CSEE5590/CS490 APS

Programming for Web/Cloud / Mobile Applications

ASSIGNMENT - 2

BALACHANDAR KULALA

(Student ID:12543027)

Part1: Video Search (using YouTube API)

Aim:

Design an application that displays the desired videos based on search keyword using YouTube API.

Description of Work:

Develop the web page that displays the desired videos based on search keyword. This task can be accomplished by using YouTube API.

Technical Section:

Majorly used technical skills.

- HTML
- JavaScript
- AngularJS
- CSS
- Bootstrap

Development Section

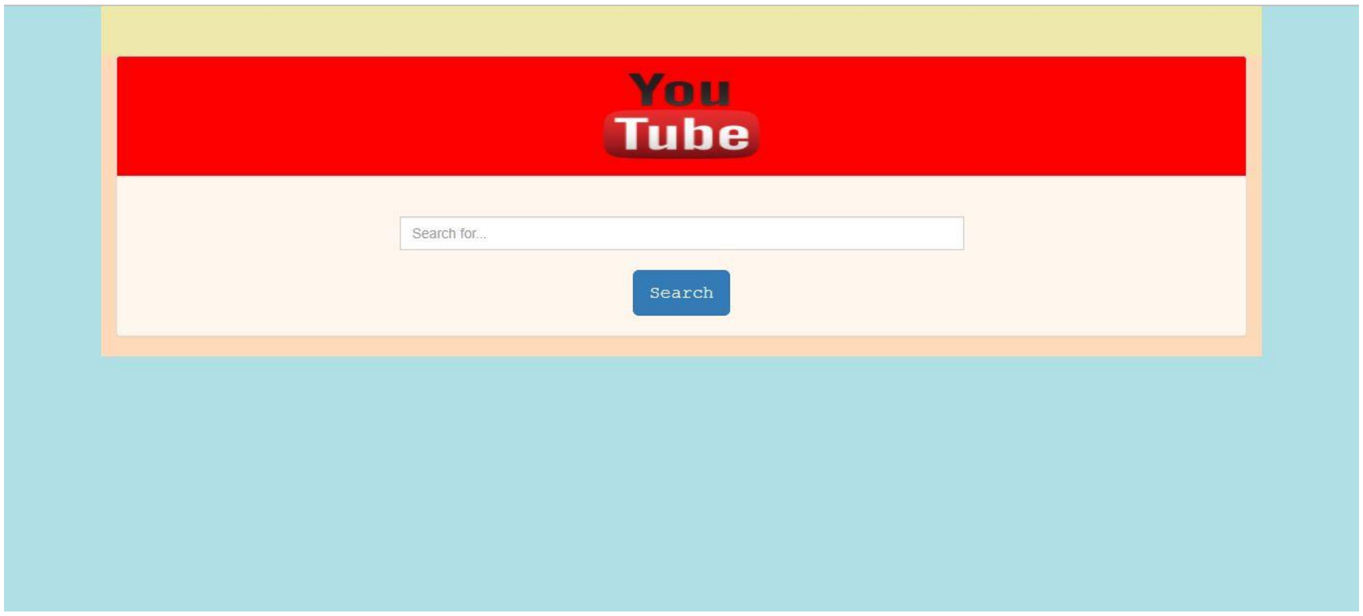
This Application contains one page:

1. Index.html :

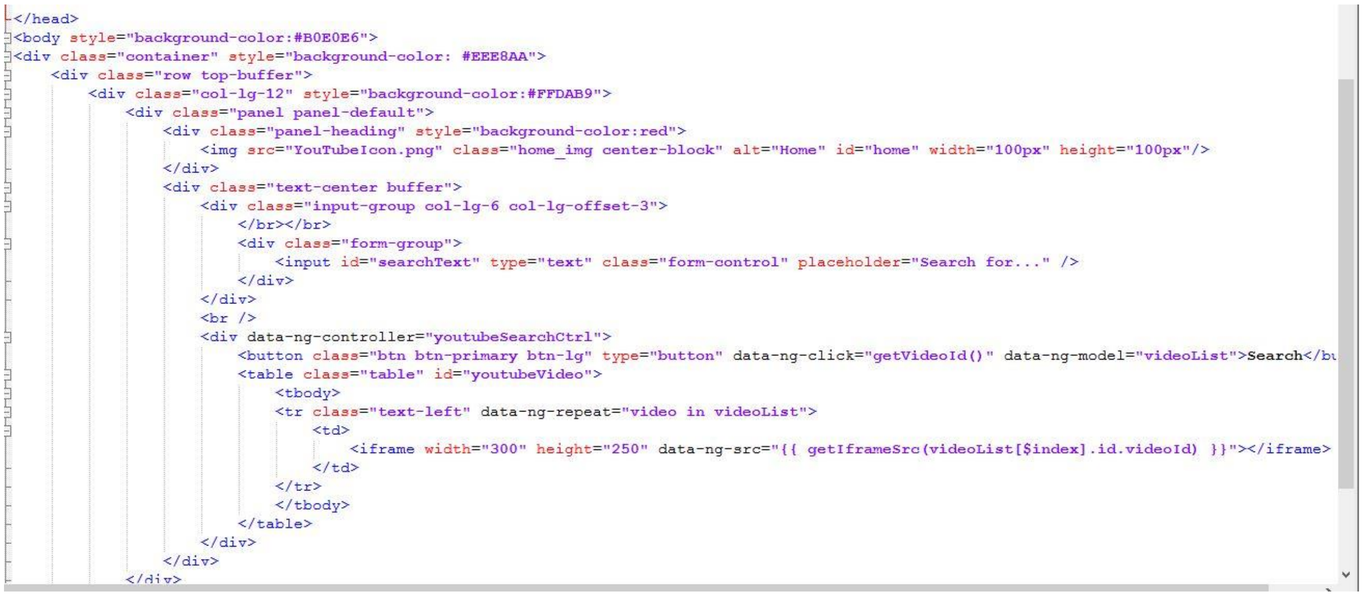
   This starting page of the application, this page displays the option for user to enter the search keyword.

   The following image shows the home section of the application.

The sample code for this index page is displayed in following image.



The corresponding HTML code for this index page is shown below.

