WEB PROGRAMMING QUESTIONS

1) **Create a navbar, enable light and dark theme, the content should adjust from horizontal to vertical stack according to the screen size.**

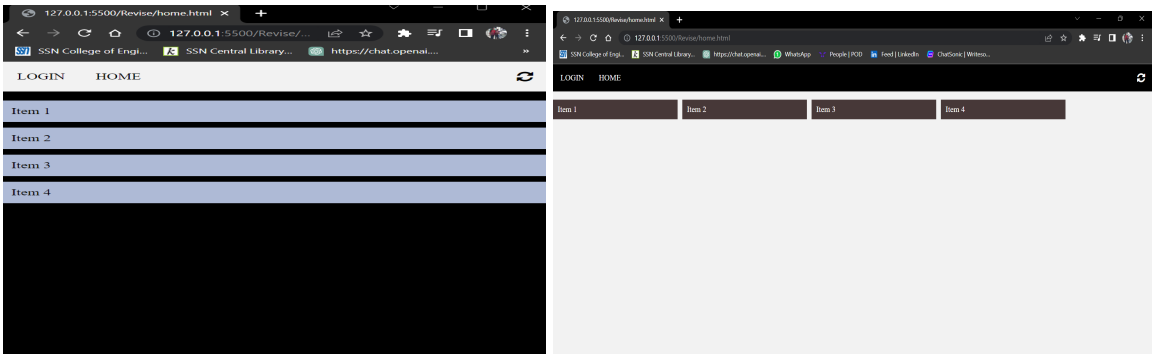
CODE:

```
<!DOCTYPE html>
<html>

<head>
    <title>Responsive Navbar</title>
    <style>
        /* Common styles for light and dark themes */
        .navbar {
            display: flex;
            justify-content: space-between;
            padding: 10px;
        }

        .navbar ul {
            margin: 0;
            padding: 0;
            display: flex;
            list-style: none;
        }

        .navbar li {
            margin-right: 10px;
        }

        .navbar a {
            text-decoration: none;
            color: #333;
            padding: 5px;
        }

        /* Light theme */
        .navbar-light {
            background-color: #f2f2f2;
```

```
            color: #333;
        }

        /* Dark theme */
        .navbar-dark {
            background-color: #333;
            color: #f2f2f2;
        }

        /* Responsive adjustments */
        @media (max-width: 768px) {
            .navbar {
                flex-direction: column;
            }

            .navbar ul {
                flex-direction: column;
            }

            .navbar li {
                margin-bottom: 10px;
                margin-right: 0;
            }
        }
    </style>
</head>

<body>
    <nav class="navbar navbar-light">
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
        <div>
            <button onclick="toggleTheme()">Toggle Theme</button>
        </div>
    </nav>

    <script>
        function toggleTheme() {
```

```
            var navbar = document.querySelector('.navbar');
            navbar.classList.toggle('navbar-light');
            navbar.classList.toggle('navbar-dark');
        }
    </script>
</body>
</html>
```

OUTPUT:



2) Teddy Bear using angular JS

CODE:

```
<!DOCTYPE html>
<html>
<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
  <style>
    #canvas {
      border: 1px solid black;
    }
    .container{
      position: absolute;
      top: 50%;
      left: 50%;
      transform: translate(-50%, -50%);
    }
    input{
      margin-bottom: 0.5rem;
```

