

FANTASY CRICKET PREDICTOR

A Project Report

Submitted in partial fulfillment of the
Requirements for the award of the Degree
of

MASTERS OF SCIENCE (COMPUTER SCIENCE)

By

Vishal Omanakuttan
Seat No: 03

Under the esteemed guidance of

Dr. Vivek Sharma



DEPARTMENT OF COMPUTER SCIENCE

NAGINDAS KHANDWALA COLLEGE

(Affiliated to University of Mumbai)

MUMBAI, 400064

MAHARASHTRA

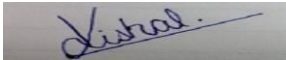
2022-2023

GR No:**89006**.....

Roll no: 03

1. Name of the Student: Vishal Omanakuttan
2. Title of the Project: Fantasy Cricket Predictor
3. Name of the Guide: Dr. Vivek Sharma
4. Teaching/Industry experience of the Guide:
5. Is this your first submission? Yes

Signature of the Student:

A rectangular box containing a handwritten signature in blue ink that reads "Vishal".

Date:

Signature of the Guide:

Date:

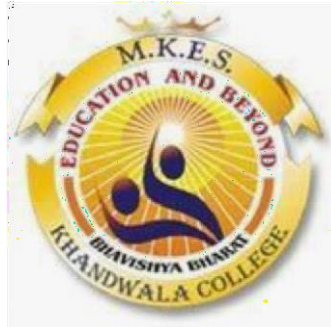
Signature of the Coordinator:

Date:

NAGINDAS KHANDWALA COLLEGE
(Affiliated to University of Mumbai)

MUMBAI-MAHARASHTRA-400068

DEPARTMENT OF COMPUTER SCIENCE



CERTIFICATE

This is to certify that the project titled, "**Fantasy Cricket Predictor**", is bonafide work of **Vishal Omanakuttan** bearing Seat No: (03) submitted in partial fulfillment of the requirements for the award of degree of MASTERS OF SCIENCE in COMPUTER SCIENCE from University of Mumbai.

Internal Guide

Coordinator

External Examiner

Date:

College Seal

ABSTRACT

Cricket is one of the most famous sports around the world. The growing interest in cricket in recent years has given rise to various formats such as T20, T10 from test and ODI formats. All these forms of cricket match madness are present in today's online fantasy cricket league games. In recent years, the fantasy sports platform has taken the Indian gaming landscape by storm and in 2021 he has a valuation of Rs 34600 crore. One of the key aspects of participating in fantasy cricket is team selection. Fantasy Cricket is gaining more users due to its popularity in India. Cricket is also one of the sports that generates a lot of data, so there are many opportunities for data analysis. Fantasy Cricket users must select the right mix of players to maximize points and earn cash rewards. This project describes a retrospective team selection approach that proposes a fantasy cricket team for the next match using real historical data collected from player performance in the last 10 matches. The technique used is linear programming implemented in Python using the Pulp library. Player points are calculated using an exponentially weighted average and then added fantasy cricket constraints using linear programming for team selection.

ACKNOWLEDGEMENT

The project of **“Fantasy Cricket Predictor”** has been a unique experience for me. I am very thankful to the Department of Computer Science Nagindas Khandwala College, who gave us the opportunity to accomplish this project.

I would like to thank all those who guided and associated me in the completion of this project. I express my sincere gratitude to our Head of Department and my project guide Ms. Elizabeth George for her invaluable support and guidance during the course of the project. I am also very much thankful to the “UNIVERSITY OF MUMBAI ” for including the project work as part of the syllabus, without which I would not have gained the experience of developing software as this.

Finally I would like to express my gratitude to my friends for their support and guidance throughout this venture. Last but not the least i would thank my family without whose support, motivation and encouragement this would not have been possible.

Thanking you,

Vishal Omanakuttan

DECLARATION

I hereby declare that the project entitled, “**Fantasy Cricket Predictor**” done at the place **where the project is done**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **MASTERS OF SCIENCE (COMPUTER SCIENCE)** to be submitted as a final semester project as part of our curriculum.

Name and signature of the student

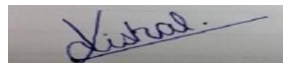
A rectangular box containing a handwritten signature in blue ink. The signature appears to be 'Kishal' followed by a horizontal line.

TABLE OF CONTENTS

Chapter 1: Introduction

- 1.1 Background**
- 1.2 Objectives**
- 1.3 Purpose, Scope and Applicability**
 - 1.3.1 Purpose**
 - 1.3.2 Scope**
 - 1.3.3 Applicability**
- 1.4 Organization of Report**

Chapter 2: System Analysis

- 2.1 Existing System**
- 2.2 Proposed System**
- 2.3 Requirement Analysis**
- 2.4 Hardware Requirements**
- 2.5 Software Requirements**
- 2.6 Justification of selection of Technology**

Chapter 3: System Design

- 3.1 Module Division**
- 3.2 Data Dictionary**
- 3.3 ER Diagrams**
- 3.4 DFD/UML Diagrams**

Chapter 4: Implementation and Testing

- 4.1 Code (Place Core segments)**
- 4.2 Testing Approach**
 - 4.2.1 Unit Testing (Test cases and Test Results)**
 - 4.2.2 Integration System (Test cases and Test**

Results) Chapter 5: Results and Discussions (Output Screens)

Chapter 6: Conclusion and Future Work

Chapter 7: References

Chapter 8: Glossary

List of Tables

1. Table: Event
2. Table: User Information

List of Figures

1. Figure: ER Diagram
2. Figure: UML Use Case Diagram
3. Figure: Activity Diagram
4. Figure: Sequence Diagram

Chapter 1

Introduction

1.1 Background

Fantasy Cricket Predictor as the name suggests is a Fantasy game prediction website. This website predicts a team that proposes a fantasy cricket team for the next match using real historical data collected from player performance in the last 10 matches.

In this website the user has to select Team A and Team B and then click on the predict button to get a list of top 11 players and there are charts which show No. of Batsmen, No. of Bowlers and No. of All-Rounders selected and also total no of players selected from team A and team B.

1.2 Objectives

- Historical Data Driven result of top 11 players in two teams.
- Selecting top performing players from a pool of 30 players.
- Using the models results to earn maximum reward in fantasy cricket games.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

Fantasy Sports is a new and exciting game where players can select fantasy teams from real-world professional sports. The aim is to build a team that can score the most points compared to another team. Fantasy sports allow users to make the best use of their sports knowledge and earn real cash by taking part in fantasy sports leagues. Fantasy cricket predictor uses historical data to predict top 11 performing players and then user can use this result to take part in fantasy cricket games and earn maximum reward.

1.3.2 Scope

The scope of the project is already pressed on a laptop, pc or mobile phone with an internet connection. First the user, need to access the website

After that user has to select two teams and then click on the predict button and the algorithm spits out the Dream team that maximizes the probability of scoring the highest points. In addition to that, it also conveys the estimated points that can be accrued through the selected team. The value would help us check the accuracy of the system.

1.3.3 Applicability

Fantasy Cricket Predictor as the name suggests is a Fantasy game prediction website. This website predicts a team that proposes a fantasy cricket team for the next match using real historical data collected from player performance in the last 10 matches.

1.4 Organization of Report

Fantasy Cricket Predictor is a Web Application which is used to provide a Dream team that maximizes the probability of scoring the highest points. In addition to that, it also conveys the estimated points that can be accrued through the selected team. The value would help us check the accuracy of the system.

Chapter 2

System Analysis

2.1 Existing System

- It doesn't focus on historical data driven approach
- Does not support all the platforms.
- Doesn't show graphical representation of predicted data

2.2 Proposed System

- Predicts the output of top 11 players from the pool of 30 players
- It is easy to use
- Produce Charts which show graphical representation of the data

2.3 Requirement Analysis

Existing system of Fantasy Predictor doesn't show the Graphical Representation of Predicted data.

