EnglandTeam;
    }

    public ArrayList getEnglandBowling() {
        return EnglandBowling;
    }
}
```

```
public englandPlayerProfile() {
    //When user create an object with this
class, automatically dictionaries get updates

super.addCommonValuesToTheDictionary(JoeRoot, "Joe
Root", "Batsman", "England");

super.addCommonValuesToTheDictionary(BenStokes, "Ben
Stokes", "All Rounder", "England");

super.addCommonValuesToTheDictionary(JosButtler, "Jo
s Buttler", "Batsman", "England");

super.addCommonValuesToTheDictionary(MoeenAli, "Moe
en Ali", "All Rounder", "England");

super.addCommonValuesToTheDictionary(EoinMorgan, "Eo
in Morgan", "Batsman", "England");

super.addCommonValuesToTheDictionary(JonnyBairstow,
"Jonny Bairstow", "Batsman", "England");

super.addCommonValuesToTheDictionary(ChrisWoakes, "C
hris Woakes", "Bowler", "England");

super.addCommonValuesToTheDictionary(JofraArcher, "J
ofra Archer", "Bowler", "England");

super.addCommonValuesToTheDictionary(JasonRoy, "Jaso
n Roy", "Batsman", "England");

super.addCommonValuesToTheDictionary(AdilRashid, "Ad
il Rashid", "Bowler", "England");

super.addCommonValuesToTheDictionary(DomSibley, "Dom
Sibley", "Batsman", "England");
```

```
super.addCommonValuesToTheDictionary(MarkWood, "Mark  
Wood", "Bowler", "England");  
  
    EnglandTeam.add(0, JoeRoot);  
    EnglandTeam.add(1, BenStokes);  
    EnglandTeam.add(2, JosButtler);  
    EnglandTeam.add(3, MoeenAli);  
    EnglandTeam.add(4, EoinMorgan);  
    EnglandTeam.add(5, JonnyBairstow);  
    EnglandTeam.add(6, ChrisWoakes);  
    EnglandTeam.add(7, JofraArcher);  
    EnglandTeam.add(8, JasonRoy);  
    EnglandTeam.add(9, AdilRashid);  
    EnglandTeam.add(10, DomSibley);  
    EnglandTeam.add(11, MarkWood);  
  
}  
public Hashtable getJoeRoot() {  
    return JoeRoot;  
}  
  
public Hashtable getBenStokes() {  
    return BenStokes;  
}  
  
public Hashtable getJosButtler() {  
    return JosButtler;  
}  
  
public Hashtable getMoeenAli() {  
    return MoeenAli;  
}  
  
public Hashtable getEoinMorgan() {  
    return EoinMorgan;  
}  
  
public Hashtable getJonnyBairstow() {
```

```
        return JonnyBairstow;
    }

    public Hashtable getChrisWoakes() {
        return ChrisWoakes;
    }

    public Hashtable getJofraArcher() {
        return JofraArcher;
    }

    public Hashtable getJasonRoy() {
        return JasonRoy;
    }

    public Hashtable getAdilRashid() {
        return AdilRashid;
    }

    public Hashtable getDomSibley() {
        return DomSibley;
    }

    public Hashtable getMarkWood() {
        return MarkWood;
    }
}
```

indiaPlayerProfile class:

```
package sample;

import java.util.ArrayList;
import java.util.Hashtable;

public class indiaPlayerProfile extends
playerProfileDictionaryStructure {
    private Hashtable ViratKohli =new Hashtable();
    private Hashtable RohitSharma =new Hashtable();
    private Hashtable KLRahul =new Hashtable();
    private Hashtable MSDhoni =new Hashtable();
    private Hashtable RishabhPant =new Hashtable();
    private Hashtable HardikPandya =new
Hashtable();
    private Hashtable MohammadShami = new
Hashtable();
    private Hashtable RavindraJadeja =new
Hashtable();
    private Hashtable JaspritBumrah =new
Hashtable();
    private Hashtable SharbulThakur =new
Hashtable();
    private Hashtable YuzvendraChahal =new
Hashtable();
    private Hashtable BhuvneshwarKumar =new
Hashtable();

    //Creating an arrayList consisting of all the
    India players dictionaries
    private ArrayList IndiaTeam = new ArrayList();
    private ArrayList IndiaBowling = new
ArrayList();

    public ArrayList getIndiaTeam() {
        return IndiaTeam;
    }
}
```

```
public ArrayList getIndiaBowling() {
    return IndiaBowling;
}
public indiaPlayerProfile() {
    //When user create an object with this
class, automatically dictionaries get updates

super.addCommonValuesToTheDictionary(ViratKohli,"Vi
rat Kohli","Batsman","India");

super.addCommonValuesToTheDictionary(RohitSharma,"R
ohit Sharma","Batsman","India");

super.addCommonValuesToTheDictionary(KLRahul,"K L
Rahul","Batsman","India");

super.addCommonValuesToTheDictionary(MSDhoni,"MS
Dhoni","Batsman","India");

super.addCommonValuesToTheDictionary(RishabhPant,"R
ishabh Pant","Batsman","India");

super.addCommonValuesToTheDictionary(HardikPandya,"
Hardik Pandya","All Rounder","India");

super.addCommonValuesToTheDictionary(MohammadShami,
"Mohammad Shami","Bowler","India");

super.addCommonValuesToTheDictionary(RavindraJadeja
,"Ravindra Jadeja","All Rounder","India");

super.addCommonValuesToTheDictionary(JaspritBumrah,
"Jasprit Bumrah","Bowler","India");

super.addCommonValuesToTheDictionary(SharbulThakur,
"Sharbul Thakur","All Rounder","India");
```

```
super.addCommonValuesToTheDictionary(YuzvendraChaha  
l,"Yuzvendra Chahal","Bowler","India");  
  
super.addCommonValuesToTheDictionary(BhuvneshwarKum  
ar,"Bhuvneshwar Kumar","Bowler","India");  
  
    IndiaTeam.add(0,ViratKohli);  
    IndiaTeam.add(1,RohitSharma);  
    IndiaTeam.add(2,KLRahul);  
    IndiaTeam.add(3,MSDhoni);  
    IndiaTeam.add(4,RishabhPant);  
    IndiaTeam.add(5,HardikPandya);  
    IndiaTeam.add(6,MohammadShami);  
    IndiaTeam.add(7,RavindraJadeja);  
    IndiaTeam.add(8,JaspritBumrah);  
    IndiaTeam.add(9,SharbulThakur);  
    IndiaTeam.add(10,YuzvendraChahal);  
    IndiaTeam.add(11,BhuvneshwarKumar);  
}  
public Hashtable getViratKohli() {  
    return ViratKohli;  
}  
  
public Hashtable getRohitSharma() {  
    return RohitSharma;  
}  
  
public Hashtable getKLRahul() {  
    return KLRahul;  
}  
  
public Hashtable getMSDhoni() {  
    return MSDhoni;  
}  
  
public Hashtable getRishabhPant() {  
    return RishabhPant;  
}
```

```
public Hashtable getHardikPandya() {  
    return HardikPandya;  
}  
  
public Hashtable getMohammadShami() {  
    return MohammadShami;  
}  
  
public Hashtable getRavindraJadeja() {  
    return RavindraJadeja;  
}  
  
public Hashtable getJaspritBumrah() {  
    return JaspritBumrah;  
}  
  
public Hashtable getSharbulThakur() {  
    return SharbulThakur;  
}  
  
public Hashtable getYuzvendraChahal() {  
    return YuzvendraChahal;  
}  
  
public Hashtable getBhuvneshwarKumar() {  
    return BhuvneshwarKumar;  
}  
}
```

newZealandPlayerProfile class:

```
package sample;

import java.util.ArrayList;
import java.util.Hashtable;

public class newZealandPlayerProfile extends
playerProfileDictionaryStructure {
    private Hashtable KaneWilliamson =new
Hashtable();
    private Hashtable IshSodhi =new Hashtable();
    private Hashtable MartinGuptill =new
Hashtable();
    private Hashtable RossTailor =new Hashtable();
    private Hashtable TomLatham =new Hashtable();
    private Hashtable TrentBoult =new Hashtable();
    private Hashtable ColinDeGrandhomme = new
Hashtable();
    private Hashtable TimSouthee =new Hashtable();
    private Hashtable LockieFerguson =new
Hashtable();
    private Hashtable JamesNeesham =new
Hashtable();
    private Hashtable MichellSantner =new
Hashtable();
    private Hashtable ColinMunro =new Hashtable();

    //Creating an arrayList consisting of all the
New Zealand players dictionaries
    private ArrayList NewZealandTeam = new
ArrayList();
    private ArrayList NewZealandBowling = new
ArrayList();

    public ArrayList getNewZealandTeam() {
        return NewZealandTeam;
    }
}
```

```
public ArrayList getNewZealandBowling() {  
    return NewZealandBowling;  
}  
  
public newZealandPlayerProfile() {  
    //When user create an object with this  
    class, automatically dictionaries get updates  
  
super.addCommonValuesToTheDictionary(KaneWilliamson,  
,"Kane Williamson","Batsman","New Zealand");  
  
super.addCommonValuesToTheDictionary(IshSodhi,"Ish  
Sodhi","Bowler","New Zealand");  
  
super.addCommonValuesToTheDictionary(MartinGuptill,  
"Martin Guptill","Batsman","New Zealand");  
  
super.addCommonValuesToTheDictionary(RossTailor,"Ro  
ss Tailor","Batsman","New Zealand");  
  
super.addCommonValuesToTheDictionary(TomLatham,"Tom  
Latham","Batsman","New Zealand");  
  
super.addCommonValuesToTheDictionary(TrentBoult,"Tr  
ent Boult","Bowler","New Zealand");  
  
super.addCommonValuesToTheDictionary(ColinDeGrandho  
mme,"Colin De Grandhomme","All Rounder","New  
Zealand");  
  
super.addCommonValuesToTheDictionary(TimSouthee,"Ti  
m Southee","Bowler","New Zealand");  
  
super.addCommonValuesToTheDictionary(LockieFerguson  
,"Lockie Ferguson","Bowler","New Zealand");  
  
super.addCommonValuesToTheDictionary(JamesNeesham,"
```

```
James Neesham", "All Rounder", "New Zealand") ;  
  
super.addCommonValuesToTheDictionary(MichellSantner  
, "Michell Santner", "All Rounder", "New Zealand") ;  
  
super.addCommonValuesToTheDictionary(ColinMunro, "Co  
lin Munro", "All Rounder", "New Zealand") ;  
  
NewZealandTeam.add(0, KaneWilliamson) ;  
NewZealandTeam.add(1, IshSodhi) ;  
NewZealandTeam.add(2, MartinGuptill) ;  
NewZealandTeam.add(3, RossTailor) ;  
NewZealandTeam.add(4, TomLatham) ;  
NewZealandTeam.add(5, TrentBoult) ;  
NewZealandTeam.add(6, ColinDeGrandhomme) ;  
NewZealandTeam.add(7, TimSouthee) ;  
NewZealandTeam.add(8, LockieFerguson) ;  
NewZealandTeam.add(9, JamesNeesham) ;  
NewZealandTeam.add(10, MichellSantner) ;  
NewZealandTeam.add(11, ColinMunro) ;  
}  
public Hashtable getKaneWilliamson() {  
    return KaneWilliamson;  
}  
  
public Hashtable getIshSodhi() {  
    return IshSodhi;  
}  
  
public Hashtable getMartinGuptill() {  
    return MartinGuptill;  
}  
  
public Hashtable getRossTailor() {  
    return RossTailor;  
}  
  
public Hashtable getTomLatham() {
```

```
        return TomLatham;
    }

    public Hashtable getTrentBoult() {
        return TrentBoult;
    }

    public Hashtable getColinDeGrandhomme() {
        return ColinDeGrandhomme;
    }

    public Hashtable getTimSouthee() {
        return TimSouthee;
    }

    public Hashtable getLockieFerguson() {
        return LockieFerguson;
    }

    public Hashtable getJamesNeesham() {
        return JamesNeesham;
    }

    public Hashtable getMichellSantner() {
        return MichellSantner;
    }

    public Hashtable getColinMunro() {
        return ColinMunro;
    }
}
```

pakistanPlayerProfile class:

```
package sample;

import java.util.ArrayList;
import java.util.Hashtable;

public class pakistanPlayerProfile extends
playerProfileDictionaryStructure {
    private Hashtable BabarAzam =new Hashtable();
    private Hashtable ShaheenAfridi =new
Hashtable();
    private Hashtable MohammadAmir =new
Hashtable();
    private Hashtable ShoaibMalik =new Hashtable();
    private Hashtable AsifAli =new Hashtable();
    private Hashtable HasanAli =new Hashtable();
    private Hashtable MohammadHafeez = new
Hashtable();
    private Hashtable FakharZaman =new Hashtable();
    private Hashtable SarfarazAhmad =new
Hashtable();
    private Hashtable ShadabKhan =new Hashtable();
    private Hashtable ImadWasim =new Hashtable();
    private Hashtable WahabRiaz =new Hashtable();

    //Creating an arrayList consisting of all the
    Pakistan players dictionaries
    private ArrayList PakistanTeam = new
ArrayList();
    private ArrayList PakistanBowling = new
ArrayList();

    public ArrayList getPakistanTeam() {
        return PakistanTeam;
    }

    public ArrayList getPakistanBowling() {
```

```
        return PakistanBowling;
    }

    public pakistanPlayerProfile() {
        //When user create an object with this
        class, automatically dictionaries get updates

        super.addCommonValuesToTheDictionary(BabarAzam, "Bab
ar Azam", "Batsman", "Pakistan");

        super.addCommonValuesToTheDictionary(ShaheenAfridi,
"Shaheen Afridi", "Bowler", "Pakistan");

        super.addCommonValuesToTheDictionary(MohammadAmir, "M
ohammad Amir", "Bowler", "Pakistan");

        super.addCommonValuesToTheDictionary(ShoaibMalik, "S
hoaib Malik", "All Rounder", "Pakistan");

        super.addCommonValuesToTheDictionary(AsifAli, "Asif
Ali", "Batsman", "Pakistan");

        super.addCommonValuesToTheDictionary(HasanAli, "Hasa
n Ali", "Bowler", "Pakistan");

        super.addCommonValuesToTheDictionary(MohammadHafeez
, "Mohammad Hafeez", "All Rounder", "Pakistan");

        super.addCommonValuesToTheDictionary(FakharZaman, "F
akhar Zaman", "Batsman", "Pakistan");

        super.addCommonValuesToTheDictionary(SarfrazAhmad,
"Sarfraz Ahmad", "Batsman", "Pakistan");

        super.addCommonValuesToTheDictionary(ShadabKhan, "Sh
adab Khan", "All Rounder", "Pakistan");

        super.addCommonValuesToTheDictionary(ImadWasim, "Ima
```

```
d Wasim", "All Rounder", "Pakistan");  
  
super.addCommonValuesToTheDictionary(WahabRiaz, "Wah  
ab Riaz", "All Rounder", "Pakistan");  
  
PakistanTeam.add(0, BabarAzam);  
PakistanTeam.add(1, ShaheenAfridi);  
PakistanTeam.add(2, MohammadAmir);  
PakistanTeam.add(3, ShoaibMalik);  
PakistanTeam.add(4, AsifAli);  
PakistanTeam.add(5, HasanAli);  
PakistanTeam.add(6, MohammadHafeez);  
PakistanTeam.add(7, FakharZaman);  
PakistanTeam.add(8, SarfarazAhmad);  
PakistanTeam.add(9, ShadabKhan);  
PakistanTeam.add(10, ImadWasim);  
PakistanTeam.add(11, WahabRiaz);  
}  
public Hashtable getBabarAzam() {  
    return BabarAzam;  
}  
  
public Hashtable getShaheenAfridi() {  
    return ShaheenAfridi;  
}  
  
public Hashtable getMohammadAmir() {  
    return MohammadAmir;  
}  
  
public Hashtable getShoaibMalik() {  
    return ShoaibMalik;  
}  
  
public Hashtable getAsifAli() {  
    return AsifAli;  
}
```

```
public Hashtable getHasanAli() {  
    return HasanAli;  
}  
  
public Hashtable getMohammadHafeez() {  
    return MohammadHafeez;  
}  
  
public Hashtable getFakharZaman() {  
    return FakharZaman;  
}  
  
public Hashtable getSarfrazAhmad() {  
    return SarfrazAhmad;  
}  
  
public Hashtable getShadabKhan() {  
    return ShadabKhan;  
}  
  
public Hashtable getImadWasim() {  
    return ImadWasim;  
}  
  
public Hashtable getWahabRiaz() {  
    return WahabRiaz;  
}  
}
```

sriLankaPlayerProfile class:

```
package sample;

import java.util.ArrayList;
import java.util.Hashtable;

public class sriLankaPlayerProfile extends
playerProfileDictionaryStructure {
    private Hashtable DimuthKarunaratne =new
Hashtable();
    private Hashtable LasithMalinga =new
Hashtable();
    private Hashtable DushmanthaChameera =new
Hashtable();
    private Hashtable KusalPerera =new Hashtable();
    private Hashtable KusalMendis =new Hashtable();
    private Hashtable AvishkaFernando =new
Hashtable();
    private Hashtable MaheeshTheekshana = new
Hashtable();
    private Hashtable DhananjayaDeSilva =new
Hashtable();
    private Hashtable AnjeloMathews =new
Hashtable();
    private Hashtable IsuruUdana =new Hashtable();
    private Hashtable ThisaraPerera =new
Hashtable();
    private Hashtable SurangaLakmal =new
Hashtable();

    //Creating an arrayList consisting of all the
Sri Lankan players dictionaries
    private ArrayList SriLankaTeam = new
ArrayList();
    private ArrayList SriLankaBowling = new
ArrayList();
```

```
public ArrayList getSriLankaTeam() {  
    return SriLankaTeam;  
}  
  
public ArrayList getSriLankaBowling() {  
    return SriLankaBowling;  
}  
public sriLankaPlayerProfile() {  
    //When user create an object with this  
class, automatically dictionaries get updates  
  
super.addCommonValuesToTheDictionary(DimuthKarunara  
tne,"Dimuth Karunaratne","Batsman","Sri Lanka");  
  
super.addCommonValuesToTheDictionary(LasithMalinga,  
"Lasith Malinga","Bowler","Sri Lanka");  
  
super.addCommonValuesToTheDictionary(DushmanthaCham  
eera,"Dushmantha Chameera","Bowler","Sri Lanka");  
  
super.addCommonValuesToTheDictionary(KusalPerera,"K  
usal Perera","Batsman","Sri Lanka");  
  
super.addCommonValuesToTheDictionary(KusalMendis,"K  
usal Mendis","Batsman","Sri Lanka");  
  
super.addCommonValuesToTheDictionary(AvishkaFernand  
o,"Avishka Fernando","Batsman","Sri Lanka");  
  
super.addCommonValuesToTheDictionary(MaheeshTheeksh  
ana,"Maheesh Theekshana","Bowler","Sri Lanka");  
  
super.addCommonValuesToTheDictionary(DhananjayaDeSi  
lva,"Dhananjaya De Silva","All Rounder","Sri  
Lanka");  
  
super.addCommonValuesToTheDictionary(AnjeloMathews,  
"Anjelo Mathews","All Rounder","Sri Lanka");
```

```
super.addCommonValuesToTheDictionary(IsuruUdana, "Is  
uru Udana", "All Rounder", "Sri Lanka");  
  
super.addCommonValuesToTheDictionary(ThisaraPerera,  
"Thisara Perera", "Bowler", "Sri Lanka");  
  
super.addCommonValuesToTheDictionary(SurangaLakmal,  
"Suranga Lakmal", "All Rounder", "Sri Lanka");  
  
SriLankaTeam.add(0, DimuthKarunaratne);  
SriLankaTeam.add(1, LasithMalinga);  
SriLankaTeam.add(2, DushmanthaChameera);  
SriLankaTeam.add(3, KusalPerera);  
SriLankaTeam.add(4, KusalMendis);  
SriLankaTeam.add(5, AvishkaFernando);  
SriLankaTeam.add(6, MaheeshTheekshana);  
SriLankaTeam.add(7, DhananjayaDeSilva);  
SriLankaTeam.add(8, AnjeloMathews);  
SriLankaTeam.add(9, IsuruUdana);  
SriLankaTeam.add(10, ThisaraPerera);  
SriLankaTeam.add(11, SurangaLakmal);  
}  
public Hashtable getDimuthKarunaratne() {  
    return DimuthKarunaratne;  
}  
  
public Hashtable getLasithMalinga() {  
    return LasithMalinga;  
}  
  
public Hashtable getDushmanthaChameera() {  
    return DushmanthaChameera;  
}  
  
public Hashtable getKusalPerera() {  
    return KusalPerera;  
}
```

```
public Hashtable getKusalMendis() {  
    return KusalMendis;  
}  
  
public Hashtable getAvishkaFernando() {  
    return AvishkaFernando;  
}  
  
public Hashtable getMaheeshTheekshana() {  
    return MaheeshTheekshana;  
}  
  
public Hashtable getDhananjayaDeSilva() {  
    return DhananjayaDeSilva;  
}  
  
public Hashtable getAnjeloMathews() {  
    return AnjeloMathews;  
}  
  
public Hashtable getIsuruUdana() {  
    return IsuruUdana;  
}  
  
public Hashtable getThisaraPerera() {  
    return ThisaraPerera;  
}  
  
public Hashtable getSurangaLakmal() {  
    return SurangaLakmal;  
}  
}
```

best5BowlersAndBatsman class:

```
package sample;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.Dictionary;

public class best5BowlersAndBatsman {
    private ArrayList allThePlayers = new
ArrayList();
    private int[]
maximumWicketsOrScoreTakenByPlayers = new int[5];

    public best5BowlersAndBatsman(ArrayList
allThePlayers) {
        this.allThePlayers = allThePlayers;
    }

    public void displayBest5Bowlers() {
addValuesToTheArrayAndSortInDescendingOrder("ball")
;
        int
newLengthOfTheArray=removeDuplicateElementInAnArray
(maximumWicketsOrScoreTakenByPlayers,
maximumWicketsOrScoreTakenByPlayers.length);
        int noOfBestBowlers = 0;
        for (int i=0;i< newLengthOfTheArray;i++) {
            for (int
j=0;j<allThePlayers.size();j++) {
                Dictionary playerProfile =
(Dictionary) allThePlayers.get(j);
                if
(maximumWicketsOrScoreTakenByPlayers[i]==(int)playe
rProfile.get("Wickets")) {
```

```
System.out.println(playerProfile.get("Name")+" has
taken "+ maximumWicketsOrScoreTakenByPlayers[i]+"
wickets.");
                noOfBestBowlers++;
                if (noOfBestBowlers==5) {

System.out.println("*****");
                    break;
                }
            }
        }
    }

public void displayBest5Batsmen() {

addValuesToTheArrayAndSortInDescendingOrder("bat");
    int
newLengthOfTheArray=removeDuplicateElementInAnArray
(maximumWicketsOrScoreTakenByPlayers,
maximumWicketsOrScoreTakenByPlayers.length);
    int noOfBestBatsmen = 0;
    for (int i=0;i< newLengthOfTheArray;i++) {
        for (int
j=0;j<allThePlayers.size();j++) {
            Dictionary playerProfile =
(Dictionary) allThePlayers.get(j);
            if
(maximumWicketsOrScoreTakenByPlayers[i]==(int)playe
rProfile.get("Runs")) {

System.out.println(playerProfile.get("Name")+" has
taken "+ maximumWicketsOrScoreTakenByPlayers[i]+"
runs.");
                noOfBestBatsmen++;
                if (noOfBestBatsmen==5) {
                    break;
                }
            }
        }
    }
}
```

