

```
1 using System;
2 using System.Collections.Generic;
3 using System.Diagnostics;
4 using System.Linq;
5 using System.Web;
6 using System.Web.UI;
7 using System.Web.UI.HtmlControls;
8 using System.Web.UI.WebControls;
9 using DAL;
10 using DAL.apiClases;
11
12 public partial class Betting : System.Web.UI.Page
13 {
14     public string fixtures;
15     List<Root> fixs;
16     List<Bet> lb;
17     Dictionary<int, string> clubs;
18     public string currgameId;
19     public string error;
20     protected void Page_Load(object sender, EventArgs e)
21     {
22         //only people logged in can use this page
23         if (Session["username"] == null)
24         {
25             Response.Redirect("~/initPage.aspx");
26         }
27
28         addFixtures();
29         bets.Visible = false;
30         gameid.Visible = false;
31         //important to remove later!!!!!!!!!!!!!!!!!!!!!!
32         //Session["username"] = "ort";
33         if (ViewState["State"] != null && ViewState["State"].ToString() == "2")
34         {
35             state2();
36         }
37     }
38     public void addFixtures()
39     {
40         int x = 1;
41         fixs = APICall.GetCall();
42         List<Root> lr = APICall.sortbyNextGameWeek(fixs,
43             APICall.getCurrentGameweek(0));
44         clubs = FPLFunctions.getdicOfClubs();
45         foreach (Root fixture in lr)
46         {
47             //fixtures += GlobalFunctions.CreateGame(fixture, clubs);
48             CreateGame(fixture, clubs);
49             Debug.WriteLine("added game: " + x.ToString());
50             x += 1;
51         }
52     }
53 }
```

```

52     public void CreateGame(Root game, Dictionary<int, string> clubs)
53     {
54         //creating vairables
55         string homeT = clubs[game.team_h];
56         string awayT = clubs[game.team_a];
57
58         DateTime time = game.kickoff_time ?? DateTime.Now.AddYears(-19999999);
59
60         int? homeS = 0;
61         if (game.team_h_score != null)
62             homeS = game.team_h_score;
63         int? awayS = 0;
64         if (game.team_a_score != null)
65             awayS = game.team_a_score;
66
67         //create fixture
68         HtmlGenericControl gameWrap = new HtmlGenericControl("DIV");
69         gameWrap.Attributes.Add("class", "row");
70         mfix.Controls.Add(gameWrap);
71
72         HtmlGenericControl homeTClass = new HtmlGenericControl("DIV");
73         homeTClass.Attributes.Add("class", "col - sm");
74         homeTClass.InnerHtml = "<img src='images/badges/' + homeT + ".png'>"; ;
75         gameWrap.Controls.Add(homeTClass);
76
77
78         HtmlGenericControl score = new HtmlGenericControl("DIV");
79         score.Attributes.Add("class", "col - sm");
80         gameWrap.Controls.Add(score);
81         HtmlGenericControl scoreInRow = new HtmlGenericControl("DIV");
82         scoreInRow.Attributes.Add("class", "row");
83         score.Controls.Add(scoreInRow);
84         HtmlGenericControl homeTName = new HtmlGenericControl("DIV");
85         homeTName.Attributes.Add("class", "col - sm");
86         homeTName.InnerHtml = "<p>" + homeT + "</p>";
87         scoreInRow.Controls.Add(homeTName);
88         HtmlGenericControl sOrT = new HtmlGenericControl("DIV");
89         sOrT.Attributes.Add("class", "col - sm");
90         if (game.started == true /*chage to false at end of season to check*/)
91         {
92             sOrT.InnerHtml = "<p> (" + homeS + " - " + awayS + ") </p>";
93         }
94         else
95         {
96             sOrT.InnerHtml = "<p> " + time.ToString("dddd, dd MMMM h:mm tt") + " ↗
97                 </p>";
98             Button b = new Button();
99             b.Text = "bet";
100             b.ID = game.id.ToString();
101             b.Click += new EventHandler(bet_Click);
102             b.CssClass = "btn btn-primary";
103             sOrT.Controls.Add(b);

```

```

103     }
104     scoreInRow.Controls.Add(sOrT);
105     HtmlGenericControl awayTName = new HtmlGenericControl("DIV");
106     awayTName.Attributes.Add("class", "col - sm");
107     awayTName.InnerHtml = "<p>" + awayT + "</p>";
108     scoreInRow.Controls.Add(awayTName);
109
110     HtmlGenericControl awayTClass = new HtmlGenericControl("DIV");
111     awayTClass.Attributes.Add("class", "col - sm");
112     awayTClass.InnerHtml = "<img src='images/badges/' + awayT + ".png'>"; ;
113     gameWrap.Controls.Add(awayTClass);
114 }
115 public void addclub(string club)
116 {
117     foreach (string playerName in CardFunctions.getByClubNames(club))
118     {
119         playersdl.Items.Add(playerName);
120     }
121 }
122 protected void bet_Click(object sender, EventArgs e)
123 {
124