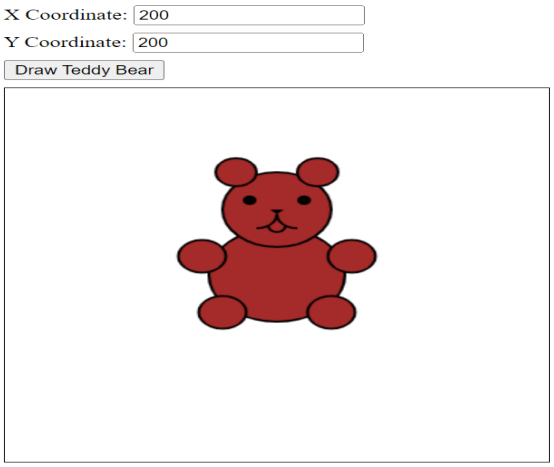
```
    }
    button{
      margin-bottom: 0.5rem;
    }
  </style>
</head>

<body  ng-app="teddyBearApp" ng-controller="TeddyBearController">
  <div class="container">
  <div>
    <label for="xCoordinate">X Coordinate:</label>
    <input type="number" id="xCoordinate" ng-model="x" placeholder="Enter X
Coordinate">
  </div>
  <div>
    <label for="yCoordinate">Y Coordinate:</label>
    <input type="number" id="yCoordinate" ng-model="y" placeholder="Enter Y
Coordinate">
  </div>
  <div>
    <button ng-click="drawTeddyBear()">Draw Teddy Bear</button>
  </div>
  <div>
    <canvas id="canvas" width="400" height="400"></canvas>
  </div>
  </div>

  <script>
    angular.module('teddyBearApp', [])
      .controller('TeddyBearController', function($scope) {
        var canvas = document.getElementById('canvas');
        var context = canvas.getContext('2d');

        $scope.drawTeddyBear = function() {
          context.clearRect(0, 0, canvas.width, canvas.height);

          // Draw body
          context.beginPath();
          context.arc($scope.x, $scope.y, 50, 0, 2 * Math.PI);
          context.fillStyle = 'brown';
          context.fill();
          context.strokeStyle = 'black'; // Border color
```

```
        context.lineWidth = 2; // Border width
        context.stroke();
        context.closePath();

        // Draw head
        context.beginPath();
        context.arc($scope.x, $scope.y - 70, 40, 0, 2 * Math.PI);
        context.fillStyle = 'brown';
        context.fill();
        context.strokeStyle = 'black'; // Border color
        context.lineWidth = 2; // Border width
        context.stroke();
        context.closePath();

        //Draw left ear
        context.beginPath();
        context.arc($scope.x - 30, $scope.y - 110, 15, 0, 2 * Math.PI);
        context.fillStyle = 'brown';
        context.fill();
        context.strokeStyle = 'black'; // Border color
    context.lineWidth = 2; // Border width
    context.stroke();
        context.closePath();

        //draw right ear
        context.beginPath();
        context.arc($scope.x + 30, $scope.y - 110, 15, 0, 2 * Math.PI);
        context.fillStyle = 'brown';
        context.fill();
        context.strokeStyle = 'black'; // Border color
    context.lineWidth = 2; // Border width
    context.stroke();
        context.closePath();

        // Draw eyes
        context.beginPath();
        context.arc($scope.x - 20, $scope.y - 80, 5, 0, 2 * Math.PI);
        context.fillStyle = 'black';
        context.fill();
        context.closePath();

        context.beginPath();
```

```
        context.arc($scope.x + 20, $scope.y - 80, 5, 0, 2 * Math.PI);
        context.fillStyle = 'black';
        context.fill();
        context.closePath();

        // Draw nose
        context.beginPath();
        context.moveTo($scope.x - 5, $scope.y - 70);
        context.lineTo($scope.x + 5, $scope.y - 70);
        context.lineTo($scope.x, $scope.y - 65);
        context.fillStyle = 'black';
        context.fill();
        context.closePath();

        //Draw hands
        context.beginPath();
        context.arc($scope.x - 55, $scope.y -20, 17.5, 0, 2*Math.PI);
        context.fillStyle = 'brown';
        context.fill();
        context.strokeStyle = 'black'; // Border color
    context.lineWidth = 2; // Border width
    context.stroke();
        context.closePath();
        context.beginPath();
        context.arc($scope.x + 55, $scope.y -20, 17.5, 0, 2*Math.PI);
        context.fillStyle = 'brown';
        context.fill();
        context.strokeStyle = 'black'; // Border color
    context.lineWidth = 2; // Border width
    context.stroke();
        context.closePath();

        //Draw feet
        context.beginPath();
        context.arc($scope.x - 40, $scope.y +40, 17.5, 0, 2*Math.PI);
        context.fillStyle = 'brown';
        context.fill();
        context.strokeStyle = 'black'; // Border color
    context.lineWidth = 2; // Border width
    context.stroke();
        context.closePath();
```

```
          context.beginPath();
          context.arc($scope.x + 40, $scope.y + 40, 17.5, 0, 2*Math.PI);
          context.fillStyle = 'brown';
          context.fill();
          context.strokeStyle = 'black'; // Border color
      context.lineWidth = 2; // Border width
      context.stroke();
          context.closePath();

          //Draw mouth
          context.beginPath();
          context.arc($scope.x-15, $scope.y-65, 15, 0, Math.PI/2);
          context.strokeStyle = 'black';
context.lineWidth = 2;
context.stroke();
context.closePath();
context.beginPath();
          context.arc($scope.x+15, $scope.y-65, 15, Math.PI, Math.PI/2, true);
          context.strokeStyle = 'black';
context.lineWidth = 2;
context.stroke();
context.closePath();
context.beginPath();
          context.arc($scope.x, $scope.y-52, 6,Math.PI, 0, true);
          context.strokeStyle = 'black';
context.lineWidth = 2;
context.stroke();
context.closePath();
        };
      });
  </script>
</body>

</html>
```