```

        }
    }

    private void
addValuesToTheArrayAndSortInDescendingOrder (String
option) {
    String feature = null;
    if (option.equals ("bat")) {
        feature = "Runs";
    } else if (option.equals ("ball")) {
        feature = "Wickets";
    }
    Integer[] wicketsOrScoreOfAllThePlayers =
new Integer [allThePlayers.size ()];
    for (int i=0;i<allThePlayers.size ();i++) {
        Dictionary playerProfile =
(Dictionary) allThePlayers.get (i);
        wicketsOrScoreOfAllThePlayers [i] =
(Integer) playerProfile.get (feature);
    }
    Arrays.sort (wicketsOrScoreOfAllThePlayers,
Collections.reverseOrder()); //Sort the values in
descending order
    for (int i=0;i<5;i++) {
        maximumWicketsOrScoreTakenByPlayers [i]
= wicketsOrScoreOfAllThePlayers [i];
    }
}

private int removeDuplicateElementInAnArray (int
arr[], int arrayLength) {
    if (arrayLength==0 || arrayLength==1)
        return arrayLength;

    int[] temp = new int [arrayLength];
    int j = 0;
    for (int i=0; i<arrayLength-1; i++)
        if (arr[i] != arr[i+1])

```

```

        temp[j++] = arr[i];

        temp[j++] = arr[arrayLength-1];

        // Changing original array
        for (int i=0; i<j; i++)
            arr[i] = temp[i];

        return j;
    }
}

```

Controller class:

```

package sample;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;

import java.io.IOException;
import java.net.URL;
import java.util.Locale;
import java.util.ResourceBundle;

public class Controller implements Initializable{
    private Stage stage;

```

```
private Scene scene;
private Parent root;
Main main = new Main();
//These text fields are to edit or delete team
profiles
@FXML
TextField
askUserToEditTeamProfile,getOption,teamName,
fieldNameTeamProfile,newValue,doneWithEditing,doneW
ithDeleting;

//These text fields are for edit or delete
player profiles
@FXML
TextField
askUserToEditPlayerProfile,playerName,fieldNamePlay
erProfile,doneWithEditingPlayer,doneWithDeletingPla
yer;
//These text fields are for display team
profiles
@FXML
TextField
askToDisplayTeamProfile,teamNameToDisplay;
//These text fields are for display player
profiles
@FXML
TextField askToDisplayPlayerProfile;
//These text fields are to get the match
summery,best 5 bowlers and batsmen,tournament
standing
@FXML
TextField
askToDisplayMatchSummery,askToDisplayBest5Batsmen,a
skToDisplayBest5Bowlers,askToDisplayTournamentStand
ing;
@FXML
private Label errorMessageLabel;
@FXML
```

```
private Button addButton;

private String errorMessage = "";
//To check is the fields are empty, used when
display the match summery
private boolean ifFieldFilled() {
    boolean isFilled = true;
    try {
        if
(askToDisplayMatchSummery.getText().isEmpty()) {
            isFilled = false;
            errorMessage = "First text field is
empty!";
        }

        if
(askToDisplayBest5Batsmen.getText().isEmpty()) {
            isFilled = false;
            if (errorMessage.isEmpty()) {
                errorMessage = "Second text
field is empty!";
            }else{
                errorMessage+="\nSecond text
field is empty!";
            }
        }
        if
(askToDisplayBest5Bowlers.getText().isEmpty()) {
            isFilled = false;
            if (errorMessage.isEmpty()) {
                errorMessage = "Third text
field is empty!";
            }else
                errorMessage+="\nThird text
field is empty!";
        }
    }
}
```

```
(askToDisplayTournamentStanding.getText().isEmpty())
) {
    isFilled = false;
    if (errorMessage.isEmpty()){
        errorMessage = "Fourth text
field is empty!";
    }else
        errorMessage+="\nFourth text
field is empty!";
}

errorMessageLabel.setText(errorMessage);
}catch (NullPointerException e){
    //Nothing happened
}
return isFilled;
}

private boolean isValid(){
    boolean isValid = true;
    try {
        if
(! (askToDisplayMatchSummery.getText().toLowerCase()
.equals("yes") ||
askToDisplayMatchSummery.getText().toLowerCase().eq
uals("no"))){
            isValid = false;
            errorMessage = "Invalid input in
first text field!";
        }
        if
(! (askToDisplayBest5Batsmen.getText().toLowerCase()
.equals("yes") ||
askToDisplayBest5Batsmen.getText().toLowerCase().eq
uals("no"))){
            isValid = false;
            errorMessage = "Invalid input in
second text field!";
        }
    }
}
```

```

        if
        (! (askToDisplayBest5Bowlers.getText() .toLowerCase()
        .equals("yes") ||

askToDisplayBest5Bowlers.getText() .toLowerCase() .eq
uals("no"))){

            isValid = false;
            errorMessage = "Invalid input in
third text field!";

        }

        if
        (! (askToDisplayTournamentStanding.getText() .toLowerCase()
        .equals("yes") ||

askToDisplayTournamentStanding.getText() .toLowerCaseCas
e() .equals("no"))){

            isValid = false;
            errorMessage = "Invalid input in
fourth text field!";

        }

errorMessageLabel.setText(errorMessage);
    }
    catch (NullPointerException e){
        //Nothing happened
    }
    return isValid;
}

```

```

public void switchToScene2(ActionEvent event)
throws IOException {
    if
(askUserUpEditTeamProfile.getText() .toLowerCase() .e
quals("yes") &&
getOption.getText() .toLowerCase() .equals("edit")) {

displayScene(event, "guiForEditTeamProfile.fxml");
} else if

```

```
(askUserToEditTeamProfile.getText().toLowerCase().equals("no")) {  
  
    displayScene(event, "guiAskingForEditDeletePlayerProfile.fxml");  
    } else if  
(askUserToEditTeamProfile.getText().toLowerCase().equals("yes") &&  
getOption.getText().toLowerCase().equals("delete"))  
{  
  
    displayScene(event, "guiForDeleteTeamProfile.fxml");  
    } else if (getOption.getText().isEmpty() ||  
askUserToEditTeamProfile.getText().isEmpty()) {  
    System.out.println("Text field is  
empty!");  
    }  
    else  
        System.out.println("Invalid input");  
}  
public void switchToScene3(ActionEvent event)  
throws IOException{  
    if  
(askUserToEditPlayerProfile.getText().toLowerCase()  
.equals("yes") &&  
getOption.getText().toLowerCase().equals("edit")) {  
  
    displayScene(event, "guiForEditPlayerProfile.fxml");  
    } else if  
(askUserToEditPlayerProfile.getText().toLowerCase()  
.equals("no")) {  
  
    displayScene(event, "guiToDisplayTeamProfile.fxml");  
    } else if  
(askUserToEditPlayerProfile.getText().toLowerCase()  
.equals("yes") &&  
getOption.getText().toLowerCase().equals("delete"))  
{
```

```
displayScene(event, "guiForDeletePlayerProfile.fxml");
}
} else
    System.out.println("Invalid input");
}

public void switchToScene4(ActionEvent event)
throws IOException{
    if
(askToDisplayTeamProfile.getText().toLowerCase().equals("yes")){
main.displayTeamProfile(teamNameToDisplay.getText());
}
} else if
(askToDisplayTeamProfile.getText().toLowerCase().equals("no")){
displayScene(event, "guiToDisplayPlayerProfile.fxml");
}
} else
    System.out.println("Invalid input");
}

public void switchToScene5(ActionEvent event)
throws IOException{
    if
(askToDisplayPlayerProfile.getText().toLowerCase().equals("yes")){
main.displayPlayerProfile(teamNameToDisplay.getText());
}
} else if
(askToDisplayPlayerProfile.getText().toLowerCase().equals("no")){
displayScene(event, "guiToDisplayMatchSummary.fxml");
}
} else
```

```
        System.out.println("Invalid input");
    }
    public void deletePlayerProfile(ActionEvent event) throws IOException{
        main.deletePlayerProfile(playerName.getText());
        if
        (doneWithDeletingPlayer.getText().toLowerCase().equals("yes")){
            displayScene(event,"guiAskingForEditDeletePlayerProfile.fxml");
        }
    }

    public void editPlayerProfile(ActionEvent event) throws IOException{
        main.editPlayerProfile(playerName.getText(),
        fieldNamePlayerProfile.getText(),newValue.getText());
        if
        (doneWithEditingPlayer.getText().toLowerCase().equals("yes")){
            displayScene(event,"guiAskingForEditDeletePlayerProfile.fxml");
        }
    }

    public void editTeamProfile(ActionEvent event)
throws IOException{
        main.editTeamProfile("edit",teamName.getText(),
        fieldNameTeamProfile.getText(),newValue.getText());
        if
        (doneWithEditing.getText().toLowerCase().equals("yes")){

```

```
displayScene(event, "guiAskingForEditAndDeleteTeamPr
ofile.fxml");
    }
}
public void deleteTeamProfile(ActionEvent
event) throws IOException{
    main.deleteTeamProfile(teamName.getText());
    if
(doneWithDeleting.getText().toLowerCase().equals("y
es")) {

displayScene(event, "guiAskingForEditAndDeleteTeamPr
ofile.fxml");
    }
}
public void displayScene(ActionEvent
event, String fxmlFileName) throws IOException {
    Parent root =
FXMLLoader.load(getClass().getResource(fxmlFileName
));
    stage = (Stage)
((Node)event.getSource()).getScene().getWindow();
    scene = new Scene(root,1300,700);
    stage.setScene(scene);
    stage.show();
}

int outerLoopCount = 0; //These variables are
for match generation
int innerLoopCount = outerLoopCount+1;
String group = "A";
@Override
public void initialize(URL location,
 ResourceBundle resources) {
    addButton.setOnMouseClicked(new
EventHandler<MouseEvent>() {
```

```

@Override
public void handle(MouseEvent event) {
    System.exit(0);
}
});

addButton.setOnMouseClicked(new
EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        errorMessage = "";
        try {
            if (ifFieldFilled() &&
isValid()) {
                /* Below mechanism is equal to
                   for (int
i=0;i<3;i++) {
                           for (int
j=i+1;j<4;j++) {
main.clickedOnce(askToDisplayMatchSummery.getText()
, askToDisplayBest5Bowlers.getText(), askToDisplayBes
t5Batsmen.getText(), askToDisplayTournamentStanding.
getText(), outerLoopCount, innerLoopCount);
}
}
Did'n't use for loop
here because each time user click the button,new
values should manually pass to the relevant methods
*/
                if (outerLoopCount<3) {
                    if (innerLoopCount<4) {
main.clickedOnce(askToDisplayMatchSummery.getText()
, askToDisplayBest5Bowlers.getText(), askToDisplayBes
t5Batsmen.getText(), askToDisplayTournamentStanding.
getText(), outerLoopCount, innerLoopCount, group);
                        innerLoopCount+=1;
} else if

```

dictionariesForEachTeamProfiles class:

```
package sample;
import java.util.*;

public class dictionariesForEachTeamProfiles {
    //This class will create dictionaries for each
team and we can use this class
    //to edit the dictionary and to display
dictionaries of specific team
    private Hashtable Afganistan=new Hashtable();
    private Hashtable Australia=new Hashtable();
    private Hashtable Bangladesh=new Hashtable();
```

```
private Hashtable England=new Hashtable();
private Hashtable India=new Hashtable();
private Hashtable NewZealand=new Hashtable();
private Hashtable Pakistan=new Hashtable();
private Hashtable SriLanka=new Hashtable();

public dictionariesForEachTeamProfiles() {
    //When a object was created, relevant
dictionaries also will be updated

insertValuesForTheRelavantDictionary(Afganistan,"Af
ganistan",2,8);

insertValuesForTheRelavantDictionary(Australia,"Aus
tralia",7,3);

insertValuesForTheRelavantDictionary(Bangladesh,"Ba
ngladesh",4.1,6);

insertValuesForTheRelavantDictionary(India,"India",
6.5,4);

insertValuesForTheRelavantDictionary(England,"Engla
nd",8,2);

insertValuesForTheRelavantDictionary(NewZealand,"Ne
w Zealand",10,1);

insertValuesForTheRelavantDictionary(Pakistan,"Paki
stan",4,7);

insertValuesForTheRelavantDictionary(SriLanka,"Sri
Lanka",4.45,5);
```

```
}

    private void
insertValuesForTheRelavantDictionary(Hashtable
country, String countryName, double rating,int
position) {
    country.put("Name",countryName);
    country.put("No of matches",0);
    country.put("Position",position);
    country.put("Rating",rating);
}
public Hashtable getAfganistan() {
    return Afganistan;
}

public Hashtable getAustralia() {
    return Australia;
}

public Hashtable getBangladesh() {
    return Bangladesh;
}

public Hashtable getEngland() {
    return England;
}

public Hashtable getIndia() {
    return India;
}

public Hashtable getNewZealand() {
    return NewZealand;
}

public Hashtable getPakistan() {
    return Pakistan;
}
```

```

    }

    public Hashtable getSriLanka() {
        return SriLanka;
    }
}

```

editDeleteAndDisplayPlayerProfiles class:

```

package sample;

import java.util.*;

public class editDeleteAndDisplayPlayerProfiles {
    private Scanner scanner=new Scanner(System.in);
    private ArrayList allThePlayers=new
ArrayList();

    Dictionary playerProfileForEditOrDelete; //This
is used to edit player profile

    public
editDeleteAndDisplayPlayerProfiles(ArrayList
allThePlayers) {
        this.allThePlayers = allThePlayers;
    }

    public void displayPlayerProfile(Object
countryName) {
        for(int
i=0;i<this.allThePlayers.size();i++) {
            Dictionary playerDictionary =
(Dictionary) allThePlayers.get(i);
            Object country=
playerDictionary.get("country");
            if (country.equals(countryName)) {
                printTeam(playerDictionary);
            }
        }
    }
}

```

```
        }

    }

    public void
getUsersInputAndEditPlayerProfile(String
playerName, String feature, String newVal) {
    String player=playerName;
    editPlayerProfile(player, feature, newVal);
}

    public void
getUsersInputAndDeletePlayerProfiles(String
playerName) {
    deletePlayerProfile(playerName);
}

    private void editPlayerProfile(String
playerName, String feature1, String newVal1) {
    boolean isNameUpdated = false;
    if (isPlayerExist(playerName)) {
        String feature = feature1;
        if
(feature.toLowerCase().equals("name")) {
            String newVal = newVal1;

playerProfileForEditOrDelete.put(feature, newVal);
            isNameUpdated = true;
            System.out.println("Player profile
updated.");
        } else if
(feature.toLowerCase().equals("role")) {
            String newVal = newVal1;

playerProfileForEditOrDelete.put(feature, newVal);
            System.out.println("Player profile
updated.");
        } else {
            System.out.println("Invalid
feature");
        }
    }
}
```

```

        }
    }
    if (isNameUpdated==false) {
        if (isPlayerExist(playerName) ==
false) {
            System.out.println("Invalid player
name.");
        }
    }
}

private void deletePlayerProfile(String
playerName){
    boolean isDeletedPlayerProfile = false;
    if (isPlayerExist(playerName)) {
        for (int
i=0;i<allThePlayers.size();i++) {
            if (playerProfileForEditOrDelete ==
(Dictionary) allThePlayers.get(i)) {
                allThePlayers.remove(i);
                isDeletedPlayerProfile = true;
                System.out.println("Deleted
successfully.");
                break;
            }
        }
    }
    if (isDeletedPlayerProfile == false) {
        System.out.println("Invalid player
name");
    }
}
private boolean isPlayerExist(String
playerName) {
    boolean playerFound = false;
    for (int i=0;i<allThePlayers.size();i++) {
        Dictionary playerProfile = (Dictionary)
allThePlayers.get(i);

```

```
        Object player =
playerProfile.get("Name");
        if (player.equals(playerName)) {
            this.playerProfileForEditOrDelete =
playerProfile;
            playerFound=true;
            break;
        }
    }if (playerFound) {
        return true;
}else
    return false;
}

private void printTeam(Dictionary
playerDictionary) {
    System.out.println(("Name =
"+playerDictionary.get("Name")));
    System.out.println("Role =
"+playerDictionary.get("Role"));
    System.out.println("Matches =
"+playerDictionary.get("Matches"));
    System.out.println("Innings =
"+playerDictionary.get("Innings"));
    System.out.println("Runs =
"+playerDictionary.get("Runs"));
    System.out.println("50s =
"+playerDictionary.get("50s"));
    System.out.println("Best =
"+playerDictionary.get("Best"));
    System.out.println("Wickets =
"+playerDictionary.get("Wickets"));
    System.out.println();
}
}
```

editDeleteAndDisplayTeamProfile class:

```
package sample;

import java.util.*;

public class editDeleteAndDisplayTeamProfile {
    private Scanner scanner=new Scanner(System.in);
    private ArrayList allTheTeams = new
ArrayList();

    public
editDeleteAndDisplayTeamProfile(ArrayList
allTheTeams) {
        this.allTheTeams = allTheTeams;
    }
    public void displayTeamProfile(Object
countryName) {
        boolean isTeamExist = false;
        for(int i=0;i<this.allTheTeams.size();i++) {
            Dictionary playerDictionary =
(Dictionary) allTheTeams.get(i);
            Object country=
playerDictionary.get("Name");
            if (country.equals(countryName)) {
                isTeamExist = true;
                printTeam(playerDictionary);
                break;
            }
        }if (!isTeamExist) {
            System.out.println("Team not found.");
        }
    }

    public void
getUsersInputAndEditTeamProfile(String option,
String teamName, String fieldName, String newVal) {
        String team=teamName;
```

```
        if (isValidTeam(team)) {
            if
(option.toLowerCase().equals("edit")) {
                String key =fieldName;
                if (key.equals("No of matches"))
|| key.equals("Position")) {
                    try {
                        int value
=Integer.parseInt(newVal);
editTeamProfile(team,key,value);
displayTeamProfile(team);
                }catch
(NumberFormatException e) {
                    System.out.println("You
have entered an invalid value or the text field is
empty");
                }
            }else {
                System.out.println("Invalid
field.");
            }
        }
    }
    public void
getUsersInputAndDeleteTeamProfile(String team) {
    deleteTeamProfile(team);
}
private void printTeam(Dictionary
playerDictionary) {
    System.out.println(("Name =
"+playerDictionary.get("Name")));
    System.out.println("Number of matches =

```

```

"+playerDictionary.get("No of matches"));
        System.out.println("Position =
"+playerDictionary.get("Position"));
        System.out.println("Rating =
"+playerDictionary.get("Rating"));
        System.out.println();
    }

private boolean isValidTeam(String team) {
    if (team.equals("Afghanistan") ||
team.equals("Australia") ||
team.equals("Bangladesh") || team.equals("England") ||
team.equals("India") || team.equals("New Zealand") || team.equals("Pakistan") || team.equals("Sri Lanka")){
        return true;
    }else
        return false;
}

private boolean isValidOption(String option){
    if (option.toLowerCase().equals("edit") ||
option.toLowerCase().equals("delete")){
        return true;
    }else
        return false;
}

private void editTeamProfile(String
countryName, String key, int value){
    boolean isTeamExist = false;
    for (int i=0;i<allTheTeams.size();i++) {
        Dictionary
teamProfile=(Dictionary) allTheTeams.get(i);
        Object country =
teamProfile.get("Name");
        if (country.equals(countryName)){
            isTeamExist = true;
            teamProfile.put(key,value);
        }
    }
}

```

```
        System.out.println("Successfully
updated");
        break;
    }
} if (!isTeamExist) {
    System.out.println("Team not found.");
}
}

private void deleteTeamProfile(String
teamName) {
    int i;
    for (i=0;i<allTheTeams.size();i++) {
        Dictionary
teamProfile=(Dictionary) allTheTeams.get(i);
        Object country =
teamProfile.get("Name");
        if (country.equals(teamName)) {
            teamProfile.remove("No of
matches");
            System.out.println("Deleted
successfully");
            break;
        }
    } if (i==allTheTeams.size()) {
        System.out.println("Invalid Team
Name.");
    }
}
```

listOfAllThePlayers class:

```
package sample;

import java.util.ArrayList;

public class listOfAllThePlayers {
    //This class will create an arrayList
    //consisting of dictionaries of all the players
    //and return the arrayList
    //Using Composition
    private afghanistanPlayerProfile
afghanistanPlayerProfile;
    private australiaPlayerProfile
australiaPlayerProfile;
    private bangladeshPlayerProfile
bangladeshPlayerProfile;
    private englandPlayerProfile
englandPlayerProfile;
    private indiaPlayerProfile indiaPlayerProfile;
    private newZealandPlayerProfile
newZealandPlayerProfile;
    private pakistanPlayerProfile
pakistanPlayerProfile;
    private sriLankaPlayerProfile
sriLankaPlayerProfile;

    public
listOfAllThePlayers(sample.afghanistanPlayerProfile
afghanistanPlayerProfile,
sample.australiaPlayerProfile
australiaPlayerProfile,
sample.bangladeshPlayerProfile
bangladeshPlayerProfile,
sample.englandPlayerProfile englandPlayerProfile,
sample.indiaPlayerProfile indiaPlayerProfile,
sample.newZealandPlayerProfile
newZealandPlayerProfile,
```

```
sample.pakistanPlayerProfile pakistanPlayerProfile,
sample.sriLankaPlayerProfile sriLankaPlayerProfile)
{
    //Initializing instance variable
    this.afghanistanPlayerProfile =
afghanistanPlayerProfile;
    this.australiaPlayerProfile =
australiaPlayerProfile;
    this.bangladeshPlayerProfile =
bangladeshPlayerProfile;
    this.englandPlayerProfile =
englandPlayerProfile;
    this.indiaPlayerProfile =
indiaPlayerProfile;
    this.newZealandPlayerProfile =
newZealandPlayerProfile;
    this.pakistanPlayerProfile =
pakistanPlayerProfile;
    this.sriLankaPlayerProfile =
sriLankaPlayerProfile;
}

private ArrayList allThePlayers=new
ArrayList();
public void insertValuesToTheArrayList() {
    //Inserting values to the array

allThePlayers.add(afghanistanPlayerProfile.getRashid
Khan());
allThePlayers.add(afghanistanPlayerProfile.getMohomm
adNabi());
allThePlayers.add(afghanistanPlayerProfile.getAsghar
Afghan());
allThePlayers.add(afghanistanPlayerProfile.getHashma
```