125     if (BettingFunctions.didBet(Session["username"].ToString(),
126         Convert.ToInt32(((Button)sender).ID))
127     {
128         error = "already bet on this game";
129     }
130     else
131     {
132         bets.Visible = true;
133         mfix.Visible = false;
134         Root fixture = getgamebyId(((Button)sender).ID);
135         hteamname.Text = clubs[fixture.team_h];
136         ateamname.Text = clubs[fixture.team_a];
137         gameid.Text = ((Button)sender).ID;
138
139         addclub(clubs[fixture.team_h]);
140         addclub(clubs[fixture.team_a]);
141         Debug.WriteLine(((Button)sender).ID);
142     }
143 }
144 protected void subbtn_Click(object sender, EventArgs e)
145 {
146     bets.Visible = false;
147     mfix.Visible = true;
148
149     string score = hteamscore.Text + " - " + ateamscore.Text;
150     string winner = "";
151     if (Convert.ToInt32(hteamscore.Text) > Convert.ToInt32(ateamscore.Text))
152     {
153         winner = hteamname.Text;

```

```
154     }
155     else if (Convert.ToInt32(hteamscore.Text) < Convert.ToInt32
156             (ateamscore.Text))
157     {
158         winner = ateamname.Text;
159     }
160     else
161     {
162         winner = "draw";
163     }
164     BettingFunctions.AddBet(Session["username"].ToString(), Convert.ToInt32
165                             (gameid.Text), winner, score, playersdl.Selected.Value.ToString());
166
167     playersdl.Items.Clear();
168     hteamscore.Text = 0.ToString();
169     ateamscore.Text = 0.ToString();
170 }
171 public Root getgamebyId(string id)
172 {
173     foreach (Root fixture in fixs)
174     {
175         if (fixture.id.ToString() == id)
176         {
177             return fixture;
178         }
179     }
180     return new Root();
181 }
182 public void CreateBet(Bet lb, Root game)
183 {
184     //creating vairables
185     string homeT = clubs[game.team_h];
186     string awayT = clubs[game.team_a];
187
188     DateTime time = game.kickoff_time ?? DateTime.Now.AddYears(-19999999);
189
190     int? homeS = 0;
191     if (game.team_h_score != null)
192     {
193         homeS = game.team_h_score;
194     }
195     int? awayS = 0;
196     if (game.team_a_score != null)
197     {
198         awayS = game.team_a_score;
199     }
200
201     //create fixture
202     HtmlGenericControl gameWrap = new HtmlGenericControl("DIV");
203     gameWrap.Attributes.Add("class", "row");
204     bfix.Controls.Add(gameWrap);
205
206     HtmlGenericControl homeTClass = new HtmlGenericControl("DIV");
207     homeTClass.Attributes.Add("class", "col - sm");
208     homeTClass.InnerHtml = "<img src='images/badges/' + homeT + ".png'>";
```

```
204     gamewrap.Controls.Add(homeTClass);
205
206
207     HtmlGenericControl score = new HtmlGenericControl("DIV");
208     score.Attributes.Add("class", "col - sm");
209     gamewrap.Controls.Add(score);
210     HtmlGenericControl scoreInRow = new HtmlGenericControl("DIV");
211     scoreInRow.Attributes.Add("class", "row");
212     score.Controls.Add(scoreInRow);
213     HtmlGenericControl homeTName = new HtmlGenericControl("DIV");
214     homeTName.Attributes.Add("class", "col - sm");
215     homeTName.InnerHtml = "<p>" + homeT + "</p>";
216     scoreInRow.Controls.Add(homeTName);
217     HtmlGenericControl sOrT = new HtmlGenericControl("DIV");
218     sOrT.Attributes.Add("class", "col - sm");
219     if (game.finished)
220     {
221         sOrT.InnerHtml = "<p> Full Time </p>";
222
223         bool isrlscore = false;
224         string rlscore = homeS + " - " + awayS;
225         if (rlscore == lb.score)
226         {
227             sOrT.InnerHtml += "<p style='color: green;'>Fulltime score: [" + ↗
                rlscore + "]" </p>";
228             isrlscore = true;
229         }
230         else
231             sOrT.InnerHtml += "<p style='color: red;'>Fulltime score: [" + ↗
                rlscore + "]" </p>";
232
233         Button b = new Button();
234         b.Attributes.Add("winside", "0");
235
236         bool iswteam = false;
237         string winningTeam = "";
238         if (game.team_h_score > game.team_a_score)
239             winningTeam = clubs[game.team_h];
240         else if (game.team_a_score > game.team_h_score)
241         {
242             winningTeam = clubs[game.team_a];
243             b.Attributes["winside"] = "2";
244         }
245         else
246         {
247             winningTeam = "draw";
248             b.Attributes["winside"] = "1";
249         }
250         if (winningTeam == lb.winner)
251         {
252             sOrT.InnerHtml += "<p style='color: green;'>Winning team: [" + ↗
                lb.winner + "]" </p>";
```

```

253         iswteam = true;
254     }
255     else