2.4 Hardware Requirements

The most well-known arrangement of necessities characterized by any working framework or programming application is the physical PC assets, otherwise called equipment. An equipment necessities list is frequently joined by an equipment similarity list (HCL), particularly in the event of working frameworks.

A HCL records tried, similarity and now and again incongruent equipment gadgets for a specific working framework or application. The accompanying sub-segments talk about the different parts of equipment prerequisites

- 2.90Ghz Intel core i3 processor or other compatible
- Minimum 1GB RAM

2.5 Software Requirements

Programming Requirements manage characterizing programming asset necessities and essentials that should be introduced on a PC to give ideal working of an application. These necessities or pre-essentials are commonly excluded from the product establishment bundle and should be introduced independently before the product is introduced

Programming languages: Django, Python, HTML, CSS.

Browser: Google Chrome

Front End Technology: VS Code

Back End Technology: Python

2.6 Justification of selection of Technology

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Chapter 3

System Design

3.1 Module Division

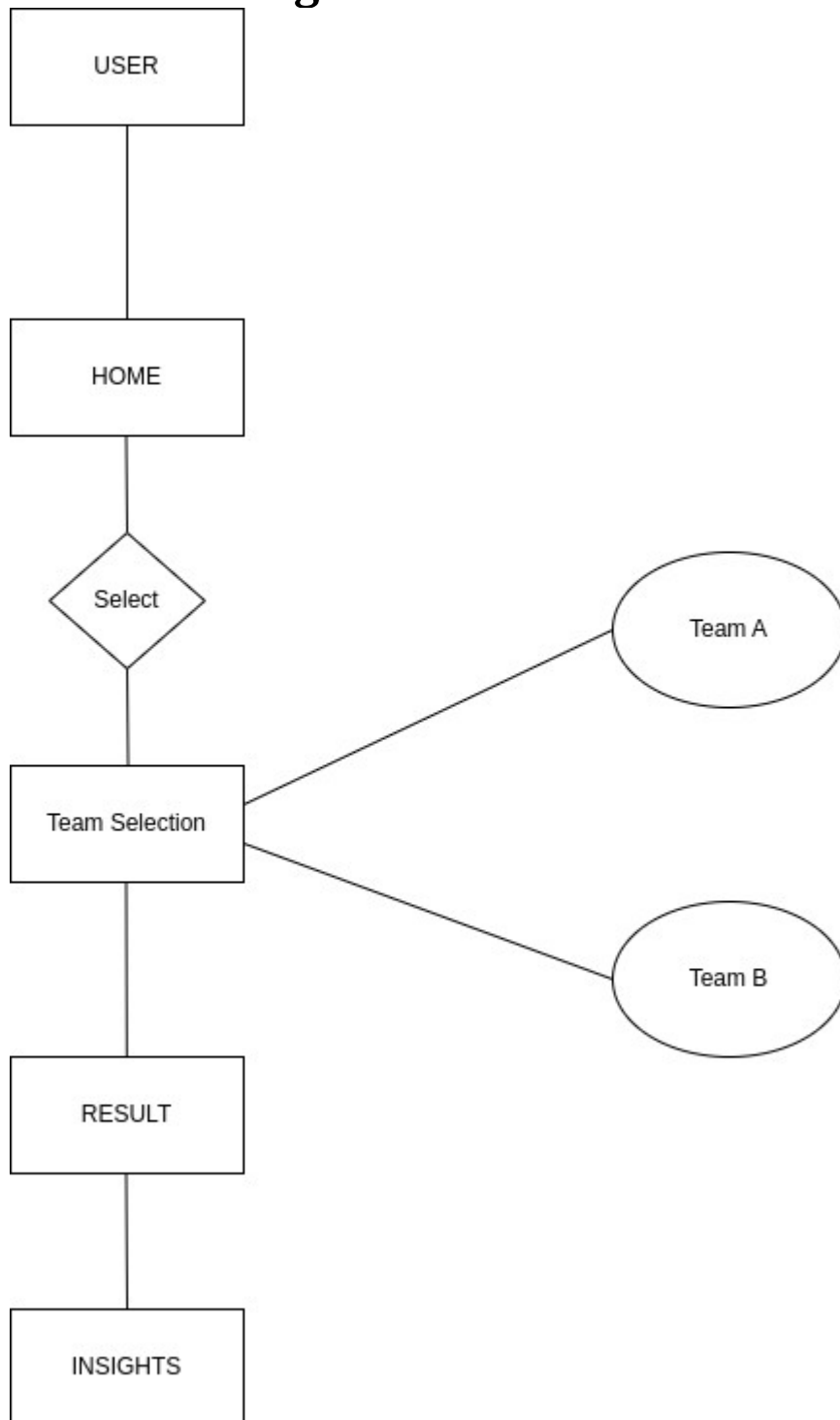
User modules:

