

# **SPORTS SCHEDULER: Scheduling Sports Fixture**

A PROJECT REPORT

BACHELOR OF TECHNOLOGY IN  
INFORMATION TECHNOLOGY

3<sup>rd</sup> Semester

Submitted by:

***Mrityunjay Mishra (2k19/IT/080)***

***Rachit Parwanda (2k19/IT/102)***

Under the Supervision of  
**Mrs. Swati Sharda**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi  
110042

NOVEMBER, 2020

# **DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

## **CANDIDATE'S DECLARATION**

We, MRITYUNJAY MISHRA (Roll No. 2K19/IT/080) and RACHIT PARWANDA (Roll No. 2K19/IT/102), students of B.Tech. (INFORMATION TECHNOLOGY), hereby declare that the project Dissertation titled "SPORTS SCHEDULER: Scheduling Sports Fixture" which is submitted by us to the Department of INFORMATION TECHNOLOGY, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the 3<sup>rd</sup> semester of the Bachelor of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date: 18-11-2020

Mrityunjay Mishra

Rachit Parwanda

# **DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

## **CERTIFICATE**

I, hereby certify that the Project: **SPORTS SCHEDULER: Scheduling Sports Fixture**, which is submitted by Mrityunjay Mishra, Roll No – 2K19/IT/080 and Rachit Parwanda, Roll No – 2K19/IT/102, INFORMATION TECHNOLOGY, Delhi Technological University, Delhi in fulfillment of the requirement for the 3<sup>rd</sup> semester of Bachelor of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

**Supervisor :**

Mrs. Swati Sharda

Date: 18-November-2020

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

**ACKNOWLEDGEMENT**

We would like to convey our heartfelt thanks to our supervisor Mrs. Swati Sharda for her ingenious ideas, tremendous help and cooperation. We are extremely grateful to our friends who gave valuable suggestions and guidance for completion of our project. The cooperation and healthy criticism came handy and useful with them.

# *Table of Contents*

OBJECTIVE.....	6
INTRODUCTION .....	7
• Introduction to HTML, CSS and JavaScript:.....	7
• Introduction to Bootstrap:.....	7
• Introduction to NodeJS:.....	8
• Introduction to Handlebars: .....	8
• Introduction to C++ Programming Language: .....	8
METHODOLOGY .....	9
• FRONTEND:-.....	9
HTML/CSS/BOOTSRAP :-.....	9
JAVASCRIPT and NodeJS:-.....	9
• Backend .....	9
C++ .....	9
CONCEPTS OF DISCRETE STRUCTURES USED: .....	12
RESULT AND ANALYSIS.....	14
CHALLENGES FACED.....	17
GITHUB LINK- .....	18
WEBSITE LINK: .....	18
BIBLIOGRAPHY .....	18

# OBJECTIVE

The problem statement is quite simple. We have to schedule games or make a fixture for a league or a tournament. Like its IPL time and which team will play with whom and when, is what we have to decide considering factors like there is no team which plays matches on continuous days, the break for each team after a match should be almost similar for a fair league.

We also have covered a wide variety of Sports like Cricket, Hockey, Football, Chess, Swimming, 400-m race, Marathon, Carrom and Basketball.

We have also covered 3 types of Tournament formats, like Round Robin, League and Knockout.

# INTRODUCTION

- Introduction to HTML, CSS and JavaScript:

HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript are the languages that run the web. They're very closely related, but they're also designed for very specific tasks. Understanding how they interact is a key to become a good web developer. Simply stating :

- HTML is for adding meaning to raw content by marking it up.
- CSS is for formatting that marked up content.
- JavaScript is for making that content and formatting interactive.

Think of HTML as the skeleton of a web page that contains abstract text and images behind the scenes, CSS as the page that actually gets displayed, and JavaScript as the behaviors that can manipulate both HTML and CSS. All 3 of them are totally different languages, but they all refer to one another in some way. Most websites rely on all three to make them user friendly and visually appealing.