OUTPUT:



3) **Create an Angular Js application to make a canvas and draw a teddy bear by getting input coordinates and values to draw circles and lines from the user. In the application first get the size of the canvas from the user and then only show the canvas and hide the input fields for the canvas size and show the input fields for getting the parameters for drawing the lines and the circles (perform this using ng-if). Use separate buttons for drawing lines and circles using the input parameters got from the user. Bind all this functionality inside one controller and then draw the teddy bear as per your styling requirements. The teddy bear does not necessarily need to be coloured, just the outline is fine.**

```
<!DOCTYPE html>
<html ng-app="teddyBearApp">

<head>
    <meta charset="utf-8" content="width=device-width, height=device-height,
initial-scale=1.0">
    <title>Teddy Bear Drawing</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.8/angular.min.js"></s
cript>
    <script>
        angular.module('teddyBearApp', [])
            .controller('teddyBearCtrl', function ($scope) {
                $scope.canvas = {};
                $scope.canvasSizeSet = false;
                $scope.setCanvasSize = function () {
```

```
                    console.log($scope.canvas.height);
                    console.log($scope.canvas.width);
                    $scope.canvasSizeSet = true;
                };
                $scope.draw = function () {
                    var canvas = document.querySelector('canvas');
                    var ctx = canvas.getContext('2d');
                    ctx.clearRect(0, 0, $scope.canvas.width,
$scope.canvas.height);

                    var headstartX = $scope.canvas.width / 2;
                    var headstartY = $scope.canvas.height / 3;

                    var bodystartX = $scope.canvas.width / 2;
                    var bodystartY = $scope.canvas.height / 3 +
$scope.canvas.radius;

                    // Draw the head
                    ctx.beginPath();
                    ctx.arc(headstartX, headstartY, $scope.canvas.radius, 0,
Math.PI * 2);
                    ctx.stroke();

                    // Draw the ears
                    ctx.beginPath();
                    ctx.arc(headstartX - $scope.canvas.radius, headstartY -
$scope.canvas.radius, $scope.canvas.radius / 3, 0, Math.PI * 2);
                    ctx.stroke();
                    ctx.beginPath();
                    ctx.arc(headstartX + $scope.canvas.radius, headstartY -
$scope.canvas.radius, $scope.canvas.radius / 3, 0, Math.PI * 2);
                    ctx.stroke();

                    // Draw the body
                    ctx.beginPath();
                    ctx.moveTo(bodystartX, bodystartY);
                    ctx.lineTo(bodystartX, bodystartY + $scope.canvas.length);
                    ctx.stroke();

                    // Draw the arms
                    ctx.beginPath();
                    ctx.moveTo(bodystartX, bodystartY);
```

```
                    ctx.lineTo(bodystartX - $scope.canvas.length / 2,
bodystartY + $scope.canvas.length / 3);
                    ctx.moveTo(bodystartX, bodystartY);
                    ctx.lineTo(bodystartX + $scope.canvas.length / 2,
bodystartY + $scope.canvas.length / 3);
                    ctx.stroke();

                    // Draw the legs
                    ctx.beginPath();
                    ctx.moveTo(bodystartX, bodystartY + $scope.canvas.length);
                    ctx.lineTo(bodystartX - $scope.canvas.length / 2,
bodystartY + $scope.canvas.length + $scope.canvas.length / 3);
                    ctx.moveTo(bodystartX, bodystartY + $scope.canvas.length);
                    ctx.lineTo(bodystartX + $scope.canvas.length / 2,
bodystartY + $scope.canvas.length + $scope.canvas.length / 3);
                    ctx.stroke();

                }
            });
    </script>
    <style>
        html {
            width: 100vw;
            height: 100vh;
        }

        body {
            width: 100%;
            height: 100%;
            background-color: brown;
            font-family: 'Century Schoolbook', "serif";
            color: antiquewhite;
        }

        #canvas {
            background: ivory;
        }
    </style>
</head>

<body ng-controller="teddyBearCtrl">
    <div>
```

```
        <div ng-if="!canvasSizeSet">
            <label>Canvas Width:</label>
            <input type="number" ng-model="canvas.width" min="1" max="1500">
            <label>Canvas Height:</label>
            <input type="number" ng-model="canvas.height" min="1" max="700">
            <button ng-click="setCanvasSize()">Set Size</button>
        </div>
        <canvas id="canvas" ng-show="canvasSizeSet" width="{{canvas.width}}"
height="{{canvas.height}}"></canvas>
        <div ng-show="canvasSizeSet">
            <label>Line length(max:{{canvas.width/2}}):</label>
            <input type="number" ng-model="canvas.length" min="1"
max="{{canvas.width/2}}">
            <br>
            <label>Circle radius(max:{{canvas.length/2}})</label>
            <input type="number" ng-model="canvas.radius" min="1"
max="{{canvas.length}}/2">
            <button ng-click="draw()">Set parameters</button>
            {{canvas.width}} x {{canvas.height}} px
        </div>
    </div>
</body>

</html>
```

