Name: Lakshmana Kumar Mettu

Class ID:11

Introduction:

- This lab is about to get idea about all the elements such are html, CSS ,angular and also making every page as responsive using bootstrap elements.

Programming elements:

- HTML5
- CSS
- Angular JS
- Bootstrap
- JavaScript
- Google Knowledge Graph Search API

Platform Required:

- JetBrains-WebStorm

Objective-1:

- Create a MOOC web application which will render below details in different web pages.
- This web application has login page as user as well as guest.
- Corresponding login page using html as shown below.

```
</style>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAX
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<script>
    var app = angular.module('myApp', []);
    app.controller('initCtrl',['$scope', '$window', function($scope,$window) {
        $scope.users=[
            {username:'Kumar',password:'12345'},
            {username:'Lakshman',password:'12345'},
            {username:'Mettu',password:'12345'}
        ];
        $scope.err_msg='';
        $scope.url=$window.location.href;
        $scope.url=$scope.url.substring(0,$scope.url.indexOf('Login.htm'))
        $scope.login = function() {
            i=0;

            for (;i< $scope.users.length;i++)
            {
                if($scope.myName==$scope.users[i].username && $scope.myPassword==$scope.users[i].password)
                {
                    $scope.err_msg='';

                    $scope.url=$scope.url+'Home.html?user='+$scope.myName;
                    $window.location.href =$scope.url;

                    break;
```

Username *

Enter the Username

Password *

Enter the Password

Login SignUp

Guest User

Username *

Lakshman

Password *

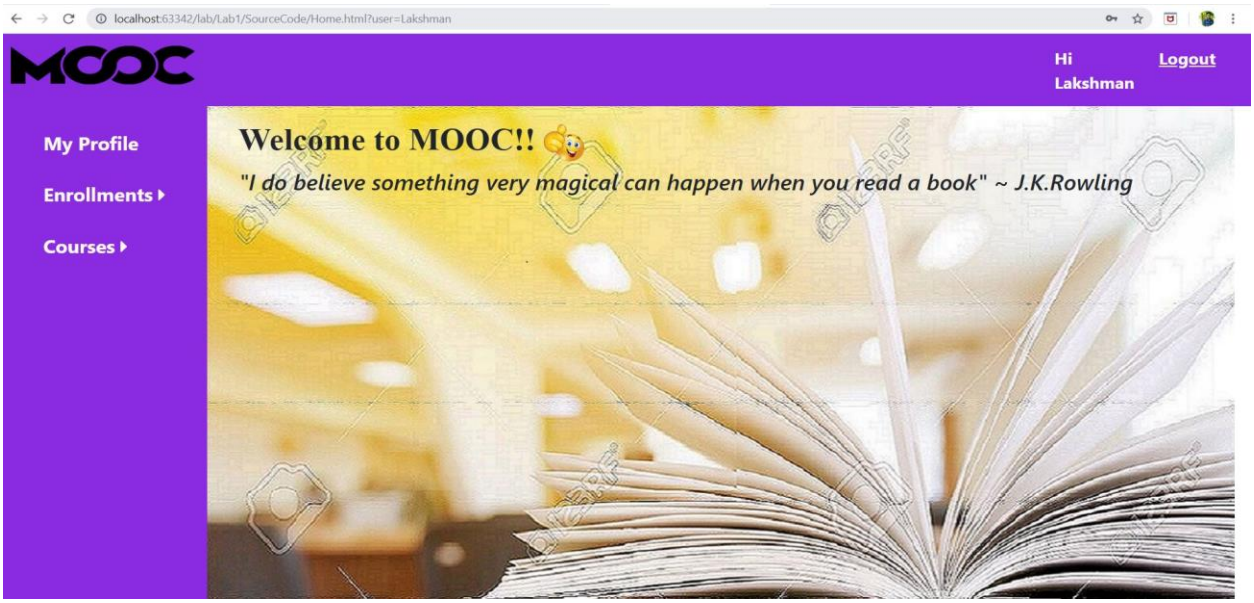
•••••

Login SignUp

Guest User

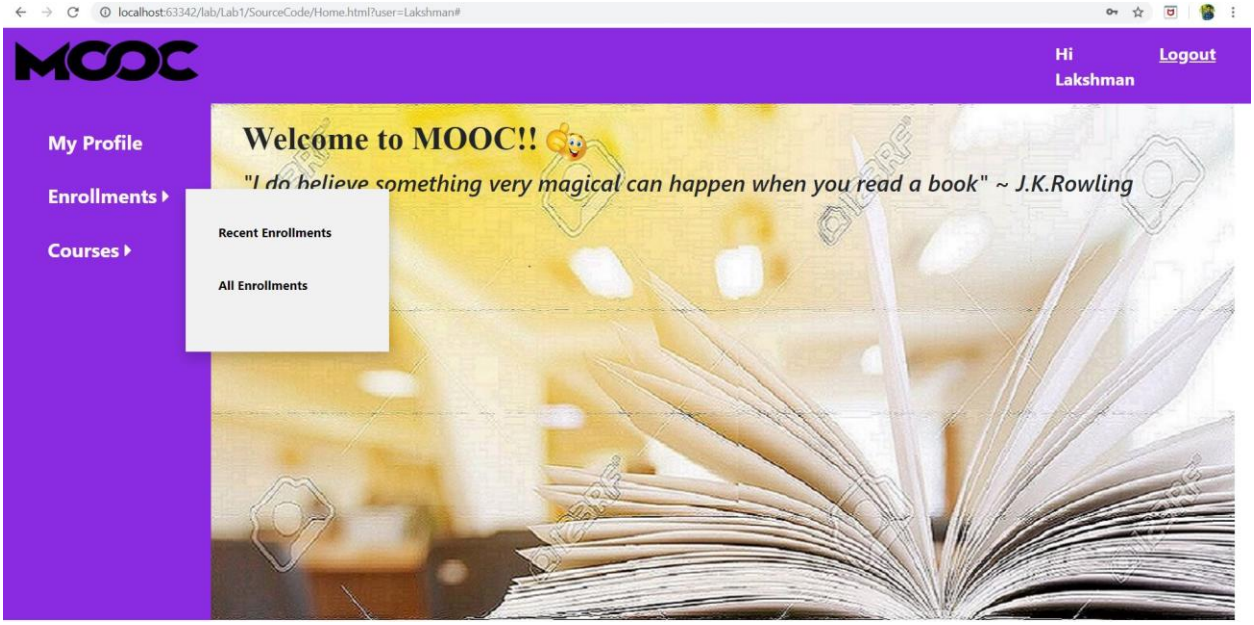
- Whenever you login using user details which are shown in login.html ,user can re-direct to home page which contains following features.
  i. My Profile
  ii. Enrollments
  iii. Courses
- The code snippet for home page as well as user interface are shown below.