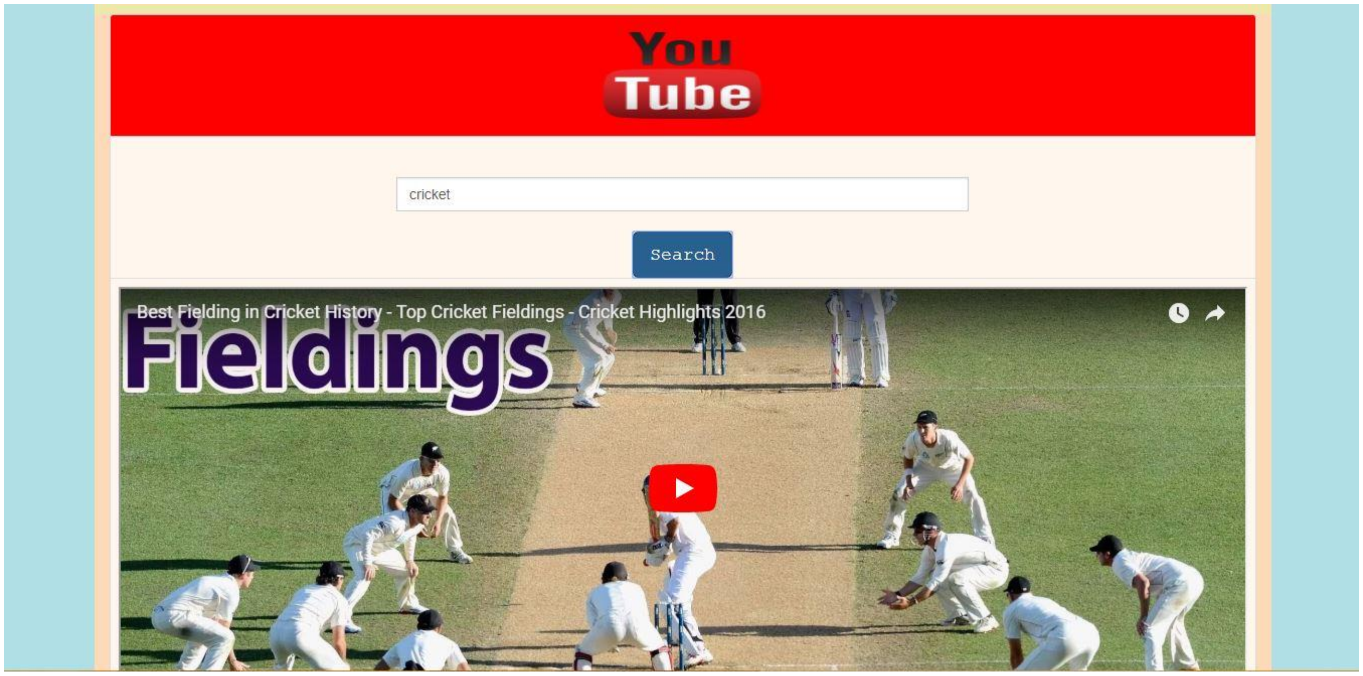


```
</head>
<body style="background-color:#B0E0E6">
<div class="container" style="background-color: #EEE8AA">
    <div class="row top-buffer">
        <div class="col-lg-12" style="background-color:#FFDAB9">
            <div class="panel panel-default">
                <div class="panel-heading" style="background-color:red">
                    <img src="YouTubeIcon.png" class="home_img center-block" alt="Home" id="home" width="100px" height="100px"/>
                </div>
                <div class="text-center buffer">
                    <div class="input-group col-lg-6 col-lg-offset-3">
                        </br></br>
                        <div class="form-group">
                            <input id="searchText" type="text" class="form-control" placeholder="Search for..." />
                        </div>
                    </div>
                    <br />
                    <div data-ng-controller="youtubeSearchCtrl">
                        <button class="btn btn-primary btn-lg" type="button" data-ng-click="getVideoId()" data-ng-model="videoList">Search</bu
                        <table class="table" id="youtubeVideo">
                            <tbody>
                            <tr class="text-left" data-ng-repeat="video in videoList">
                                <td>
                                    <iframe width="300" height="250" data-ng-src="{{ getIframeSrc(videoList[$index].id.videoId) }}"></iframe>
                                </td>
                            </tr>
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
```

This sample code describes the design of search textbox, search button and YouTube icon display.

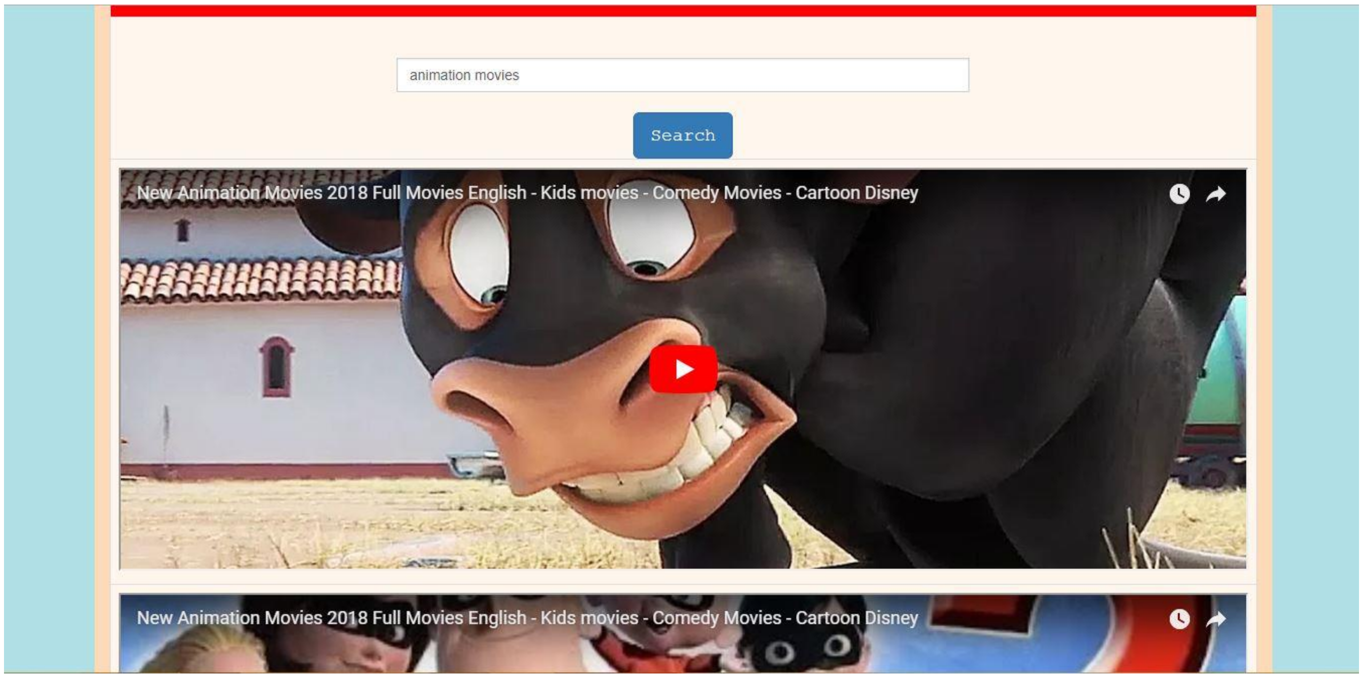
When user enters the search keyword and clicks on search button displayed in the screen, it displays the related videos for the search keyword.

The output screen shown below.