4) **Song playlist using AngularJS: Create a main playlist with the fields Song Name, Song Genre and Song Author from an input form. You also want to search the list using genre and author fields. If any of the songs from the filtered results is clicked, that song should be added to your personal playlist. Display all the playlists as lists. Use AngularJS directives like ng-click, custom services and custom filters to do these. No need to use PHP or JSON files. Use AngularJS lists to store the song details.**

```
<!DOCTYPE html>
<html ng-app="myApp">

<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular.min.js"></s
cript>
</head>
```

```
<body ng-controller="myController">
  <h2>Main Playlist</h2>
  <form>
    <label>Song Name:</label>
    <input type="text" ng-model="newSong.name">
    <label>Song Genre:</label>
    <input type="text" ng-model="newSong.genre">
    <label>Song Author:</label>
    <input type="text" ng-model="newSong.author">
    <button ng-click="addSong()">Add</button>
  </form>

  <h2>Main Playlist</h2>
  <ul>
    <li ng-repeat="song in songs">{{ song.name }} - {{ song.genre }} - {{
song.author }}</li>
  </ul>

  <h2>Search</h2>
  <label>Search by Genre:</label>
  <input type="text" ng-model="searchGenre">
  <label>Search by Author:</label>
  <input type="text" ng-model="searchAuthor">

  <h2>Filtered Songs</h2>
  <ul>
    <li ng-repeat="song in filteredSongs = songs | filter: {genre:
searchGenre, author: searchAuthor}">
      {{ song.name }} - {{ song.genre }} - {{ song.author }}
      <button ng-click="addToPlaylist(song)">Add to Personal Playlist</button>
    </li>
  </ul>

  <h2>Personal Playlist</h2>
  <ul>
    <li ng-repeat="song in personalPlaylist">{{ song.name }} - {{ song.genre
}} - {{ song.author }}</li>
  </ul>

  <script>
    angular.module('myApp', [])
```

```
      .controller('myController', function($scope, playlistService) {
        $scope.songs = playlistService.getMainPlaylist();
        $scope.newSong = {};

        $scope.addSong = function() {
          playlistService.addSong($scope.newSong);
          $scope.newSong = {};
        };

        $scope.addToPlaylist = function(song) {
          playlistService.addToPlaylist(song);
        };

        $scope.personalPlaylist = playlistService.getPersonalPlaylist();
      })
      .service('playlistService', function() {
        var mainPlaylist = [];
        var personalPlaylist = [];

        this.addSong = function(song) {
          mainPlaylist.push(song);
        };

        this.addToPlaylist = function(song) {
          personalPlaylist.push(song);
        };

        this.getMainPlaylist = function() {
          return mainPlaylist;
        };

        this.getPersonalPlaylist = function() {
          return personalPlaylist;
        };
      });
  </script>
</body>
</html>
```