1. Insert the name of two teams in the select team section
2. Click on more details to view Chart

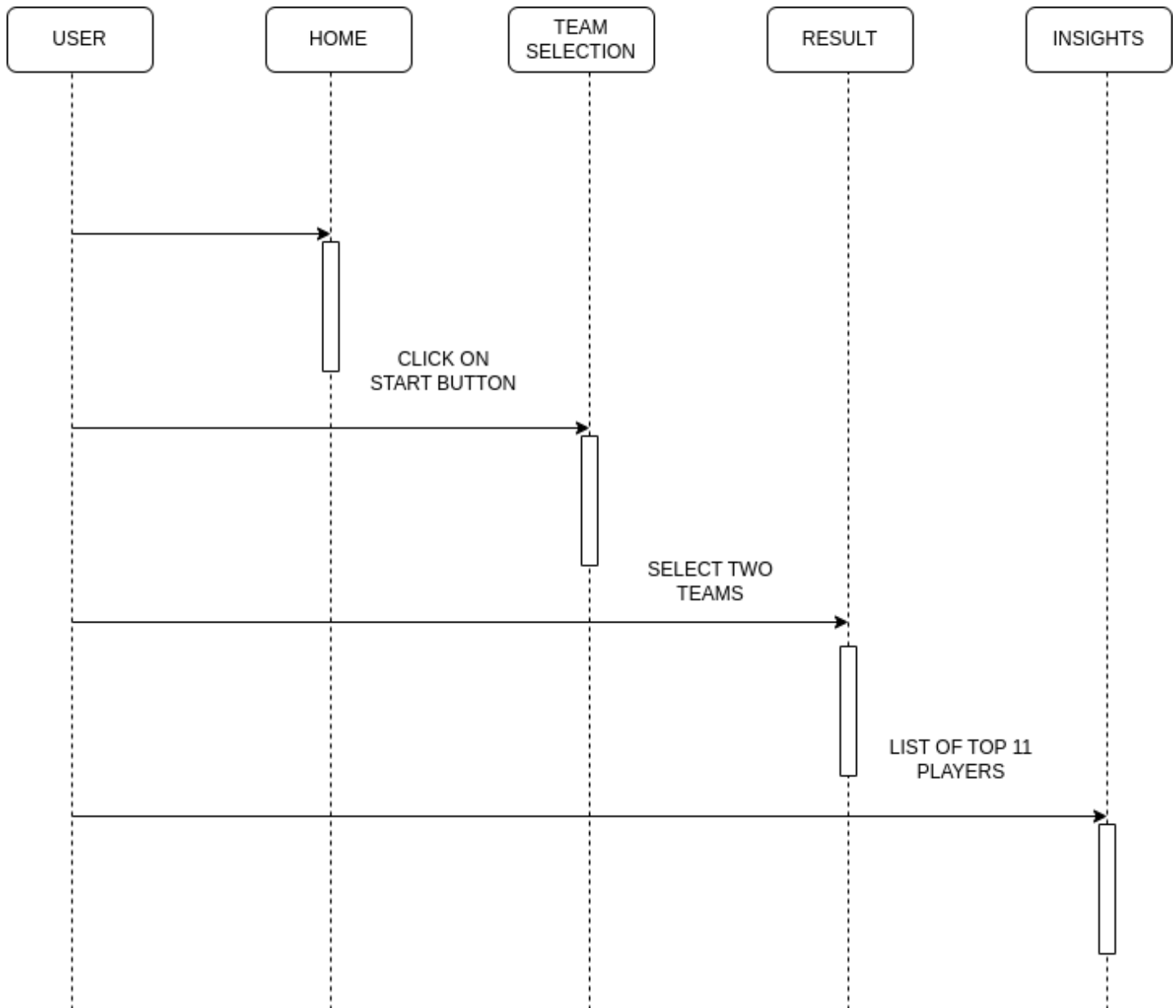
Event Table

| Action | Source | Activity | Response |
|----------------|--------|--------------------------|---|
| Start | User | Team Selection | Takes to Team Selection Page |
| Team Selection | User | Insert name of two teams | Takes to Result page |
| Result Page | User | Click on Charts Button | Show the Graphical Representation of predicted data |