- Introduction to Bootstrap:

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. It solves many problems, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

## **Why Bootstrap?**

- Faster and Easier Web-Development.
- It creates Platform-independent web-pages.
- It creates Responsive Web-pages.
- It designed to be responsive to mobile devices too.
- It is Free! Available on [www.getbootstrap.com](http://www.getbootstrap.com)

- [Introduction to NodeJS:](#)

NodeJS is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36.

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

- [Introduction to Handlebars:](#)

Handlebars is a logic-less templating engine that dynamically generates the HTML pages.

It keeps the HTML page clean and separates the logic-less templates from the business logic in our JavaScript files, thus improving the structure of the application (and also its maintainability and scalability)

It simplifies the task of manually updating the data on the view.

- [Introduction to C++ Programming Language:](#)

C++ is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a **compiled** language. It is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.

Some of the *features & key-points* to note about the programming language are: It is **Simple, Machine Independent but Platform Dependent, Mid-level language**, has **Rich library support, high speed of execution**, supports **Pointer and direct Memory-Access**, it is **Object-Oriented** and it is a **Compiled Language**, therefore, contributing to its speed.



# METHODOLOGY

- FRONTEND:-

## HTML/CSS/BOOTSTRAP :-

We have around 10 pages in our website.

To create the structure of website we used **HTML**.

To design the website, we use **CSS**.

To implement the navigation bar on the website we are using bootstrap features and **BOOTSTRAP** is also used to implement the form functionality on pages as well.

## JAVASCRIPT and NodeJS:-

To add the functionalities to our webpage of user interaction we have used JavaScript.

We have implemented a tournament scheduler in our website in which, the user has to select the sport from the dropdown menu (out of 9 sports) after which they can select the type of schedule ("Knockout", "Round Robin", "League") then the user needs to select the number of pools and number of teams and click on "Create Schedule" button.

The new webpage will show the desired schedule.

The schedule is created by providing the input from our main page to the backend where C++ file processes it and provides the output which is printed on the desired page. This whole system in the backend is controlled by the NodeJS server.

- Backend

## C++

C++ is used to build behind-the-scenes logic for scheduling the tournaments. It takes necessary inputs from the frontend, generates schedule according to them and renders the schedule back to frontend, where it is displayed.

## Round Robin Scheduling:

For this type of scheduling, we have used the concept of Complete Graph.

To implement the Graph in C++, we have used the Adjacency Matrix Representation of Graph.

The vertices/nodes of a graph represent a unique team/player number, and the edges of a graph represent a Match between the two teams/players whom the edge is connecting.

For scheduling Round Robin format, we have used Edge-Colouring, where weight of an edge represents a Unique colour (from 0 to k, where k is the Chromatic Number).

Each edge coloured with same colour, represents that those matches can happen simultaneously, without any conflict/overlap.

**-Chromatic Number (k):** It is the minimum number of colours required to colour a graph, such that no two adjacent vertices, adjacent edges, or adjacent regions are colored with same colour.

For complete graphs: - Chromatic Number (k)

$$= n-1, \text{ if } n \text{ is even,}$$

$$= n, \text{ if } n \text{ is odd,}$$

where,  $n$  = number of vertices/nodes in a graph = number of teams/players

## Knockout Scheduling:

For scheduling knockout tournaments, we have used the concept of grouping all teams/players in pairs, and then, winners of each match progressing to next round, therefore reducing the teams in successive rounds.

This process continues until 1team/player remains, which is then declared as the Winner!

If, number of teams are not equal to some power of 2, we have then used the concept of 'Byes', which basically means that a team/player given a 'Bye', will not be playing the first round, but will start playing from second round onwards.

Distribution of 'Byes' is such that number of teams in second round become equal to some power of 2.

In Knockout tournament:

Total rounds =  $\text{ceil}(\log_2(n))$ , where  $\text{ceil}()$  is the 'ceiling function' and  $n$  is the total number of teams.

Total byes =  $2^{\text{total rounds}} - n$ , where  $n$  is the total number of teams.

### **League Tournaments:**

League tournaments are an extension of Round Robin tournaments.

Leagues are basically, Round Robin tournaments followed by eliminators / quarter finals / semi finals / finals, between top teams of the leaderboard.