```
tullahShahidi()) ;  
  
allThePlayers.add(afghanistanPlayerProfile.getMujeeb  
Rahman()) ;  
  
allThePlayers.add(afghanistanPlayerProfile.getHamidH  
asan()) ;  
  
allThePlayers.add(afghanistanPlayerProfile.getRahmat  
Shah()) ;  
  
allThePlayers.add(afghanistanPlayerProfile.getNoorAl  
i()) ;  
  
allThePlayers.add(afghanistanPlayerProfile.getGulbad  
inNaib()) ;  
  
allThePlayers.add(afghanistanPlayerProfile.getSayedS  
hirzad()) ;  
  
allThePlayers.add(afghanistanPlayerProfile.getIkramA  
likhil()) ;  
  
allThePlayers.add(afghanistanPlayerProfile.getSamiul  
lahShinwari()) ;  
  
  
allThePlayers.add(australiaPlayerProfile.getSteveSm  
ith()) ;  
  
allThePlayers.add(australiaPlayerProfile.getDavidWa  
rner()) ;  
  
allThePlayers.add(australiaPlayerProfile.getAaronFi  
nch()) ;  
  
allThePlayers.add(australiaPlayerProfile.getMitchel  
Starc()) ;
```

```
allThePlayers.add(australiaPlayerProfile.getAdamZam  
pa());  
  
allThePlayers.add(australiaPlayerProfile.getGlennMa  
xwell());  
  
allThePlayers.add(australiaPlayerProfile.getMatthew  
Wade());  
  
allThePlayers.add(australiaPlayerProfile.getPatCumm  
ins());  
  
allThePlayers.add(australiaPlayerProfile.getMarcusS  
toinis());  
  
allThePlayers.add(australiaPlayerProfile.getAlexCar  
ey());  
  
allThePlayers.add(australiaPlayerProfile.getNathanL  
yon());  
  
allThePlayers.add(australiaPlayerProfile.getKaneRic  
hardson());  
  
  
allThePlayers.add(bangladeshPlayerProfile.getShakib  
AlHasan());  
  
allThePlayers.add(bangladeshPlayerProfile.getMahmud  
ullah());  
  
allThePlayers.add(bangladeshPlayerProfile.getMushfi  
qurRahim());  
  
allThePlayers.add(bangladeshPlayerProfile.getLitonD  
as());
```

```
allThePlayers.add(bangladeshPlayerProfile.getMustafizurRahman()) ;  
  
allThePlayers.add(bangladeshPlayerProfile.getTamimIqbal()) ;  
  
allThePlayers.add(bangladeshPlayerProfile.getSaumyaSarkar()) ;  
  
allThePlayers.add(bangladeshPlayerProfile.getMashrafeMortaza()) ;  
  
allThePlayers.add(bangladeshPlayerProfile.getMehidyHasan()) ;  
  
allThePlayers.add(bangladeshPlayerProfile.getMohammadSaifuddin()) ;  
  
allThePlayers.add(bangladeshPlayerProfile.getSabbirRahman()) ;  
  
allThePlayers.add(bangladeshPlayerProfile.getRubelHossain()) ;  
  
allThePlayers.add(englandPlayerProfile.getJoeRoot()) ;  
  
allThePlayers.add(englandPlayerProfile.getBenStokes()) ;  
  
allThePlayers.add(englandPlayerProfile.getJosButtler()) ;  
  
allThePlayers.add(englandPlayerProfile.getMoeenAli()) ;  
  
allThePlayers.add(englandPlayerProfile.getEoinMorgan)
```

```
n() ;  
  
allThePlayers.add(englandPlayerProfile.getJonnyBair  
stow());  
  
allThePlayers.add(englandPlayerProfile.getChrisWoak  
es());  
  
allThePlayers.add(englandPlayerProfile.getJofraArch  
er());  
  
allThePlayers.add(englandPlayerProfile.getJasonRoy(  
));  
  
allThePlayers.add(englandPlayerProfile.getAdilRashi  
d());  
  
allThePlayers.add(englandPlayerProfile.getDomSibley  
());  
  
allThePlayers.add(englandPlayerProfile.getMarkWood(  
));  
  
  
allThePlayers.add(indiaPlayerProfile.getViratKohli(  
));  
  
allThePlayers.add(indiaPlayerProfile.getRohitSharma  
());  
  
allThePlayers.add(indiaPlayerProfile.getKL Rahul());  
  
allThePlayers.add(indiaPlayerProfile.getMSDhoni());  
  
allThePlayers.add(indiaPlayerProfile.getRishabhPant  
());  
  
allThePlayers.add(indiaPlayerProfile.getHardikPandy
```

```
a() ;  
  
allThePlayers.add(indiaPlayerProfile.getMohommadSha  
mi());  
  
allThePlayers.add(indiaPlayerProfile.getRavindraJad  
eja());  
  
allThePlayers.add(indiaPlayerProfile.getJaspritBumr  
ah());  
  
allThePlayers.add(indiaPlayerProfile.getSharbulThak  
ur());  
  
allThePlayers.add(indiaPlayerProfile.getYuzvendraCh  
ahal());  
  
allThePlayers.add(indiaPlayerProfile.getBhuvneshwar  
Kumar());  
  
  
allThePlayers.add(newZealandPlayerProfile.getKaneWi  
lliamson());  
  
allThePlayers.add(newZealandPlayerProfile.getIshSod  
hi());  
  
allThePlayers.add(newZealandPlayerProfile.getMartin  
Guptill());  
  
allThePlayers.add(newZealandPlayerProfile.getRossTa  
ilor());  
  
allThePlayers.add(newZealandPlayerProfile.getTomLat  
ham());  
  
allThePlayers.add(newZealandPlayerProfile.getTrentB  
oult());
```

```
allThePlayers.add(newZealandPlayerProfile.getColinD  
eGrandhomme());  
  
allThePlayers.add(newZealandPlayerProfile.getTimSou  
thee());  
  
allThePlayers.add(newZealandPlayerProfile.getLockie  
Ferguson());  
  
allThePlayers.add(newZealandPlayerProfile.getJamesN  
eesham());  
  
allThePlayers.add(newZealandPlayerProfile.getMichel  
lSantner());  
  
allThePlayers.add(newZealandPlayerProfile.getColinM  
unro());  
  
allThePlayers.add(pakistanPlayerProfile.getBabarAza  
m());  
  
allThePlayers.add(pakistanPlayerProfile.getShaheenA  
fridi());  
  
allThePlayers.add(pakistanPlayerProfile.getMohammad  
Amir());  
  
allThePlayers.add(pakistanPlayerProfile.getShoaibMa  
lik());  
  
allThePlayers.add(pakistanPlayerProfile.getAsifAli(  
));  
  
allThePlayers.add(pakistanPlayerProfile.getHasanAli  
());
```

```
allThePlayers.add(pakistanPlayerProfile.getMohammad
Hafeez()) ;

allThePlayers.add(pakistanPlayerProfile.getFakharZa
man()) ;

allThePlayers.add(pakistanPlayerProfile.getSarfraz
Ahmad()) ;

allThePlayers.add(pakistanPlayerProfile.getShadabKh
an()) ;

allThePlayers.add(pakistanPlayerProfile.getImadWasi
m()) ;

allThePlayers.add(pakistanPlayerProfile.getWahabRia
z()) ;

allThePlayers.add(sriLankaPlayerProfile.getDimuthKa
runaratne()) ;

allThePlayers.add(sriLankaPlayerProfile.getLasithMa
linga()) ;

allThePlayers.add(sriLankaPlayerProfile.getDushman
tHaChameera()) ;

allThePlayers.add(sriLankaPlayerProfile.getKusalMen
dis()) ;

allThePlayers.add(sriLankaPlayerProfile.getKusalPer
era()) ;

allThePlayers.add(sriLankaPlayerProfile.getAvishkaF
ernando()) ;

allThePlayers.add(sriLankaPlayerProfile.getMaheeshT
```

```
heekshana()) ;  
  
allThePlayers.add(sriLankaPlayerProfile.getDhananja  
yaDeSilva());  
  
allThePlayers.add(sriLankaPlayerProfile.getAnjeloMa  
thews());  
  
allThePlayers.add(sriLankaPlayerProfile.getIsuruUda  
na());  
  
allThePlayers.add(sriLankaPlayerProfile.getThisaraP  
erera());  
  
allThePlayers.add(sriLankaPlayerProfile.getSurangaL  
akmal());  
}  
  
public ArrayList getAllThePlayers() {  
    return allThePlayers;  
}  
}
```

listOfAllTheTeams class:

```
package sample;

import java.util.ArrayList;

public class listOfAllTheTeams {
    //This class is used to create an arrayList of
dictionaries of all the teams
    //and return the arrayList
    //using composition
    private dictionariesForEachTeamProfiles
dictionariesForEachTeamProfiles;

    public
listOfAllTheTeams(dictionariesForEachTeamProfiles
dictionariesForEachTeamProfiles) {
        this.dictionariesForEachTeamProfiles =
dictionariesForEachTeamProfiles;
    }
    private ArrayList allTheTeams=new ArrayList();

    //Creating an ArrayList which consists of group
A teams dictionaries
    private ArrayList groupA = new ArrayList();

    //Creating an ArrayList which consists of group
B teams dictionaries
    private ArrayList groupB = new ArrayList();
    public void addValuesToTheArray() {
        //Inserting values to the array

allTheTeams.add(dictionariesForEachTeamProfiles.get
Afghanistan());
    }

    allTheTeams.add(dictionariesForEachTeamProfiles.get
Australia());
}
```

```
allTheTeams.add(dictionariesForEachTeamProfiles.get
Bangladesh()) ;

allTheTeams.add(dictionariesForEachTeamProfiles.get
England()) ;

allTheTeams.add(dictionariesForEachTeamProfiles.get
India()) ;

allTheTeams.add(dictionariesForEachTeamProfiles.get
NewZealand()) ;

allTheTeams.add(dictionariesForEachTeamProfiles.get
Pakistan()) ;

allTheTeams.add(dictionariesForEachTeamProfiles.get
SriLanka()) ;

        //Adding elements to the group A
        groupA.add(0,
dictionariesForEachTeamProfiles.getAfganistan());
        groupA.add(1,
dictionariesForEachTeamProfiles.getAustralia());
        groupA.add(2,
dictionariesForEachTeamProfiles.getBangladesh());
        groupA.add(3,
dictionariesForEachTeamProfiles.getEngland());

        //Adding elements to the group B
        groupB.add(0,
dictionariesForEachTeamProfiles.getIndia());
        groupB.add(1,
dictionariesForEachTeamProfiles.getNewZealand());
        groupB.add(2,
dictionariesForEachTeamProfiles.getPakistan());
        groupB.add(3,
dictionariesForEachTeamProfiles.getSriLanka());
    }
```

```

public ArrayList getAllTheTeams() {
    return allTheTeams;
}

public ArrayList getGroupA() {
    return groupA;
}

public ArrayList getGroupB() {
    return groupB;
}
}

```

Main class:

```

package sample;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.util.ArrayList;
import java.util.Dictionary;
import java.util.Scanner;

public class Main extends Application {
    //Creating objects using classes
    private static dictionariesForEachTeamProfiles
dictionariesForEachTeamProfiles =new
dictionariesForEachTeamProfiles();
    private static afghanistanPlayerProfile
afghanistanPlayerProfile=new
afghanistanPlayerProfile();
    private static australiaPlayerProfile
australiaPlayerProfile=new

```

```
australiaPlayerProfile();  
    private static bangladeshPlayerProfile  
bangladeshPlayerProfile=new  
bangladeshPlayerProfile();  
    private static englandPlayerProfile  
englandPlayerProfile=new englandPlayerProfile();  
    private static indiaPlayerProfile  
indiaPlayerProfile=new indiaPlayerProfile();  
    private static newZealandPlayerProfile  
newZealandPlayerProfile=new  
newZealandPlayerProfile();  
    private static pakistanPlayerProfile  
pakistanPlayerProfile=new pakistanPlayerProfile();  
    private static sriLankaPlayerProfile  
sriLankaPlayerProfile=new sriLankaPlayerProfile();  
  
    private static listOfAllThePlayers  
listOfAllThePlayers=new  
listOfAllThePlayers(afghanistanPlayerProfile,austral  
iaPlayerProfile,bangladeshPlayerProfile,englandPlay  
erProfile,indiaPlayerProfile,newZealandPlayerProfil  
e,pakistanPlayerProfile,sriLankaPlayerProfile);  
  
    //This object will be used to display player  
profiles where ever it is needed  
    private static  
editDeleteAndDisplayPlayerProfiles  
editDeleteAndDisplayPlayerProfiles =new  
editDeleteAndDisplayPlayerProfiles(listofAllThePlay  
ers.getAllThePlayers());  
  
    //This object will store dictionaries created  
for each team profile in an arrayList  
    private static listOfAllTheTeams  
listOfAllTheTeams=new  
listOfAllTheTeams(dictionariesForEachTeamProfiles);
```

```
//This object will be used to display player profiles where ever it is needed
    private static editDeleteAndDisplayTeamProfile editDeleteAndDisplayTeamProfile =new editDeleteAndDisplayTeamProfile(listOfAllTheTeams.getListAllTheTeams());  
  
    //This object will be used to get the match summery of 2 teams
    private static teamWonLooseDecisionInOneMatch teamWonLooseDecisionInOneMatch = new teamWonLooseDecisionInOneMatch(listOfAllThePlayers, listOfAllTheTeams);  
  
    //This object will be used to get the the position of the teams according to the rating
    private static positionOfTeams positionOfTeams = new positionOfTeams(listOfAllTheTeams);  
  
    //This object will display a match summery for a given match
    private static overviewOfAMatchSummery matchSummery = new overviewOfAMatchSummery(teamWonLooseDecisionInOneMatch);  
  
    //This object will display best 5 bowlers and batsmen
    private static best5BowlersAndBatsman best5BowlersAndBatsman = new best5BowlersAndBatsman(listOfAllThePlayers.getAllThePlayers());  
  
    //This object will display match information of group A and B matches
    private static tournamentMatches tournamentMatches = new tournamentMatches(teamWonLooseDecisionInOneMatch, ma
```

```
tchSummary,
positionOfTeams, best5BowlersAndBatsman, afghanistanPlayerProfile, australiaPlayerProfile, bangladeshPlayerProfile, englandPlayerProfile, indiaPlayerProfile, newZealandPlayerProfile, pakistanPlayerProfile, sriLankaPlayerProfile);

@Override
public void start(Stage primaryStage) throws Exception{
    Parent root =
FXMLLoader.load(getClass().getResource("guiAskingForEditAndDeleteTeamProfile.fxml"));
    primaryStage.setScene(new Scene(root, 1300,
700));
    primaryStage.show();
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

        //Adding values to the allThePlayers
arrayList
listOfAllThePlayers.insertValuesToTheArrayList();

        //Adding values to the allTheTeams
arrayList
listOfAllTheTeams.addValuestoTheArrayList();

    launch(args);
}

public void editPlayerProfile(String playerName, String feature, String newVal) {
editDeleteAndDisplayPlayerProfiles.getUsersInputAnd
```

```
        EditPlayerProfile(playerName, feature, newVal);
    }
    public void deletePlayerProfile(String
playerName) {
editDeleteAndDisplayPlayerProfiles.getUsersInputAnd
DeletePlayerProfiles(playerName);
}
public void editTeamProfile(String
option, String teamName, String fieldName, String
newValue) {
editDeleteAndDisplayTeamProfile.getUsersInputAndEdi
tTeamProfile(option, teamName, fieldName, newValue);
}
public void deleteTeamProfile(String team) {
editDeleteAndDisplayTeamProfile.getUsersInputAndDel
eteTeamProfile(team);
}
public void displayTeamProfile(String team) {
editDeleteAndDisplayTeamProfile.displayTeamProfile(
team);
}
public void displayPlayerProfile(String team) {
    if (team.equals("Afghanistan") ||
team.equals("Australia") ||
team.equals("Bangladesh") || team.equals("England") ||
team.equals("India") || team.equals("New
Zealand") || team.equals("Pakistan") ||
team.equals("Sri Lanka")){
editDeleteAndDisplayPlayerProfiles.displayPlayerPro
file(team);
}else {
System.out.println("Invalid team
```

```
name." );
        }
    }

    public void clickedOnce(String askToDisplayMatchSummery, String askToDisplayBest5Bowlers, String askToDisplayBest5Batsmen, String askToDisplayTournamentStanding, int outerLoopCount, int innerLoopCount, String groupAOrB) {
        ArrayList group = new ArrayList();
        if (groupAOrB.equals("A")) {
            group = listOfAllTheTeams.getGroupA();
        }else {
            group = listOfAllTheTeams.getGroupB();
        }

        Dictionary team1 = (Dictionary) group.get(outerLoopCount);
        Dictionary team2 = (Dictionary) group.get(innerLoopCount);

        tournamentMatches.matchBetweenTwoTeams(team1, team2);
    }

    getUserInputAndDisplayMatchSummery(askToDisplayMatchSummery);

    getUserInputAndDisplayBest5BatsmenAndBowlers(askToDoDisplayBest5Batsmen, askToDisplayBest5Bowlers);
    System.out.println();

    getUserInputAndDisplayTournamentStanding(askToDisplayTournamentStanding);

    tournamentMatches.resetPlayerFeaturesInOneTeam(team1.get("Name").toString()); //Reset some player features after each match
```

```
tournamentMatches.resetPlayerFeaturesInOneTeam(team
2.getName().toString());
}
public void
displayTeamWhoQualifiedForFinals(String groupAOrB) {
    ArrayList group = new ArrayList();
    if (groupAOrB.equals("A")) {
        group = listOfAllTheTeams.getGroupA();
    }else {
        group = listOfAllTheTeams.getGroupB();
    }

tournamentMatches.displayTeamWhoQualifiedForFinals(
group);
}
public void
getUserInputAndDisplayMatchSummery(String
askUserToDisplayMatchSummery) {

tournamentMatches.getUserInputAndDisplayMatchSummer
y(askUserToDisplayMatchSummery);
}
public void
getUserInputAndDisplayBest5BatsmenAndBowlers(String
askUserToDisplayBest5Batsmen, String
askUserToDisplayBest5Bowlers) {

tournamentMatches.displayBest5BowlersOrBatsmen(askU
serToDisplayBest5Batsmen, askUserToDisplayBest5Bowle
rs);
}
public void
getUserInputAndDisplayTournamentStanding(String
askUserInputAndDisplayTournamentStanding) {

tournamentMatches.displayTournamentStanding(askUser
InputAndDisplayTournamentStanding);
```

```
    }
}
```

overviewOfAMatchSummery class:

```
package sample;

import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Dictionary;
import java.util.Hashtable;

public class overviewOfAMatchSummery {
    teamWonLooseDecisionInOneMatch
teamWonLooseDecisionInOneMatch;

    public
overviewOfAMatchSummery(sample.teamWonLooseDecision
InOneMatch teamWonLooseDecisionInOneMatch) {
        this.teamWonLooseDecisionInOneMatch =
teamWonLooseDecisionInOneMatch;
    }

    public void displayMatchSummery() {
        System.out.println();
        //Printing who won the toss and the
decision of the team who won the toss
        for (String
name:teamWonLooseDecisionInOneMatch.getInnings().ke
ySet()) {
            String key = name.toString();
            String value =
teamWonLooseDecisionInOneMatch.getInnings().get(nam
e).toString();
            System.out.println(key+": "+ value);
        }

        //Match summery in Team1 side
```

```
System.out.println();
System.out.println("Team
"+teamWonLooseDecisionInOneMatch.team1+": ");
System.out.println();
//Printing each bowler's wickets with the
name in team 1
bowlerWickets(1);

//Printing total wickets taken by bowling
team when bowling:
displayTotalWickets(1);

System.out.println("*****");
//Printing each batsman's score with the
name and balls took for scoring and method of
dismissal if out
displayBatsmanDetail(1);

//Printing total runs taken by batting
side:
totalRunsOfTheBattingSide(1);

System.out.println("*****");
//Match summary in Team2 Side:
System.out.println();
System.out.println("Team
"+teamWonLooseDecisionInOneMatch.team2+": ");
//Printing each bowler's wickets with the
name in team 2
bowlerWickets(2);

//Printing total wickets taken by bowling
team when bowling:
displayTotalWickets(2);
```

```
System.out.println("*****");  
  
        //Printing each batsman's score with the  
name and balls took for scoring and method of  
dismissal if out  
        System.out.println();  
        displayBatsmanDetail(2);  
  
        //Printing total runs taken by batting  
side:  
        totalRunsOfTheBattingSide(2);  
  
System.out.println("*****");  
  
        //Printing which team won the match:  
        displayTeamWhoWonTheMatch();  
    }  
    private void bowlerWickets(int teamNumber){  
        ArrayList Array = new ArrayList();  
        Array =  
getTeamBowlingOrBatting(teamNumber,"ball");  
        for (int i=0;i<Array.size();i++){  
            Hashtable playerProfile = (Hashtable)  
Array.get(i);  
            for (Object  
name:playerProfile.keySet()) {  
                String key = name.toString();  
                String value =  
playerProfile.get(name).toString();  
                if (key.equals("Name") ||  
(key.equals("Wickets taken in a match")) ||  
(key.equals("Overs Bowled"))){  
                    System.out.println(key + ":" +  
value);  
                }  
            }  
        System.out.println();  
    }  
}
```

```
    }

    private void displayTotalWickets(int teamNumber) {
        String countryName=null;
        if (teamNumber==1) {
            countryName =
teamWonLooseDecisionInOneMatch.team1;
        }else if (teamNumber == 2) {
            countryName =
teamWonLooseDecisionInOneMatch.team2;
        }
        int totWickets =
calculateTotalWicketsTakenByOneSide(teamNumber);
        System.out.println("Total wickets taken by
"+countryName+" team is "+totWickets);
    }

    private int
calculateTotalWicketsTakenByOneSide(int teamNumber) {
        int totWickets = 0;
        ArrayList teamBowling = new ArrayList();
        teamBowling =
getTeamBowlingOrBatting(teamNumber,"ball");
        for (int i=0;i<teamBowling.size();i++) {
            Dictionary playerProfile = (Dictionary)
teamBowling.get(i);
            totWickets+=(Integer)
playerProfile.get("Wickets taken in a match");
        }
        return totWickets;
    }

    private void displayBatsmanDetail(int teamNumber) {
        ArrayList teamBatting = new ArrayList();
        teamBatting =
getTeamBowlingOrBatting(teamNumber,"bat");
        for (int i=0;i<teamBatting.size();i++) {
```

```

        Hashtable playerProfile = (Hashtable)
teamBatting.get(i);
        for (Object
name:playerProfile.keySet())){
            String key = name.toString();
            String value =
playerProfile.get(name).toString();
            if (key.equals("Name") ||
key.equals("Scores taken in a match") ||
key.equals("Balls took for scoring") ||
key.equals("Method of dismissal")){
                System.out.println(key +":
"+value);
            }
        }
        System.out.println();
    }

}
private void totalRunsOfTheBattingSide(int
teamNumber){
    if (teamNumber==1){
        System.out.println("Total marks of
"+teamWonLooseDecisionInOneMatch.team1+ " team is
"+teamWonLooseDecisionInOneMatch.totalMarksScoredBy
Team1);
    }else if (teamNumber==2){
        System.out.println("Total marks of
"+teamWonLooseDecisionInOneMatch.team2+ " team is
"+teamWonLooseDecisionInOneMatch.totalMarksScoredBy
Team2);
    }
}
private void displayTeamWhoWonTheMatch() {
    int totalInTeam1 =
teamWonLooseDecisionInOneMatch.totalMarksScoredByTe
am1;
    String team1 =

```

```
teamWonLooseDecisionInOneMatch.team1;
        int totalInTeam2 =
teamWonLooseDecisionInOneMatch.totalMarksScoredByTe
am2;
        String team2 =
teamWonLooseDecisionInOneMatch.team2;
        if (totalInTeam1>totalInTeam2) {
            System.out.println(team1+" won the
match");
        }else
            System.out.println(team2+" won the
match");
    }

    private ArrayList getTeamBowlingOrBatting(int
teamNumber, String option){
        if (teamNumber==1 &&
option.equals("ball")){
            return
teamWonLooseDecisionInOneMatch.getTeam1Bowling();
        }else if (teamNumber==2 &&
option.equals("ball")){
            return
teamWonLooseDecisionInOneMatch.getTeam2Bowling();
        }else if (teamNumber==1 &&
option.equals("bat")){
            return
teamWonLooseDecisionInOneMatch.getTeam1Batting();
        }else
            return
teamWonLooseDecisionInOneMatch.getTeam2Batting();
    }
}
```

playerProfileDictionaryStructure class:

```
package sample;
import java.util.Hashtable;

public class playerProfileDictionaryStructure {
    //Used protected specifier therefore,only sub
    classes can access this method
    protected void
addCommonValuesToTheDictionary(Hashtable
player, String name, String role, String country) {
    player.put("Name", name);
    player.put("Role", role);
    player.put("country", country);
    player.put("Matches", 0);
    player.put("Innings", 0);
    player.put("Runs", 0);
    player.put("50s", 0);
    player.put("Best", 0);
    player.put("Wickets", 0);
    player.put("Scores taken in a match", 0);
    player.put("Wickets taken in a match", 0);
    player.put("Balls took for scoring", 0);
    player.put("Method of dismissal", "Not
played");
    player.put("Overs Bowled", 0);
}
}
```

positionOfTeams class:

```
package sample;

import java.util.Arrays;
import java.util.Collections;
import java.util.Dictionary;
```