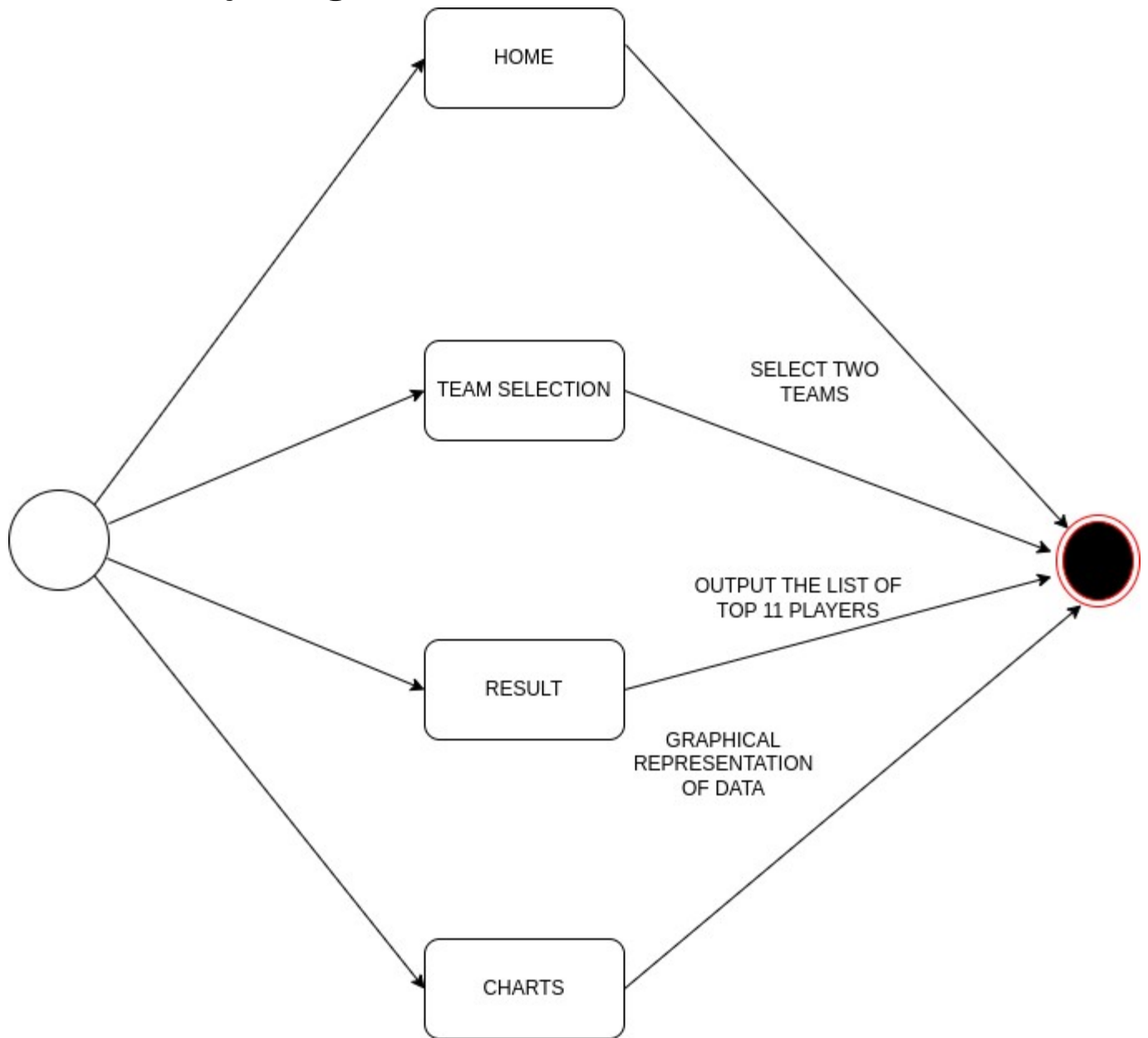
3.2 E-R Diagram



3.3 Sequence Diagram



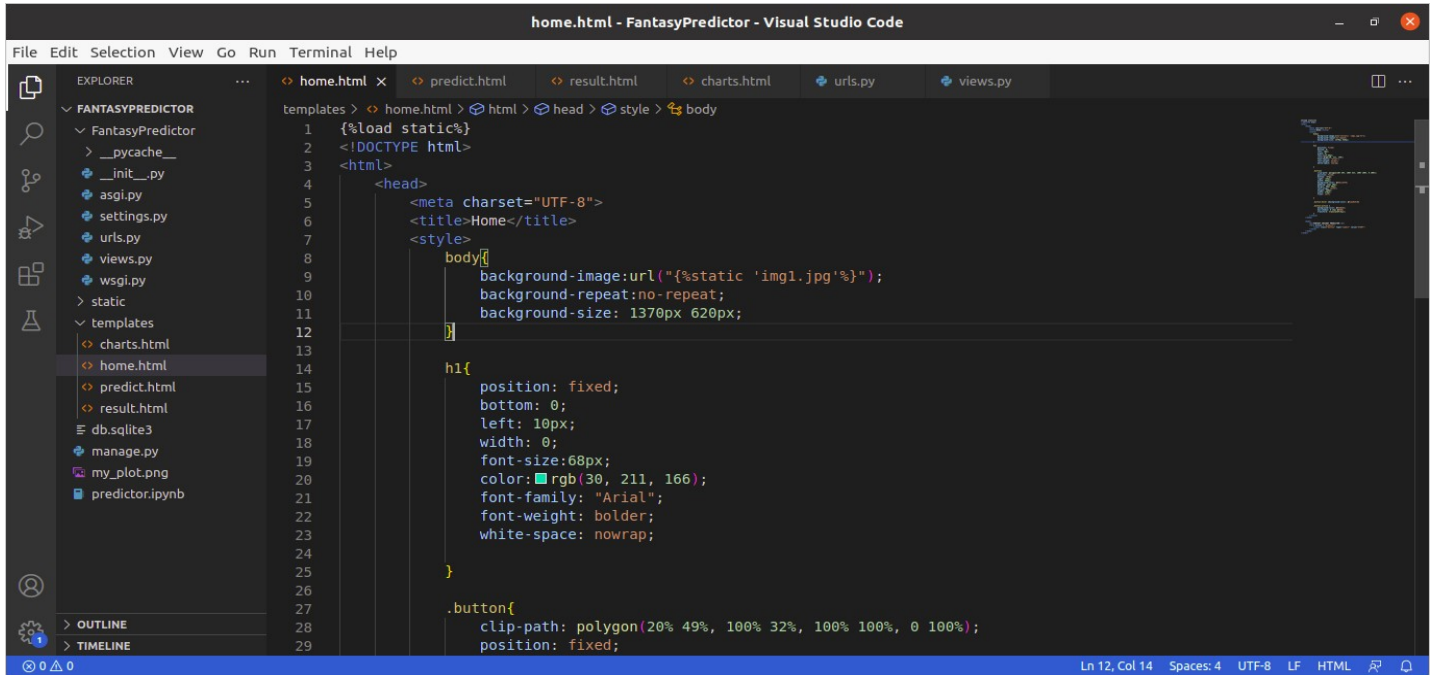
3.6 Activity Diagram



Chapter 4 Implementation and Testing

4.1 Code

home.html



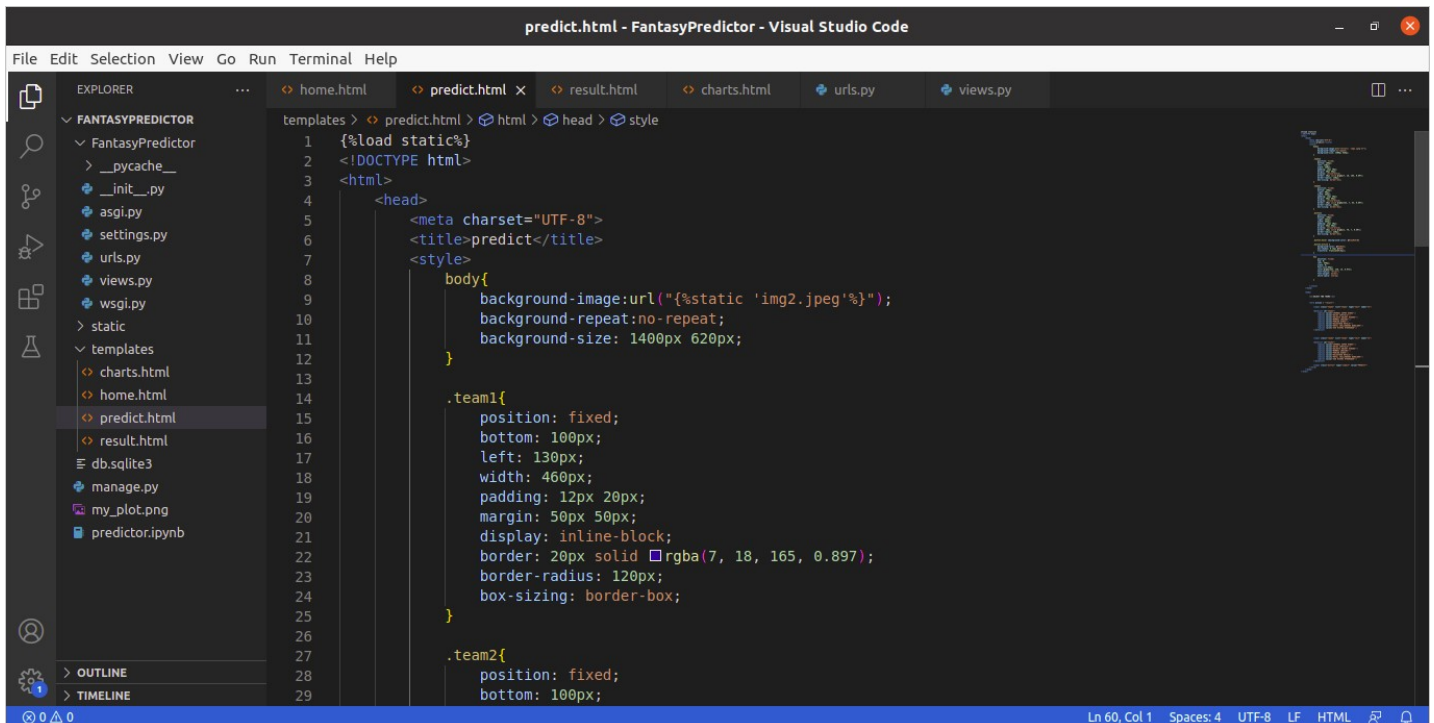
```
home.html - FantasyPredictor - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
FANTASYPREDICTOR
  FantasyPredictor
    __pycache__
    __init__.py
    asgi.py
    settings.py
    urls.py
    views.py
    wsgi.py
    static
    templates
      charts.html
      home.html
      predict.html
      result.html
    db.sqlite3
    manage.py
    my_plot.png
    predictor.ipynb

templates > home.html > html > head > style > body
1 {%load static%}
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title>Home</title>
7     <style>
8       body{
9         background-image:url("{%static 'img1.jpg'%}");
10        background-repeat:no-repeat;
11        background-size: 1370px 620px;
12      }
13
14      h1{
15        position: fixed;
16        bottom: 0;
17        left: 10px;
18        width: 0;
19        font-size:60px;
20        color:rgb(30, 211, 166);
21        font-family: "Arial";
22        font-weight: bolder;
23        white-space: nowrap;
24      }
25
26      .button{
27        clip-path: polygon(20% 49%, 100% 32%, 100% 100%, 0 100%);
28        position: fixed;
29      }

```

predict.html



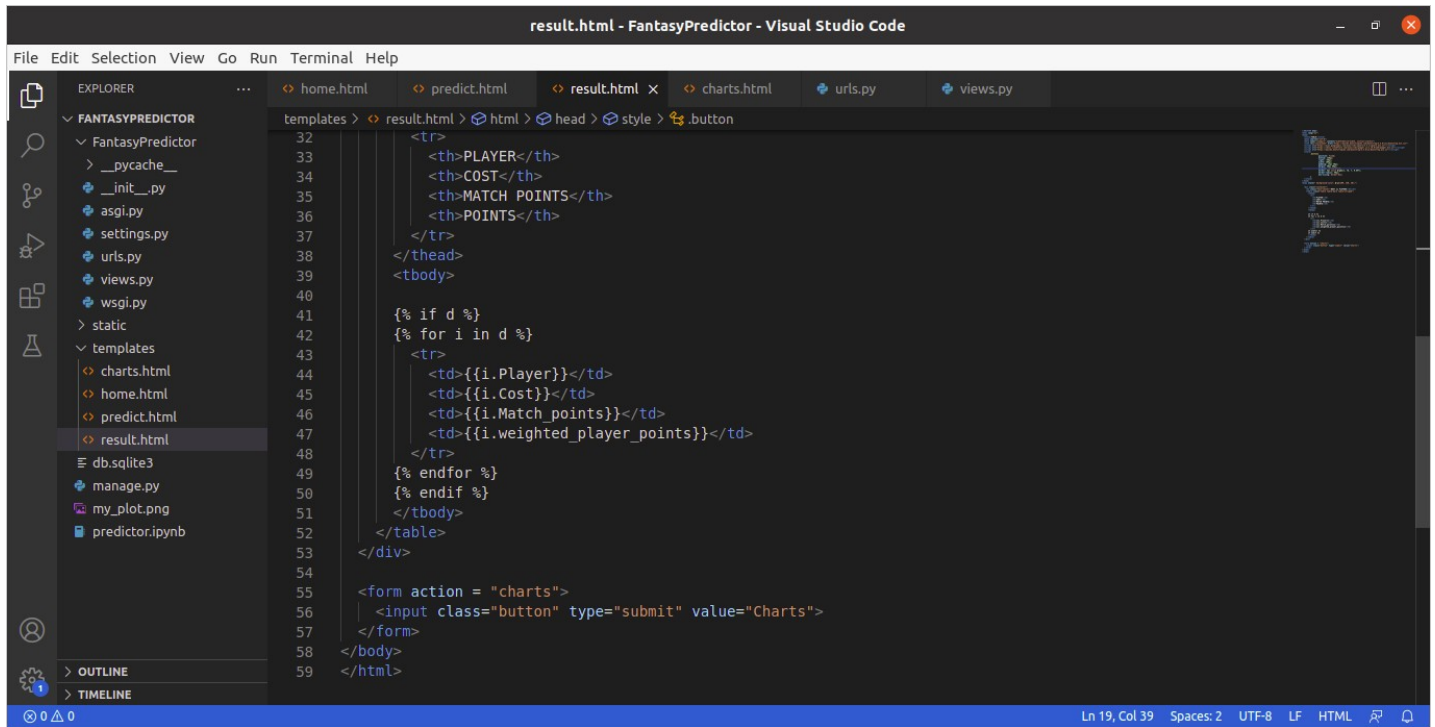
```
predict.html - FantasyPredictor - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
FANTASYPREDICTOR
  FantasyPredictor
    __pycache__
    __init__.py
    asgi.py
    settings.py
    urls.py
    views.py
    wsgi.py
    static
    templates
      charts.html
      home.html
      predict.html
      result.html
    db.sqlite3
    manage.py
    my_plot.png
    predictor.ipynb

templates > predict.html > html > head > style
1 {%load static%}
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title>predict</title>
7     <style>
8       body{
9         background-image:url("{%static 'img2.jpeg'%}");
10        background-repeat:no-repeat;
11        background-size: 1400px 620px;
12      }
13
14      .team1{
15        position: fixed;
16        bottom: 100px;
17        left: 130px;
18        width: 460px;
19        padding: 12px 20px;
20        margin: 50px 50px;
21        display: inline-block;
22        border: 20px solid rgba(7, 18, 165, 0.897);
23        border-radius: 120px;
24        box-sizing: border-box;
25      }
26
27      .team2{
28        position: fixed;
29        bottom: 100px;

