Lab 4 : CS 387 Spring 2022 - IITCricInfo
IPL Web Application using Node.js, Postgres and Angular

Team Size : 2

Deadline : 5th February, 2022

Overview: Frontend (Angular/ReactJs you can use any) applications connecting to a Node.js backend server with connectiservervity to postgresql database

This lab is meant to expose you to technologies that many of you will use in your projects and beyond as you enter the industry. Doing this assignment will help you develop reusable code modules that you can employ directly in your projects as well.

Note : We will use lab3 IPL Schema and data for this lab.

Section A - Learning NodeJs Basics, React and Angular Basics

Getting Started:

1. **Node.js:** You can learn more about Node.js here.

   Node.js Setup:
   For Node.js development it is recommended to use Virtual Studio Code or Intellij Idea Ultimate. You don't have to use these, you can do it just as well with a text editor such as VIM or Emacs but these IDEs make life simpler.

   Installation Instructions:
   > sudo apt update
   > sudo apt install nodejs
   > nodejs -v (Confirm that the installation was successful)
   > sudo apt install npm
   > mkdir IITCricInfo && cd IITCricInfo // this is for the specific application you will build for this assignment.
   > npm init
   > npm -v // (Verify the installed version)
   If successful, you will see the version number of npm installed

   Node.js Modules Installations: here
   Node.js Hello World: here
   Node.js Callbacks: here

Node.js Express: here
Node.js module to interact with postgres: here

Frontend(NOT mandated to use only these frameworks) :
EJS: here
Angular with Nodejs: here
React with Nodejs: here
Data visualization: chart.js, victory

Section B
Match Information and Statistics

1. **List the matches in order of year (recent first) as a table, one match per row:**
   CricBuzz example to show matches for a given season

   Hint : Use a server side cursor to show 10 entries per page and next button.
   you can pass this information either as a query parameter in link
   Example /matches/?skip=0&limit=10
   Or you can pass this information as a parameter for get query/
   URL : /matches

   Each element in the list will contain basic details of the match and the link to details of the match.
   Match Name or complete entry should be a hyperlink to the details of the match

Team1 vs Team2 Stadium name, city name Result

   When you click on the above link it should go to a page displaying match info
   URL: /matches/match_id

   The page should contain 3 sections
   1. scorecard.
   2. score comparison.
   3. summary

2. **Scorecard : It should display a score card for a match with a id taken from [Cricbuzz ScoreCard Example]**

There are 3 sections in the scorecard : inning1, inning2 and match_info.
For each inning section you have to create 2 tables as shown below.
(Note : You can Ignore Fall of wickets, powerplay)
Each player should have a hyperlink on their name, clicking on it should land to the player details page.

Innings Sections:
First innings (name of team batting first) :
Batting:

Batter	Runs	fours	sixes	Balls faced
xyz	10	1	0	15

Extras: //total extras in that innings
Total: //Total score (wickets)

Bowling:

Bowler	Balls Bowled	runs_given	wickets
abc	18	41	0

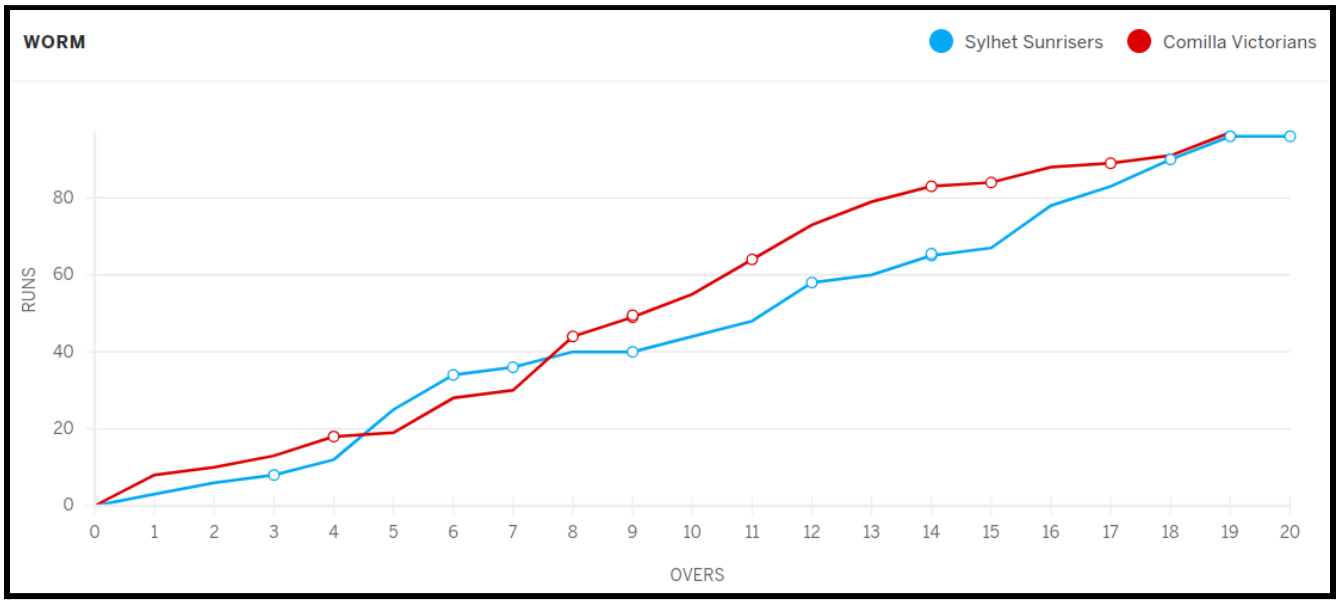
Batter/Bowler: name of the batter/bowler
Runs : Total Runs scored by the batter in the match
Fours/Sixes: Number of Fours/sixes in the match
Balls_faced: Number of balls faced
Balls_bowler: Number of balls bowled
Runs_given: total runs scored by any batter when the bowler has bowled.
Wickets: Total number of wickets taken in the match.

match_info section:
Add only the following details:
*/
Match : match_id, team1_name vs team2_name, year of playing
Toss:
Venue:

Umpires :
Playing XI of team1:
Playing XI of team2:
*/
Need not add any hyperlinks here.

3. **Score Comparison: On selecting a specific match and score comparison button, display a score comparison for each match for each ball . (20 marks)**

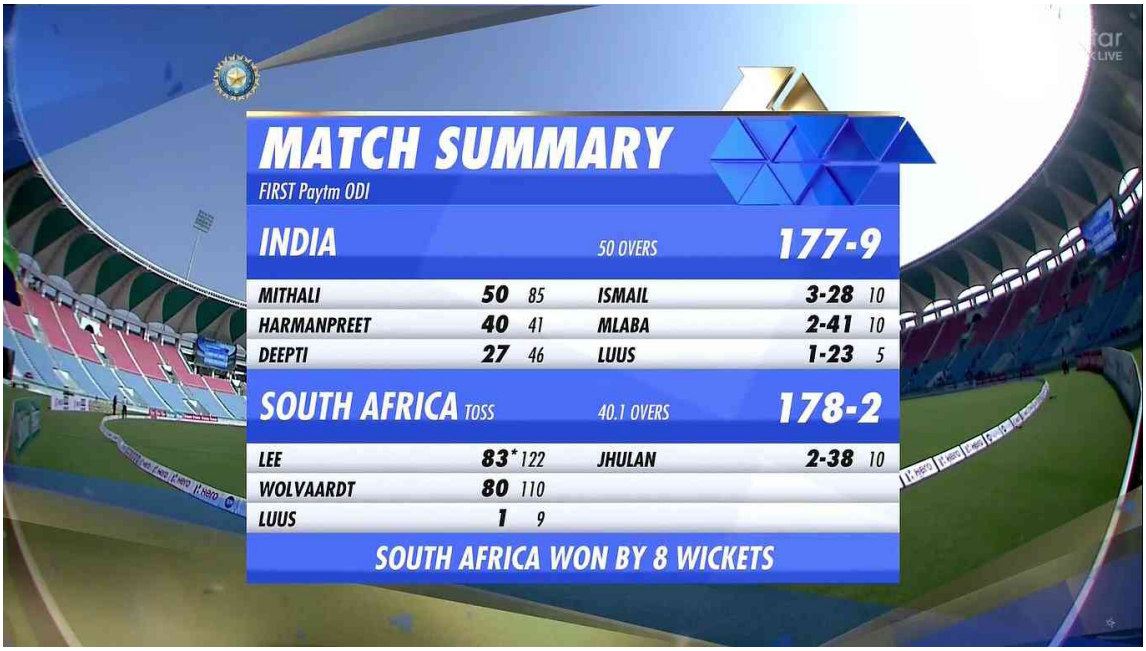
   Y axis : Runs
   x axis: Number of overs completed
   The graph should display line changes after each ball. **Highlight the points when a wicket goes.** Your graph should be very similar to the picture below. You should add the legends without fail. At the bottom you should add the line which team won by what margin or how many no of wickets as given in the image