```
public class positionOfTeams {
    listOfAllTheTeams listOfAllTheTeams;

    public positionOfTeams(sample.listOfAllTheTeams
listOfAllTheTeams) {
        this.listOfAllTheTeams = listOfAllTheTeams;
    }

    private Double[] rating = new Double[8];

    public void printPosition() {
        calculatePosition();
        Integer[] positions ={1,2,3,4,5,6,7,8};
        for (int i = 0;i< positions.length;i++) {
            for (int j =
0;j<listOfAllTheTeams.getAllTheTeams().size();j++) {
                Dictionary teamProfile =
(Dictionary)
listOfAllTheTeams.getAllTheTeams().get(j);
                if
((Integer)teamProfile.get("Position")==positions[i])
                {
                    System.out.println("Name: "+
teamProfile.get("Name"));
                    System.out.println("Number of
matches: "+teamProfile.get("No of matches"));
                    System.out.println("Position:
"+teamProfile.get("Position"));
                    System.out.println("Rating:
"+teamProfile.get("Rating"));
                    System.out.println();
                    break;
                }
            }
        }
    }

    private void
```

```
addValuesToTheArrayAndSortInDescendingOrder () {
    for (int i=0;i<listOfAllTheTeams.getAllTheTeams().size();i++) {
        Dictionary teamProfile = (Dictionary)listOfAllTheTeams.getAllTheTeams().get(i);
        rating[i] = (Double)teamProfile.get("Rating");
    }
    //This will sort the array in descending order
    Arrays.sort(rating,
    Collections.reverseOrder());
}

private void calculatePosition() {

addValuesToTheArrayAndSortInDescendingOrder ();
    for (int i=0;i<8;i++) {
        for (int j=0;j<listOfAllTheTeams.getAllTheTeams().size();j++)
    {
        Dictionary teamProfile =
(Dictionary)listOfAllTheTeams.getAllTheTeams().get(j);
        if ((Double)teamProfile.get("Rating")==(Double) rating[i]){
            teamProfile.put("Position",i+1);
        }
    }
}
}
```

teamWonLooseDecisionInOneMatch class:

```
package sample;

import java.util.ArrayList;
import java.util.Dictionary;
import java.util.Hashtable;
import java.util.Random;

public class teamWonLooseDecisionInOneMatch {
    private listOfAllThePlayers
listOfAllThePlayers;
    private listOfAllTheTeams listOfAllTheTeams;
    private WeightedRandom
possibleThingsThatCanHappenToABall = new
WeightedRandom();
    private ArrayList Team1Batting = new
ArrayList();
    private ArrayList Team1Bowling = new
ArrayList();
    private ArrayList Team2Batting = new
ArrayList();
    private ArrayList Team2Bowling = new
ArrayList();
    Random random = new Random();

    private Hashtable <String, String> innings = new
Hashtable<String, String>();

    private final String[] toss = {"Head", "Tail"};
    private final String[] Play = {"Bat", "Ball"};

    //Assigning initial values for the specific
variables
    int ballsTookForScoring = 0;
    int scoresTakenInAMatchByOnePlayer = 0;
    int player = 0;
```

```
    int totalBallsBowled = 0; //This is used to
find if a player stays till the last over
    int totalMarksScoredByTeam1 = 0;
    int totalMarksScoredByTeam2 = 0;
    String team1; //These variables are used by
displayMatchSummeryAndStoringToFile class
    String team2;

    public
teamWonLooseDecisionInOneMatch(sample.listOfAllTheP
layers listOfAllThePlayers, sample.listOfAllTheTeams
listOfAllTheTeams) {
        this.listOfAllThePlayers =
listOfAllThePlayers;
        this.listOfAllTheTeams = listOfAllTheTeams;
    }

    public void wonLooseDecision(String team1,
String team2) {
        this.team1 = team1;
        this.team2 = team2;

assigningBowlersAndBatsman(team1, Team1Batting, Team1
Bowling);

assigningBowlersAndBatsman(team2, Team2Batting, Team2
Bowling);

findTeamWhoWonTheTossAndTheirDecision(team1, team2);
    resetInitialValues();
    for (int i=0;i<20;i++) {
        //1st innings (Team1 bat and Team2
ball)

getSummeryOfAnOverAndDoChangesAccordingly(Team1Batt
ing, Team2Bowling);
    }
    //Resetting initial values after 1st
```

```

innings
    resetInitialValues();
    for (int i=0;i<20;i++) {
        //2nd innings (Team2 bat and Team1
ball)

getSummeryOfAnOverAndDoChangesAccordingly(Team2Batt
ing,Team1Bowling);
}
//Updating relevant features from player
profile after the end of the match

updatingRelevantFeaturesFromEachAndEveryPlayerProfi
le();

updatingRelevantFeaturesFromTeamProfile(listOfAllTh
eTeams.getAllTheTeams(),team1,team2);

/*
    This method will reset some of the
temporary features of player profile to
    default values.If we want to display player
profiles,then we should call
    player profiles before calling the below
method
*/
//resetPlayerFeaturesAfterEachMatch(Team1Batting);
//resetPlayerFeaturesAfterEachMatch(Team2Batting);
}

public void
resetPlayerFeaturesAfterEachMatch(ArrayList team) {
    for (int i=0;i<team.size();i++){
        Dictionary playerProfile = (Dictionary)
team.get(i);
        playerProfile.put("Scores taken in a

```

```
match", 0);
        playerProfile.put("Wickets taken in a
match", 0);
        playerProfile.put("Balls took for
scoring", 0);
        playerProfile.put("Overs Bowled", 0);
        playerProfile.put("Method of
dismissal", "Not played");
    }
}

private void assigningBowlersAndBatsman (String
team, ArrayList teamBatting, ArrayList teamBowling) {
    //This method will assign all the players
in a country to Team1Batting or Team2Batting
    //and also will assign players for
team1Bowling or team2Bowling
    switch (team) {
        case "Afghanistan":
storePlayersOfASpecificCountryToAnArray ("Afghanistan"
", teamBatting, teamBowling);
            break;
        case "Australia":
storePlayersOfASpecificCountryToAnArray ("Australia"
, teamBatting, teamBowling);
            break;
        case "Bangladesh":
storePlayersOfASpecificCountryToAnArray ("Bangladesh"
, teamBatting, teamBowling);
            break;
        case "England":
storePlayersOfASpecificCountryToAnArray ("England", t
eamBatting, teamBowling);
            break;
```

```
        case "India":  
  
    storePlayersOfASpecificCountryToAnArray("India", tea  
mBatting, teamBowling);  
        break;  
    case "New Zealand":  
  
    storePlayersOfASpecificCountryToAnArray("New  
Zealand", teamBatting, teamBowling);  
        break;  
    case "Pakistan":  
  
    storePlayersOfASpecificCountryToAnArray("Pakistan",  
teamBatting, teamBowling);  
        break;  
    case "Sri Lanka":  
  
    storePlayersOfASpecificCountryToAnArray("Sri  
Lanka", teamBatting, teamBowling);  
        break;  
    }  
}  
  
    private void  
findTeamWhoWonTheTossAndTheirDecision(String  
team1, String team2){  
    String[] teams = {team1, team2};  
    int randomIndex =  
random.nextInt(teams.length);  
    String TeamWhoCallsTheToss =  
teams[randomIndex]; //Always the team who calls the  
toss select their choice as head  
  
    int randomIndex1 =  
random.nextInt(toss.length); //To randomize the  
result of the coin flip  
    String choice = toss[randomIndex1];  
    if (choice.equals("Head")) {
```

```

        if
(TeamWhoCallsTheToss.equals(teams[0])) {
            innings.put("Team who won the
toss", TeamWhoCallsTheToss);
        }else
            innings.put("Team who won the
toss", teams[1]);
    }else {
        if
(TeamWhoCallsTheToss.equals(teams[0])) {
            innings.put("Team who won the
toss", teams[1]);
        }else
            innings.put("Team who won the
toss", teams[0]);
    }
    int randomIndexForPlay =
random.nextInt(Play.length);
    innings.put("Decision of the team who won
the toss(Bat/Ball)", Play[randomIndexForPlay]);
}

private void
getSummeryOfAnOverAndDoChangesAccordingly(ArrayList
teamBatting, ArrayList teamBowling) {
    //This will randomly pick a bowler from
opposite team
    Dictionary randomBowlerFromTeam2
=(Dictionary)
teamBowling.get(random.nextInt(teamBowling.size()))
;

    ArrayList
ArrayOfPossibleThingsThatCanHappenToABallInAnOver =
new ArrayList();

ArrayOfPossibleThingsThatCanHappenToABallInAnOver =

```

```
getAnArrayOfPossibleThingsThatCanHappenToABallInAnOver();
    //This will increment the number of overs bowled by a bowler

        randomBowlerFromTeam2.put("Overs Bowled",
        (Integer)randomBowlerFromTeam2.get("Overs Bowled") + 1);
        for (int j = 0; j < ArrayOfPossibleThingsThatCanHappenToABallInAnOver.size(); j++) {
            if (this.player != 10) {
                Dictionary playerProfile =
                (Dictionary) teamBatting.get(player);

                doingChangesIfThePlayerScoredInABall((String)
                ArrayOfPossibleThingsThatCanHappenToABallInAnOver.get(j),
                playerProfile);
            }

            doingChangesIfThePlayerGotOutInABall((String)
            ArrayOfPossibleThingsThatCanHappenToABallInAnOver.get(j),
            playerProfile, randomBowlerFromTeam2);

            doingChangesIfAPlayerStaysTillLastOver(playerProfile);
        }
        else if (this.player == 10) {
            break;
        }
    }
}

private void resetInitialValues() {
    this.ballsTookForScoring = 0;
    this.scoresTakenInAMatchByOnePlayer = 0;
    this.player = 0;
    this.totalBallsBowled = 0;
    this.totalMarksScoredByTeam1 = 0;
```

```
        this.totalMarksScoredByTeam2 = 0;
    }

    private void
updatingRelevantFeaturesFromEachAndEveryPlayerProfile() {
    for (int i=0;i<Team1Batting.size();i++) {
        Dictionary playerProfile = (Dictionary)
Team1Batting.get(i);

playerProfile.put("Innings", (Integer)playerProfile.
get("Innings")+2); //Increment by 2 because there
are 2 innings in a match

playerProfile.put("Matches", (Integer)playerProfile.
get("Matches")+1);
        totalMarksScoredByTeam1 +=(Integer)
playerProfile.get("Scores taken in a match");

    }
    for (int i=0;i<Team2Batting.size();i++) {
        Dictionary playerProfile = (Dictionary)
Team2Batting.get(i);

playerProfile.put("Innings", (Integer)playerProfile.
get("Innings")+2); //Increment by 2 because there
are 2 innings in a match

playerProfile.put("Matches", (Integer)playerProfile.
get("Matches")+1);
        totalMarksScoredByTeam2 +=(Integer)
playerProfile.get("Scores taken in a match");

    }
}

private void
updatingRelevantFeaturesFromTeamProfile (ArrayList
allTheTeams, String team1, String team2) {
```

```

        for (int i=0;i<allTheTeams.size() ;i++) {
            Dictionary teamProfile = (Dictionary)
allTheTeams.get(i);
            if
(teamProfile.get("Name") .equals(team1) ||
teamProfile.get("Name") .equals(team2)) {
                teamProfile.put("No of
matches", (Integer)teamProfile.get("No of
matches")+1);
            }
            if
(teamProfile.get("Name") .equals(team1)) {
                if
(totalMarksScoredByTeam1>totalMarksScoredByTeam2) {

teamProfile.put("Rating", (Double)teamProfile.get("R
ating")+(totalMarksScoredByTeam1/10));
            }else {
                teamProfile.put("Rating",
(Double) teamProfile.get("Rating") +
(totalMarksScoredByTeam1 / 20));
            }
            }else if
(teamProfile.get("Name") .equals(team2)) {
                if
(totalMarksScoredByTeam2>totalMarksScoredByTeam1) {

teamProfile.put("Rating", (Double)teamProfile.get("R
ating")+(totalMarksScoredByTeam2/10));
            }else {
                teamProfile.put("Rating",
(Double) teamProfile.get("Rating") +
(totalMarksScoredByTeam2 / 20));
            }
        }
    }
}

```

```
    private void
doingChangesIfThePlayerScoreInABall (String
score, Dictionary playerProfile) {
    //If player scored 0,1,2,3,4,5 or 6, the
relevant fields will be updated.
    for (int i=0;i<=6;i++) {
        try {
            if (i==Integer.parseInt(score)) {
                this.ballsTookForScoring+=1;
                this.totalBallsBowled+=1;

this.scoresTakenInAMatchByOnePlayer+=i;
                playerProfile.put("Balls took
for scoring",ballsTookForScoring);
                playerProfile.put("Scores taken
in a match", scoresTakenInAMatchByOnePlayer);
                break;
            }
        } catch (NumberFormatException e) {
            break;
        }
    }
}

private void
doingChangesIfThePlayerGotOutInABall (String
option, Dictionary playerProfile, Dictionary
randomBowlerFromTeam2) {
    if (option.equals("caught") ||

option.equals("bowled") || option.equals("leg by
wicket") || option.equals("run out") ||

option.equals("stumped")){
        playerProfile.put("Method of
dismissal",option);
        if ((Integer)playerProfile.get("Scores
taken in a match")>=50){
            int previousValue = (Integer)
playerProfile.get("50s");
```

```

playerProfile.put("50s", previousValue+1);
    }
    if
((Integer)playerProfile.get("Best")<(Integer)
playerProfile.get("Scores taken in a match")){
playerProfile.put("Best", (Integer)playerProfile.get
("Scores taken in a match"));
}
randomBowlerFromTeam2.put("Wickets
taken in a
match", (Integer)randomBowlerFromTeam2.get("Wickets
taken in a match")+1); //This variable resets to 0
after each match

randomBowlerFromTeam2.put("Wickets", (Integer)random
BowlerFromTeam2.get("Wickets")+1); //Total wickets
gained by bowlers in each match, used to get best 5
bowlers
        int totalRunsInPreviousMatch =
(Integer) playerProfile.get("Runs");

playerProfile.put("Runs", totalRunsInPreviousMatch+
(Integer) playerProfile.get("Scores taken in a
match"));
        this.ballsTookForScoring += 1;
        this.totalBallsBowled+=1;
        playerProfile.put("Balls took for
scoring", ballsTookForScoring);
        this.player+=1;
        this.ballsTookForScoring = 0;
        this.scoresTakenInAMatchByOnePlayer =
0;

    }
}

private void

```

```
doingChangesIfAPlayerStaysTillLastOver(Dictionary
playerProfile) {
    if (this.totalBallsBowled==120 &&
this.player!=10) {

playerProfile.put("Runs", (Integer)playerProfile.get
("Runs")+(Integer) playerProfile.get("Scores taken
in a match"));
    playerProfile.put("Method of
dismissal","Played till the last over.");
    playerProfile.put("Balls took for
scoring",ballsTookForScoring);
    if
((Integer)playerProfile.get("Best")<(Integer)
playerProfile.get("Scores taken in a match")){
playerProfile.put("Best", (Integer)playerProfile.get
("Scores taken in a match"));
    }
}
}

private ArrayList
getAnArrayOfPossibleThingsThatCanHappenToABallInAnO
ver() {
    ArrayList
ArrayOfPossibleThingsThatCanHappenToABallInAnOver =
new ArrayList();

possibleThingsThatCanHappenToABall.addEntry("0",
28);

possibleThingsThatCanHappenToABall.addEntry("1",
23);

possibleThingsThatCanHappenToABall.addEntry("2",
22);
```

```
possibleThingsThatCanHappenToABall.addEntry("3",
5);

possibleThingsThatCanHappenToABall.addEntry("4",
5);

possibleThingsThatCanHappenToABall.addEntry("5",
5);

possibleThingsThatCanHappenToABall.addEntry("6",
5);

possibleThingsThatCanHappenToABall.addEntry("caught",
2);

possibleThingsThatCanHappenToABall.addEntry("bowled",
2);

possibleThingsThatCanHappenToABall.addEntry("leg by
wicket", 1);

possibleThingsThatCanHappenToABall.addEntry("run
out", 1);

possibleThingsThatCanHappenToABall.addEntry("stumpe
d", 1);

    for (int i = 0; i < 6; i++) {

ArrayOfPossibleThingsThatCanHappenToABallInAnOver.a
dd(possibleThingsThatCanHappenToABall.getRandom());
}

return
ArrayOfPossibleThingsThatCanHappenToABallInAnOver;
}

private void
storePlayersOfASpecificCountryToArray (String
```

```
countryName,ArrayList teamBatting,ArrayList
teamBowling) {
    //This method will filter players related
    to one country and store that dictionaries
    //to one list and again filter bowler and
    all rounders and store that in another array
    for (int
i=0;i<listOfAllThePlayers.getAllThePlayers().size()
;i++) {
        Dictionary playerProfile = (Dictionary)
listOfAllThePlayers.getAllThePlayers().get(i);
        Object country =
playerProfile.get("country");
        Object role =
playerProfile.get("Role");
        if (country.equals(countryName)) {
            teamBatting.add(playerProfile);
        }
        if (country.equals(countryName) &&
(role.equals("Bowler") || role.equals("All
Rounder"))){
            teamBowling.add(playerProfile);
        }
    }
}

public Hashtable<String, String> getInnings() {
    return innings;
}

public ArrayList getTeam1Batting() {
    return Team1Batting;
}

public ArrayList getTeam1Bowling() {
    return Team1Bowling;
}
```

```
public ArrayList getTeam2Batting() {
    return Team2Batting;
}

public ArrayList getTeam2Bowling() {
    return Team2Bowling;
}
```

tournamentMatches class:

```
package sample;

import java.util.*;

public class tournamentMatches {
    private Scanner scanner = new
Scanner(System.in);
    private teamWonLooseDecisionInOneMatch
teamWonLooseDecisionInOneMatch;
    private overviewOfAMatchSummery
overviewOfAMatchSummery;
    private positionOfTeams positionOfTeams;
    private best5B bowlersAndBatsman
best5B bowlersAndBatsman;
    private afghanistanPlayerProfile
afghanistanPlayerProfile;
    private australiaPlayerProfile
australiaPlayerProfile;
    private bangladeshPlayerProfile
bangladeshPlayerProfile;
    private englandPlayerProfile
englandPlayerProfile;
    private indiaPlayerProfile indiaPlayerProfile;
    private newZealandPlayerProfile
newZealandPlayerProfile;
    private pakistanPlayerProfile
pakistanPlayerProfile;
```

```
    private sriLankaPlayerProfile
sriLankaPlayerProfile;

    public
tournamentMatches(sample.teamWonLooseDecisionInOneMatch teamWonLooseDecisionInOneMatch,
overviewOfAMatchSummery overviewOfAMatchSummery,
positionOfTeams positionOfTeams,
sample.best5BowlersAndBatsman
best5BowlersAndBatsman,
sample.afghanistanPlayerProfile
afghanistanPlayerProfile,
sample.australiaPlayerProfile
australiaPlayerProfile,
sample.bangladeshPlayerProfile
bangladeshPlayerProfile,
sample.englandPlayerProfile englandPlayerProfile,
sample.indiaPlayerProfile indiaPlayerProfile,
sample.newZealandPlayerProfile
newZealandPlayerProfile,
sample.pakistanPlayerProfile pakistanPlayerProfile,
sample.sriLankaPlayerProfile sriLankaPlayerProfile)
{
        this.teamWonLooseDecisionInOneMatch =
teamWonLooseDecisionInOneMatch;
        this.overviewOfAMatchSummery =
overviewOfAMatchSummery;
        this.positionOfTeams = positionOfTeams;
        this.best5BowlersAndBatsman =
best5BowlersAndBatsman;
        this.afghanistanPlayerProfile =
afghanistanPlayerProfile;
        this.australiaPlayerProfile =
australiaPlayerProfile;
        this.bangladeshPlayerProfile =
bangladeshPlayerProfile;
        this.englandPlayerProfile =
```

```

englandPlayerProfile;
        this.indiaPlayerProfile =
indiaPlayerProfile;
        this.newZealandPlayerProfile =
newZealandPlayerProfile;
        this.pakistanPlayerProfile =
pakistanPlayerProfile;
        this.sriLankaPlayerProfile =
sriLankaPlayerProfile;
    }

String firstTeamWhoQualifiedForFinals;
String secondTeamWhoQualifiedForFinals;
public void groupMatches(ArrayList group,int
count) {
    for (int i=0;i<3;i++) {
        for (int j=i+1;j<4;j++) {
            matchBetweenTwoTeams( (Dictionary)
group.get(i) , (Dictionary) group.get(j));
        }
    }
    if (count==1) {
        firstTeamWhoQualifiedForFinals =
displayTeamWhoQualifiedForFinals(group);
    }else if (count==2) {
        secondTeamWhoQualifiedForFinals =
displayTeamWhoQualifiedForFinals(group);
    }
}
public void matchBetweenTwoTeams(Dictionary
firstTeamProfile,Dictionary secondTeamProfile){
    Dictionary teamProfile1 = firstTeamProfile;
    Dictionary teamProfile2 =
secondTeamProfile;

    //Resetting arraylists after each match
teamWonLooseDecisionInOneMatch.getTeam1Batting().re

```

```
moveAll(teamWonLooseDecisionInOneMatch.getTeam1Batting());  
  
teamWonLooseDecisionInOneMatch.getTeam2Batting().removeAll(teamWonLooseDecisionInOneMatch.getTeam2Batting());  
  
teamWonLooseDecisionInOneMatch.getTeam1Bowling().removeAll(teamWonLooseDecisionInOneMatch.getTeam1Bowling());  
  
teamWonLooseDecisionInOneMatch.getTeam2Bowling().removeAll(teamWonLooseDecisionInOneMatch.getTeam2Bowling());  
  
teamWonLooseDecisionInOneMatch.wonLooseDecision(teamProfile1.get("Name").toString(), teamProfile2.get("Name").toString());  
    System.out.println("The match between  
"+teamProfile1.get("Name").toString()+" and  
"+teamProfile2.get("Name").toString());  
//  
getUserInputAndDisplayMatchSummery(option);  
    //Resetting some values after the match  
from player profiles  
//  
resetPlayerFeaturesInOneTeam(teamProfile1.get("Name").toString());  
//  
resetPlayerFeaturesInOneTeam(teamProfile2.get("Name").toString());  
  
//  
displayBest5B bowlersOrBatsmen(askUserToDisplayBest5Batsmen, askUserToDisplayBest5Bowlers);  
//  
displayTournamentStanding(askUserToDisplayTournamen
```

```
tStanding);  
  
    }  
    public void  
getUserInputAndDisplayMatchSummery(String  
askUserToDisplayMatchSummery){  
    String askForMatchSummery =  
askUserToDisplayMatchSummery;  
    if  
(askForMatchSummery.toLowerCase() .equals("yes")) {  
  
overviewOfAMatchSummery.displayMatchSummery();  
    }  
}  
    public void resetPlayerFeaturesInOneTeam(String  
team) {  
    switch (team) {  
        case "Afghanistan":  
  
teamWonLooseDecisionInOneMatch.resetPlayerFeaturesA  
fterEachMatch(afghanistanPlayerProfile.getAfghanistan  
Team());  
        break;  
        case "Australia":  
  
teamWonLooseDecisionInOneMatch.resetPlayerFeaturesA  
fterEachMatch(australiaPlayerProfile.getAustraliaTe  
am());  
        break;  
        case "Bangladesh":  
  
teamWonLooseDecisionInOneMatch.resetPlayerFeaturesA  
fterEachMatch(bangladeshPlayerProfile.getBangladesh  
Team());  
        break;  
        case "England":  
  
teamWonLooseDecisionInOneMatch.resetPlayerFeaturesA
```

```
fterEachMatch(englandPlayerProfile.getEnglandTeam()
) ;
        break;
    case "India":  
  
teamWonLooseDecisionInOneMatch.resetPlayerFeaturesA
fterEachMatch(indiaPlayerProfile.getIndiaTeam());
        break;
    case "New Zealand":  
  
teamWonLooseDecisionInOneMatch.resetPlayerFeaturesA
fterEachMatch(newZealandPlayerProfile.getNewZealand
Team());
        break;
    case "Pakistan":  
  
teamWonLooseDecisionInOneMatch.resetPlayerFeaturesA
fterEachMatch(pakistanPlayerProfile.getPakistanTeam
());
        break;
    case "Sri Lanka":  
  
teamWonLooseDecisionInOneMatch.resetPlayerFeaturesA
fterEachMatch(sriLankaPlayerProfile.getSriLankaTeam
());
        break;
    }
}
}  

public void displayBest5BowlersOrBatsmen(String
askUserToDisplayBest5Batsmen, String
askUserToDisplayBest5Bowlers){
    String choiceToDisplayBest5Bowlers =
askUserToDisplayBest5Bowlers;
    String choiceToDisplayBest5Batsmen =
askUserToDisplayBest5Batsmen;
    if
(choiceToDisplayBest5Bowlers.toLowerCase().equals("y
es")) {
```