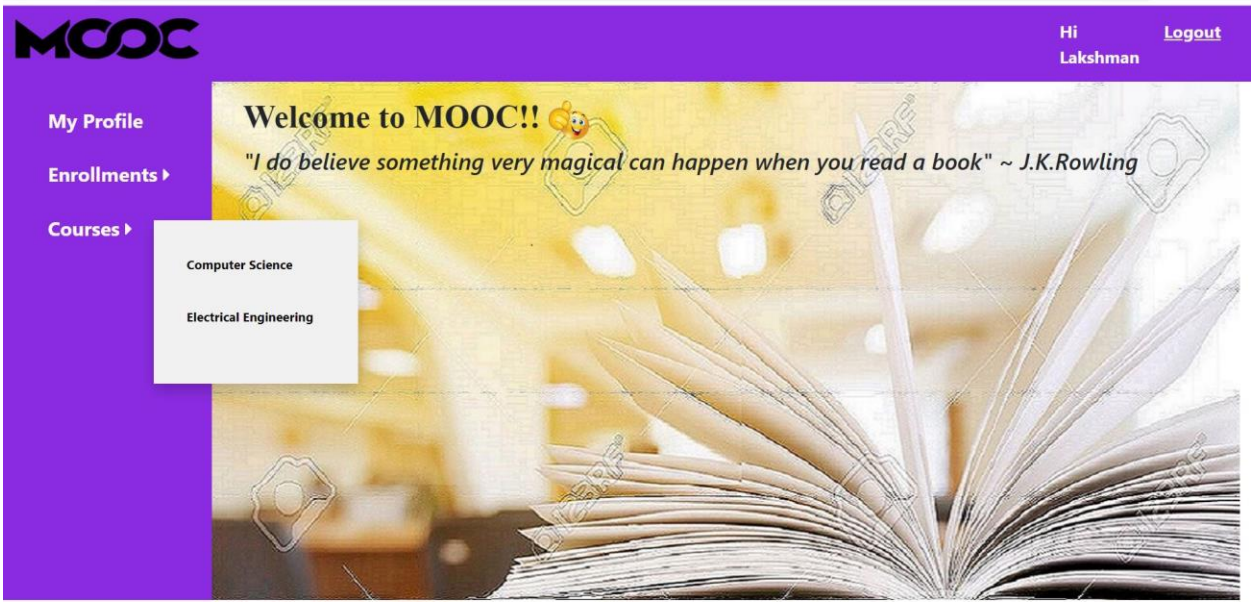
```
</head>
<body ng-app="myapph" ng-controller="initCtrl">
<div class="container-fluid" ng-init="Order=false Category=false" >
<div class="row hd" style="...">
<!--<strong class=" col-1" style="font-family: 'Times New Roman'">MOOC</strong>-->
    <img src="mooc1.png" width="150" height="50" class="col-2">
<span class = "offset-sm-7 offset-md-8 col-1">Hi {{user}}</span>
    <!--<span class = "col-1 fa fa-shopping-cart fa-lg" style="margin-top: 0.5%; margin-bottom: 0.5%" ></span>-->
    <a class = "col-sm-2 col-md-1 " href="{{log_url}}"><u>Logout</u></a>
</div>
    <div class="row">
    <div  class="col-2" style="...">
        <ul id="UL1" class="dropdown" style="...">
            <li >
                <a ng-click="load_content('P')" href="#" > My Profile</a>
            </li>
            <li>
                <a ng-click="Category=false;Order=!Order" href="#">Enrollments <i class="fa fa-caret-right"></i></a>
                <ul  ng-show="Order" class="dropdown-content order">
                    <li><a ng-click="load_content('R')" href="#">Recent Enrollments</a></li>
                    <li><a ng-click="load_content('A')" href="#">All Enrollments</a></li>
                </ul>
            </li>

            <li>
                <a href="#" ng-click="Order=false;Category=!Category">Courses <i class="fa fa-caret-right"></i></a>
                <ul  ng-show="Category" class="dropdown-content category">
                    <li><a ng-click="load_content('F')" href="#">Computer Science</a></li>
                    <li><a ng-click="load_content('S')" href="#">Electrical Engineering</a></li>

                </ul>

            </li>
        </ul>
    </div>
```



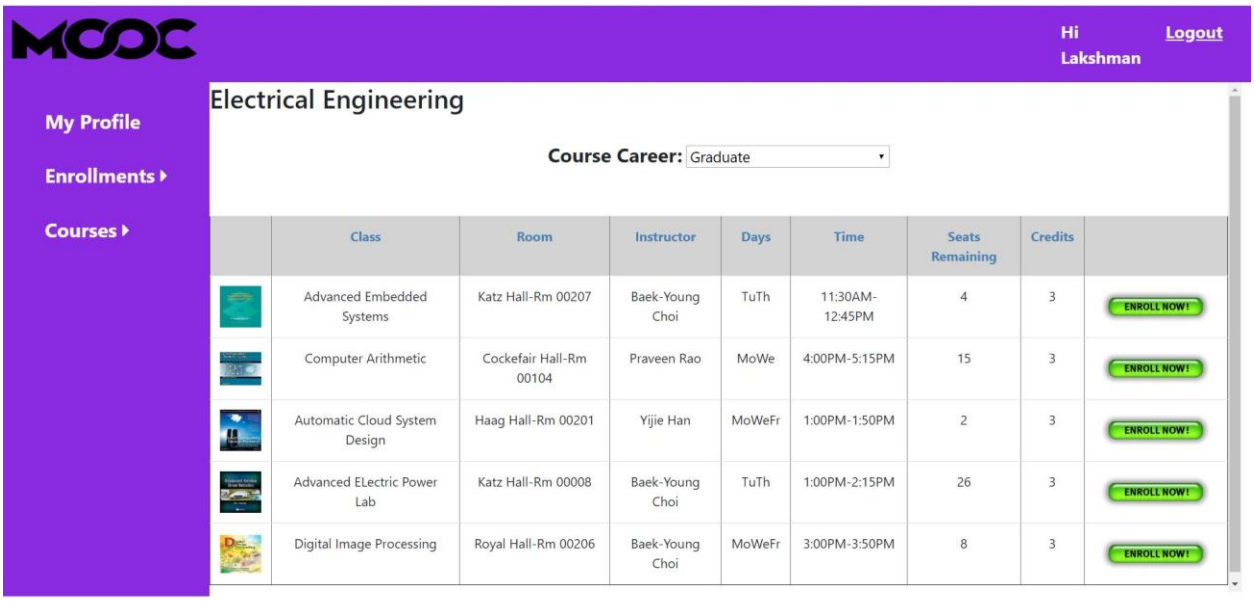
- In home page I have developed html pages for following.
- Enrollments have two catogiries.
    i. All enrollments
    ii. Recent Enrollments
- Course also have two fields.
    i. Computer Science
    ii. Electrical Engineering