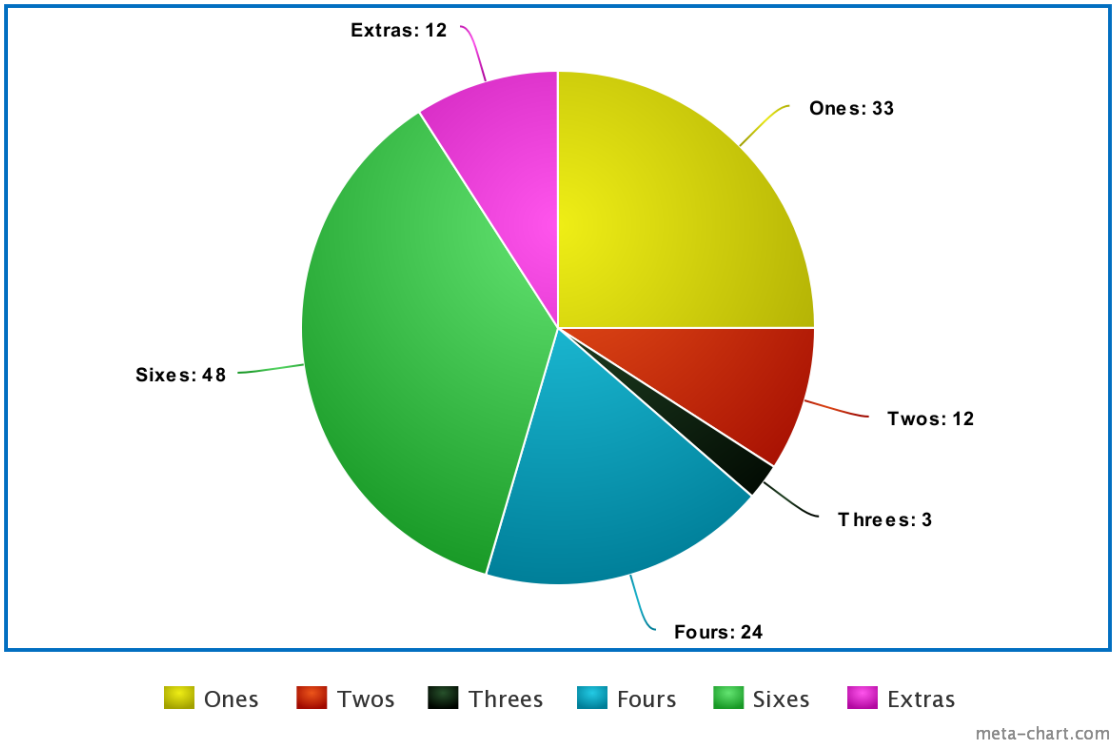
4. **Match Summary: On selecting a specific match and summary button, display a summary for each match.**

   The summary should be similar to the below table given in the image. **(instead of showing First Paytm ODI, display "match_id", IPL, "season_year").**
   Top 3 batters' names should be selected in the order of highest score (Tie breakers: lowest number of balls first, alphabetical order first).A batter should have played at least one ball to be displayed in the summary.