```

result.html



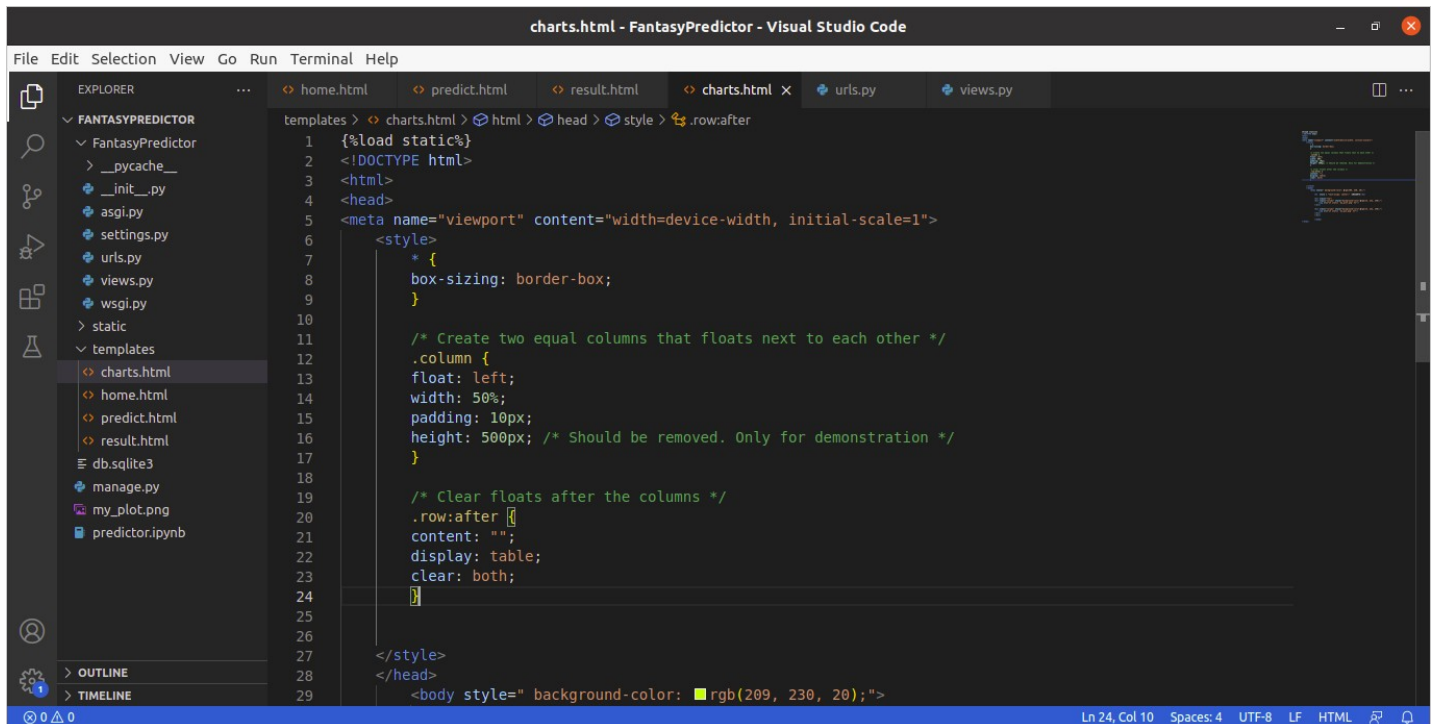
```
File Edit Selection View Go Run Terminal Help
EXPLORER
FANTASYPREDICTOR
  FantasyPredictor
    __pycache__
    __init__.py
    asgi.py
    settings.py
    urls.py
    views.py
    wsgi.py
  static
  templates
    charts.html
    home.html
    predict.html
    result.html
  db.sqlite3
  manage.py
  my_plot.png
  predictor.ipynb
OUTLINE
TIMELINE

templates > result.html > html > head > style > .button
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
<tr>
  <th>PLAYER</th>
  <th>COST</th>
  <th>MATCH_POINTS</th>
  <th>POINTS</th>
</tr>
</thead>
<tbody>

{% if d %}
{% for i in d %}
  <tr>
    <td>{{i.Player}}</td>
    <td>{{i.Cost}}</td>
    <td>{{i.Match_points}}</td>
    <td>{{i.weighted_player_points}}</td>
  </tr>
{% endfor %}
{% endif %}
</tbody>
</table>
</div>

<form action = "charts">
  <input class="button" type="submit" value="Charts">
</form>
</body>
</html>
```