```
best5BowlersAndBatsman.displayBest5Bowlers();  
  
    }  
    if  
(choiceToDisplayBest5Batsmen.toLowerCase().equals("yes")) {  
  
best5BowlersAndBatsman.displayBest5Batsmen();  
    }  
}  
  
    public void displayTournamentStanding(String  
askUserToDisplayTournamentStanding) {  
        String option =  
askUserToDisplayTournamentStanding;  
        if (option.toLowerCase().equals("yes")) {  
            positionOfTeams.printPosition();  
        }  
    }  
  
    public String  
displayTeamWhoQualifiedForFinals(ArrayList group) {  
        int[] position = new int[group.size()];  
        for (int i=0;i<group.size();i++) {  
            Dictionary teamProfile = (Dictionary)  
group.get(i);  
            position[i] = (Integer)  
teamProfile.get("Position");  
        }  
        Arrays.sort(position); //Sorting the array  
in ascending order  
        for (int i=0;i<group.size();i++) {  
            Dictionary teamProfile = (Dictionary)  
group.get(i);  
            if ((Integer)  
teamProfile.get("Position") == position[0]) {
```

```

        System.out.println(teamProfile.get("Name") + " "
qualified for finals");
                return (String)
teamProfile.get("Name");
            }
        }return null;
    }

}

```

WeightedRandom class:

```

package sample;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class WeightedRandom<T extends Object> {

    private class Entry {
        double accumulatedWeight;
        T object;
    }

    private List<Entry> entries = new
ArrayList<>();
    private double accumulatedWeight;
    private Random rand = new Random();

    public void addEntry(T object, double weight) {
        accumulatedWeight += weight;
        Entry e = new Entry();
        e.object = object;
        e.accumulatedWeight = accumulatedWeight;
    }
}

```

```
        entries.add(e);
    }

    public T getRandom() {
        double r = rand.nextDouble() *
accumulatedWeight;

        for (Entry entry: entries) {
            if (entry.accumulatedWeight >= r) {
                return entry.object;
            }
        }
        return null; //should only happen when
there are no entries
    }
}
```

FXML FILES:

1.guiAskingForEditAndDeleteTeamProfile.fxml:

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.text.Font?>

<GridPane alignment="center" hgap="10" vgap="10"
xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <columnConstraints>
        <ColumnConstraints />
        <ColumnConstraints />
    </columnConstraints>
    <rowConstraints>
        <RowConstraints />
        <RowConstraints />
    </rowConstraints>
    <children>
        <AnchorPane prefHeight="746.0"
prefWidth="1300.0">
            <children>
                <ImageView fitHeight="780.0"
fitWidth="1344.0" layoutX="-19.0">
                    <image>
                        <Image url="@Images/ajay-
```

```
parthasarathy-I27kjbkwLAo-unsplash.jpg" />
        </image>
    </ImageView>
    <AnchorPane layoutX="378.0"
layoutY="142.0" prefHeight="420.0"
prefWidth="541.0" style="-fx-background-color:
white;">
        <children>
            <Label layoutX="32.0"
layoutY="25.0" text="EDIT DELETE TEAM PROFILE">
                <font>
                    <Font name="Tahoma"
size="36.0" />
                </font>
            </Label>
            <TextField
fx:id="askUserToEditTeamProfile" layoutX="14.0"
layoutY="144.0" prefHeight="31.0" prefWidth="216.0"
promptText="Yes or No" styleClass="border"
stylesheets="@../values/style.css" />
                <Button id="dark-blue"
fx:id="addButton" layoutX="192.0" layoutY="340.0"
mnemonicParsing="false" onAction="#switchToScene2"
prefHeight="57.0" prefWidth="157.0" style="-fx-
background-color: #166e9a;"
stylesheets="@../values/style.css" text="SUBMIT"
textFill="#fffffff4">
                    <font>
                        <Font name="Tahoma"
size="18.0" />
                    </font>
                </Button>
                <Label layoutX="14.0"
layoutY="205.0" text="If yes, What do you want to
do?">
                    <font>
                        <Font name="Tahoma"
size="18.0" />
                    </font>
                </Label>
            </children>
        </AnchorPane>
```

```

                </font>
            </Label>
            <TextField fx:id="getOption"
layoutX="14.0" layoutY="238.0" prefHeight="31.0"
prefWidth="216.0" promptText="Edit or Delete"
styleClass="border"
stylesheets="@../values/style.css" />
                <Pane layoutY="68.0"
prefHeight="93.0" prefWidth="541.0">
                    <children>
                        <Label layoutX="14.0"
layoutY="35.0" text="Do you want to edit/delete
team profile?">
                            <font>
                                <Font name="Tahoma"
size="18.0" />
                            </font>
                        </Label>
                    </children>
                </Pane>
            </children>
        </AnchorPane>
    </children>
</AnchorPane>
</children>
</GridPane>

```

2.guiAskingForEditDeletePlayerProfile.fxml:

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>

```

```
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <children>
        <GridPane alignment="center" hgap="10"
vgap="10">
            <columnConstraints>
                <ColumnConstraints />
                <ColumnConstraints />
            </columnConstraints>
            <rowConstraints>
                <RowConstraints />
                <RowConstraints />
            </rowConstraints>
            <children>
                <AnchorPane prefHeight="746.0"
prefWidth="1300.0">
                    <children>
                        <ImageView fitHeight="780.0"
fitWidth="1344.0" layoutX="-19.0">
                            <image>
                                <Image
url="@Images/cfphotosin-photography-HJbv998de9s-
unsplash.jpg" />
                            </image>
                        </ImageView>
                        <AnchorPane layoutX="392.0"
layoutY="149.0" prefHeight="417.0"
prefWidth="541.0" style="-fx-background-color:
white;">
                            <children>
```

```
        <Label layoutX="22.0"
layoutY="21.0" text="EDIT DELETE PLAYER PROFILE">
            <font>
                <Font name="Tahoma"
size="36.0" />
            </font>
        </Label>
        <TextField
fx:id="askUserToEditPlayerProfile" layoutX="26.0"
layoutY="131.0" prefHeight="46.0" prefWidth="257.0"
promptText="Yes or No" styleClass="border"
stylesheets="@../values/style.css" />
            <Button id="dark-blue"
fx:id="addButton" layoutX="193.0" layoutY="324.0"
mnemonicParsing="false" onAction="#switchToScene3"
prefHeight="57.0" prefWidth="157.0" style="-fx-
background-color: #166e9a;"'
stylesheets="@../values/style.css" text="SUBMIT"
textFill="#ffffffffd4">
                <font>
                    <Font name="Tahoma"
size="18.0" />
                </font>
            </Button>
            <Label layoutX="26.0"
layoutY="205.0" prefHeight="31.0" prefWidth="264.0"
text="If yes, What do you want to do?">
                <font>
                    <Font name="Tahoma"
size="18.0" />
                </font>
            </Label>
            <Label layoutX="27.0"
layoutY="97.0" text="Do you want to edit/delete
player profile?">
                <font>
                    <Font name="Tahoma"
size="18.0" />
                </font>
            </Label>
```

```

                </font>
            </Label>
            <TextField
fx:id="getOption" layoutX="27.0" layoutY="250.0"
prefHeight="46.0" prefWidth="264.0"
promptText="Edit or Delete" styleClass="border"
stylesheets="@../values/style.css" />
                </children>
            </AnchorPane>
            <children>
        </AnchorPane>
        <children>
    </AnchorPane>
    <children>
</GridPane>
</children>
</AnchorPane>

```

3.guiForDeletePlayerProfile.fxml:

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <children>
        <AnchorPane prefHeight="400.0"
prefWidth="600.0">

```

```
<children>
    <AnchorPane prefHeight="400.0"
prefWidth="600.0">
        <children>
            <GridPane alignment="center"
hgap="10" vgap="10">
                <columnConstraints>
                    <ColumnConstraints />
                    <ColumnConstraints />
                </columnConstraints>
                <rowConstraints>
                    <RowConstraints />
                    <RowConstraints />
                </rowConstraints>
                <children>
                    <AnchorPane
prefHeight="746.0" prefWidth="1300.0">
                        <children>
                            <ImageView
fitHeight="780.0" fitWidth="1344.0" layoutX="-
19.0">
                                <image>
                                    <Image
url="@Images/cfphotosin-photography-HJbv998de9s-
unsplash.jpg" />
                                </image>
                            </ImageView>
                            <AnchorPane
layoutX="392.0" layoutY="150.0" prefHeight="456.0"
prefWidth="541.0" style="-fx-background-color:
white;">
                                <children>
                                    <Label
layoutX="29.0" layoutY="20.0" text="DELETE PLAYER
PROFILE">
                                        <font>
                                            <Font
name="Bodoni MT" size="36.0" />
```

```
        </font>
    </Label>
    <TextField
fx:id="playerName" layoutX="29.0" layoutY="137.0"
prefHeight="65.0" prefWidth="363.0" promptText="Ex
: Rashid Khan" styleClass="border"
stylesheets="@../values/style.css" />
    <Button
fx:id="addButton" layoutX="192.0" layoutY="370.0"
mnemonicParsing="false"
onAction="#deletePlayerProfile" prefHeight="57.0"
prefWidth="157.0" style="-fx-background-color:
#166e9a;" text="SUBMIT" textFill="#fffffd4">
        <font>
            <Font
name="Tahoma" size="18.0" />
        </font>
    </Button>
    <Label
layoutX="29.0" layoutY="93.0" prefHeight="44.0"
prefWidth="250.0" text="Enter the player name: ">
        <font>
            <Font
name="Tahoma" size="17.0" />
        </font>
    </Label>
    <Label
layoutX="29.0" layoutY="228.0" prefHeight="44.0"
prefWidth="441.0" text="If you are done with
deleting player profiles, type yes">
        <font>
            <Font
name="Tahoma" size="17.0" />
        </font>
    </Label>
    <TextField
fx:id="doneWithDeletingPlayer" layoutX="29.0"
layoutY="285.0" prefHeight="65.0" prefWidth="363.0"
```

```

promptText="Yes" styleClass="border"
styleSheets="@../values/style.css" />
                                </children>
                            </AnchorPane>
                        </children>
                    </AnchorPane>
                </children>
            </GridPane>
        </children>
    </AnchorPane>
</children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</AnchorPane>

```

4.guiForDeleteTeamProfile.fxml:

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <children>
        <AnchorPane prefHeight="400.0"
prefWidth="600.0">
            <children>

```

```
<GridPane alignment="center" hgap="10"
vgap="10">
    <columnConstraints>
        <ColumnConstraints />
        <ColumnConstraints />
    </columnConstraints>
    <rowConstraints>
        <RowConstraints />
        <RowConstraints />
    </rowConstraints>
    <children>
        <AnchorPane prefHeight="746.0"
prefWidth="1300.0">
            <children>
                <ImageView
fitHeight="780.0" fitWidth="1344.0" layoutX="-
19.0">
                    <image>
                        <Image
url="@Images/ajay-parthasarathy-I27kjbkwIAo-
unsplash.jpg" />
                    </image>
                </ImageView>
                <AnchorPane layoutX="378.0"
layoutY="142.0" prefHeight="469.0"
prefWidth="570.0" style="-fx-background-color:
white;">
                    <children>
                        <Label layoutX="32.0"
layoutY="24.0" text="DELETE &quot;NUMBER OF
MATCHES&quot; FEATURE">
                            <font>
                                <Font
name="Baskerville Old Face" size="25.0" />
                            </font>
                        </Label>
                        <TextField
fx:id="teamName" layoutX="32.0" layoutY="184.0"
style="-fx-border-radius: 10px;
-fx-background-color: white;
-fx-text-fill: black;
-fx-prompt-text-color: #ccc;
-fx-prompt-text-size: 14px;
-fx-prompt-text-align: center;
-fx-prompt-text-height: 1.2;
-fx-prompt-text-width: 100%;"/>
                    </children>
                </AnchorPane>
            </children>
        </AnchorPane>
    </children>

```

```
prefHeight="46.0" prefWidth="314.0" promptText="Ex:  
Sri Lanka" styleClass="border"  
styleSheets="@../values/style.css" />  
        <Button id="dark-  
blue" fx:id="addButton" layoutX="192.0"  
layoutY="370.0" mnemonicParsing="false"  
onAction="#deleteTeamProfile" prefHeight="57.0"  
prefWidth="157.0" style="-fx-background-color:  
#166e9a;" styleSheets="@../values/style.css"  
text="SUBMIT" textFill="#fffffd4">  
            <font>  
                <Font  
name="Tahoma" size="18.0" />  
            </font>  
        </Button>  
        <Label layoutX="32.0"  
layoutY="142.0" text="Enter the team name: ">  
            <font>  
                <Font  
name="Tahoma" size="24.0" />  
            </font>  
        </Label>  
        <Label  
layoutX="140.0" layoutY="65.0" text="FROM TEAM  
PROFILE">  
            <font>  
                <Font  
name="Baskerville Old Face" size="25.0" />  
            </font>  
        </Label>  
        <Label layoutX="32.0"  
layoutY="260.0" text="If you are done with deleting  
team profiles, type yes">  
            <font>  
                <Font  
name="Tahoma" size="21.0" />  
            </font>  
        </Label>
```

```

        <TextField
fx:id="doneWithDeleting" layoutX="32.0"
layoutY="293.0" prefHeight="46.0" prefWidth="314.0"
promptText="Yes" styleClass="border"
styleSheets="@../values/style.css" />
    </children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</GridPane>
</children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</AnchorPane>

```

5.guiForEditPlayerProfile.fxml:

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <children>
        <AnchorPane prefHeight="400.0"
prefWidth="600.0">

```

```
<children>
    <GridPane alignment="center" hgap="10"
vgap="10">
        <columnConstraints>
            <ColumnConstraints />
            <ColumnConstraints />
        </columnConstraints>
        <rowConstraints>
            <RowConstraints />
            <RowConstraints />
        </rowConstraints>
        <children>
            <AnchorPane prefHeight="746.0"
prefWidth="1300.0">
                <children>
                    <ImageView
fitHeight="780.0" fitWidth="1344.0" layoutX="-
19.0">
                        <image>
                            <Image
url="@Images/cfphotosin-photography-HJbv998de9s-
unsplash.jpg" />
                        </image>
                    </ImageView>
                    <AnchorPane layoutX="392.0"
layoutY="152.0" prefHeight="521.0"
prefWidth="541.0" style="-fx-background-color:
white;">
                        <children>
                            <Label layoutX="58.0"
layoutY="14.0" text="EDIT PLAYER PROFILE">
                                <font>
                                    <Font
name="Bodoni MT" size="36.0" />
                                </font>
                            </Label>
                            <TextField
fx:id="playerName" layoutX="32.0" layoutY="122.0"
```

```
prefHeight="31.0" prefWidth="216.0" promptText="Ex
: Rashid Khan" styleClass="border"
styleSheets="@ ../values/style.css" />
<Button
fx:id="addButton" layoutX="192.0" layoutY="441.0"
mnemonicParsing="false"
onAction="#editPlayerProfile" prefHeight="57.0"
prefWidth="157.0" style="-fx-background-color:
#166e9a;" text="SUBMIT" textFill="#fffffd4">
<font>
<Font
name="Tahoma" size="18.0" />
</font>
</Button>
<Label layoutX="32.0"
layoutY="177.0" text="Enter the feature you want to
edit: ">
<font>
<Font
name="Tahoma" size="18.0" />
</font>
</Label>
<Label layoutX="32.0"
layoutY="99.0" text="Enter the player name: ">
<font>
<Font
name="Tahoma" size="18.0" />
</font>
</Label>
<TextField
fx:id="fieldNamePlayerProfile" layoutX="32.0"
layoutY="200.0" prefHeight="31.0" prefWidth="216.0"
promptText="Name/Role" styleClass="border"
styleSheets="@ ../values/style.css" />
<Label layoutX="32.0"
layoutY="248.0" text="Enter the value to update: ">
<font>
<Font
```

```
name="Tahoma" size="18.0" />
                </font>
            </Label>
            <TextField
fx:id="newValue" layoutX="32.0" layoutY="283.0"
prefHeight="31.0" prefWidth="216.0"
styleClass="border"
stylesheets="@../values/style.css" />
                <Label layoutX="33.0"
layoutY="327.0">
                    <font>
                        <Font
name="Tahoma" size="18.0" />
                            </font>
                    </Label>
                    <Label layoutX="33.0"
layoutY="339.0" text="If you are done with editing
player profiles, type yes">
                        <font>
                            <Font
name="Tahoma" size="18.0" />
                                </font>
                            </Label>
                            <TextField
fx:id="doneWithEditingPlayer" layoutX="33.0"
layoutY="378.0" prefHeight="31.0" prefWidth="216.0"
promptText="Yes" styleClass="border"
stylesheets="@../values/style.css" />
                                </children>
                            </AnchorPane>
                        </children>
                    </AnchorPane>
                </children>
            </GridPane>
        </children>
    </AnchorPane>
</children>
</AnchorPane>
```

6.guiForEditTeamProfile.fxml:

```
parthasarathy-I27kjbkwAo-unsplash.jpg" />
        </image>
    </ImageView>
<AnchorPane layoutX="378.0"
layoutY="142.0" prefHeight="518.0"
prefWidth="541.0" style="-fx-background-color:
white;">
    <children>
        <Label layoutX="82.0"
layoutY="14.0" text="EDIT TEAM PROFILE">
            <font>
                <Font name="Bodoni
MT" size="36.0" />
            </font>
        </Label>
        <TextField fx:id="teamName"
layoutX="32.0" layoutY="122.0" prefHeight="31.0"
prefWidth="216.0" promptText="Ex: Sri Lanka"
styleClass="border"
stylesheets="@../values/style.css" />
        <Button id="dark-blue"
fx:id="addButton" layoutX="192.0" layoutY="447.0"
mnemonicParsing="false" onAction="#editTeamProfile"
prefHeight="57.0" prefWidth="157.0" style="-fx-
background-color: #166e9a;"
stylesheets="@../values/style.css" text="SUBMIT"
textFill="#fffffff4">
            <font>
                <Font name="Tahoma"
size="18.0" />
            </font>
        </Button>
        <Label layoutX="32.0"
layoutY="177.0" text="Enter the field to update: ">
            <font>
                <Font name="Tahoma"
size="18.0" />
            </font>
```

```
        </Label>
        <Label layoutX="32.0"
layoutY="99.0" text="Enter the team name: ">
        <font>
            <Font name="Tahoma"
size="18.0" />
        </font>
    </Label>
    <TextField
fx:id="fieldNameTeamProfile" layoutX="32.0"
layoutY="200.0" prefHeight="31.0" prefWidth="216.0"
promptText="No of matches/Position"
styleClass="border"
stylesheets="@../values/style.css" />
        <Label layoutX="32.0"
layoutY="248.0" text="Enter the new value: ">
        <font>
            <Font name="Tahoma"
size="18.0" />
        </font>
    </Label>
    <TextField fx:id="newValue"
layoutX="32.0" layoutY="283.0" prefHeight="31.0"
prefWidth="216.0" styleClass="border"
stylesheets="@../values/style.css" />
        <Label layoutX="32.0"
layoutY="327.0" text="If you are done with editing
team profiles , type yes">
        <font>
            <Font name="Tahoma"
size="18.0" />
        </font>
    </Label>
    <TextField
fx:id="doneWithEditing" layoutX="32.0"
layoutY="363.0" prefHeight="31.0" prefWidth="216.0"
promptText="Yes" styleClass="border"
stylesheets="@../values/style.css" />
```

```

                </children>
            </AnchorPane>
            </children>
        </AnchorPane>
        <children>
    </GridPane>
</children>
</AnchorPane>
```

7.guiToDisplayMatchSummery fxml:

```

<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.String?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<HBox maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity"
prefHeight="601.0" prefWidth="1118.0"
xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <children>
        <VBox alignment="TOP_CENTER"
prefHeight="200.0" prefWidth="400.0"
styleClass="left-side-background"
stylesheets="@../values/style.css">
            <children>
                <ImageView fitHeight="283.0"
fitWidth="249.0" pickOnBounds="true"
```

```
preserveRatio="true">
    <image>
        <Image url="@Images/creative-
logo-cricket-vector-illustration-creative-abstract-
cricket-player-black-background-typopgraphy-vector-
109905399.jpg" />
    </image>
</ImageView>
<ImageView fitHeight="361.0"
fitWidth="400.0" pickOnBounds="true"
preserveRatio="true">
    <image>
        <Image url="@Images/pink-cricket-
ball-helmet-black-white-background-helmetworld-
worldcup-twenty-t-cricketworldcup-tournament-game-
play-winner-150156244.jpg" />
    </image>
</ImageView>
<Pane prefHeight="83.0"
prefWidth="400.0" VBox.vgrow="ALWAYS" />
</children>
</VBox>
<VBox prefHeight="601.0" prefWidth="434.0">
    <children>
        <Pane prefHeight="177.0"
prefWidth="400.0">
            <children>
                <Label fx:id="errorMessageLabel"
alignment="CENTER" layoutX="1.0" layoutY="57.0"
prefHeight="118.0" prefWidth="434.0"
textFill="#171717">
                    <font>
                        <Font name="Arial Bold"
size="18.0" />
                    </font>
                </Label>
                <Label layoutX="28.0"
layoutY="14.0" text="Tournament Matches">
```

```
        <font>
            <Font name="Berlin Sans FB
Demi Bold" size="36.0" />
        </font>
    </Label>
</children>
</Pane>
<Pane prefHeight="428.0"
prefWidth="434.0" styleClass="transparent"
stylesheets="@ ../values/style.css">
    <children>
        <Label layoutX="14.0"
layoutY="22.0" prefHeight="35.0" prefWidth="359.0"
text="Do you want to get the match summery?">
            <font>
                <Font name="Berlin Sans FB
Demi Bold" size="19.0" />
            </font>
        </Label>
        <TextField
fx:id="askToDisplayMatchSummery" layoutX="14.0"
layoutY="57.0" prefHeight="35.0" prefWidth="274.0"
promptText="Yes or No" styleClass="border"
stylesheets="@ ../values/style.css" />
            <Button fx:id="addButton"
layoutX="52.0" layoutY="362.0"
mnemonicParsing="false" prefHeight="50.0"
prefWidth="304.0"
stylesheets="@ ../values/style.css" text="SUBMIT"
textFill="#fcfcfc">
                <font>
                    <Font name="Arial"
size="24.0" />
                </font>
                <styleClass>
                    <String fx:value="left-
side-background" />
                    <String fx:value="button" />
                </styleClass>
            </Button>
        </children>
    </Pane>