Top 3 bowlers should be in order with the highest number of wickets. A bowler should have taken at least one wicket to be displayed in the summary (Tie breakers: lowest runs given first, alphabetical order first).
Each player name displayed above should have a hyperlink which goes to the player_inforamtion page.



b) Display a pie chart for both the teams showing the percentage of runs scored in fours, sixes, 1s, 2s and extras. A sample diagram down here

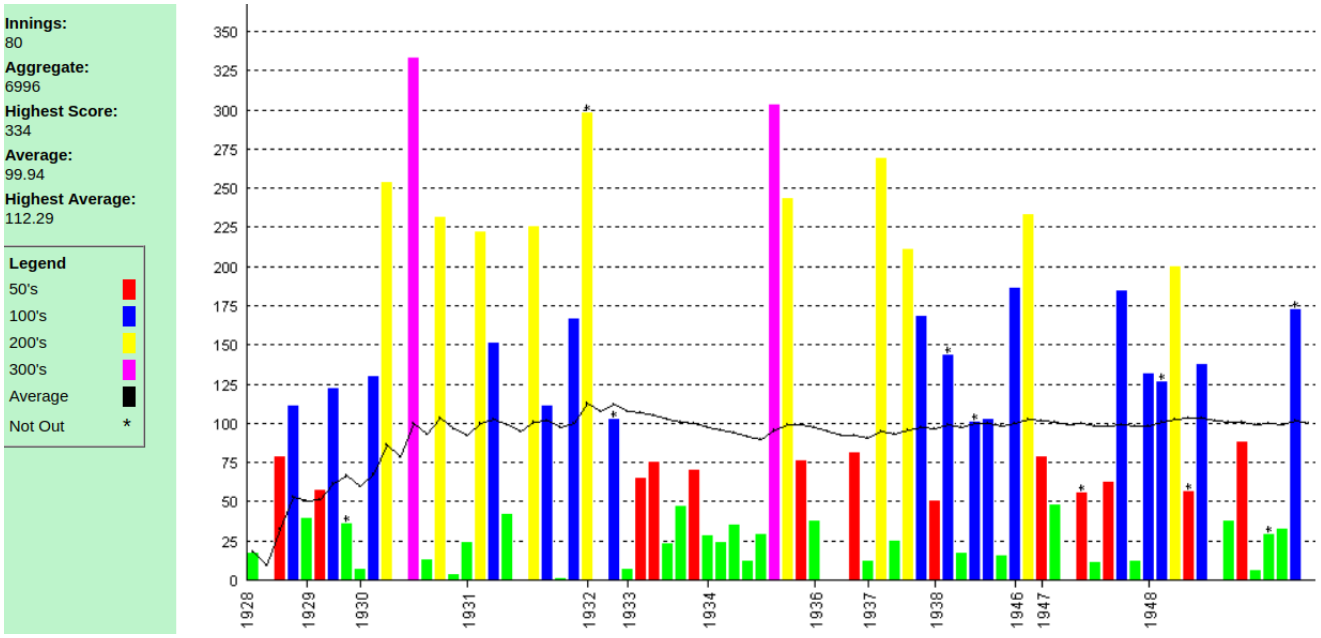


Section C
Player Information and Statistics

Player statistics (/players/<player_id>)

This page will display player information and statistics about his IPL career. It has three section :

a. **Player Basic Information**
   Player Name
   Country
   Batting Style(batting_skill field)
   Bowling Skill
b. **Player Batting Statistics**
   Hint : https://stackblitz.com/edit/ng2-charts-bar-colors



Display the following information about a player's batting career.

1. **Matches :** No of matches a player has played
2. **Runs :** Total runs scored by player
3. **Four :** No of Runs Scored in 4
4. **Six :** No of Runs Scored in 6
5. **Fifty:** No of fifty Scored
6. **HS :** Highest Score
7. **Strike Rate :** average number of runs scored per 100 balls faced.
8. **Average :** The total number of runs they have scored divided by the number of times they have been out

Information in graph : Runs Scored in each match used different colors for different ranges of total.