You
Tube
cricket
Search
Best Fielding in Cricket History - Top Cricket Fieldings - Cricket Highlights 2016
Fieldings

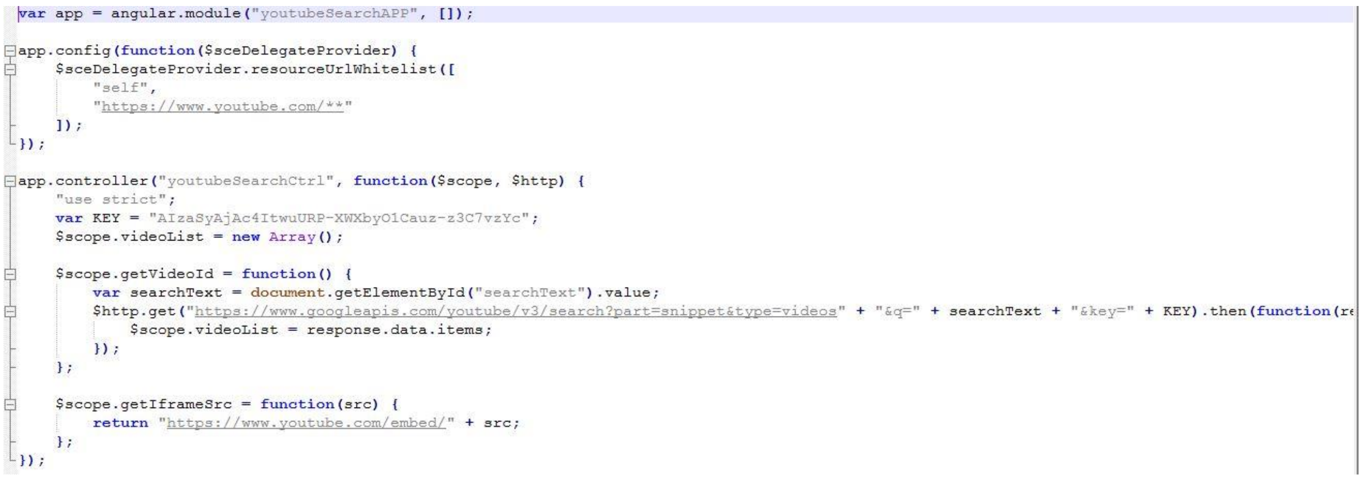
Here user entered the search keyword as “cricket”. When clicks on the search button it displays the videos related to cricket which shown in output.

Another sample output for search keyword: “Animation movies”


animation movies
Search
New Animation Movies 2018 Full Movies English - Kids movies - Comedy Movies - Cartoon Disney
New Animation Movies 2018 Full Movies English - Kids movies - Comedy Movies - Cartoon Disney

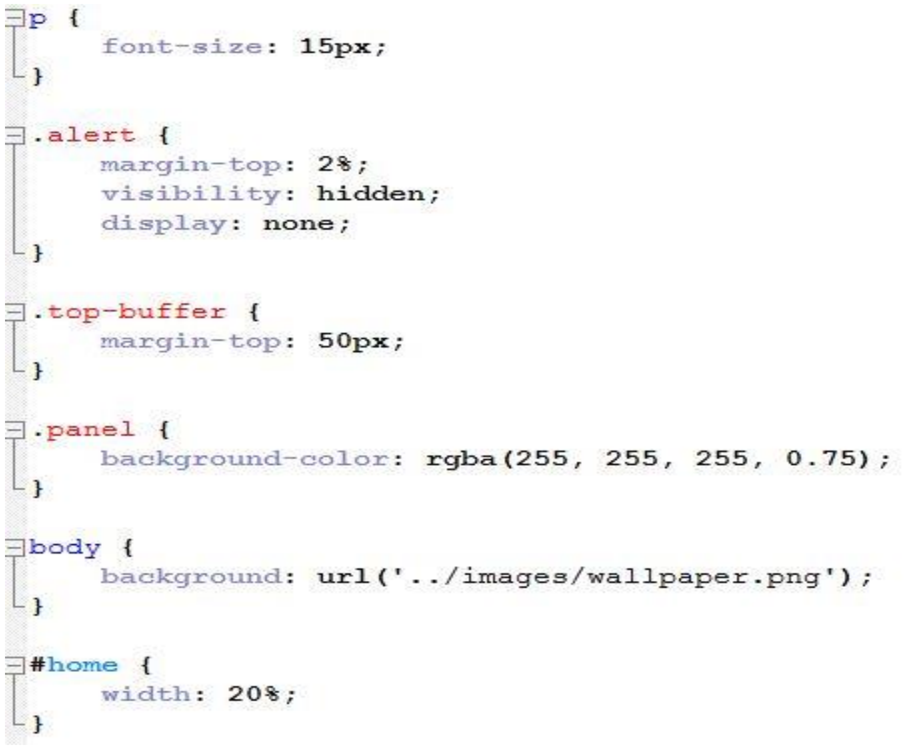
This functionality is achieved by using the YouTube API shown in below code snippet.

The angular JS code to accomplish this task is shown in following image.



```
var app = angular.module("youtubeSearchAPP", []);

app.config(function($sceDelegateProvider) {
    $sceDelegateProvider.resourceUrlWhitelist([
        "self",
        "https://www.youtube.com/**"
    ]);
});

app.controller("youtubeSearchCtrl", function($scope, $http) {
    "use strict";
    var KEY = "AIzaSyAjAc4ItwuURP-XWXbyO1Cauz-z3C7vzYc";
    $scope.videoList = new Array();

    $scope.getVideoId = function() {
        var searchText = document.getElementById("searchText").value;
        $http.get("https://www.googleapis.com/youtube/v3/search?part=snippet&type=videos" + "&q=" + searchText + "&key=" + KEY).then(function(re
            $scope.videoList = response.data.items;
        });
    };

    $scope.getIframeSrc = function(src) {
        return "https://www.youtube.com/embed/" + src;
    };
});
```

For proper designing and styling we used css file, which is shown below.



```
p {
    font-size: 15px;
}

.alert {
    margin-top: 2%;
    visibility: hidden;
    display: none;
}

.top-buffer {
    margin-top: 50px;
}

.panel {
    background-color: rgba(255, 255, 255, 0.75);
}

body {
    background: url('../images/wallpaper.png');
}

#home {
    width: 20%;
}
```

Part2: Tic-Tac-Toe game with Angular JS

Aim:

Develop a Tic-Tac-Toe game using Angular JS.

Description:

Develop the Tic-Tac-Toe game application using Angular JS. This game provides the facility to play from basic dimensions 3x3 to till 10x10. This allows to play more than one player as well. This application is developed using Angular JS, HTML and CSS.

Technical Section:

Majorly used technical skills.