charts.html



```
charts.html - FantasyPredictor - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
FANTASYPREDICTOR
  FantasyPredictor
    __pycache__
    __init__.py
    asgi.py
    settings.py
    urls.py
    views.py
    wsgi.py
  static
  templates
    charts.html
    home.html
    predict.html
    result.html
  db.sqlite3
  manage.py
  my_plot.png
  predictor.ipynb
OUTLINE
TIMELINE

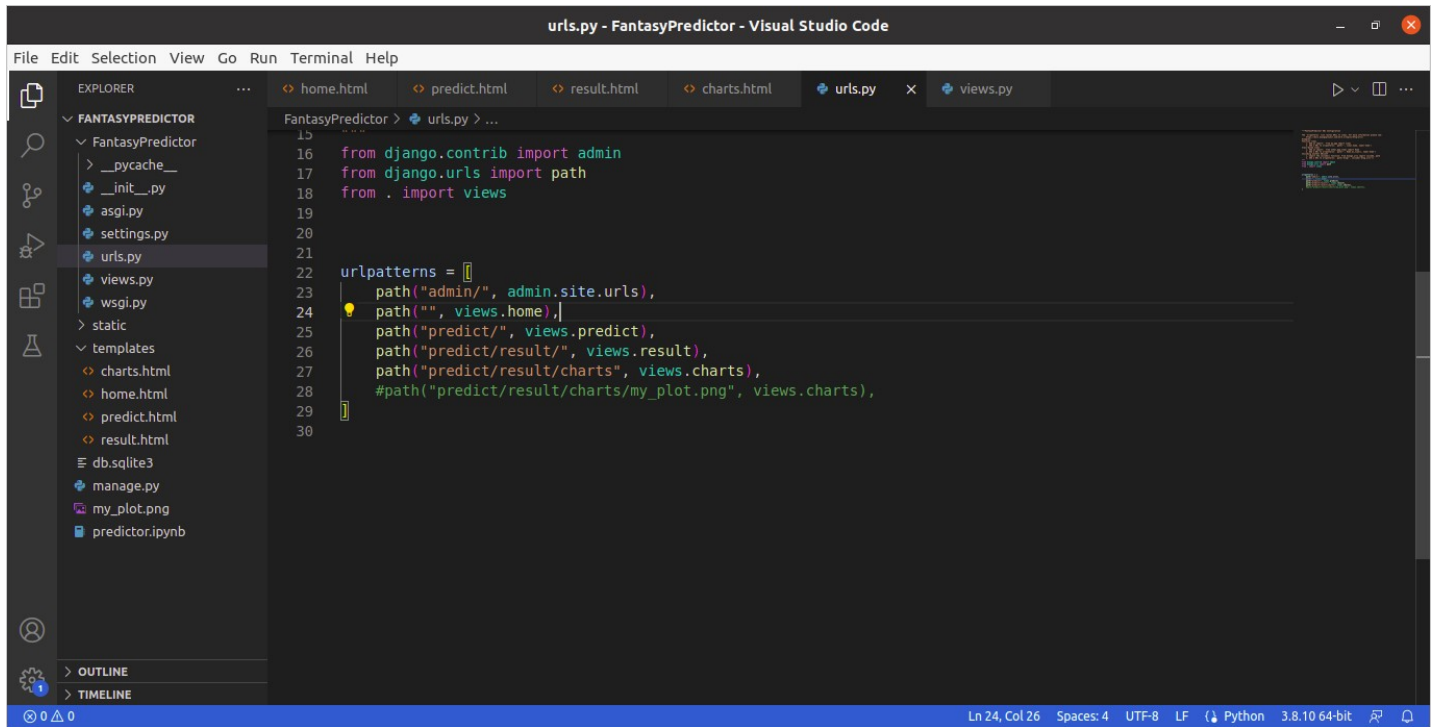
templates > charts.html > html > head > style > .row.after
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
{%load static%}
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    * {
      box-sizing: border-box;
    }

    /* Create two equal columns that floats next to each other */
    .column {
      float: left;
      width: 50%;
      padding: 10px;
      height: 500px; /* Should be removed. Only for demonstration */
    }

    /* Clear floats after the columns */
    .row:after {
      content: "";
      display: table;
      clear: both;
    }

  </style>
</head>
<body style=" background-color: #c0c0c0;">
```

urls.py

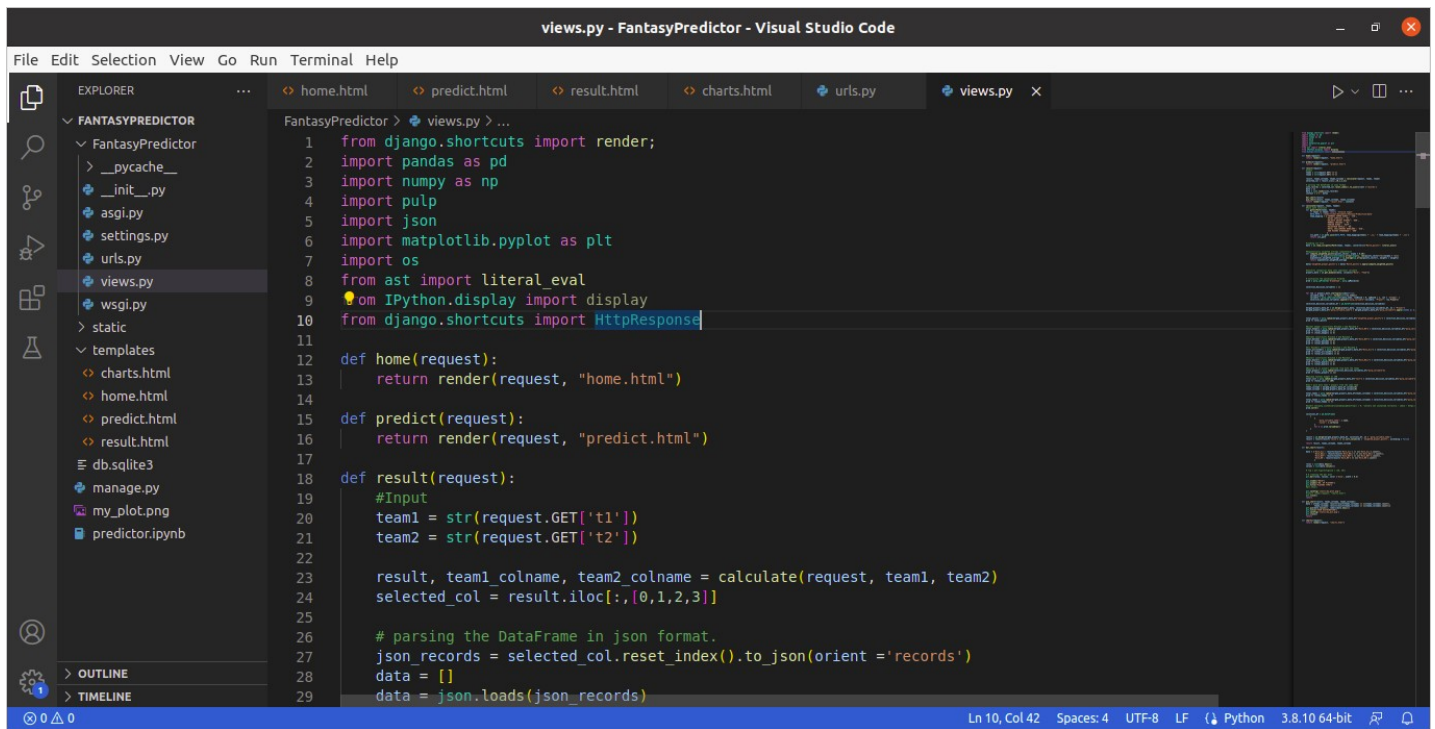


The screenshot shows the Visual Studio Code editor with the 'urls.py' file open. The Explorer sidebar on the left shows the project structure for 'FANTASYPREDICTOR', including files like __pycache__, __init__.py, asgi.py, settings.py, urls.py, views.py, wsgi.py, static, templates, charts.html, home.html, predict.html, result.html, db.sqlite3, manage.py, my_plot.png, and predictor.ipynb. The main editor area displays the following Python code:

```
15
16 from django.contrib import admin
17 from django.urls import path
18 from . import views
19
20
21 urlpatterns = [
22     path("admin/", admin.site.urls),
23     path("", views.home),
24     path("predict/", views.predict),
25     path("predict/result/", views.result),
26     path("predict/result/charts", views.charts),
27     #path("predict/result/charts/my_plot.png", views.charts),
28 ]
29
30
```

The status bar at the bottom indicates the cursor is at Line 24, Column 26, with 4 spaces, UTF-8 encoding, LF line endings, Python 3.8.10 64-bit.

views.py



The screenshot shows the Visual Studio Code editor with the 'views.py' file open. The Explorer sidebar on the left shows the project structure for 'FANTASYPREDICTOR'. The main editor area displays the following Python code:

```
1 from django.shortcuts import render;
2 import pandas as pd
3 import numpy as np
4 import pulp
5 import json
6 import matplotlib.pyplot as plt
7 import os
8 from ast import literal_eval
9 from IPython.display import display
10 from django.shortcuts import HttpResponseRedirect
11
12 def home(request):
13     return render(request, "home.html")
14
15 def predict(request):
16     return render(request, "predict.html")
17
18 def result(request):
19     #Input
20     team1 = str(request.GET['t1'])
21     team2 = str(request.GET['t2'])
22
23     result, team1_colname, team2_colname = calculate(request, team1, team2)
24     selected_col = result.iloc[:, [0,1,2,3]]
25
26     # parsing the DataFrame in json format.
27     json_records = selected_col.reset_index().to_json(orient='records')
28     data = []
29     data = json.loads(json_records)
```