```

```
/>
        </styleClass>
    </Button>
    <Label layoutX="14.0"
layoutY="105.0" prefHeight="35.0" prefWidth="359.0"
text="Do you want to view the best 5 batsmen?">
        <font>
            <Font name="Berlin Sans FB
Demi Bold" size="19.0" />
        </font>
    </Label>
    <TextField
fx:id="askToDisplayBest5Batsmen" layoutX="14.0"
layoutY="140.0" prefHeight="35.0" prefWidth="274.0"
promptText="Yes or No" styleClass="border"
stylesheets="@../values/style.css" />
        <Label layoutX="14.0"
layoutY="184.0" prefHeight="35.0" prefWidth="359.0"
text="Do you want to view the best 5 bowlers?">
        <font>
            <Font name="Berlin Sans FB
Demi Bold" size="19.0" />
        </font>
    </Label>
    <TextField
fx:id="askToDisplayBest5Bowlers" layoutX="14.0"
layoutY="219.0" prefHeight="35.0" prefWidth="274.0"
promptText="Yes or No" styleClass="border"
stylesheets="@../values/style.css" />
        <Label layoutX="14.0"
layoutY="254.0" prefHeight="35.0" prefWidth="426.0"
text="Do you want to view the tournament
standing?">
        <font>
            <Font name="Berlin Sans FB
Demi Bold" size="19.0" />
        </font>
    </Label>
```

```
        <TextField  
fx:id="askToDisplayTournamentStanding"  
layoutX="14.0" layoutY="289.0" prefHeight="35.0"  
prefWidth="274.0" promptText="Yes or No"  
styleClass="border"  
styleSheets="@../values/style.css" />  
            </children>  
        </Pane>  
    </children>  
</VBox>  
    <VBox alignment="TOP_CENTER"  
prefHeight="601.0" prefWidth="263.0"  
styleClass="left-side-background"  
styleSheets="@../values/style.css">  
        <children>  
            <ImageView fitHeight="316.0"  
fitWidth="285.0" pickOnBounds="true"  
preserveRatio="true">  
                <image>  
                    <Image url="@Images/istockphoto-  
1179575293-612x612.jpg" />  
                </image>  
            </ImageView>  
            <ImageView fitHeight="126.0"  
fitWidth="302.0" pickOnBounds="true"  
preserveRatio="true">  
                <image>  
                    <Image url="@Images/cricket-  
horizontal-banner-batsman-championship-background-  
free-vector.jpg" />  
                </image>  
            </ImageView>  
            <Pane prefHeight="159.0"  
prefWidth="285.0" VBox.vgrow="ALWAYS">  
                <children>  
                    <ImageView fitHeight="198.0"  
fitWidth="285.0" pickOnBounds="true"  
preserveRatio="true">
```

```

        <image>
            <Image
url="@Images/gettyimages-497203317-612x612.jpg" />
        </image>
    </ImageView>
</children>
</Pane>
</children>
</VBox>
</children>
</HBox>

```

8.guiToDisplayPlayerProfile.fxml:

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <children>
        <AnchorPane prefHeight="400.0"
prefWidth="600.0">
            <children>
                <AnchorPane prefHeight="400.0"
prefWidth="600.0">
                    <children>

```

```
        <GridPane alignment="center"
hgap="10" vgap="10">
            <columnConstraints>
                <ColumnConstraints />
                <ColumnConstraints />
            </columnConstraints>
            <rowConstraints>
                <RowConstraints />
                <RowConstraints />
            </rowConstraints>
            <children>
                <AnchorPane
prefHeight="746.0" prefWidth="1300.0">
                    <children>
                        <ImageView
fitHeight="790.0" fitWidth="1372.0" layoutX="-61.0"
layoutY="-5.0">
                            <image>
                                <Image
url="@Images/yogendra-singh-byvGfqk5QAY-
unsplash.jpg" />
                            </image>
                        </ImageView>
                        <AnchorPane
layoutX="383.0" layoutY="145.0" prefHeight="451.0"
prefWidth="541.0" style="-fx-background-color:
white;">
                            <children>
                                <Label
layoutX="32.0" layoutY="20.0" text="DISPLAY PLAYER
PROFILE">
                                    <font>
                                        <Font
name="Bodoni MT" size="36.0" />
                                    </font>
                                </Label>
                                <TextField
fx:id="askToDisplayPlayerProfile" layoutX="32.0"
layoutY="100.0" style="-fx-background-color: white;
-fx-border-color: black; -fx-border-width: 1px; -fx-
font-size: 14px; -fx-prompt-text-color: #ccc; -fx-
text-align: center; -fx-text-base-size: 14px; -fx-
text-color: black; -fx-text-font-family: sans-serif; -fx-
text-indent: 0px; -fx-text-overline: none; -fx-
text-decoration: none; -fx-text-underline: none; -fx-
text-wrap: nowrap; -fx-text-weight: bold; -fx-
width: 200px;" />
                            </children>
                        </AnchorPane>
                    </children>
                </AnchorPane>
            </children>
        </GridPane>
```

```
layoutY="169.0" prefHeight="44.0" prefWidth="279.0"
promptText="Yes or No" styleClass="border"
styleSheets="@../values/style.css" />
                                <Button
fx:id="addButton" layoutX="192.0" layoutY="364.0"
mnemonicParsing="false" onAction="#switchToScene5"
prefHeight="57.0" prefWidth="157.0" style="-fx-
background-color: #166e9a;" text="SUBMIT"
textFill="#ffffffffd4">
                                <font>
                                <Font
name="Tahoma" size="18.0" />
                                </font>
                            </Button>
                            <Label
layoutX="32.0" layoutY="126.0" text="Do you need to
know the details of player profiles">
                                <font>
                                <Font
name="Tahoma" size="21.0" />
                                </font>
                            </Label>
                            <Label
layoutX="32.0" layoutY="246.0" text="If yes, enter
the team name">
                                <font>
                                <Font
name="Tahoma" size="21.0" />
                                </font>
                            </Label>
                            <TextField
fx:id="teamNameToDisplay" layoutX="32.0"
layoutY="285.0" prefHeight="44.0" prefWidth="279.0"
promptText="Ex: England" styleClass="border"
styleSheets="@../values/style.css" />
                                </children>
                            </AnchorPane>
                        </children>
```

```

        </AnchorPane>
    </children>
</GridPane>
</children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</AnchorPane>

```

9.guiToDisplayTeamProfile.fxml:

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller">
    <children>
        <AnchorPane prefHeight="400.0"
prefWidth="600.0">
            <children>
                <GridPane alignment="center" hgap="10"
vgap="10">
                    <columnConstraints>
                        <ColumnConstraints />
                        <ColumnConstraints />

```

```
</columnConstraints>
<rowConstraints>
    <RowConstraints />
    <RowConstraints />
</rowConstraints>
<children>
    <AnchorPane prefHeight="746.0"
prefWidth="1300.0">
        <children>
            <ImageView
fitHeight="790.0" fitWidth="1372.0" layoutX="-61.0"
layoutY="-5.0">
                <image>
                    <Image
url="@Images/yogendra-singh-byvGfqk5QAY-
unsplash.jpg" />
                </image>
            </ImageView>
            <AnchorPane layoutX="383.0"
layoutY="145.0" prefHeight="444.0"
prefWidth="541.0" style="-fx-background-color:
white;">
                <children>
                    <Label layoutX="47.0"
layoutY="20.0" text="DISPLAY TEAM PROFILE">
                        <font>
                            <Font
name="Bodoni MT" size="36.0" />
                        </font>
                    </Label>
                    <TextField
fx:id="askToDisplayTeamProfile" layoutX="32.0"
layoutY="169.0" prefHeight="44.0" prefWidth="279.0"
promptText="Yes or No" styleClass="border"
stylesheets="@../values/style.css" />
                    <Button
fx:id="addButton" layoutX="192.0" layoutY="360.0"
mnemonicParsing="false" onAction="#switchToScene4">
                </children>
            </AnchorPane>
        </children>
    </AnchorPane>
</children>

```


CSS file:

Style.css file

```
.left-side-background{
    -fx-background-color: #030000;
}

.button{
    -fx-background-radius: 15px;
    -fx-text-fill: #ffffff;
}

.transparent{
    -fx-background-color: rgba(0,0,0,0);
}

.border{
    -fx-border-color: #544B49;
    -fx-border-width: 1.5;
}

#dark-blue {
    -fx-background-color:
        #090a0c,
        linear-gradient(#38424b 0%, #1f2429 20%,
#191d22 100%),
        linear-gradient(#20262b, #191d22),
        radial-gradient(center 50% 0%, radius 100%,
rgba(114,131,148,0.9), rgba(255,255,255,0));
    -fx-background-radius: 5,4,3,5;
    -fx-background-insets: 0,1,2,0;
    -fx-text-fill: white;
    -fx-effect: dropshadow( three-pass-box ,
rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );
    -fx-font-family: "Arial";
    -fx-text-fill: linear-gradient(white, #d0d0d0);
    -fx-font-size: 12px;
    -fx-padding: 10 20 10 20;
}
#dark-blue Text {
    -fx-effect: dropshadow( one-pass-box ,
```

```
rgba(0,0,0,0.9) , 1, 0.0 , 0 , 1 );  
}
```

Assumptions

- *Runs, Wickets of the player profile cannot be edit by the user and the values of those variables will be updated by the program after completion of each match (Reason: Runs and wickets will be used to find best batsman and bowler). Above variables are used to assess how players have performed in the T20 tournament
- *Name and role variables are the only variables which can be edit by the user in player profile
- *Wide and no balls are not considered in this T20 match
- *User is capable to delete only the player profiles, user can't delete team profiles, but user can delete some features of the team profile (Ex: No of matches)
- *Before starting the league, the 'rating' in team profiles given is based on how well the teams has performed in previous leagues
- *If a team won the toss and choose to bat first in the first innings, although the second team exceed the total score of the first team, still the second team have the chance to get more scores though that team won the match (Reason: It will increase the rating and helps to increase the position of the team)

Ex: Match between Afghanistan and Australia, Afghanistan won the toss and elected to bat first. Afghanistan scored 123 in first innings and Australia scored 124 within 18 overs in second innings. Australia won the match and still the Australian team have the chance to get more

score in the remaining 2 overs. So, the team can increase the rating which helps to maintain their position.

*First innings ends after 20 overs, 2nd innings will be started when the opposite team bats

*If a team won the match, 'rating' feature of the team profile will be updated by the total score taken by winning team divided by 10

(EX: total/10), 'rating' feature of the loosed team will be updated by dividing total score by 20(Ex: total/20)

*After 6 matches in each group, the team who got highest rating will be qualified for finals, team positions (tournament standing of teams) are based on 'rating'. Each teams play 3 matches within the group.

In other way of saying, the first 2 teams displaying in the tournament standing after the end of all matches in group A and B will be playing the finals

*There's no sequence of batting order (who's going to bat first, second, etc..). Perhaps there can be a chance a bowler can bat at the beginning.

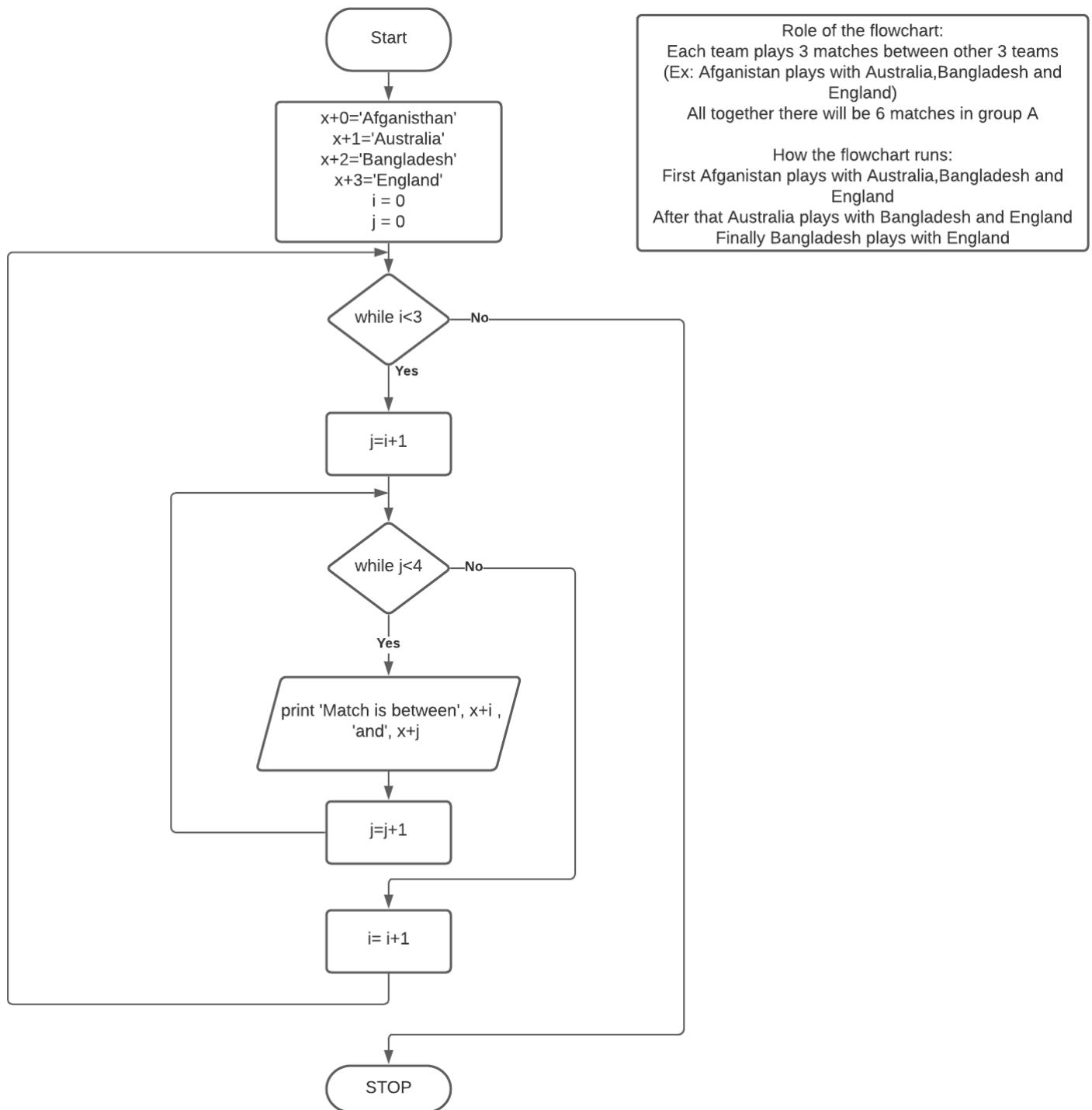
Runs=Total runs scored after each match

Ex: In 1st match player1 scored 12, in 2nd match he scored 50, so runs=12+50 after the 2nd match

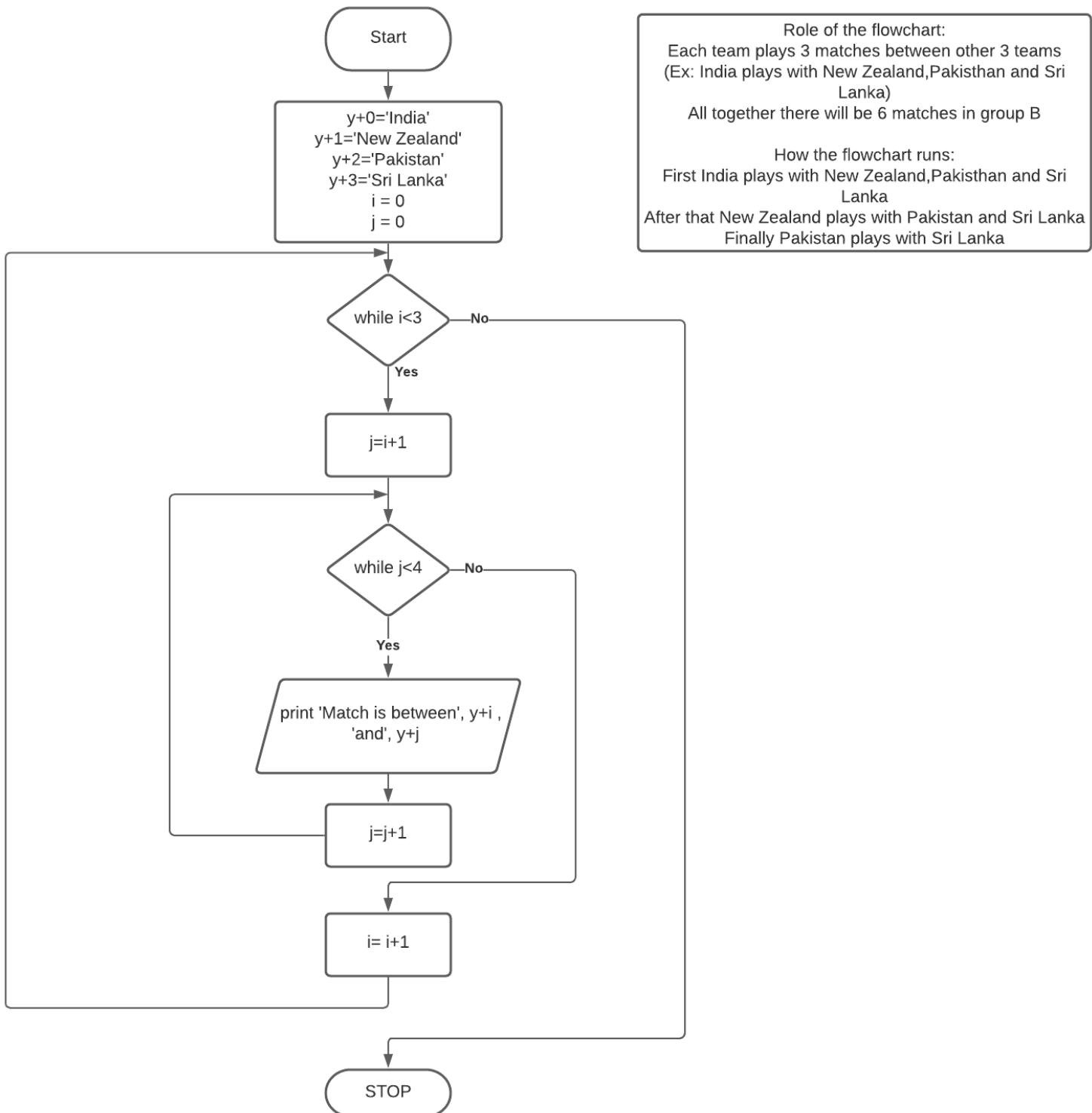
Best=Highest score taken by the player in the entire gameplay

Ex: In 1st match player1 scored 12, in 2nd match he scored 50,so his best will be 50

Random match generation for group A teams



Random match generation for group B teams



Test Cases:

Test Case	Description	Input	Expected Output	Actual Output
Toss	Toss is used to determine which team do the batting and bowling first.	No inputs given by the user. Result (Toss won/loose decision)will generate randomly	Team who calls the toss always selects the choice as head and if the coin flips as head,then the team who calls the toss will win the toss If the team who calls the won the toss,team has freedom to choose bat or ball(Generated randomly by the program) Sample Output: Team who won the toss : Afghanistan Decision of the team who won the toss(Bat/Ball) : Ball	Team who's calling the toss has the freedom to choose head or tail when the coin flips
Possible things that can happen to a ball	When a bowler put a ball, batsman can hit the ball and increase the scores or	No inputs given by the user. Result of each and every possible instances that can happen to a ball will	Output will not be displayed for each and every ball,but the final output after each match will be given when user asks to display the match summery. Wide and	Actual output will be same as the expected output,but the difference is ,in actual output wide

	batsman can out or nothing can happen(No scores or no wicket).	generate randomly. User should input yes or no(y/n) if user wants to get the match summery	no balls will not be considered by the program Sample Output of a bowler and a batsman: Name Rashid Khan wickets taken in a match 3 overs bowled 6	and no balls will also be taken for consideration.
			Name Asghar Afghan scores taken in a match 13 balls took for scoring 13 method of dismissal caught	
Number of 50's taken in matches by a batsman	Situation where batsman scores more than 50 runs in a match.	No user inputs to get only number of 50's taken by a player, number of 50's will be displayed along with the player profiles	When user wants to display the player profile and if a player scores more than 50 runs in a match, it will be displayed. Sample output: Name Dimuth Karunaratne Matches 3 Innings 6 Runs 107 50s 1 Best 61 wickets 0	Actual output will be same as the expected output

			Role batsman	
Best 5 batsmen	Filter and arrange best 5 players with highest runs taken in matches at a given point of time	Asks user whether user need to get the 5 best batsmen after the end of each matches Sample: Do you want to view the best 5 batsmen in the league(y/n): Expected Inputs: Y n	Sample output: Hashmatullah Shahidi has scored 68 runs Rashid Khan has scored 40 runs Aaron Finch has scored 39 runs Matthew Wade has scored 32 runs Mohammad Nabi has scored 30 runs	Actual output will be same as the expected output
Best 5 bowlers	Filter and arrange best 5 bowlers among the other bowlers based on the highest wickets taken in matches at a given point of time	Asks user whether user need to get the 5 best bowlers after the end of each matches Sample: Do you want to view the best 5 bowlers in the league upto now(y/n):	Sample output: Gulbadin Naib has taken 3 wickets Sayed Shirzad has taken 3 wickets Hamid Hasan has taken 2 wickets Rahmat Shah has taken 2 wickets Samiullah Shinwari has taken 2 wickets	Actual output will be same as the expected output

		Expected inputs: Y n		
How the rating is calculated	Rating is an attribute which is used to determine the positions of the teams.	User cannot specifically get the rating in a team, but it will be displayed as an instance when user needed to get the information on team profiles or tournament standing	Output will be displayed along with the team profiles or when user needed to get the tournament standing Sample output: Name India No of matches 3 position 1 rating 55.2	There's no such a variable called rating to determine the positions of the teams
Team positioning and tournament standing	Positions of the teams at a given time will be arranged based on the rating which can be achieved by teams during the matches.	Ask user if user needed to display the tournament standing Sample: Do you want to view the Tournament standing(y/n): Expected input: Y n	Sample output: Name India No of matches 3 position 1 rating 55.2 Name Australia No of matches 3 position 2 rating 50.05 Name England No of matches 3 position 3	In reality, the teams will be arranged according to the number of matches teams won and does not consider about the rating.

rating
49.10000000000001

....

....

....