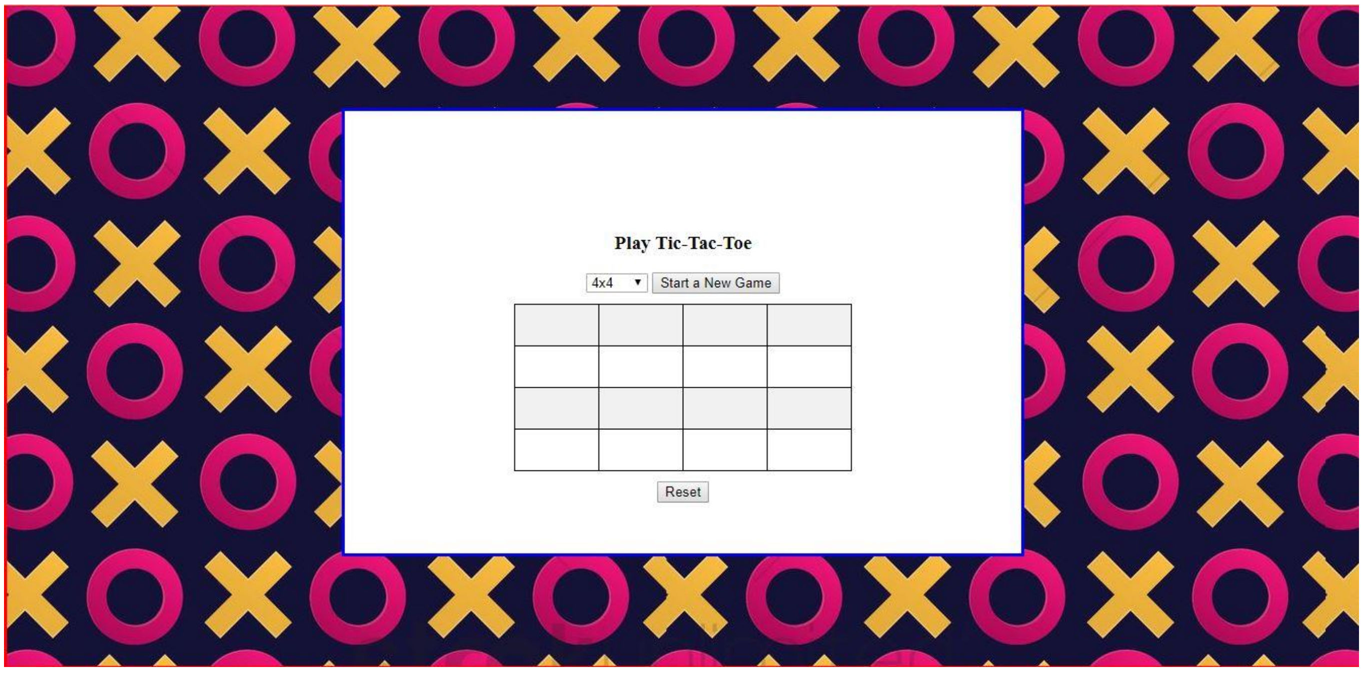
- HTML
- JavaScript
- AngularJS
- CSS
- Bootstrap

Development:

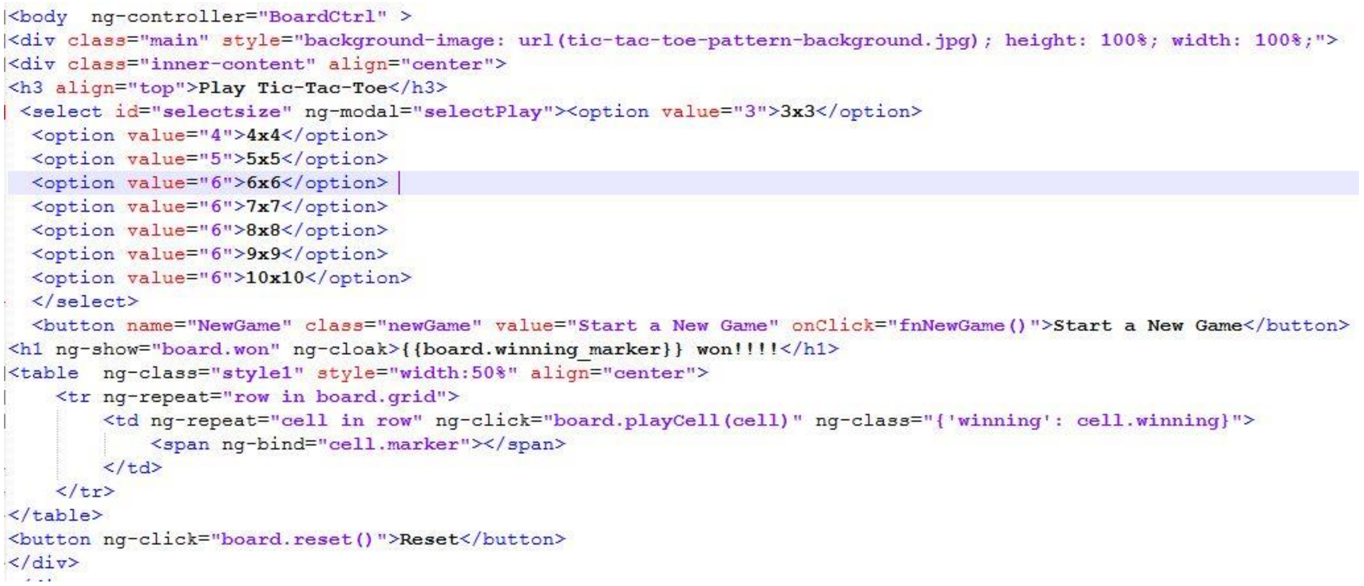
This application contains only one single page index.html. Index page displays the game selection options such as dimensions of the game and number of players.

After selection of dimensions and number of players, players should click on the cells to mark and they should play recursively until anyone of them wins the game.

The design of the html page is shown below.



The corresponding HTML code snippet is shown below.