- To get better understand output snippets for user-interface is shown below.



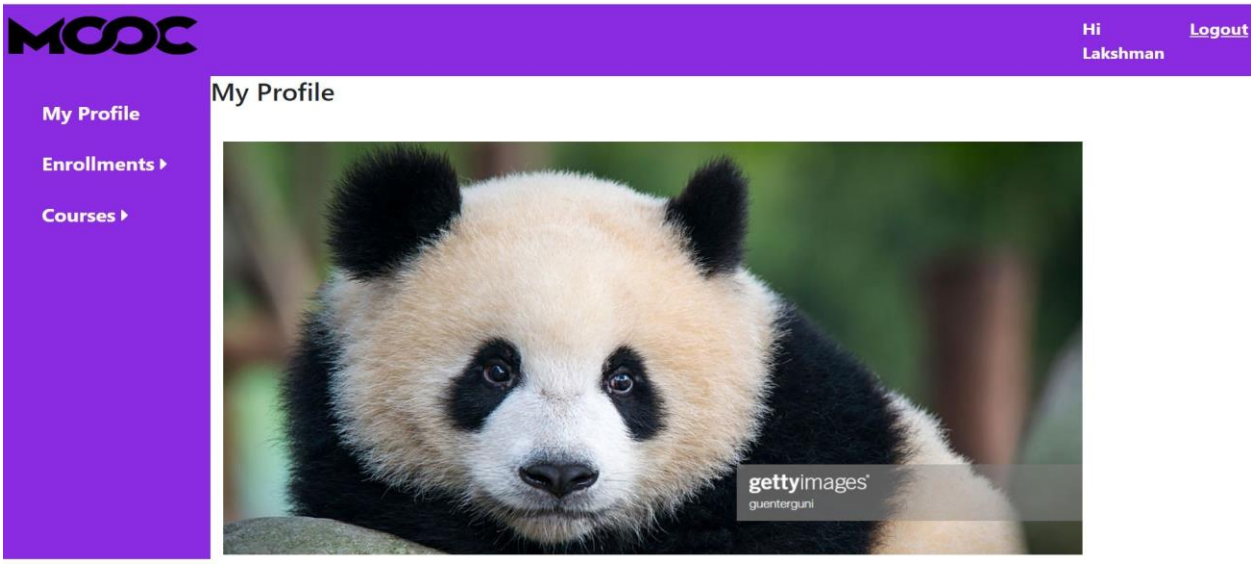


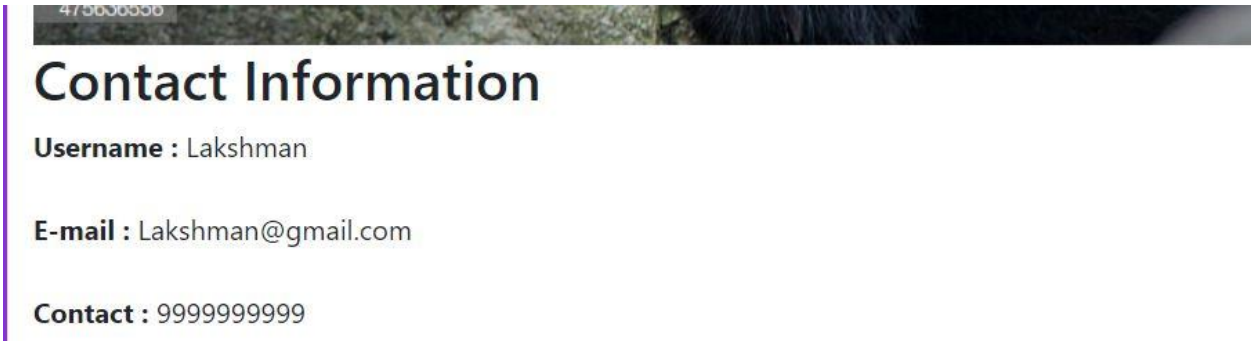
- Whenever user enters into course page it has graduate and under-graduate enrollments as drop down menu.



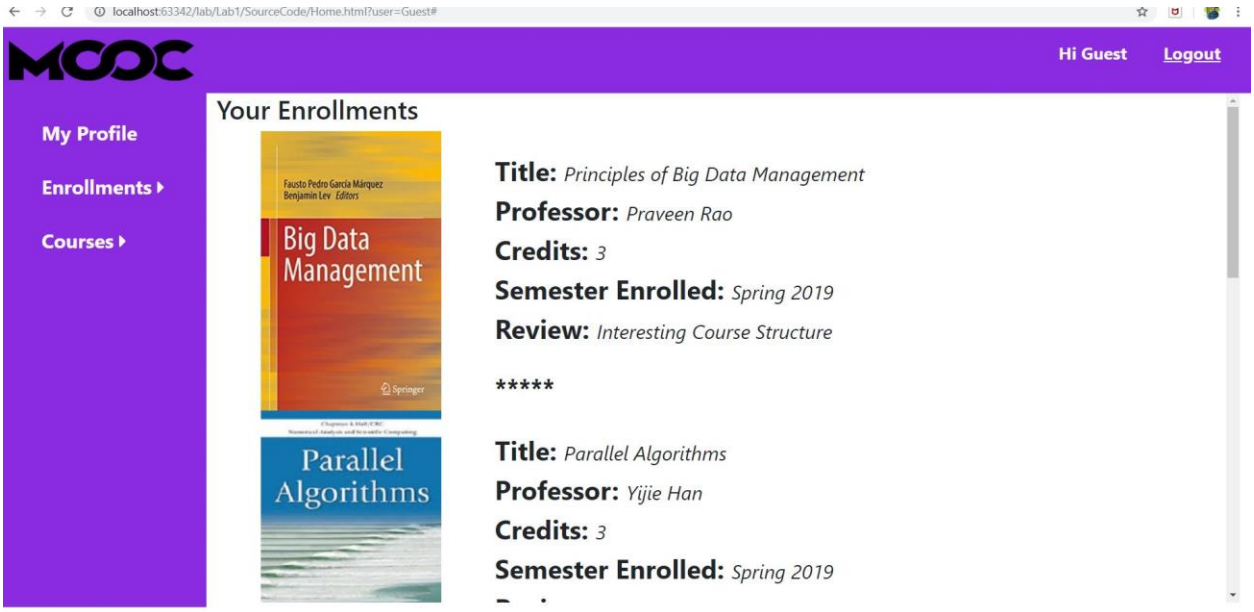
- If you click on enroll button ,the subject gets enrolled to enrollments.