The status bar at the bottom indicates the cursor is at Line 10, Column 42, with 4 spaces, UTF-8 encoding, LF line endings, Python 3.8.10 64-bit.


```
views.py - FantasyPredictor - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
FantasyPredictor
  > __pycache__
  > __init__.py
  asgi.py
  settings.py
  urls.py
  views.py
  wsgi.py
  static
  templates
    charts.html
    home.html
    predict.html
    result.html
  db.sqlite3
  manage.py
  my_plot.png
  predictor.ipynb

FantasyPredictor > views.py > ...
35
36 def calculate(request, team1, team2):
37     #Data path selection
38     def getCsvPath(team1, team2):
39         if team1 == team2: return 'Invalid Input'
40         DATA_PATH = '/home/vishal/Documents/Fantasy-Prediction/data'
41         team_mapping = {'CHENNAI SUPER KINGS': 'CSK',
42                         'DELHI CAPITALS': 'DC',
43                         'KOLKATA KNIGHT RIDERS': 'KKR',
44                         'MUMBAI INDIANS': 'MI',
45                         'PUNJAB KINGS': 'KXIP',
46                         'RAJASTHAN ROYALS': 'RR',
47                         'ROYAL CHALLENGERS BANGLORE': 'RCB',
48                         'SUN RISERS HYDERABAD': 'SRH'
49                         }
50         csv_path = os.path.join(DATA_PATH, team_mapping[team1] + '_vs_' + team_mapping[team2] + '.csv')
51         return csv_path
52
53     #Loading csv file
54     data = pd.read_csv(getCsvPath(team1, team2), converters={'Match_points': literal_eval})
55
56
57
58     #Exponentially weighted average computataion
59     def compute_weighted_points(points_vector, alpha = 0.20):
60         weights = np.exp(list(reversed(np.array(range(1, len(points_vector)+1))*alpha * -1)))
61         exponential_weighted_average = np.average(np.array(points_vector), weights = weights)
62         return exponential_weighted_average
63
```

```
views.py - FantasyPredictor - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
FantasyPredictor
  > __pycache__
  > __init__.py
  asgi.py
  settings.py
  urls.py
  views.py
  wsgi.py
  static
  templates
    charts.html
    home.html
    predict.html
    result.html
  db.sqlite3
  manage.py
  my_plot.png
  predictor.ipynb

FantasyPredictor > views.py > ...
69
70
71 # Initialize the optimization Problem
72 prob = pulp.LpProblem('Dreamteam', pulp.LpMaximize)
73
74
75 selection_decision_variables = []
76
77
78 for row in players_data.itertuples(index=True):
79     variable_name = 'x_{}'.format(str(row.Index))
80     variable = pulp.LpVariable(variable_name, lowBound = 0, upBound = 1, cat = 'Integer' )
81     selection_decision_variables.append({"pulp_variable":variable, "Player": row.Player})
82
83 selection_decision_variables_df = pd.DataFrame(selection_decision_variables)
84
85 merged_players_data_df = pd.merge(players_data, selection_decision_variables_df, on = "Player")
86 merged_players_data_df["pulp_variable_name"] = merged_players_data_df["pulp_variable"].apply(lambda x: x)
87
88
89
90 total_points = pulp.lpSum(merged_players_data_df["weighted_player_points"] * selection_decision_variables)
91 prob += total_points
92
93
94 #Wicket keeper constraints Minimum 1 and Maximum 4
95 total_keepers = pulp.lpSum(merged_players_data_df["Role_WK"] * selection_decision_variables)
96 prob += (total_keepers >= 1)
97 prob += (total_keepers <= 4)
98
```

4.2 Testing Approach

Testing Approach is one of the basic steps, which is performed in the early stages. Most of the testers prefer performing to check if a specific unit of code is functional or not. Unit Testing is one of the common steps performed for every activity because it helps in removing basic and simple errors.

4.1.1 Unit Testing

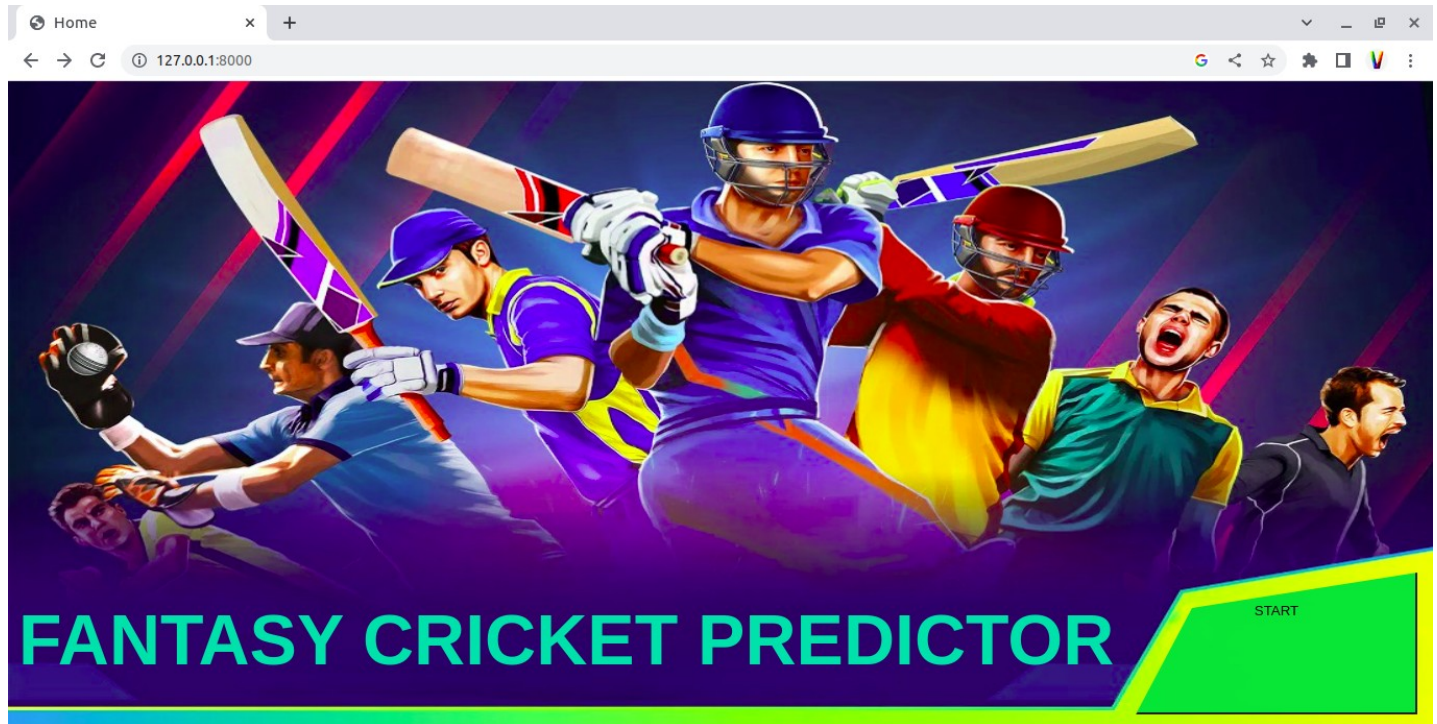
| Sr. No | Action | Inputs | Expected Output | Actual Output | Test Result | Test Comments |
|--------|----------------|--------|-----------------|---------------|-------------|--|
| 1 | Access Website | NA | Home Page | Home Page | Pass | [Stalin 07/11/2022 11:45 AM]: Launch successful |