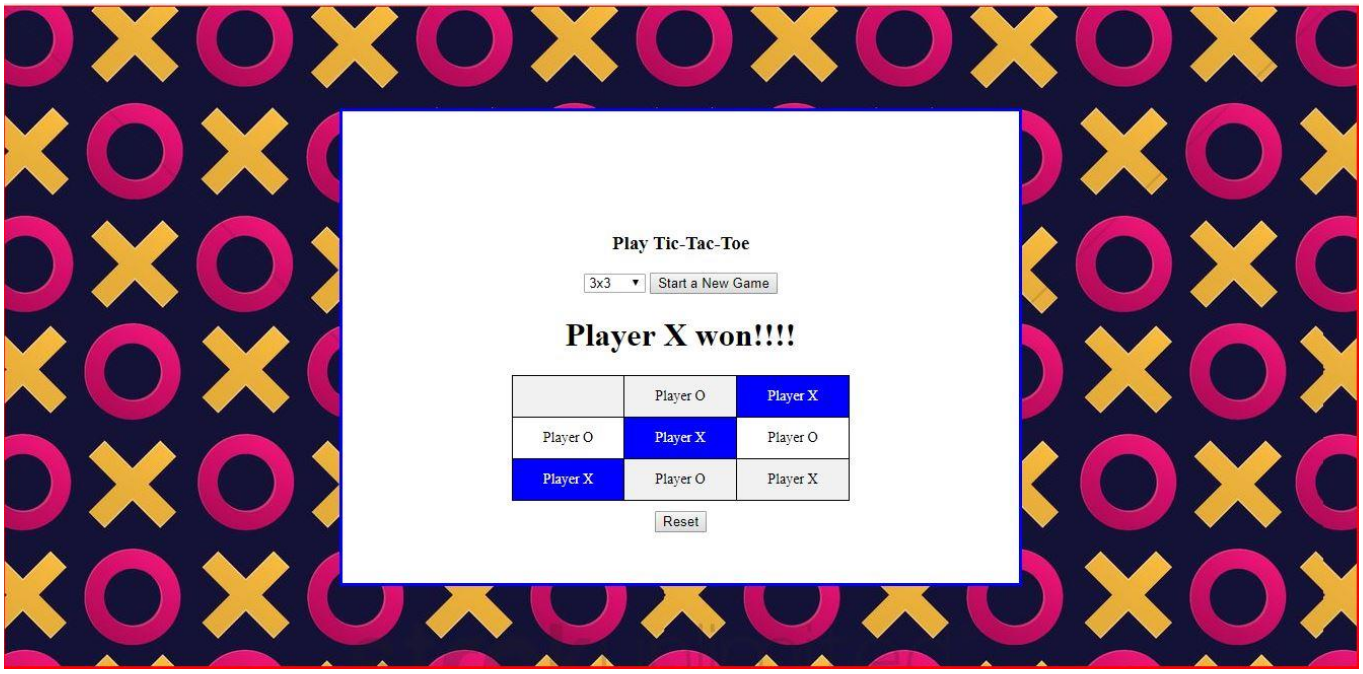


```
<body  ng-controller="BoardCtrl" >
<div class="main" style="background-image: url(tic-tac-toe-pattern-background.jpg); height: 100%; width: 100%;">
<div class="inner-content" align="center">
<h3 align="top">Play Tic-Tac-Toe</h3>
 <select id="selectsize" ng-modal="selectPlay"><option value="3">3x3</option>
  <option value="4">4x4</option>
  <option value="5">5x5</option>
  <option value="6">6x6</option>
  <option value="6">7x7</option>
  <option value="6">8x8</option>
  <option value="6">9x9</option>
  <option value="6">10x10</option>
  </select>
  <button name="NewGame" class="newGame" value="Start a New Game" onClick="fnNewGame()">Start a New Game</button>
<h1 ng-show="board.won" ng-cloak>{{board.winning_marker}} won!!!!</h1>
<table  ng-class="style1" style="width:50%" align="center">
    <tr ng-repeat="row in board.grid">
        <td ng-repeat="cell in row" ng-click="board.playCell(cell)" ng-class="{'winning': cell.winning}">
            <span ng-bind="cell.marker"></span>
        </td>
    </tr>
</table>
<button ng-click="board.reset()">Reset</button>
</div>
```

This html code illustrates the designing part of the application such as background image, table formation based on user selection, rest button.

The following two images describes the output screens.

1. <u>Winning game output with 3x3 board:</u>



Play Tic-Tac-Toe

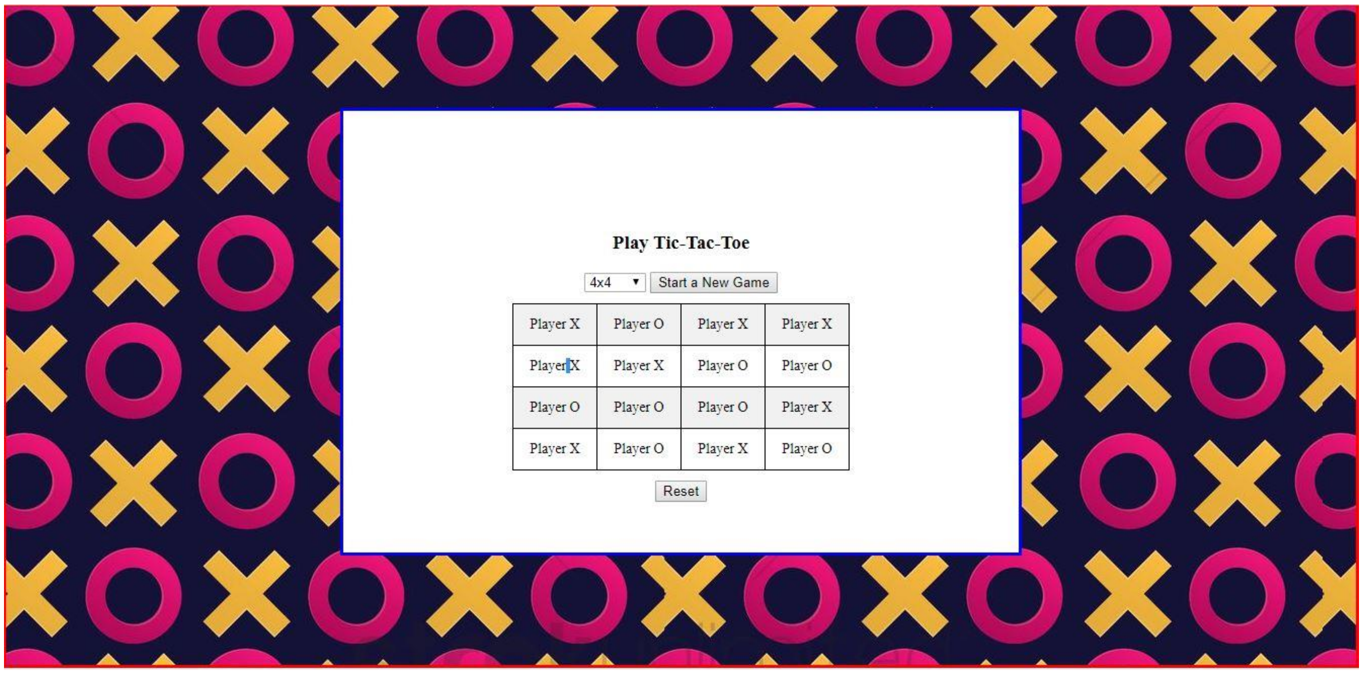
3x3 ▾ Start a New Game

Player X won!!!!