- Coming to My Profile page it has user details like name,mail-id and contact number along with user image.

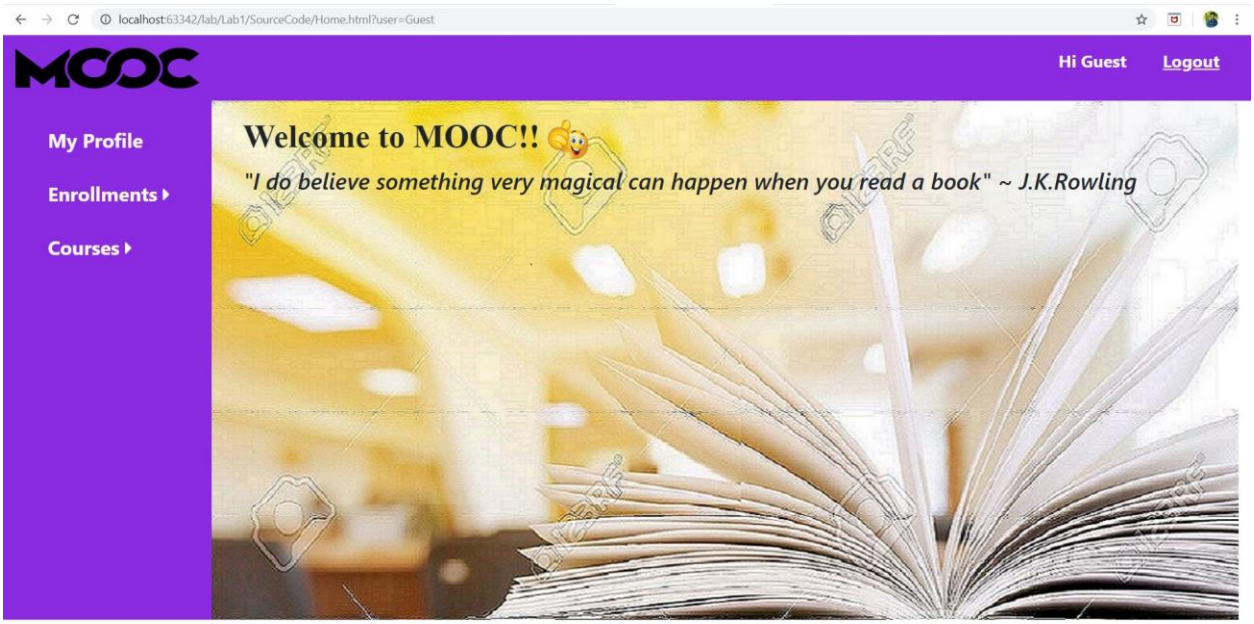


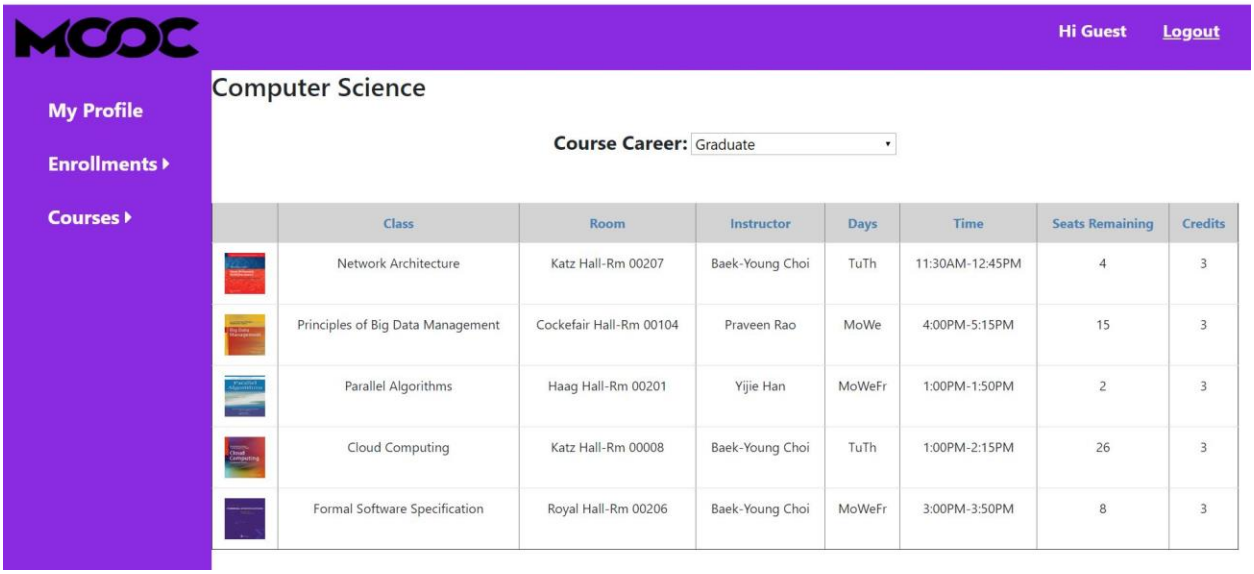


- If we go to enrollments page, it has recent enrollments and all enrollments already shown in above images.