4.1.2 Integration Testing

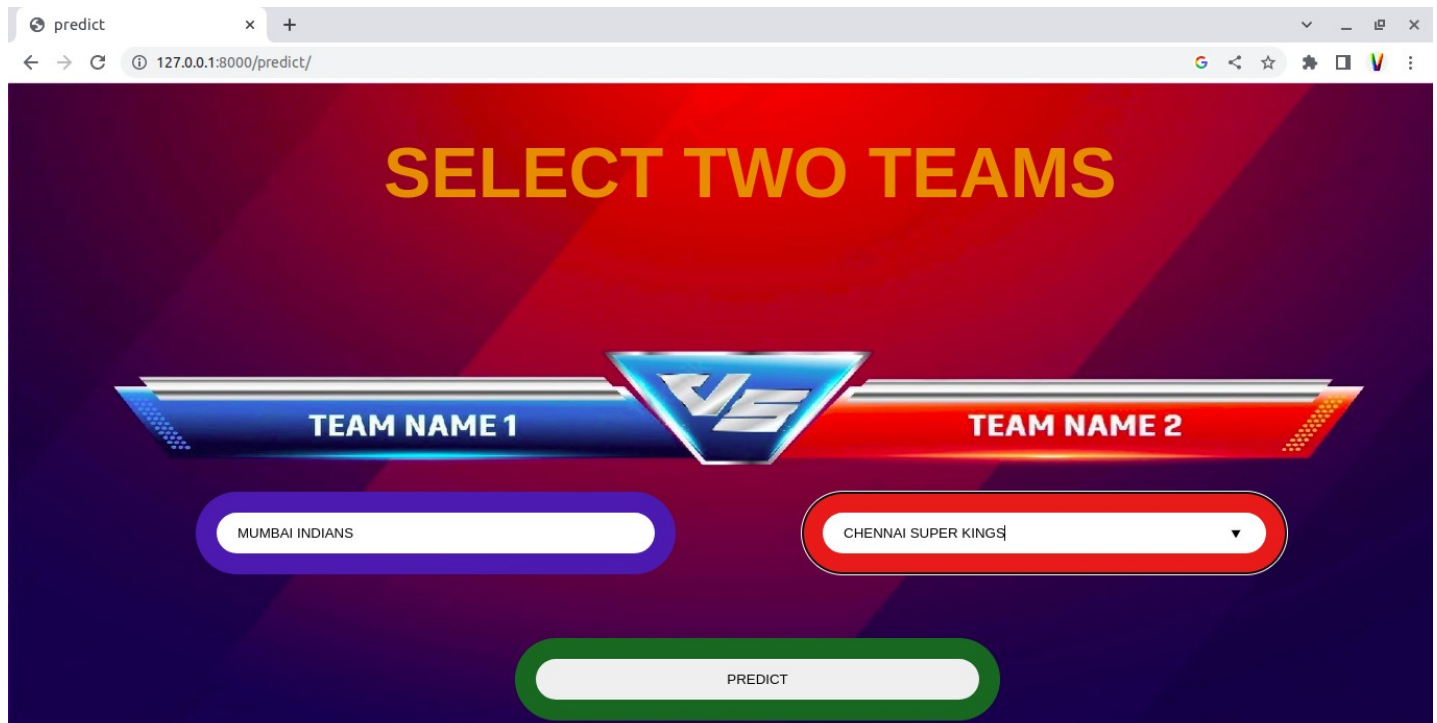
| Sr. No | Action | Inputs | Expected Output | Actual Output | Test Result | Test Comments |
|--------|---------------------|---|---------------------|---------------------|-------------|--|
| 1 | Access Website | NA | Home Page | Home Page | Pass | [Stalin 07/11/2022 11:44 AM]: Launch successful |
| 2 | Home Page | Click on start button | Team Selection Page | Team Selection Page | pass | [Stalin 07/11/2022 11:44 AM]: Launch successful |
| 3 | Team Selection Page | Add two teams name and click on predict | Result Page | Result Page | pass | [[Stalin 07/11/2022 11:44 AM]: Launch successful |
| 4 | Result Page | Click on Charts button | Insights Page | Insights Page | pass | [[Stalin 07/11/2022 11:44 AM]: Launch successful |

Chapter 5: Results and Discussions

Home Page



Select Teams



Result Page

TABLE

127.0.0.1:8000/predict/result?t1=MUMBAI+INDIANS&t2=CHENNAI+SUPER+KINGS

Google, Back, Star, Extensions, V, More

BEST 11 PLAYERS

| PLAYER | COST | MATCH POINTS | POINTS |
|-----------------|------|---|---------------|
| Rituraj Gaikwad | 8.0 | [8, 2, 2, 103, 102, 90] | 64.1992919263 |
| Bumrah | 10.0 | [30, 54, 0, 58, 50, 114, 4, 31, 81, 54, 2, 93, 95] | 60.2364264499 |
| Ngidi | 8.0 | [77, 23, 54, 77] | 58.8859178187 |
| De Kock | 9.0 | [42, 33, 19, 10, 99, 56, 75, 121, 80, 95, 12, 36, 60, 35] | 53.8755992606 |
| Ishan Kishan | 9.5 | [131, 35, 48, 2, 38, 4, 12, 96, 47, 34, 98, 42] | 51.1416086744 |
| Boult | 9.5 | [29, 62, 54, 33, 54, 54, 35, 29, 6, 124, 2, 35, 81] | 48.8430735923 |
| S.Yadav | 9.0 | [23, 59, 2, 32, 37, 106, 79, 23, 2, 20, 50, 115, 25, 45] | 47.3360021712 |
| Du Plessis | 9.0 | [100, 107, 51, 38, 112, 40, 20, 2, 96, 15, 5, 51, 60] | 45.0020803392 |
| Sam Curran | 8.5 | [68, 101, 30, 25, 8, 74, 21, 77, 35, 29, 72, 79, 26, 4] | 40.3800987822 |
| Rayudu | 8.5 | [95, 19, 12, 37, 50, 52, 66, 17, 6, 50, 57, 36] | 40.257306829 |

Charts

Insights Page

127.0.0.1:8000/predict/re: x

127.0.0.1:8000/predict/result/charts?

Google, Back, Star, Extensions, V, More

INSIGHTS

PLAYERS TYPE

| ROLE | NO. OF PLAYERS |
|-----------|----------------|
| Role_ALL | 2 |
| Role_BALL | 3 |
| Role_BAT | 4 |
| Role_WK | 2 |

TEAM DISTRIBUTION

| Team | Count |
|----------|-------|
| Team_CSK | 4 |
| Team_MI | 6 |

CHAPTER 6

CONCLUSION AND FUTURE WORK

Fantasy Cricket Predictor website is completely automated and spits out the Dream team that maximizes the probability of scoring the highest points. In addition to that, it also conveys the estimated points that can be accrued through the selected team. The value would help us check the accuracy of the system. The algorithm is sensitive to player performance trends and it adjusts accordingly.

Chapter 7: Reference

YouTube: www.youtube.com

Stackoverflow: www.stackoverflow.com

W3Schools: www.w3schools.com

Geeksforgeeks: www.geeksforgeeks.com

Chapter 8: Glossary

ER Diagram - An **Entity Relationship (ER) Diagram** is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

Activity Diagram - An **activity diagram** visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow **diagram**.

Use Case Diagram - Use case diagrams consists of **actors**, use cases and their **relationships**. The diagram is used to model the system/subsystem of an application

Sequence Diagram - A **sequence diagram** is a type of interaction **diagram** because it describes how—and in what order—a group of objects works together.

Html - **HTML** stands for Hyper Text Markup Language. **HTML** is the standard markup language for creating Web pages

Css - **CSS** is the language for describing the presentation of Web pages, including colors, layout, and fonts

Django - Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.