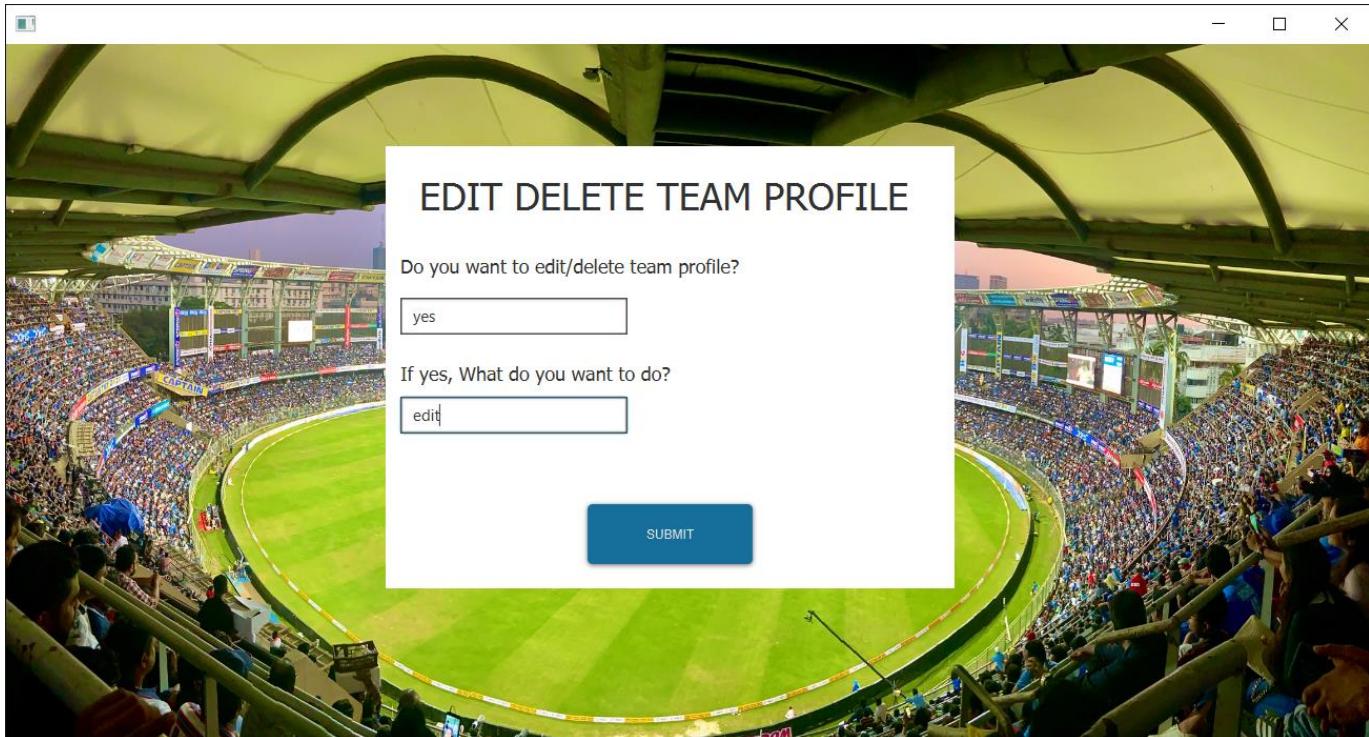
Likewise, all the team profiles will be displayed

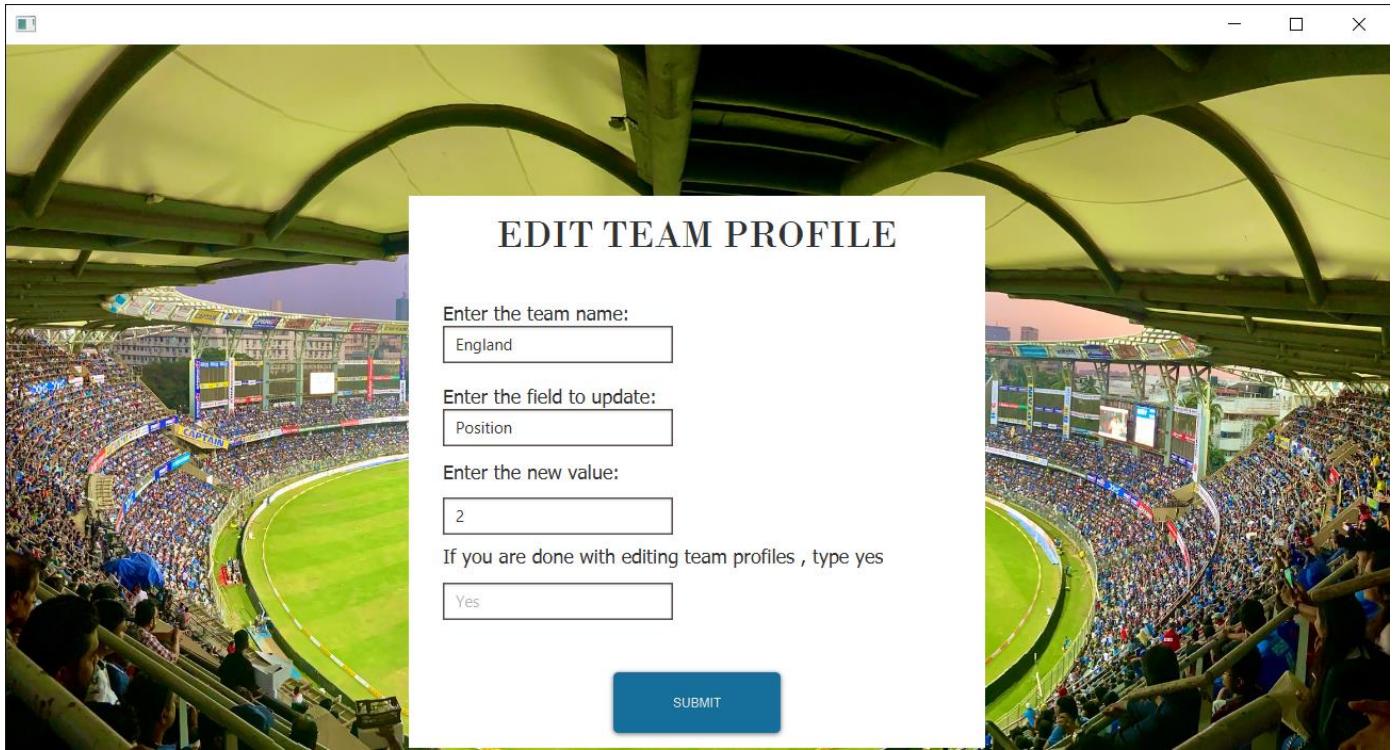
Match summery	Match summery is the summery of a match during two teams.	Sample: Do you want get the match summery between the above teams(y/n): Expected Input: Y n	Following will be displayed: <ul style="list-style-type: none">No of wickets taken in a match,overs bowled by each bowler in both teams along with bowler's namescores taken in a match ,balls took for scoring ,method of dismissal of each batsmen in both two teamsTotal marks and wickets taken by each team	Actual output will be same as the expected output
----------------------	---	---	--	---

- Team who won the match

Sample Outputs (Screen Shots):

1) Editing team profiles:





```
listOfAllThePlayers 129 tournamentMatches.displayTournamentStanding(askUserInputAndDisplayTournament
listOfAllTheTeams 130 }
Main 131 }

newZealandPlayerProfile

overviewOfAMatchSummary
```

Run: Main

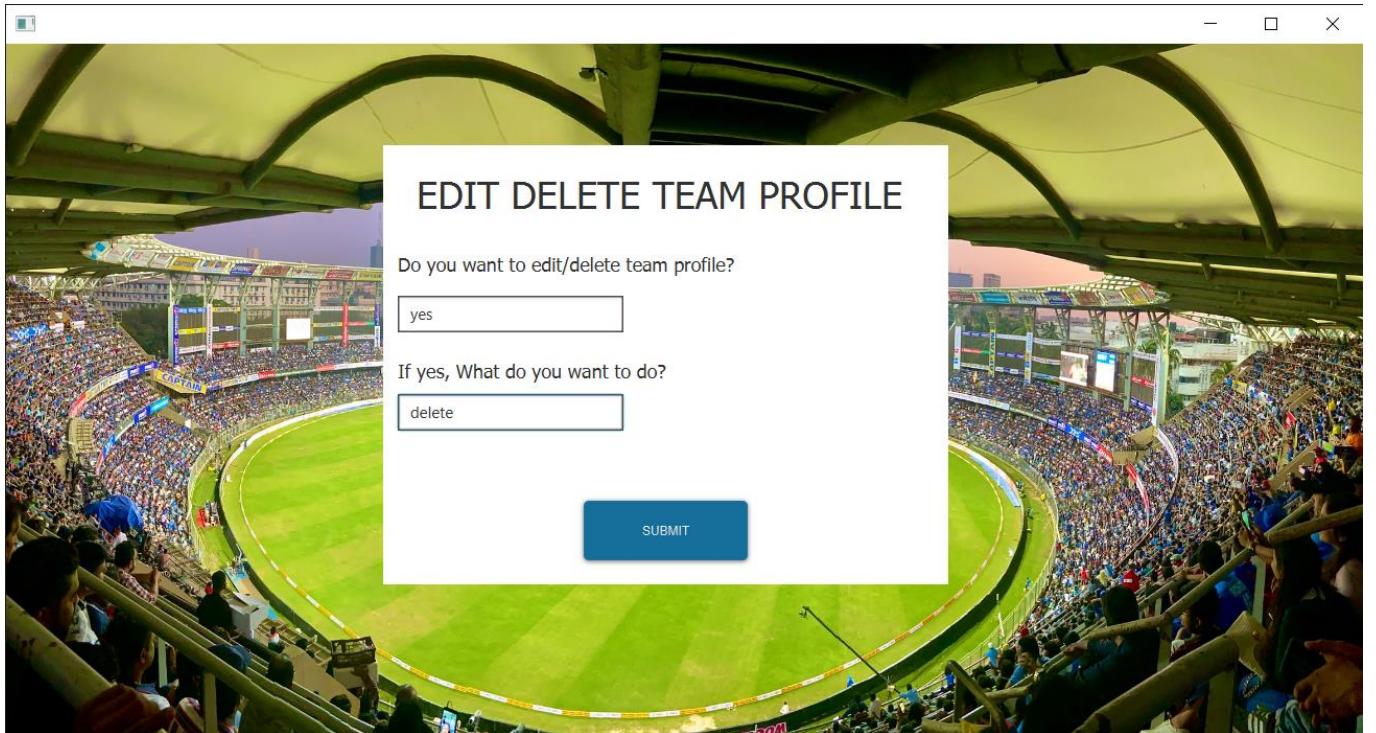
```
"C:\Program Files\Java\jdk1.8.0_321\bin\java.exe" ...
Successfully updated
Name = England
Number of matches = 0
Position = 2
Rating = 8.0
```

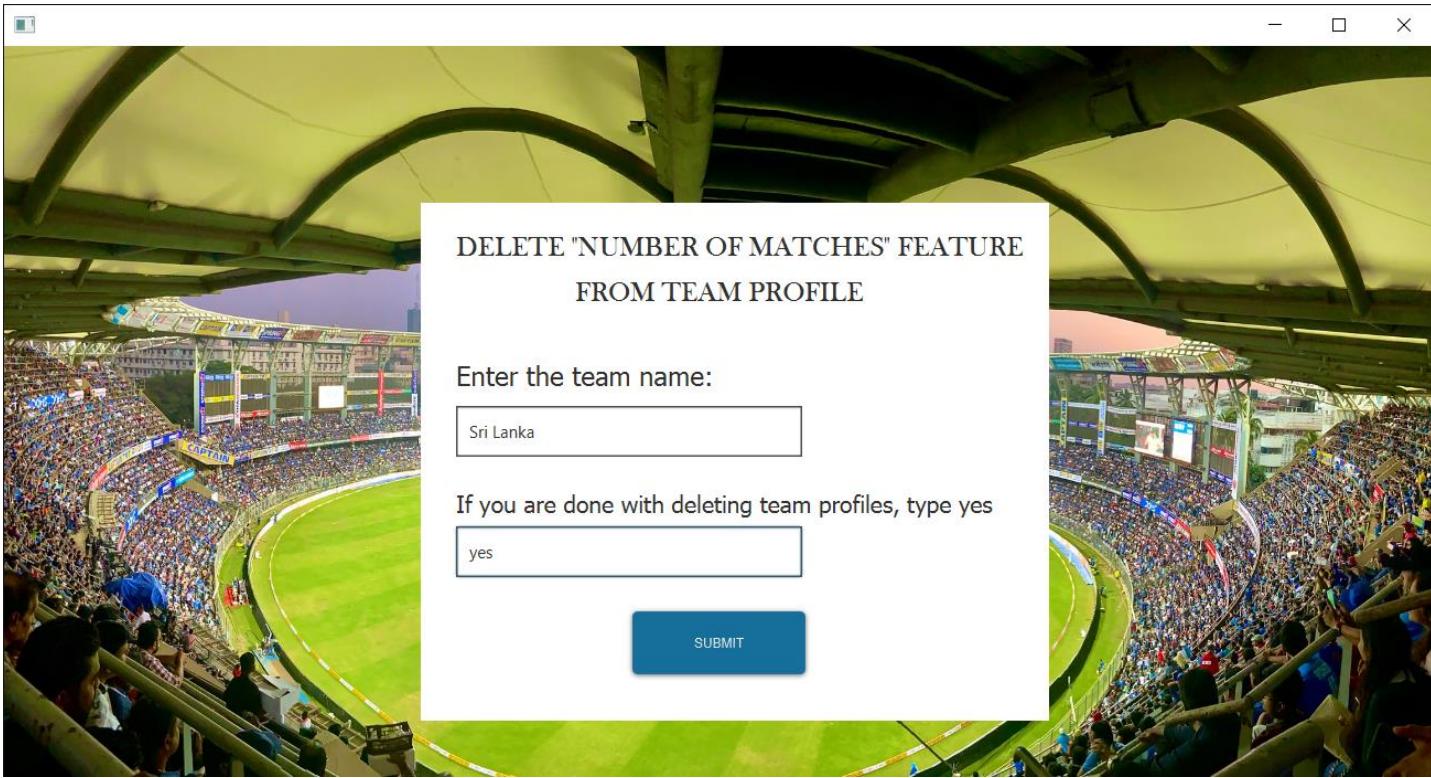
Event Log

Build completed successfully in 9 sec, 33 ms (23 minutes ago)

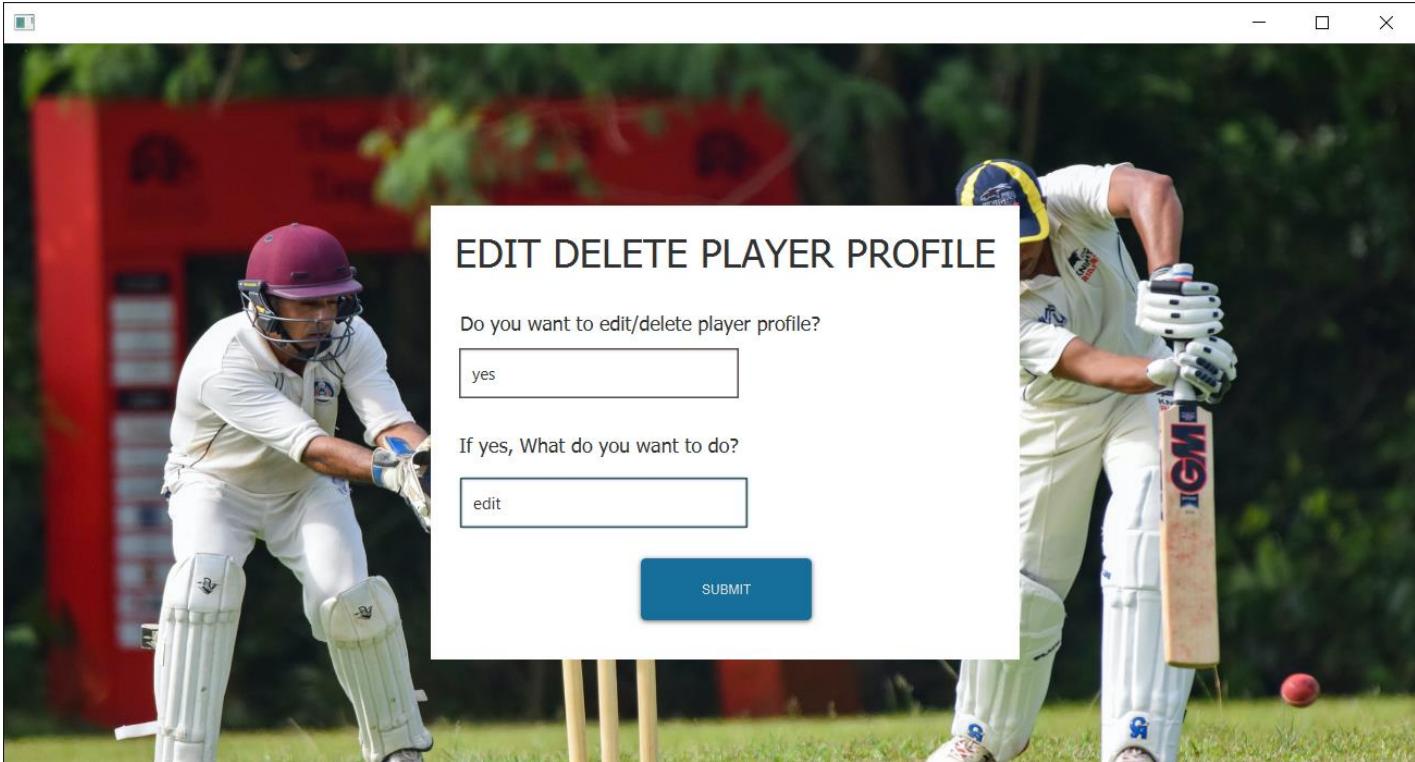
130:1 CRLF UTF-8 4 spaces

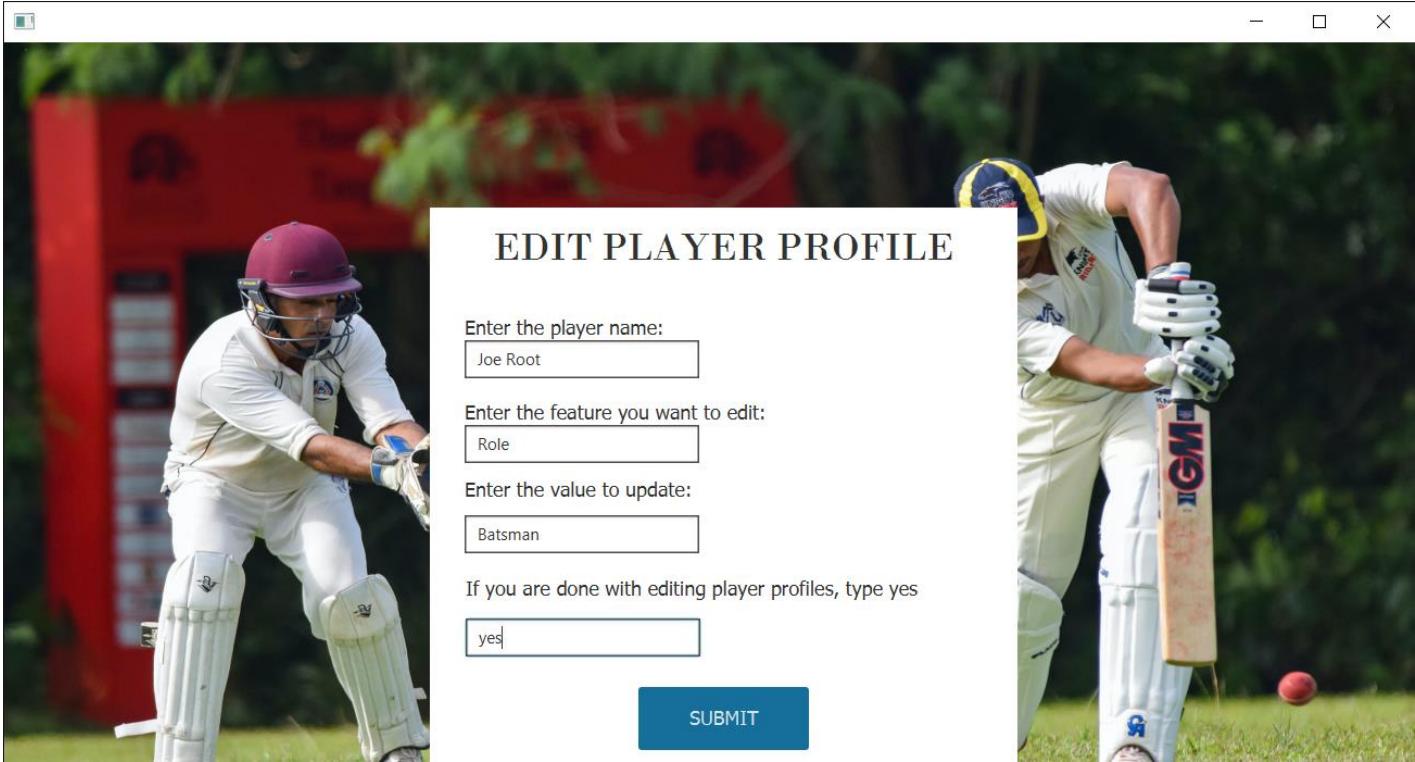
2. Deleting some features of team profiles:



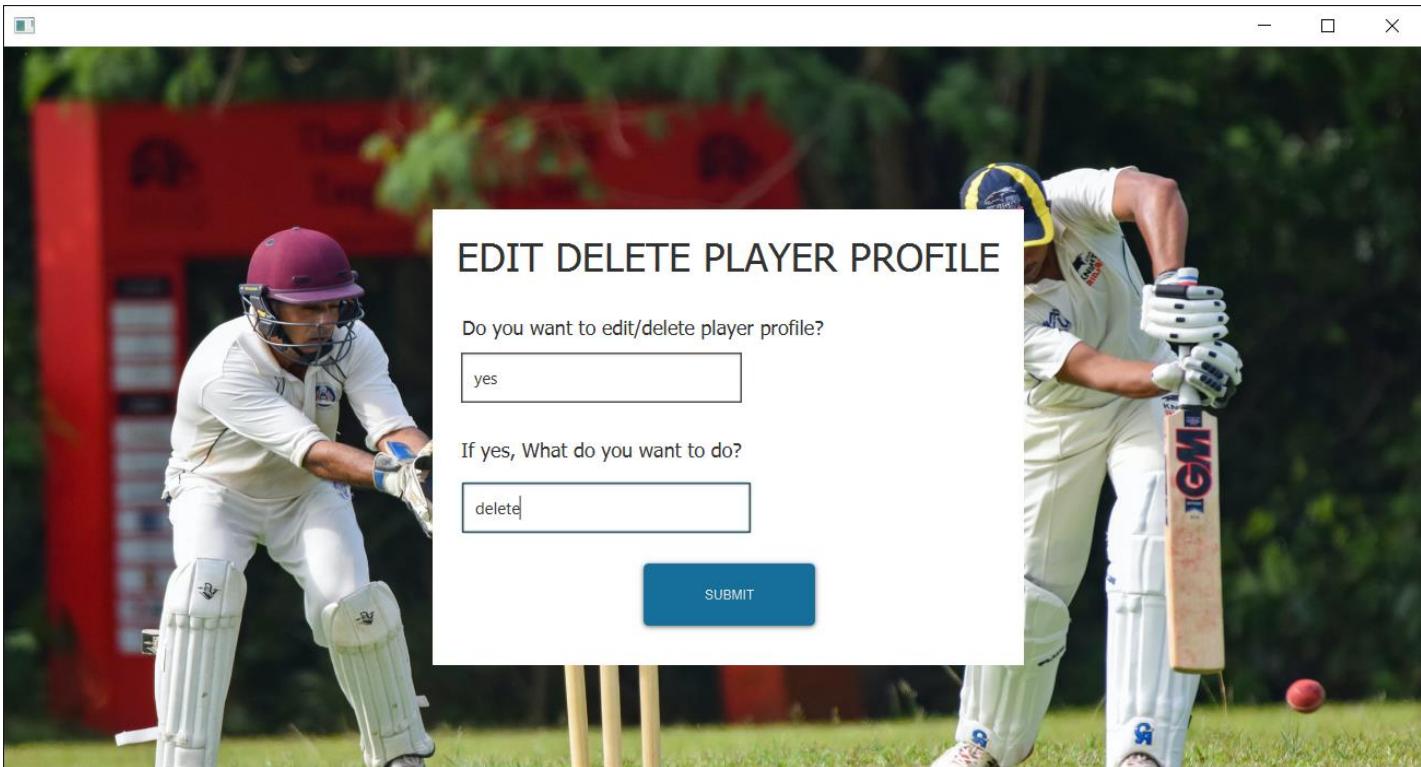


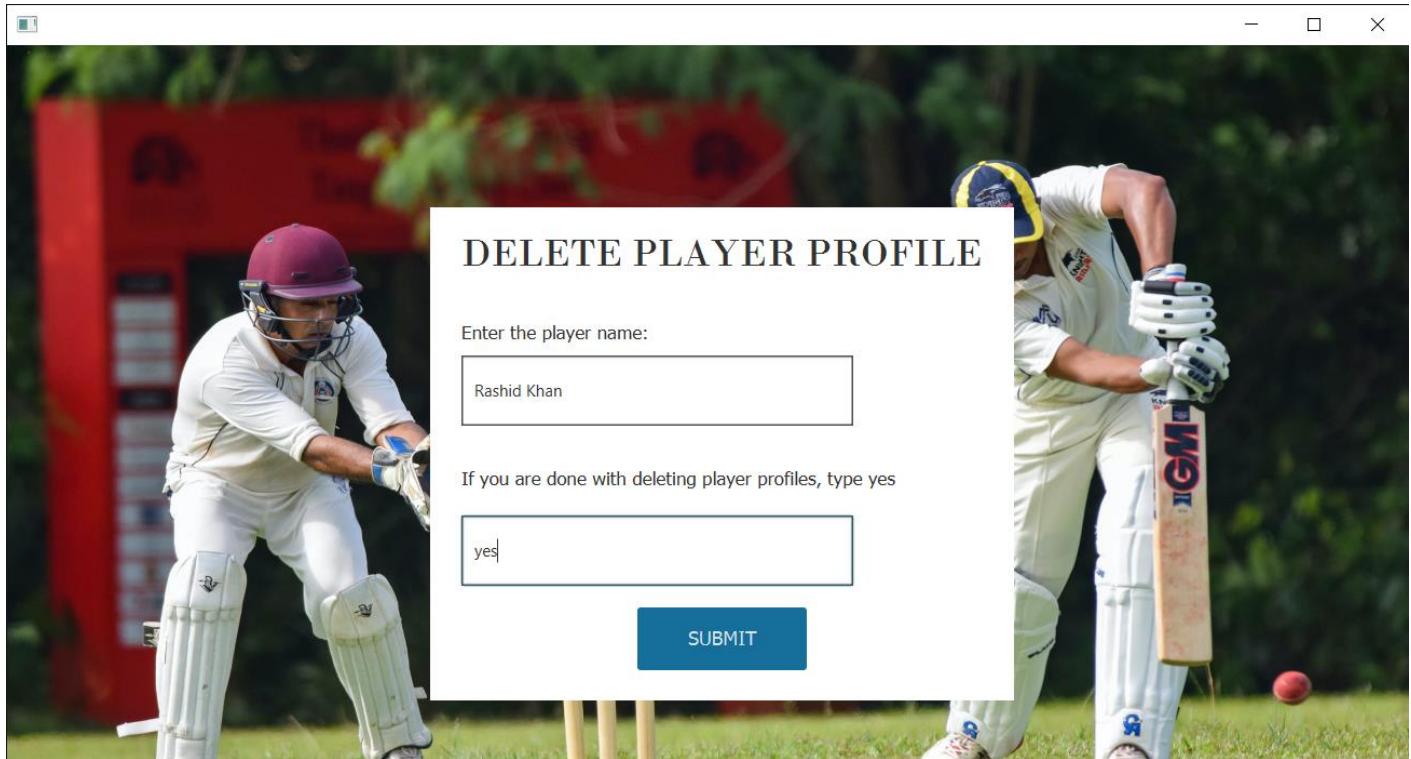
3)Editing player profiles:



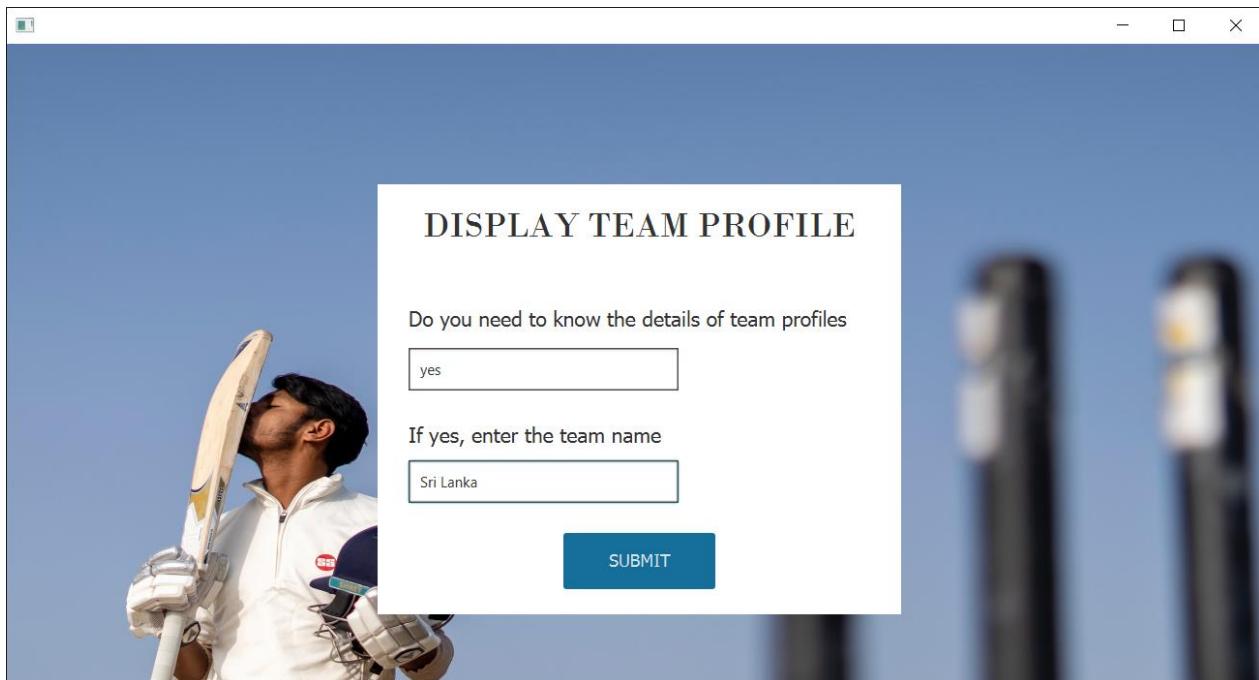


4. Deleting player profiles:





5. Display team profile:



The screenshot shows a Java IDE interface with the following details:

- Project View:** Shows files like `australiaPlayerProfile`, `bangladeshPlayerProfile`, `best5BowlersAndBatsman`, etc.
- Main.java Content:** The code is part of a class named `Main`. It contains methods for getting user input and displaying tournament standing or best 5 batsmen and bowlers.
- Terminal Output:** The terminal window shows the execution of the program, displaying player profiles for Sri Lanka and Shakib Al Hasan.
- Status Bar:** Shows "Build completed successfully in 9 sec, 33 ms (27 minutes ago)" and the current time as 4:22.

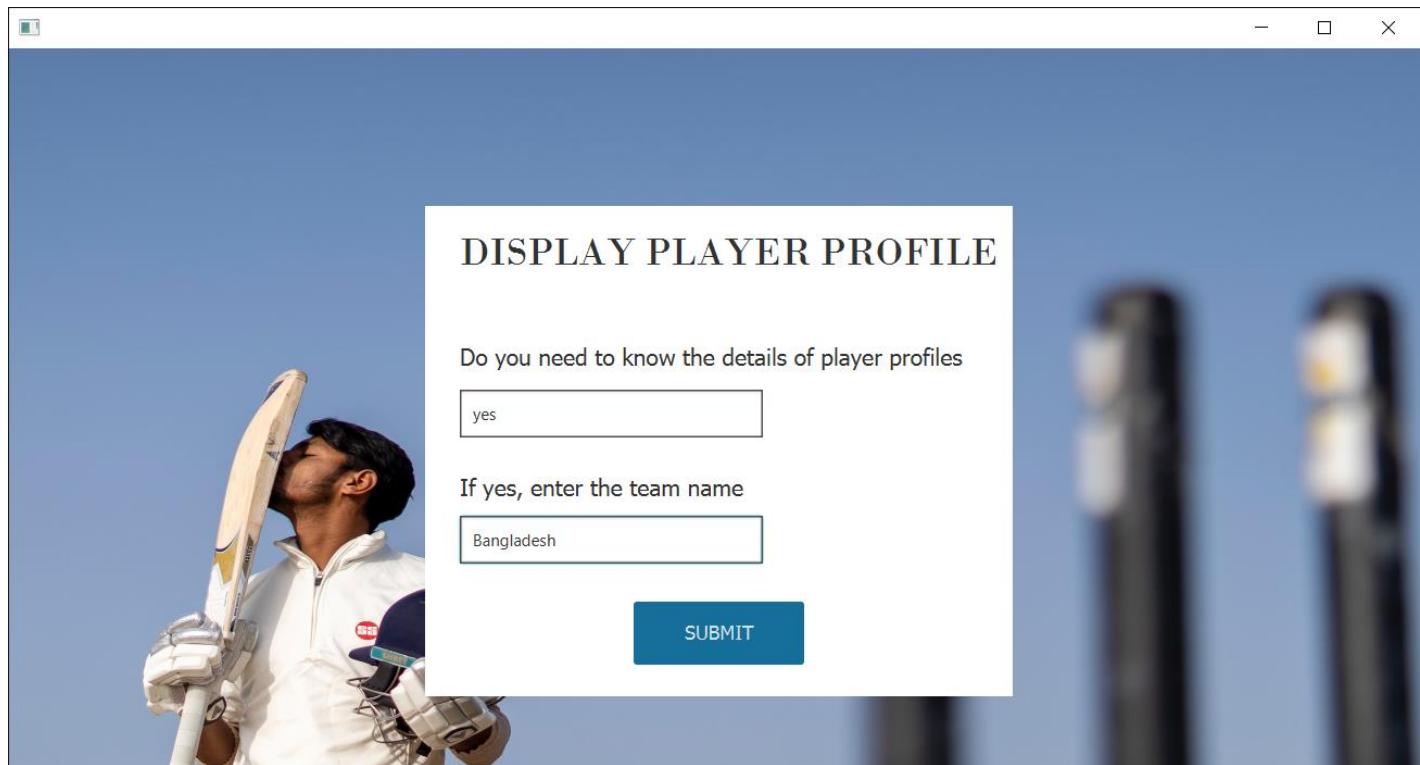
```
public void getUserInputAndDisplayBest5BatsmenAndBowlers(String askUser) {
    tournamentMatches.displayBest5BowlersOrBatsmen(askUserToDisplayBest5Batsmen, askUser);
}

public void getUserInputAndDisplayTournamentStanding(String askUserInputAndDisplayTournamentStanding) {
    tournamentMatches.displayTournamentStanding(askUserInputAndDisplayTournamentStanding);
}
```

```
Name = Sri Lanka
Number of matches = null
Position = 5
Rating = 4.45

Name = Shakib Al Hasan
```

6. Display player profile:



The screenshot shows a Java IDE interface with a dark theme. The menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The current file is CW - Main.java. The code in the editor prints player statistics for Shakib Al Hasan, Mahmudullah, and Mushfiqur Rahim. The output window shows the same information for each player. The bottom status bar indicates a successful build completed in 9 seconds.

```
Name = Shakib Al Hasan
Role = All Rounder
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

Name = Mahmudullah
Role = All Rounder
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

Name = Mushfiqur Rahim
Role = Batsman
```

This screenshot shows the same Java IDE interface as the first one, but the code has been modified to print player statistics for Mushfiqur Rahim, Liton Das, and Mustafizur Rahman, categorized by their roles. The output window displays the updated information. The bottom status bar again shows a successful build.

```
Name = Mushfiqur Rahim
Role = Batsman
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

Name = Liton Das
Role = Batsman
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

Name = Mustafizur Rahman
Role = Bowler
```

The screenshot shows a Java IDE interface with a dark theme. The menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help, and CW - Main.java. The toolbar has icons for Run, Stop, Refresh, and others. The Project tool window shows a single file named Main.java. The Run tool window also shows Main.java. The code in Main.java prints the statistics of four players:

```
Name = Mustafizur Rahman
Role = Bowler
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

Name = Tamim Iqbal
Role = Batsman
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

Name = Saumya Sarkar
Role = All Rounder
```

The status bar at the bottom indicates "Build completed successfully in 9 sec, 33 ms (31 minutes ago)" and "130:1 CRLF UTF-8 4 spaces".

This screenshot is identical to the one above, showing the same Java IDE interface and the same code output. It displays the statistics for four players: Mustafizur Rahman, Tamim Iqbal, Saumya Sarkar, and Mashrafe Mortaza. The status bar at the bottom indicates "Build completed successfully in 9 sec, 33 ms (31 minutes ago)" and "130:1 CRLF UTF-8 4 spaces".

The screenshot shows a Java IDE interface with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The current file is CW - Main.java. The code in the editor prints player statistics for three individuals: Mehidy Hasan, Mohammad Saifuddin, and Sabbir Rahman. Each player's name, role, and various performance metrics like matches, innings, runs, 50s, best score, and wickets are displayed. The output window shows the same information. The bottom status bar indicates the build completed successfully in 9 sec, 33 ms (31 minutes ago) at 130:1 CRLF UTF-8 4 spaces.

```
Name = Mehidy Hasan
Role = All Rounder
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

Name = Mohammad Saifuddin
Role = All Rounder
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

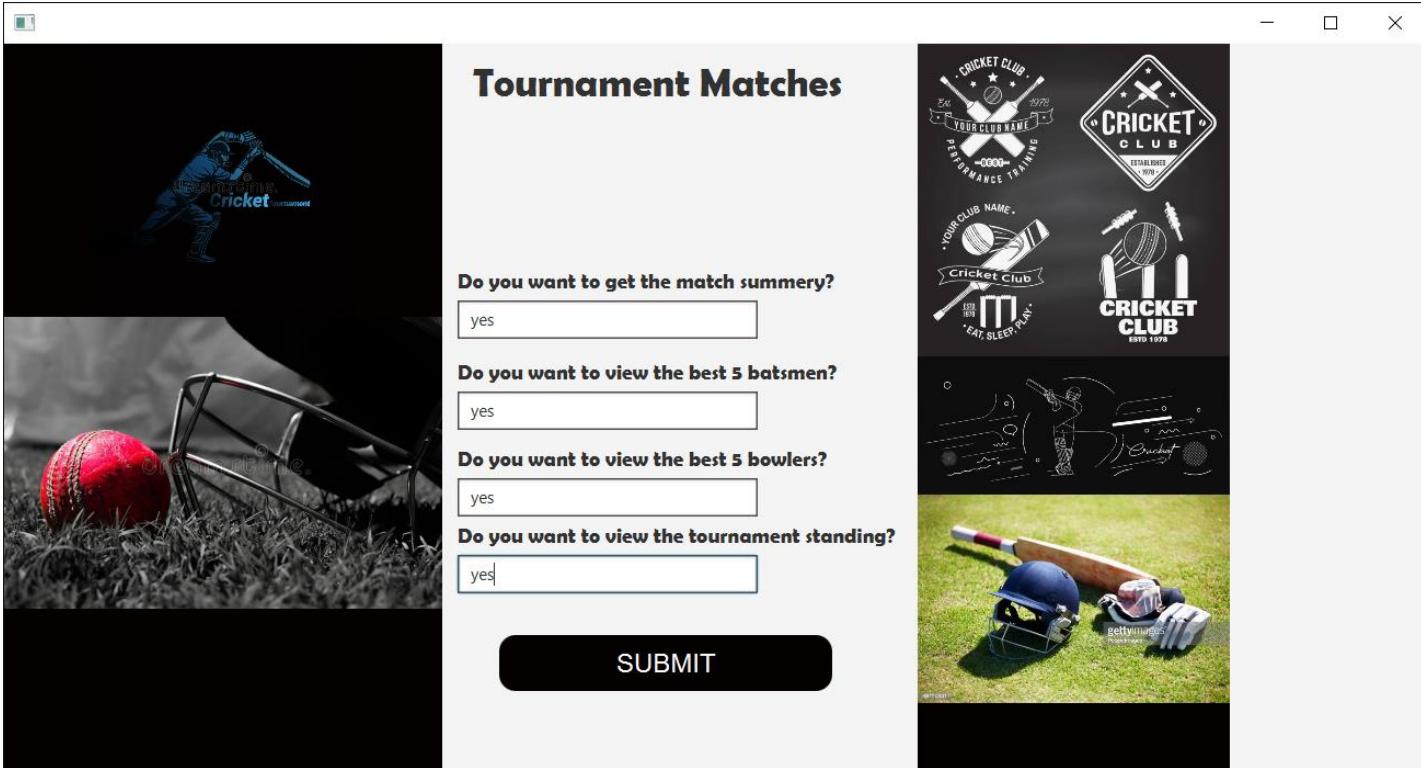
Name = Sabbir Rahman
Role = All Rounder
```

This screenshot shows the same Java IDE interface as the first one, but with different player data. It prints statistics for Sabbir Rahman and Rubel Hossain. The structure of the code and output is identical to the first screenshot, with each player's name, role, and various performance metrics being printed. The bottom status bar indicates the build completed successfully in 9 sec, 33 ms (32 minutes ago) at 130:1 CRLF UTF-8 4 spaces.

```
Name = Sabbir Rahman
Role = All Rounder
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0

Name = Rubel Hossain
Role = Bowler
Matches = 0
Innings = 0
Runs = 0
50s = 0
Best = 0
Wickets = 0
```

7. Display match summary, best 5 batsmen and bowlers, tournament standing:



```
The match between Afghanistan and Australia
Decision of the team who won the toss(Bat/Ball): Bat
Team who won the toss: Australia

Team Afghanistan:
Overs Bowled: 3
Name: Mohammad Nabi
Wickets taken in a match: 1

Overs Bowled: 1
Name: Mujeeb Rahman
Wickets taken in a match: 1

Overs Bowled: 6
Name: Hamid Hasan
Wickets taken in a match: 3

Overs Bowled: 0
```

The screenshot shows a Java IDE interface with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help, and CW - Main.java. The title bar displays "CW / src / sample / Main". The main window shows the code output in the center pane:

```
Overs Bowled: 0
Name: Rahmat Shah
Wickets taken in a match: 0

Overs Bowled: 4
Name: Gulbadin Naib
Wickets taken in a match: 3

Overs Bowled: 4
Name: Sayed Shirzad
Wickets taken in a match: 0

Overs Bowled: 2
Name: Samiullah Shinwari
Wickets taken in a match: 2

Total wickets taken by Afghanistan team is 10
*****
Name: Mohammad Nabi
```

At the bottom, the status bar shows "Build completed successfully in 9 sec, 33 ms (39 minutes ago)", "Event Log", "123:12 CRLF UTF-8 4 spaces", and a file icon.

The screenshot shows a Java IDE interface with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help, and CW - Main.java. The title bar displays "CW / src / sample / Main". The main window shows the code output in the center pane:

```
TOTAL WICKETS TAKEN BY AFGHANISTAN TEAM IS 10
*****
Name: Mohammad Nabi
Method of dismissal: caught
Scores taken in a match: 0
Balls took for scoring: 1

Name: Asghar Afghan
Method of dismissal: caught
Scores taken in a match: 24
Balls took for scoring: 13

Name: Hashmatullah Shahidi
Method of dismissal: caught
Scores taken in a match: 66
Balls took for scoring: 45

Name: Mujeeb Rahman
Method of dismissal: caught
Scores taken in a match: 21
Balls took for scoring: 15
```

At the bottom, the status bar shows "Build completed successfully in 9 sec, 33 ms (40 minutes ago)", "Event Log", "164:1 CRLF UTF-8 4 spaces", and a file icon.

The screenshot shows a Java IDE interface with a dark theme. The menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help, and CW - Main.java. The title bar indicates the project is 'CW > src > sample > Main'. The code editor displays the following output:

```
Name: Hamid Hasan  
Method of dismissal: bowled  
Scores taken in a match: 20  
Balls took for scoring: 17  
  
Name: Rahmat Shah  
Method of dismissal: caught  
Scores taken in a match: 16  
Balls took for scoring: 9  
  
Name: Noor Ali  
Method of dismissal: caught  
Scores taken in a match: 1  
Balls took for scoring: 3  
  
Name: Gulbadin Naib  
Method of dismissal: caught  
Scores taken in a match: 12  
Balls took for scoring: 7
```

The bottom status bar shows 'Build completed successfully in 9 sec, 33 ms (40 minutes ago)'. The bottom right corner shows 'Event Log' with '164:1 CRLF UTF-8 4 spaces'.

The screenshot shows a Java IDE interface with a dark theme. The menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help, and CW - Main.java. The title bar indicates the project is 'CW > src > sample > Main'. The code editor displays the following output:

```
Name: Sayed Shirzad  
Method of dismissal: Played till the last over.  
Scores taken in a match: 13  
Balls took for scoring: 10  
  
Name: Ikram Alikhil  
Method of dismissal: Not played  
Scores taken in a match: 0  
Balls took for scoring: 0  
  
Name: Samiullah Shinwari  
Method of dismissal: Not played  
Scores taken in a match: 0  
Balls took for scoring: 0  
  
Total marks of Afghanistan team is 173  
*****  
  
Team Australia:  
Overs Bowled: 5
```

The bottom status bar shows 'Build completed successfully in 9 sec, 33 ms (40 minutes ago)'. The bottom right corner shows 'Event Log' with '204:28 CRLF UTF-8 4 spaces'.

The screenshot shows a Java IDE interface with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help, and CW - Main.java. The left sidebar has sections for Project, Run, Favorites, Structure, and Event Log. The main area displays the following text output:

```
Team Australia:  
Overs Bowled: 5  
Name: Mitchel Starc  
Wickets taken in a match: 0  
  
Overs Bowled: 4  
Name: Adam Zampa  
Wickets taken in a match: 3  
  
Overs Bowled: 1  
Name: Glenn Maxwell  
Wickets taken in a match: 1  
  
Overs Bowled: 3  
Name: Pat Cummins  
Wickets taken in a match: 1  
  
Overs Bowled: 3  
Name: Marcus Stoinis  
Wickets taken in a match: 3
```

At the bottom, the status bar shows "Build completed successfully in 9 sec, 33 ms (41 minutes ago)" and "224:1 CRLF UTF-8 4 spaces".

This screenshot shows the same Java IDE interface. The output text is identical to the one above, but it concludes with a summary line:

```
Total wickets taken by Australia team is 8  
*****
```

At the bottom, the status bar shows "Build completed successfully in 9 sec, 33 ms (41 minutes ago)" and "224:1 CRLF UTF-8 4 spaces".

The screenshot shows a Java IDE interface with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The title bar indicates the current file is CW - Main.java. The left sidebar has sections for Project, Favorites, Structure, and Run. The main workspace displays the following text output from the Main.java program:

```
Name: Mitchel Starc  
Method of dismissal: bowled  
Scores taken in a match: 29  
Balls took for scoring: 28  
  
Name: Adam Zampa  
Method of dismissal: leg by wicket  
Scores taken in a match: 14  
Balls took for scoring: 11  
  
Name: Glenn Maxwell  
Method of dismissal: caught  
Scores taken in a match: 60  
Balls took for scoring: 34  
  
Name: Matthew Wade  
Method of dismissal: caught  
Scores taken in a match: 9  
Balls took for scoring: 7
```

The bottom status bar shows "Build completed successfully in 9 sec, 33 ms (41 minutes ago)" and "263:18 CRLF UTF-8 4 spaces".

This screenshot shows the same Java IDE interface as the first one, but with a different set of output text from the Main.java program:

```
Name: Pat Cummins  
Method of dismissal: bowled  
Scores taken in a match: 16  
Balls took for scoring: 6  
  
Name: Marcus Stoinis  
Method of dismissal: stumped  
Scores taken in a match: 8  
Balls took for scoring: 9  
  
Name: Alex Carey  
Method of dismissal: stumped  
Scores taken in a match: 15  
Balls took for scoring: 7  
  
Name: Nathan Lyon  
Method of dismissal: Not played  
Scores taken in a match: 0  
Balls took for scoring: 0
```

The bottom status bar shows "Build completed successfully in 9 sec, 33 ms (41 minutes ago)" and "263:18 CRLF UTF-8 4 spaces".

```
Name: Kane Richardson
Method of dismissal: Not played
Scores taken in a match: 0
Balls took for scoring: 0

Total marks of Australia team is 177
*****
Australia won the match
Hamid Hasan has taken 3 wickets.
Gulbadin Naib has taken 3 wickets.
Adam Zampa has taken 3 wickets.
Marcus Stoinis has taken 3 wickets.
Samiullah Shinwari has taken 2 wickets.
*****
Hashmatullah Shahidi has taken 66 runs.
Glenn Maxwell has taken 60 runs.
Mitchel Starc has taken 29 runs.
Asghar Afghan has taken 24 runs.
Mujeeb Rahman has taken 21 runs.
```

```
Name: Australia
Number of matches: 1
Position: 1
Rating: 24.0

Name: Afganistan
Number of matches: 1
Position: 2
Rating: 10.0

Name: New Zealand
Number of matches: 0
Position: 3
Rating: 10.0

Name: England
Number of matches: 0
Position: 4
Rating: 8.0
```

The screenshot shows a Java IDE interface with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help, and CW - Main.java. The left sidebar has a Project view showing a sample project with a Main.java file. The main editor area displays the following text output:

```
Name: India
Number of matches: 0
Position: 5
Rating: 6.5

Name: Sri Lanka
Number of matches: null
Position: 6
Rating: 4.45

Name: Bangladesh
Number of matches: 0
Position: 7
Rating: 4.1

Name: Pakistan
Number of matches: 0
Position: 8
Rating: 4.0
```

At the bottom, the status bar shows "Build completed successfully in 9 sec, 33 ms (42 minutes ago)" and "338:9 CRLF UTF-8 4 spaces".