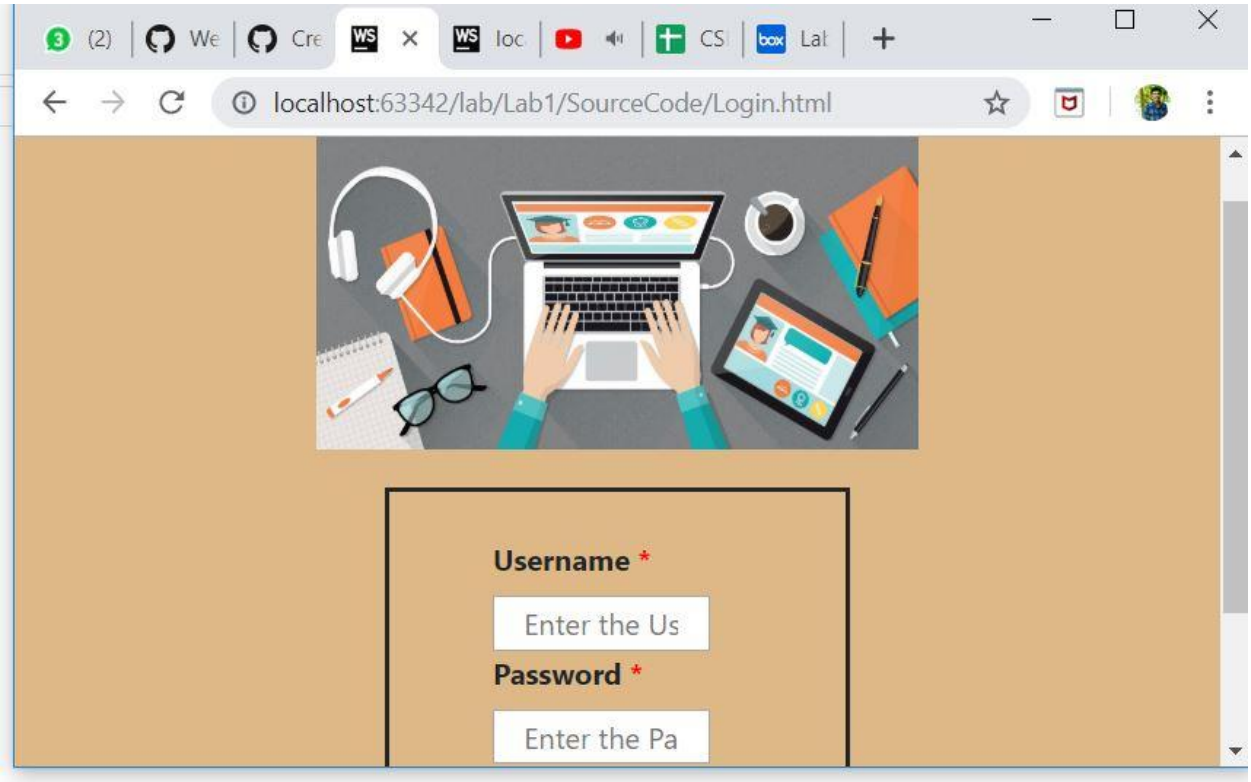
- You have option to login as guest but the guest will not able to enroll for courses but just it is showing the course details like course name,instructor name,credit hours and timings which are shown in below images.





	Class	Room	Instructor	Days	Time	Seats Remaining	Credits
	Network Architecture	Katz Hall-Rm 00207	Baek-Young Choi	TuTh	11:30AM-12:45PM	4	3
	Principles of Big Data Management	Cockefair Hall-Rm 00104	Praveen Rao	MoWe	4:00PM-5:15PM	15	3
	Parallel Algorithms	Haag Hall-Rm 00201	Yijie Han	MoWeFr	1:00PM-1:50PM	2	3
	Cloud Computing	Katz Hall-Rm 00008	Baek-Young Choi	TuTh	1:00PM-2:15PM	26	3
	Formal Software Specification	Royal Hall-Rm 00206	Baek-Young Choi	MoWeFr	3:00PM-3:50PM	8	3

- After finishing all the things when ever user or guest click on logout button it will redirect to login page.
- And finally The designed web-application is responsive has been tested.The corresponding snippet is shown below.



- Hence the creation of multi-page responsive web application is done using above programming elements.

Objective-2:

- Implementation of snake game using angular elements.
- The programming elements for this task is Angular-JS,html5 and CSS ,bootstrap.
- It will display the best score.
- The code snippets along with output snippets are shown below.
- The html page to implement user interface for snake game is given below.

```
<body ng-controller="snakeCtrl">

<div id="gameContainer">
    <div class="row header">
        <div class="column header-column" style="float: left;">
            <h3>Snake Game</h3>
        </div>
        <div class="column header-column" style="float: right; text-align: right;">
            <div class="scores">
                <span class="score">{{"Score: " + score}}</span><br>
                <span class="best-score" style="font-size: 12px;">{{"Best Score: " + bestScore}}</span>
            </div>
        </div>
    </div>

    <div style="margin-top: 5px;">
        <div class="row" style="height: 22px;" ng-repeat="column in board">
            <div class="column game-column"
                ng-style="{'background-color': setStyling($parent.$index, $index)}"
                ng-repeat="row in column track by $index"
            ></div>
        </div>
    </div>
    <div id="startGame" ng-click="startGame()">Start Game</div>
</div>

</body>

</html>
```

- It has ng-click button as start game to start the game.
- Corresponding java script page is shown below. And the ng-app is my-app and controller is my-controller to provide write access to code and $scope function is used to create score ,best score, update the snake after collision and create the fruit whenever snake eat existing fruit etc.