	Player O	Player X
Player O	Player X	Player O
Player X	Player O	Player X

Reset

2. Failure game output with 4x4 dimensions:



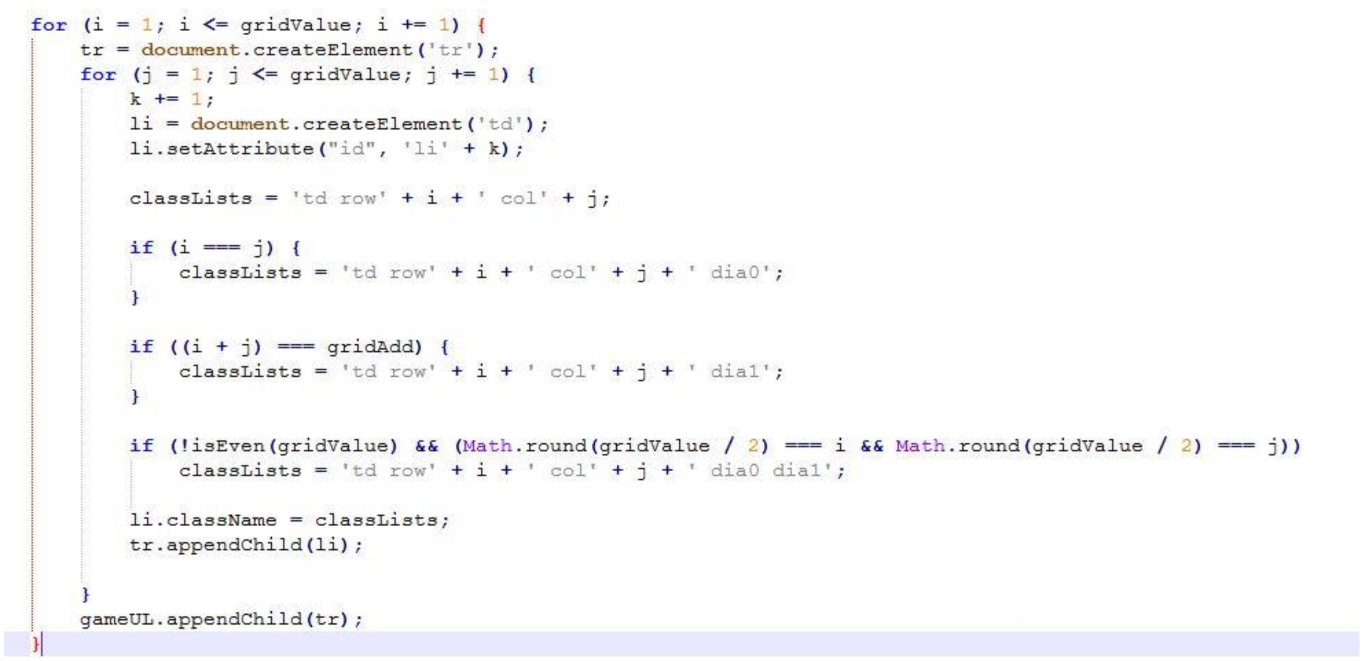
Play Tic-Tac-Toe

4x4 ▾ Start a New Game

Player X	Player O	Player X	Player X
Player X	Player X	Player O	Player O
Player O	Player O	Player O	Player X
Player X	Player O	Player X	Player O

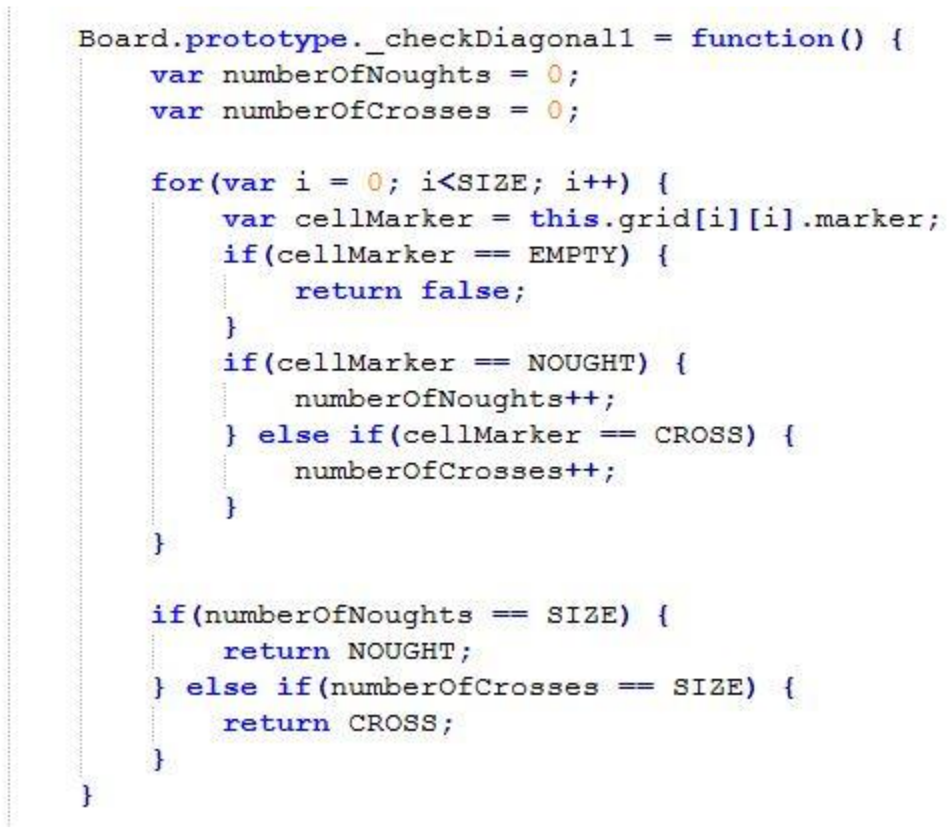
Reset

All these functionalities are developed and handled in Angular JS. This corresponding JS file is shown below.



```
for (i = 1; i <= gridValue; i += 1) {
    tr = document.createElement('tr');
    for (j = 1; j <= gridValue; j += 1) {
        k += 1;
        li = document.createElement('td');
        li.setAttribute("id", 'li' + k);

        classLists = 'td row' + i + ' col' + j;

        if (i === j) {
            classLists = 'td row' + i + ' col' + j + ' dia0';
        }

        if ((i + j) === gridAdd) {
            classLists = 'td row' + i + ' col' + j + ' dia1';
        }

        if (!isEven(gridValue) && (Math.round(gridValue / 2) === i && Math.round(gridValue / 2) === j))
            classLists = 'td row' + i + ' col' + j + ' dia0 dia1';

        li.className = classLists;
        tr.appendChild(li);

    }
    gameUL.appendChild(tr);
}
```

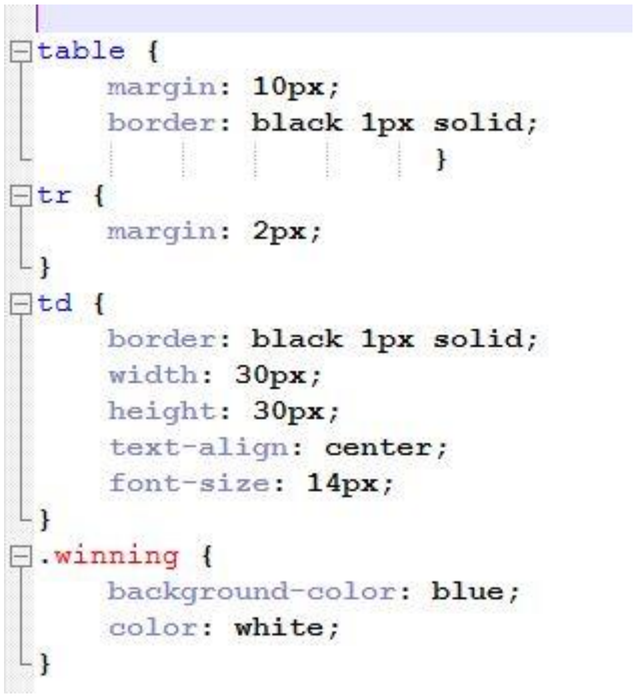
This code generates the table based on user selection.



```
Board.prototype._checkDiagonal1 = function() {
    var numberOfNoughts = 0;
    var numberOfCrosses = 0;

    for(var i = 0; i<SIZE; i++) {
        var cellMarker = this.grid[i][i].marker;
        if(cellMarker == EMPTY) {
            return false;
        }
        if(cellMarker == NOUGHT) {
            numberOfNoughts++;
        } else if(cellMarker == CROSS) {
            numberOfCrosses++;
        }
    }

    if(numberOfNoughts == SIZE) {
        return NOUGHT;
    } else if(numberOfCrosses == SIZE) {
        return CROSS;
    }
}
```

The above code checks the diagonal winning.