X -axis : Match Id

Y-axis : Runs in given match

Choose different color for runs < 20, >= 30 and <=50 and > 50

c. Player Bowling Statistics

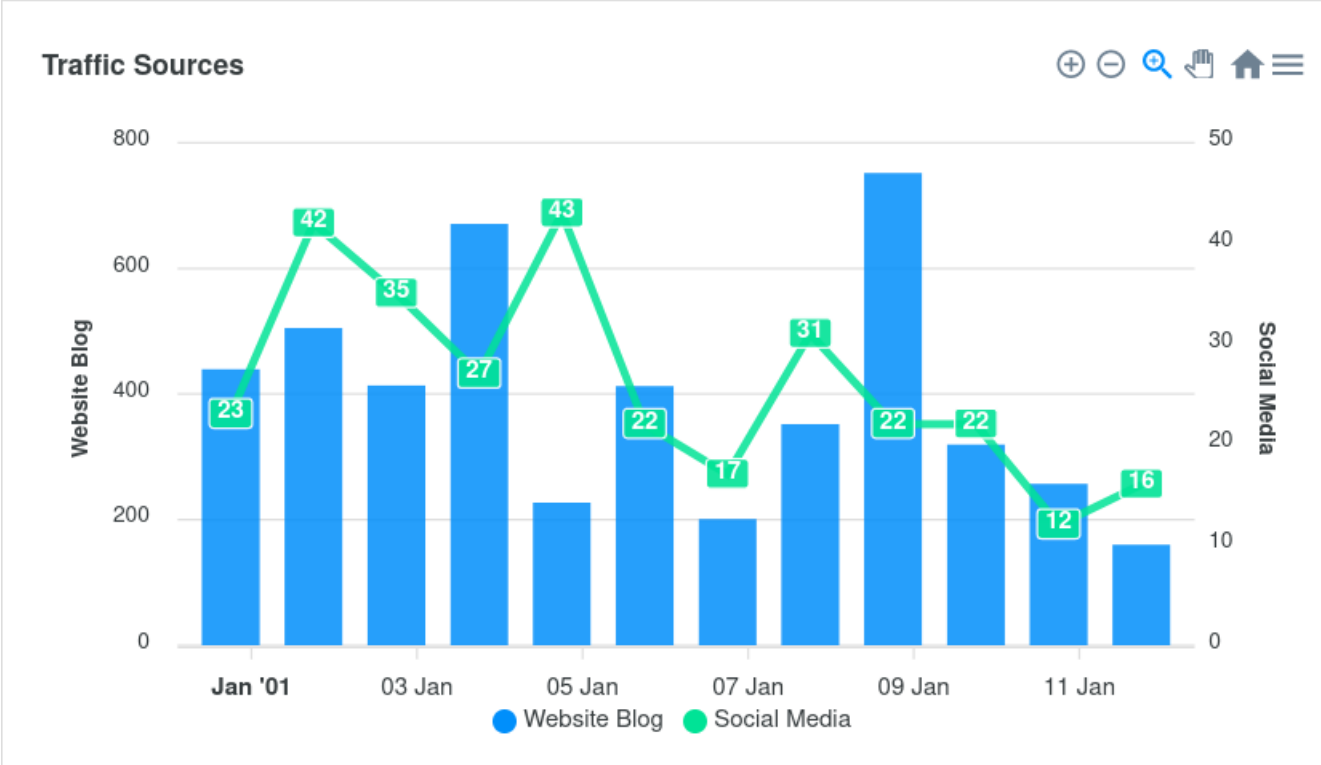
Generate the following table and graph that contains information about player's bowling career :

Matches	Runs	wickets	overs	balls	Economy	Five Wickets

Matches : Number of matches a player has bowled in.
Runs : Runs Conceded by the bowler
Balls : Total number of balls bowled
Overs : Total number of overs bowled
Wickets : Total wickets taken by the bowler
Economy : The average number of runs conceded per over (Econ = Runs/Overs bowled).
Five Wickets : The number of times in which the bowler took at least five wickets

Generate Bar Graph with stack line graph

[Hint Angular Example to create stacked line and bar graph]



Y-axis :

Line graph : wickets taken in given match

Bar graph : Runs Conceded in given match

X-axis : Match_id

Section D :
Season Points Table

Url : /pointstable/<year>

This Endpoint will show IPLpoints table for a given year
Refer : **Cribuzz IPL PointsTable Example**

Team Name	Mat	Won	Lost	Tied	NR	pts

For every match won give 2 points. Table must be sorted by the number of points.