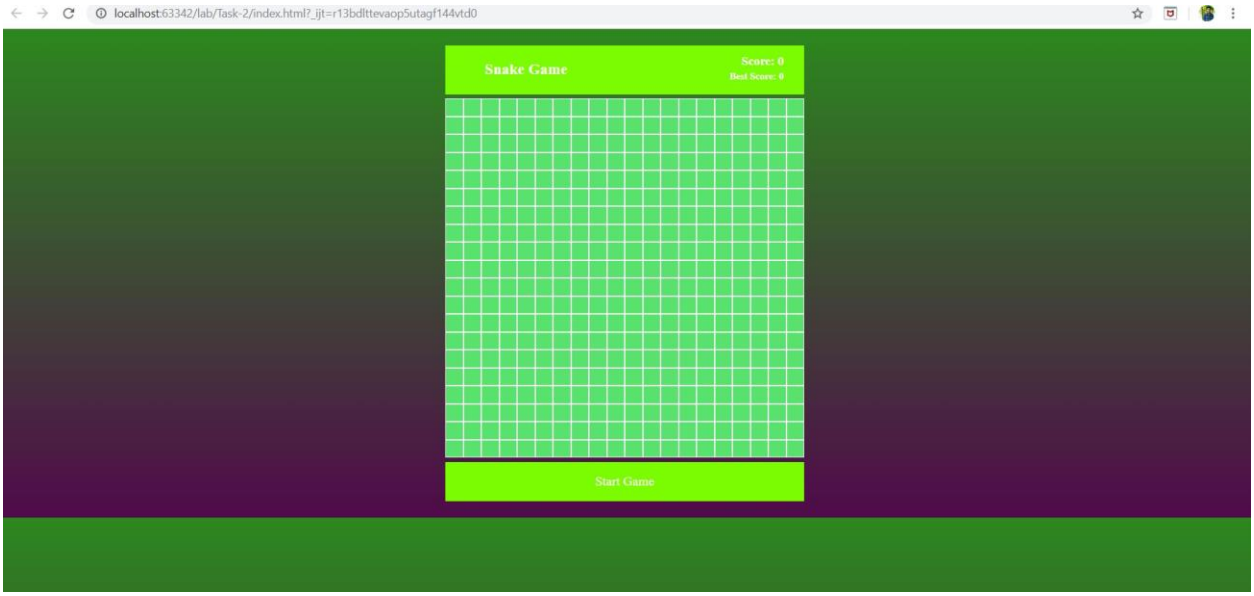
```
};

let interval, tempDirection, isGameOver;

$scope.score = 0;
$scope.bestScore = 0;

$scope.setStyling = function(col, row) {
    if (isGameOver)  {
        return COLORS.GAME_OVER;
    } else if (fruit.x == row && fruit.y == col) {
        return COLORS.FRUIT;
    } else if (snake.parts[0].x == row && snake.parts[0].y == col) {
        return COLORS.SNAKE_HEAD;
    } else if ($scope.board[col][row] === true) {
        return COLORS.SNAKE_BODY;
    }
    return COLORS.BOARD;
};

function update() {
    const newHead = getNewHead();

    if (boardCollision(newHead) || selfCollision(newHead)) {
        return gameOver();
    } else if (fruitCollision(newHead)) {
        eatFruit();
    }

    // Remove tail
```

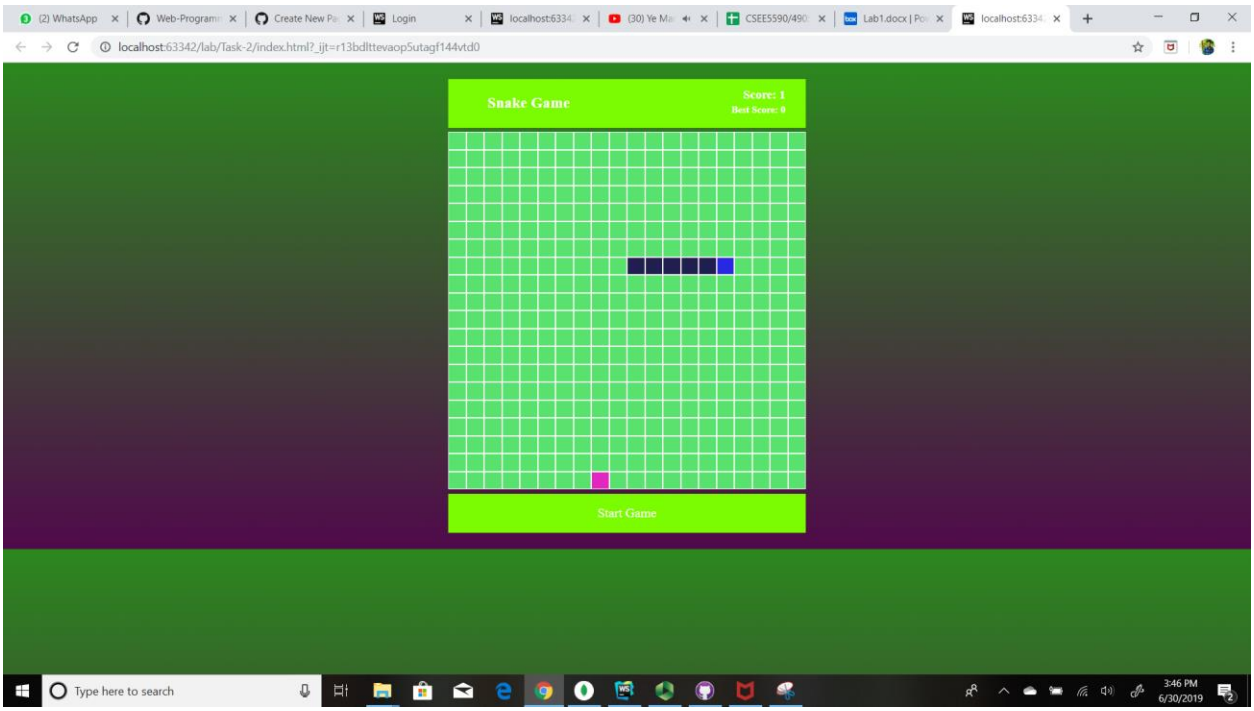
```
    }

    function eatFruit() {
        $scope.score++;

        // Grow by 1
        const tail = angular.copy(snake.parts[snake.parts.length - 1]);
        snake.parts.push(tail);
        resetFruit();

        if ($scope.score % 5 === 0) {
            interval -= 15;
        }
    }

    function gameOver() {
        isGameOver = true;

        if ($scope.score > $scope.bestScore) {
            $scope.bestScore = $scope.score;
        }

        $timeout(function() {
            isGameOver = false;
        },500);

        setupBoard();
    }

    function setupBoard() {
        $scope.board = [];
        for (let i = 0; i < BOARD_SIZE; i++) {
            $scope.board[i] = [];
```