CSS file sample used to apply the styles:



```
table {
    margin: 10px;
    border: black 1px solid;
                        }
tr {
    margin: 2px;
}
td {
    border: black 1px solid;
    width: 30px;
    height: 30px;
    text-align: center;
    font-size: 14px;
}
.winning {
    background-color: blue;
    color: white;
}
```

Part3: Friends Family

Aim:

Obtain the friends and family members list, those who are followed by me and following me in twitter using twitter API.

Description:

Develop an application that gathers the list of people who are following me and followed by me in twitter using twitter API. And visualize this using D3.js.

Twitter API link used: https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/overview

Technical Section:

Majorly used technical skills.

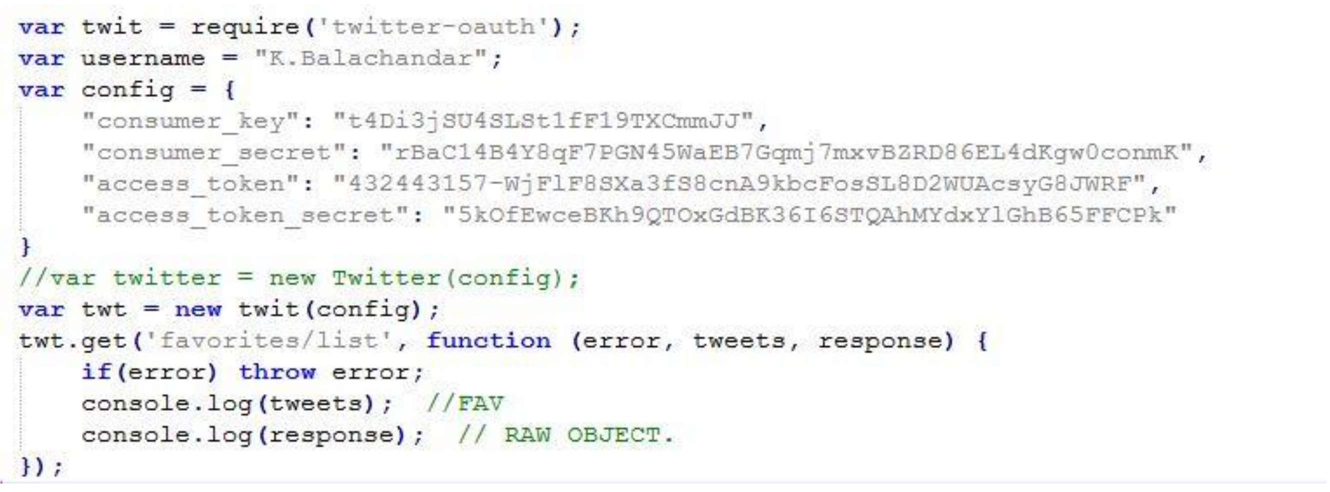
- HTML
- JavaScript
- AngularJS
- CSS
- Bootstrap

Development:

This application contains one html page that visualizes the list of people obtained from the twitter API.

This application is developed using AngularJS and D3.js .

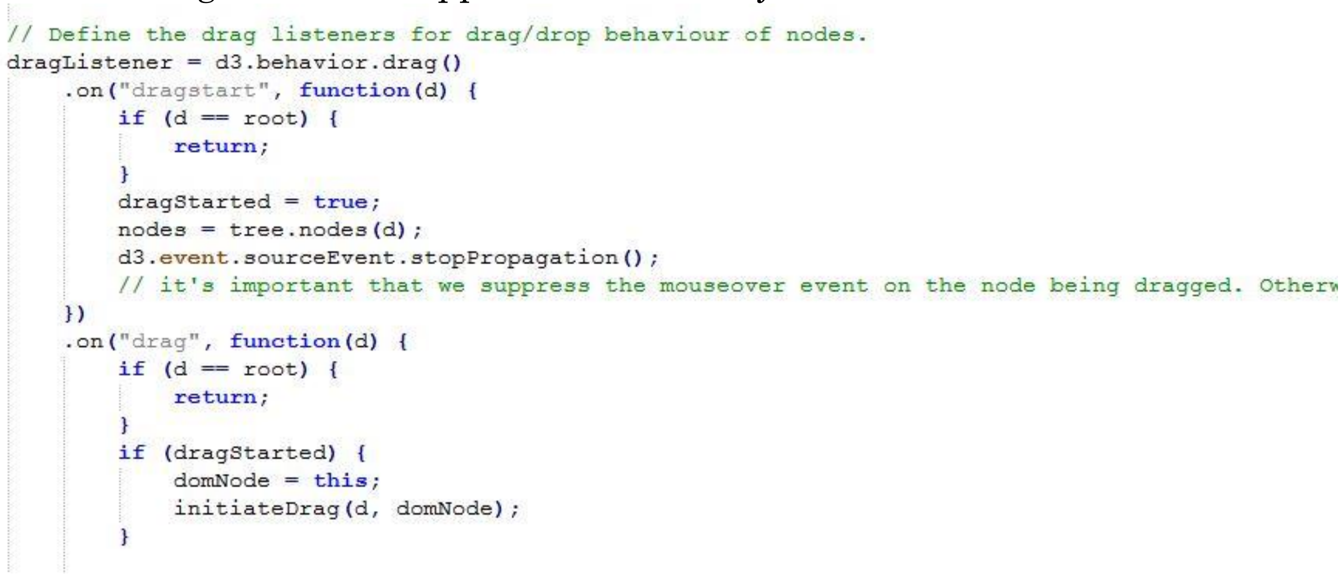
The corresponding Angular JS code file and D3.js files are shown below.



```
var twit = require('twitter-oauth');
var username = "K.Balachandar";
var config = {
    "consumer_key": "t4Di3jSU4SLSt1fF19TXCmmJJ",
    "consumer_secret": "rBaC14B4Y8qF7PGN45WaEB7Gqmj7mxvBZRD86EL4dKgw0conmK",
    "access_token": "432443157-WjFlF8SXa3fS8cnA9kbcFosSL8D2WUAcsyG8JWRF",
    "access_token_secret": "5kOfEwceBKh9QTOxGdBK36I6STQAhMYdxYlGhB65FFCPk"
}
//var twitter = new Twitter(config);
var twt = new twit(config);
twt.get('favorites/list', function (error, tweets, response) {
    if(error) throw error;
    console.log(tweets);  //FAV
    console.log(response);  // RAW OBJECT.
});
```