The NRR Formula : [(Total runs scored by team)/(Total overs played by team)] – [(Total runs scored against team)/(Total overs played against team)]

+ve and –ve run rate:

NRR signs must be mentioned. If a team's run rate is higher than their opposition then their NRR will be positive. On the other hand, if a team's run rate is lower than their opposition then their NRR will be negative.

How to calculate NRR in the entire tournament:
Now to calculate a team's NRR in a tournament, all you need to do is find out their NRR in every match in that competition and add them

Section E :
Venue Stats

1. **Getting started Venue Landing Page (/venues).**

   You will create a landing page which will show a table containing names for all the venues. Each Element in the table will show the name of the venue- on clicking on that item it will open a page which will show all the information related to the venue like venue name, address, Capacity, total matches played.

2. Venue statistics (/venue/<venue_id)

This page will contain three sections:

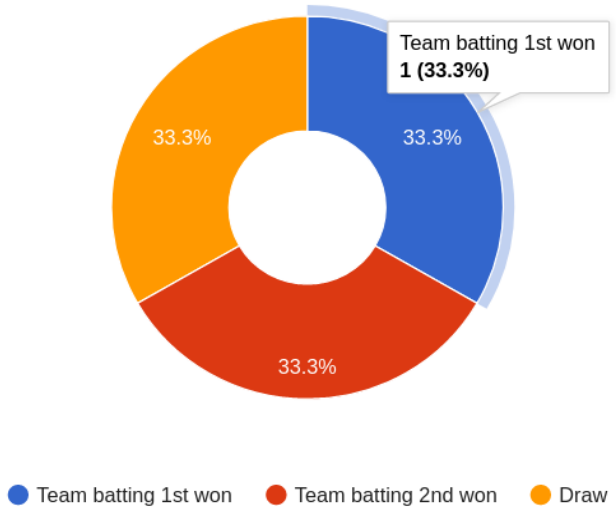
a). Basic Information

Venue name
Address
Capacity
Total matches played
Highest total recorded
Lowest total recorded
Highest score chased

b) Outlines of matches held at a given venue

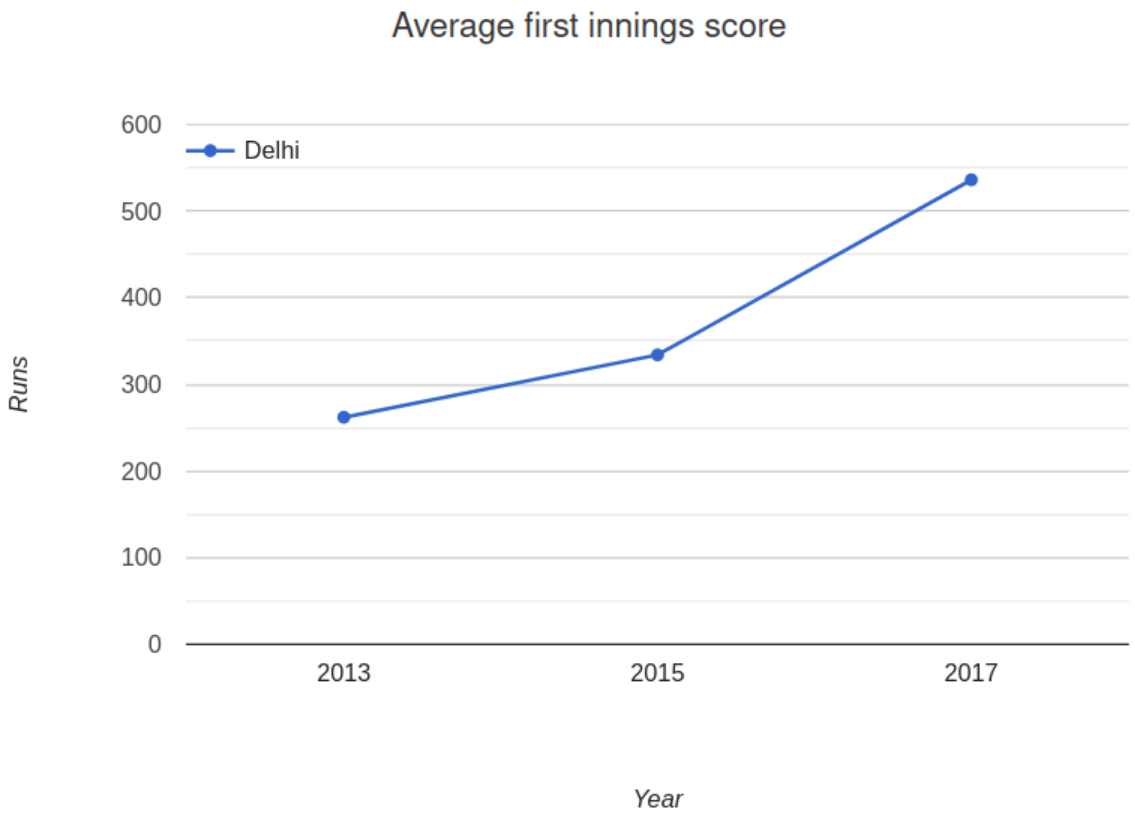
Create a pie showing percentage which shows details about matches.