256         sOrT.InnerHtml += "<p style='color: red;'>Winning team: [" + lb.winner + "] </p>";
257
258
259     bool isscorer = false;
260     foreach (Stat stt in game.stats)
261     {
262         if (stt.identifier == "goals_scored")
263         {
264             string c1 = "";
265             string c2 = "";
266             try { c1 = CardFunctions.getByCardId(stt.a [0].element).name; } catch { }
267             try { c2 = CardFunctions.getByCardId(stt.h [0].element).name; } catch { }
268
269             if (c1 == lb.scorer || c2 == lb.scorer)
270             {
271                 sOrT.InnerHtml += "<p style='color: green;'>First scorer: [" + lb.scorer + "] </p>";
272                 isscorer = true;
273             }
274             else
275                 sOrT.InnerHtml += "<p style='color: red;'>First scorer: [" + lb.scorer + "] </p>";
276             break;
277         }
278     }
279
280     b.Text = "Claim Rewards";
281     b.Attributes.Add("myid", game.id.ToString());
282     b.Attributes.Add("winner", "h");
283     b.Attributes.Add("gold", "0");
284     b.Attributes.Add("diamond", "0");
285     b.Attributes.Add("silver", "0");
286     b.CssClass = "btn btn-primary";
287     sOrT.Controls.Add(b);
288     b.Click += new EventHandler(Claim_Click);
289     if (lb.didClaim)
290         b.Enabled = false;
291     //if (isrlscore)
292     //    b.Attributes["silver"] = "1";
293     //if (isscorer)
294     //    b.Attributes["gold"] = "1";
295     if (iswteam)
296         b.Attributes["winner"] = "t";
297     else
298         b.Attributes["winner"] = "f";
299 }

```

```

300     else
301     {
302         sOrT.InnerHtml = "<p> Your bet </p>";
303         sOrT.InnerHtml += "<p>Fulltime score: [" + lb.score + "] </p>";
304         sOrT.InnerHtml += "<p>Winning team: [" + lb.winner + "] </p>";
305         sOrT.InnerHtml += "<p>First scorer: [" + lb.scorer + "] </p>";
306
307     }
308     scoreInRow.Controls.Add(sOrT);
309     HtmlGenericControl awayTName = new HtmlGenericControl("DIV");
310     awayTName.Attributes.Add("class", "col - sm");
311     awayTName.InnerHtml = "<p>" + awayT + "</p>";
312     scoreInRow.Controls.Add(awayTName);
313
314     HtmlGenericControl awayTClass = new HtmlGenericControl("DIV");
315     awayTClass.Attributes.Add("class", "col - sm");
316     awayTClass.InnerHtml = "<img src='images/badges/" + awayT + ".png'>" ;
317     gameWrap.Controls.Add(awayTClass);
318 }
319 protected void Claim_Click(object sender, EventArgs e)
320 {
321     Debug.WriteLine("in");
322     Button b = (Button)sender;
323     if (b.Attributes["gold"] == "1")
324         PackFunctions.addPack(Session["username"].ToString(), 1);
325     if (b.Attributes["diamond"] == "1")
326         PackFunctions.addPack(Session["username"].ToString(), 2);
327     if (b.Attributes["silver"] == "1")
328         PackFunctions.addPack(Session["username"].ToString(), 3);
329     int mID = Convert.ToInt32(((Button)sender).Attributes["myid"]);
330     if (b.Attributes["winner"] == "t")
331     {
332         double initialsScore = (GameFunctions.getPrecentbyMatchID
333             (mID, Convert.ToInt32(b.Attributes["winside"]))) * 4) / 100f;
334         Debug.WriteLine(GameFunctions.getPrecentbyMatchID(mID,
335             Convert.ToInt32(b.Attributes["winside"])));
336         int rounded = (int)Math.Round(initialsScore);
337         PackFunctions.addPack(Session["username"].ToString(), rounded);
338     }
339     else
340     {
341         double initialsScore = ((100 - GameFunctions.getPrecentbyMatchID(mID,
342             Convert.ToInt32(b.Attributes["winside"]))) * 4) / 100f;
343         int rounded = (int)Math.Round(initialsScore);
344         PackFunctions.addPack(Session["username"].ToString(), rounded);
345     }
346
347     BettingFunctions.claimed(Session["username"].ToString(), mID);
348     ((Button)sender).Enabled = false;
349     //Response.Redirect(Request.RawUrl, false);
350 }

```

```
349     public Root GameByGameID(string gameID)
350     {
351         foreach (Root e in fixs)
352         {
353             if ((e.id).ToString() == gameID)
354             {
355                 return e;
356             }
357         }
358         return new Root();
359     }
360     protected void myBets_Click(object sender, EventArgs e)
361     {
362         ViewState["State"] = "2";
363         state2();
364     }
365
366     void state2()
367     {
368         mfix.Visible = false;
369         bfix.Visible = true;
370
371         lb = BettingFunctions.getAllBets(Session["username"].ToString());
372         foreach (Bet b in lb)
373         {
374             CreateBet(b, GameByGameID(b.gameId));
375         }
376     }
377 }
378
```