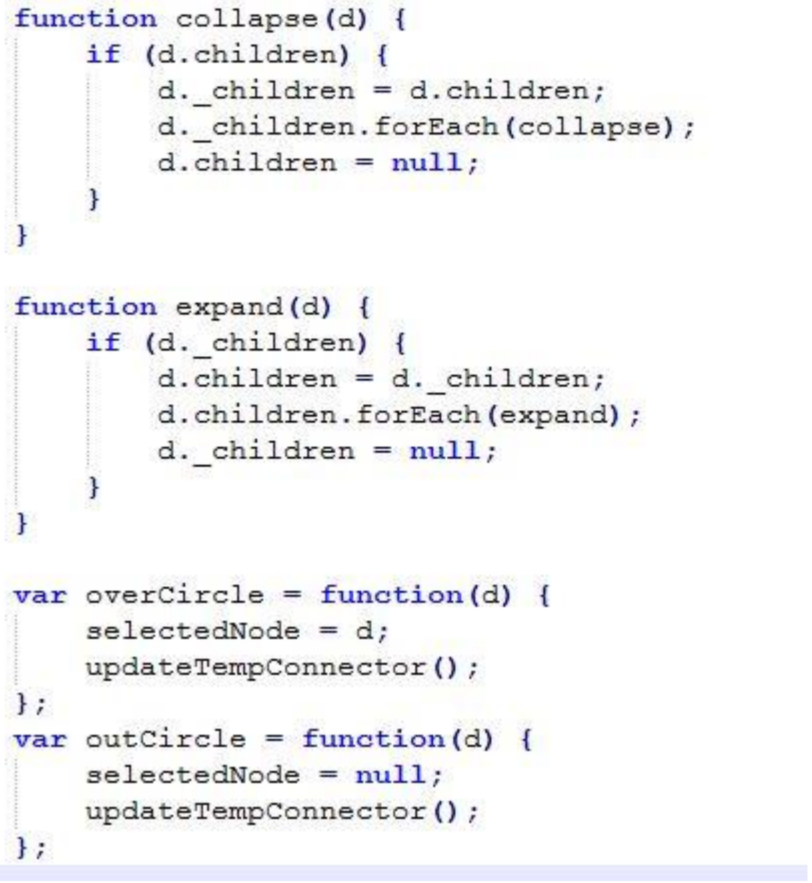
This shows the connecting to twitter with the generated keys and tokens and using the API.

The following two code snippets are from D3.js file:



```
// Define the drag listeners for drag/drop behaviour of nodes.
dragListener = d3.behavior.drag()
    .on("dragstart", function(d) {
        if (d == root) {
            return;
        }
        dragStarted = true;
        nodes = tree.nodes(d);
        d3.event.sourceEvent.stopPropagation();
        // it's important that we suppress the mouseover event on the node being dragged. Otherw
    })
    .on("drag", function(d) {
        if (d == root) {
            return;
        }
        if (dragStarted) {
            domNode = this;
            initiateDrag(d, domNode);
        }
```

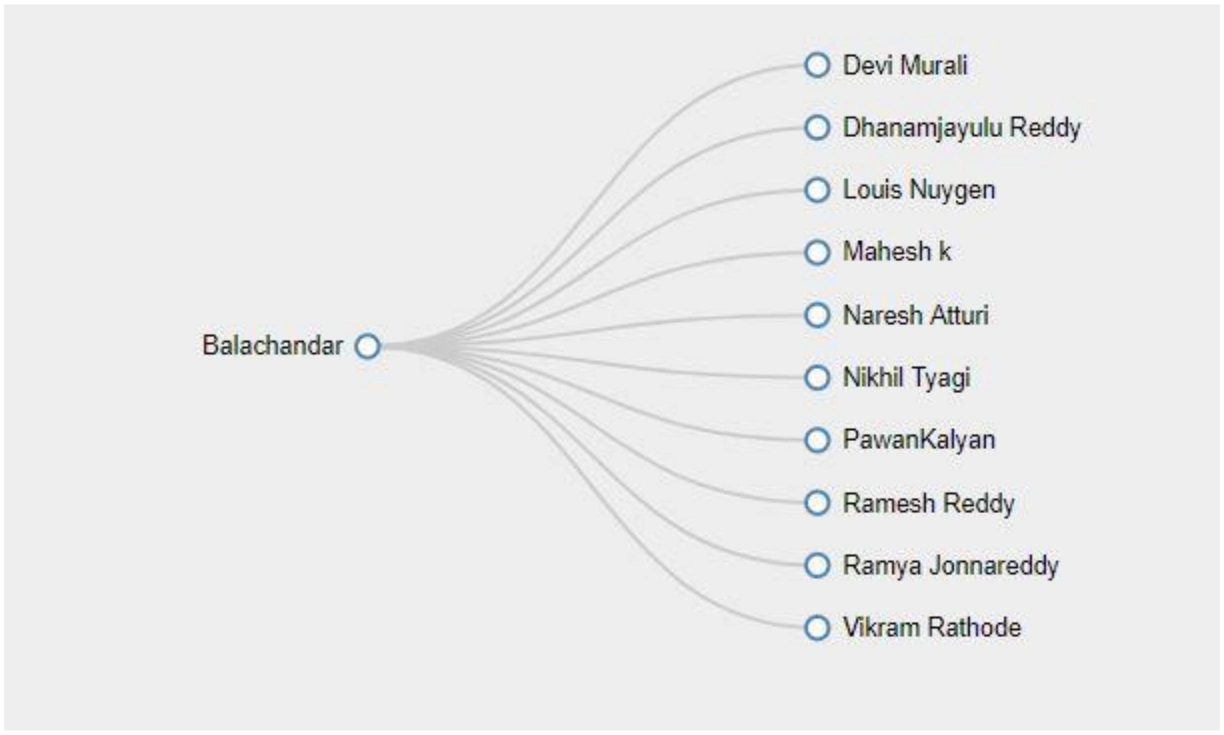
This code performs drag and drop options for the tree structure.



```
function collapse(d) {
    if (d.children) {
        d._children = d.children;
        d._children.forEach(collapse);
        d.children = null;
    }
}

function expand(d) {
    if (d._children) {
        d.children = d._children;
        d.children.forEach(expand);
        d._children = null;
    }
}

var overCircle = function(d) {
    selectedNode = d;
    updateTempConnector();
};
var outCircle = function(d) {
    selectedNode = null;
    updateTempConnector();
};
```

This code deals with the expand and collapse scenarios of the tree.

The Visualization of the application is shown below.



GitHub LINK

https://github.com/bkkhf/WebMobileProgramming/tree/master/Source/LabAssignmentWeb_2

DEMO LINK

https://youtu.be/mg9td61lgEw

References:

- https://www.w3schools.com
- https://www.udacity.com/
- http://www.hongkiat.com
- https://ariya.io
- https://css-tricks.com
- http://www.corelangs.com

- http://getbootstrap.com/
- https://jquery.com/
- https://www.javascript.com/
- http://www.tutorialsteacher.com/d3js
- http://www.tutorialsteacher.com/angularjs/angularjs-tutorials