# CONCEPTS OF DISCRETE STRUCTURES USED:

## **Graphs :**

Concept of graphs is used to make the base for scheduling Round Robin Tournaments.

## **Adjacency Matrix Representation :**

Computers cannot understand pictorial representation of graphs like us, therefore we need some other form of representation to work upon.

Among various representations of graphs which computers can understand and operate on, we are using Adjacency Matrix Representation of Graphs, which is basically a 2-D matrix, in which each row and column represents a unique team/player and cell value corresponding to a particular row and column represents the weight/colour of the edges connecting those 2 teams/players.

Since graph is a complete graph, therefore, this matrix is a Symmetric Matrix.

## **Complete Graphs :**

The graph generated while scheduling a Round Robin Tournament is a Complete Graph, and all other operations are based on properties of complete graphs.

## **Graph Colouring :**

Concept of Graph colouring is used so that 2 matches with a common team cannot occur simultaneously, ie, there should be no collision/overlaps of teams.

Chromatic number for complete graph =  $n-1$ , if  $n$  is even

=  $n$ , if  $n$  is odd

where,  $n$  = number of vertices/nodes in a graph = number of teams/players

The problem of graph coloring is mainly of 3 types:

**1. Vertex coloring:** It is the most common graph coloring problem. The problem is, given  $m$  colors, find a way of coloring the vertices of a graph such that no two adjacent vertices are colored using same color.

**2. Edge Coloring:** In this problem, the aim is to colour the graph such that no vertex is incident to two edges of same color. (or, such that no two adjacent edges have the same color with an optimal number of colors. Two edges are said to be adjacent if they are connected to the same vertex.)

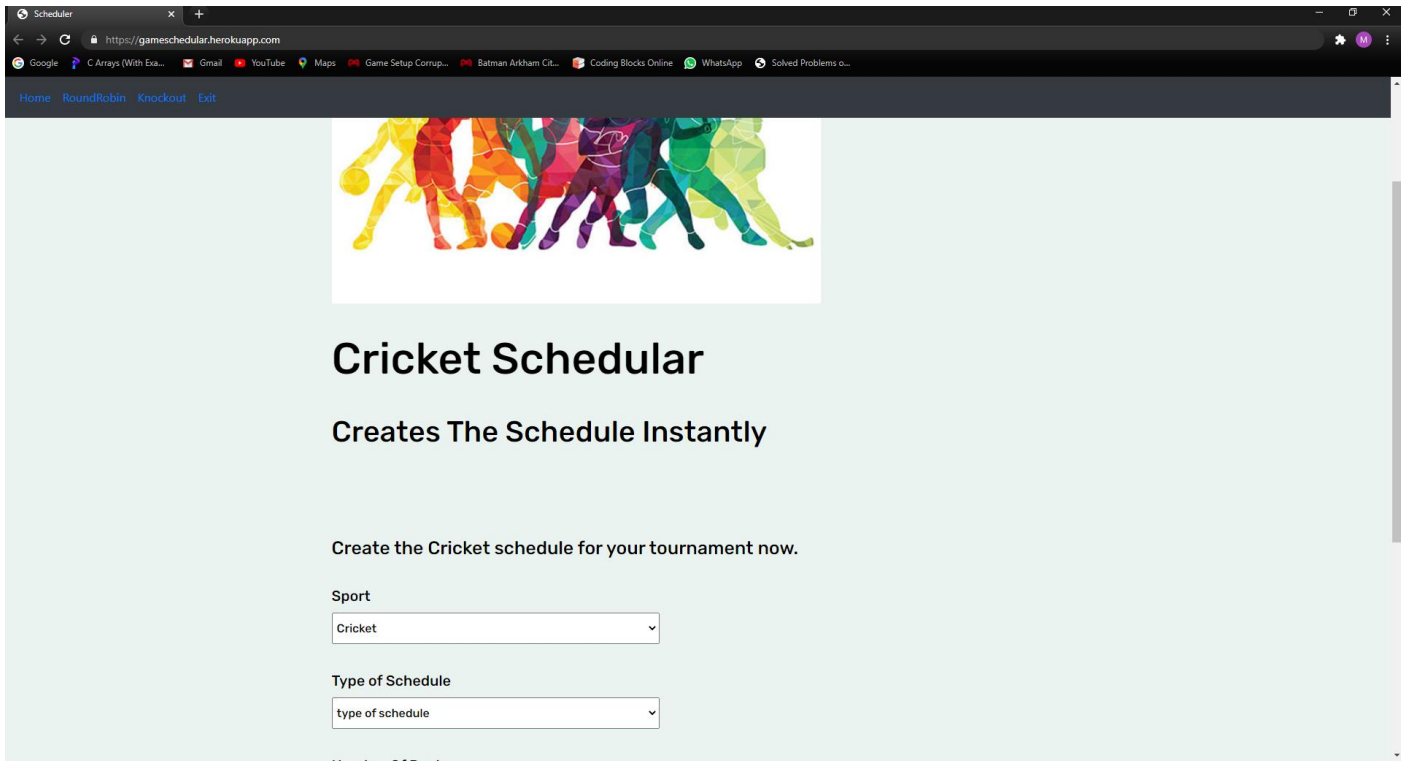
**3. Face Coloring:** (Geographical Map Coloring) can be transformed into vertex coloring.