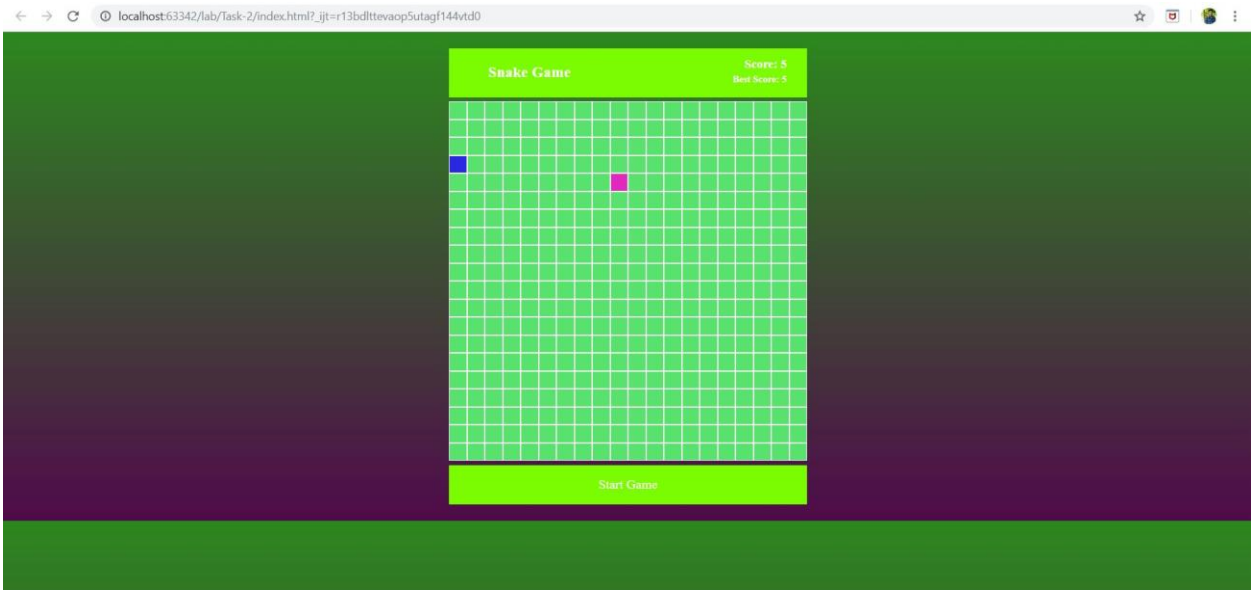
IDE and Plugin Updates
WebStorm is ready to update.

callback for controller() › startGame()

- Corresponding user interface is shown below.







- The score has been changed from one to five hence the best score replaced as five shown in above images.
- Hence implementation of snake game using angular-js has been done.

Objective-3:

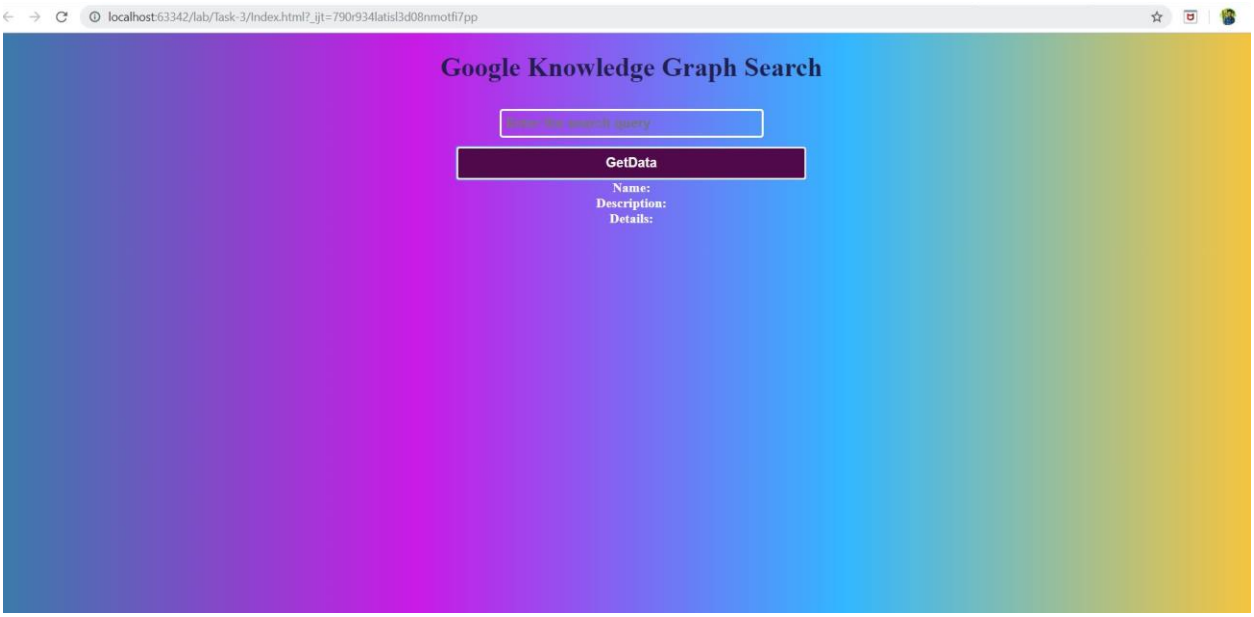
- To create an application which will display the description about the entities like people, places etc. that match certain criteria using Google knowledge graph search API.
- Code snippet for user-interface using html is given below. It has ng-click button get data, whenever user enter their interest as input query and click on get data it will display the details.