1. Matches won where team had chosen batting first
2. Matches won where team had chosen bowling first
3. Matches who were drawn. (If no matches drawn at a given venue use zero)



c) Average First Innings Score :

X axis : Alternate season year starting from 2011 to 2018
Y axis : Average score in first innings for each season.



Section F :
Altering the database by adding data
and
AutoSequencing Primary Key

In this section you will learn how to create auto incrementing primary keys for a given table and sending data to the server using the post query.

1. **Change Primary key (venue_id) of venue table to automatically incrementing primary key :**

   Steps to create auto increment key :
   1. create sequence auto_increment_id_seq;
   2. alter table xyz alter column_x set default nextval('auto_increment_id_seq');
   3. Select setval('auto_increment_id_seq', 119);

2. **Add a form to create a new venue.**

   Form must contain the following fields.
   venue Name : string
   County Name : string
   City Name : string
   Capacity : number
   Submit : button

After clicking the submit button you should use a post query to send data to the server and the server will add a new venue in the venue table.

Note : venue_id should be auto generated
Hint : Angular Example to send data to server :

Submission instructions:

Each team should submit one tarball named <rollnum1>-<rollnum2>-lab4.tar.gz. Consisting of the following:

1. Submit the DDL of your database in a file named **lab4db.ddl** .
   Example to generate DDL file : pg_dump -U cinderella -h 127.0.0.1 lab3db -s > lab4db.ddl
2. Submit the frontend in directory named **frontend**

   For Angular : Need to submit the **src folder** of your angular project. It should contain the following files/folder :
   a. Services
   b. Routes
   c. Utils : All static content and .ts/.js files should be provided.
   d. app.component.html
   e. app.component.css
   f. app.component.ts
   g. app.module.ts
   h. app-routing.module.ts
   i. main.ts

Note :

1. **You can group components and services into subfolders for each different component of your application. Ex : PlayerInfoComponent and MatchInfoComponent**
2. **You can use global routing files as well as components specific routing files.**

   For ReactJs : Need to submit the **src folder** of your reactjs project. It should contain the following files/folder.
   a. Assets : all static contents such as images, json files.
   b. Components
   c. app.js
   d. app.css
   e. index.js
   f. index.css
   g. package.json
   h. package-lock.json
3. Submit the **src folder** of your nodejs backend in a directory named **backend.** It should contain the following files/folder.
   a. Controllers
   b. Models
   c. Services
   d. package.json
4. Readme.md file
   It will contain all the references you are using for your template and other static resources.

Grading Rubric:

Total Marks 100

Section B 1	**5**
Section B 2	**5**
Section B 3	**5**
Section B 4	**10**
Section C	**25**
Section D	**20**
Section E Q1	**5**
Section E Q2	**15**
Section F Q1	**5**
Section F Q2	**5**
Bonus points for look and feel	**10**

Note : You can refer to any standard Angular/react template as long as you cite the sources in the readme file.