**For our project, we are using Edge Coloring of graphs.**

# RESULT AND ANALYSIS

Some Screenshots of our working website are:

## Home Page



## Selecting Number of teams:

Scheduler

https://gamescheduler.herokuapp.com

Home RoundRobin Knockout Exit

Create the Cricket schedule for your tournament now.

Sport  
Cricket

Type of Schedule  
Knockout

Number Of Pools  
3

Number of Teams

- 6
- 6
- 9
- 12
- 15
- 18
- 21
- 24
- 27
- 30


... of schedule

## Cricket Schedule Part-1:

Cricket Scheduling

gamescheduler.herokuapp.com/Cricket?numofteam=6&numofpools=3&type=Knockout

Home RoundRobin Knockout International Cricket Council Board of Cricket Council India Exit



**Cricket**

**Schedule**

**POOL 1:-**

Round 1:  
Match-1: Team-1 v/s Team-2

**POOL 2:-**

## Cricket Schedule Part-2 :

Cricket Scheduling

gamescheduler.herokuapp.com/Cricket/?numofteams=6&numofpools=3&type=Knockout

Home RoundRobin Knockout International Cricket Council Board of Cricket Council India Exit

**POOL 1:-**

Round 1:

Match-1: Team-1 v/s Team-2

**POOL 2:-**

Round 1:

Match-1: Team-3 v/s Team-4

**POOL 3:-**

Round 1:

Match-1: Team-5 v/s Team-6

**Important Links :**

International Cricket Council

Board of Cricket Council India



# CHALLENGES FACED

We faced many challenges during the development of this project and we also enjoyed and learnt many things during this period. At first, we didn't know about NodeJS, so first challenge for us was to learn this framework. We were also not comfortable with JavaScript as well.

Then we learnt some advance HTML/CSS and Handlebars. We also learnt basic Bootstrap to add some style to our website.

Many times, we got stuck in between then we used some online resources for help.

In C++, first we thought of adding two teams in one vertex and to use the concept of vertex colouring but it did not work as expected, so we redesigned our approach and found one which was to use edge colouring approach with each vertex of graph which solved our problem.

In frontend, we tried to use HTML but it creates a static webpage and we cannot update or change something on that. To overcome this, we used the concept of dynamic webpage i.e. handlebars.

## *GITHUB LINK-*

<https://github.com/Mrityunjay19/Scheduler>

## *WEBSITE LINK:*

<https://gamescheduler.herokuapp.com/>

## *BIBLIOGRAPHY*

1. <https://www.geeksforgeeks.org/>
2. <https://www.programiz.com/>
3. <https://www.tutorialspoint.com/>
4. <https://www.wikipedia.org/>
5. <https://getbootstrap.com/>
6. <https://www.w3schools.com/>
7. <https://stackoverflow.com/>
8. <https://nodejs.org/en/>