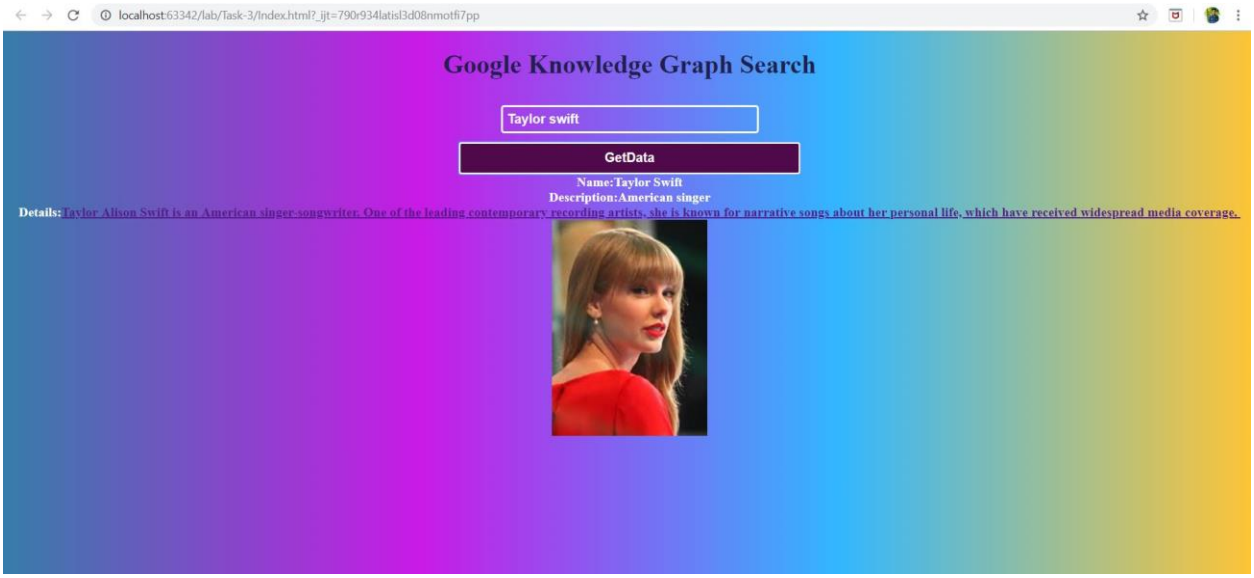
```
</head>
<body class="body" ng-app="myapp" ng-controller="mycntrl">


<div style="top: 200px">
    <div class="menu" id="div_SearchContainer" align="center">
        <h1 align="absolute" style="text-align: center; color: #211e50"> Google Knowledge Graph Search</h1>
        <input type="text" id="txt_searchFilter" placeholder="Enter the search query" /><br>
        <button id="btn_getReview" ng-click="Searchis()">GetData</button><br>
        <b style="color: #fbffff"> Name:{{name1}}</b><br>
        <b style="color: white">Description:{{desc1}}</b><br>
        <b style="color: white">Details:<a href="" ng-click="getaudio()">{{ddesc1}}</a></b><br>
        <img src ="{{imgurl}}">

    </div>
</div>
```

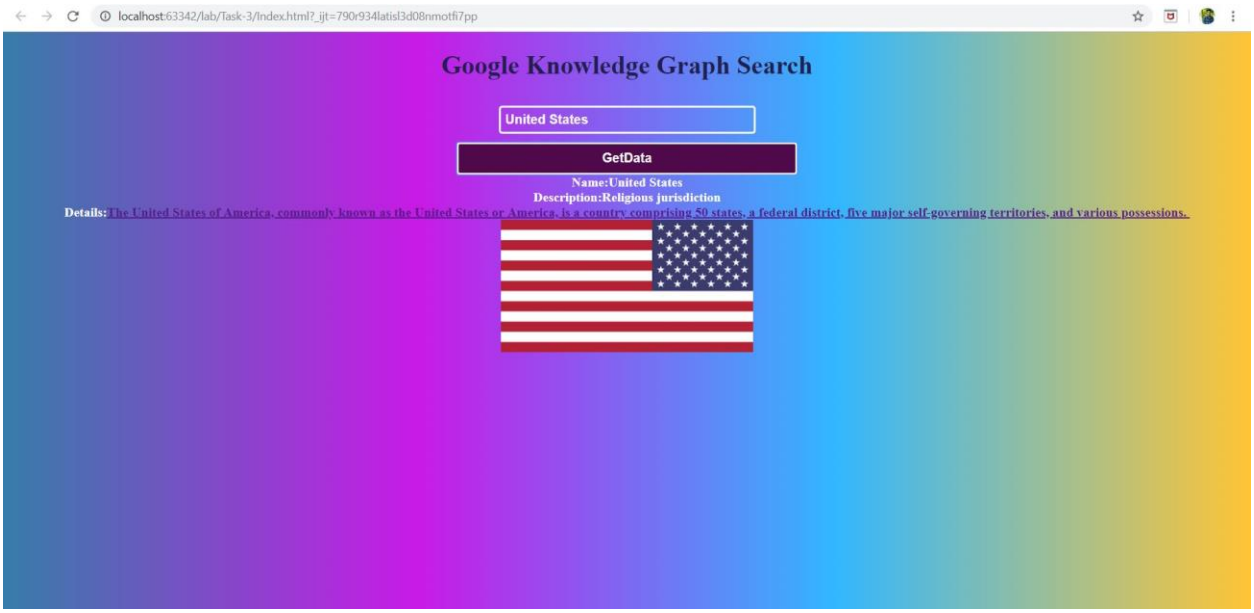
- Coming to java script page the user gets the details using get HTTP request using API URL and $scope is created the functions to display details about query as description, image and name.

```
/**
 * Created by lakshman on 6/29/2019.
 */
angular.module("myapp",[])
    .controller("mycntrl",function($scope,$http)
    {
        $scope.Searchis = function() {
            $scope.name1='';
            $scope.desc1='';
            $scope.ddesc1='';
            $scope.imgurl = ' ';
            var inp = document.getElementById( elementId: "txt_searchFilter").value;
            var result = $http.get("https://kgsearch.googleapis.com/v1/entities:search?query="+ inp +
                "&key=AIzaSyBB2HbgMzhKzC05E9ATKzCMKMJw6zepIXc&limit=1&indent=True");
            result.success(function(data)
                {
                    $scope.name1 ="Data Not Found";
                    if(data.itemListElement[0].result.name != null && data.itemListElement[0].result.name != " ")
                    {
                        $scope.name1 =data.itemListElement[0].result.name;
                    }
                    $scope.name1 = data.itemListElement[0].result.name;
                    $scope.desc1 = data.itemListElement[0].result.description;
                    $scope.ddesc1 = data.itemListElement[0].result.detailedDescription.articleBody;
                    $scope.imgurl = data.itemListElement[0].result.image.contentUrl;

                }
            )
            result.error(function (data) {
                alert("There was some error processing your request. Please try after some time");

            })
        }


    });
```

- User Interface for Task-3 is shown below.







- Hence It is about creating web application using google knowledge graph search API to display details based on user interest.

Conclusion:

- Finally, through lab got complete idea on usage of coding elements while creating web applications.

References:

- https://developers.google.com/knowledge-graph/
- https://www.mooc-list.com/tags/web-applications