5) AJAX get input from file

```html
<!DOCTYPE html>
<html>

<head>
    <title>Song Information</title>
</head>
<body>
    <div id="song-info">
    </div>

    <script>
        var xhr = new XMLHttpRequest();
        var filePath = 'musics.json';
        xhr.open('GET', filePath, true);
        xhr.responseType = 'json';
        xhr.onload = function () {
            if (xhr.status === 200) {
                var musicData = xhr.response;
                var songs = musicData; // Assume the JSON file contains an array of songs

                // Get the song-info div element
                var songInfoDiv = document.getElementById('song-info');

                // Iterate through each song and create HTML elements dynamically
                songs.forEach(song => {
                    var songName = song.songName;
                    var lyrics = song.lyrics;

                    var songElement = document.createElement('div');
                    songElement.innerHTML = '<h2>' + songName + '</h2><p>' + lyrics + '</p>';

                    // Append the song element to the song-info div
                    songInfoDiv.appendChild(songElement);
                });
            }
        };
        xhr.send();
    </script>
</body>
```

```
</html>
```

6) **Append table items according to entered details of the song(name, author) and display & apply custom filter using angularjs**

```
<!-- Using AngularJS -->
<!DOCTYPE html>
<html ng-app="myApp">

<head>
    <title>Song Information</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>

<body ng-controller="SongController">
    <div>
        <label for="songName">Song Name:</label>
        <input type="text" id="songName" ng-model="newSongName">
        <label for="author">Author:</label>
        <input type="text" id="author" ng-model="newAuthor">
        <button ng-click="addSong()">Add Song</button>
    </div>

    <table>
        <thead>
            <tr>
                <th>Song Name</th>
                <th>Author</th>
            </tr>
        </thead>
        <tbody>
            <tr ng-repeat="song in songs | filter: customFilter">
                <td>{{ song.name }}</td>
                <td>{{ song.author }}</td>
            </tr>
        </tbody>
    </table>

    <script>
```

```
        var app = angular.module('myApp', []);

        app.controller('SongController', function ($scope) {
            $scope.songs = [
                { name: 'Song A', author: 'Author A' },
                { name: 'Song B', author: 'Author B' },
                { name: 'Song C', author: 'Author C' }
            ];

            $scope.addSong = function () {
                var newSong = {
                    name: $scope.newSongName,
                    author: $scope.newAuthor
                };
                $scope.songs.push(newSong);
                $scope.newSongName = '';
                $scope.newAuthor = '';
            };

            $scope.customFilter = function (song) {
                // Implement your custom filter logic here
                // For example, filtering based on song name starting with 'A'
                return song.name.startsWith('A');
            };
        });
    </script>
</body>
</html>
```

7) Navigation menu using angular, other question was HTML5 elements in webpage

```
<!DOCTYPE html>
<html ng-app="myApp">

<head>
    <title>AngularJS Navigation Menu</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <style>
```

```
        .nav-menu {
            background-color: #f2f2f2;
            padding: 10px;
            display: flex;
            justify-content: center;
        }

        .nav-menu ul {
            list-style-type: none;
            margin: 0;
            padding: 0;
            display: flex;
        }

        .nav-menu li {
            margin-right: 10px;
        }

        .nav-menu a {
            text-decoration: none;
            color: #333;
            padding: 5px;
        }
    </style>
</head>

<body ng-controller="NavController">
    <div class="nav-menu">
        <ul>
            <li ng-repeat="item in menuItems">
                <a href="#/{{item.url}}">{{item.label}}</a>
            </li>
        </ul>
    </div>

    <script>
        var app = angular.module('myApp', []);

        app.controller('NavController', function ($scope) {
            $scope.menuItems = [
                { label: 'Home', url: 'home' },
                { label: 'About', url: 'about' },
```

```
                { label: 'Services', url: 'services' },
                { label: 'Contact', url: 'contact' }
            ];
        });
    </script>
</body>

</html>
```

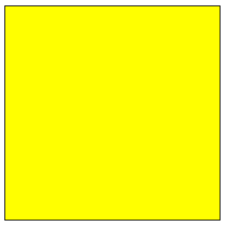
8) **Angular js animation - create two checkboxes with labels and create a div box with black border, when the first checkbox is checked the div box should move up and when unchecked should come down. When the second box is clicked the div box color should turn yellow. Make the transitions smooth.**

```
<!DOCTYPE html>
<html ng-app="myApp">

<head>
    <title>AngularJS Animation</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular-animate.js"
></script>
    <style>
        .container {
            width: 200px;
            height: 200px;
            border: 1px solid black;
            background-color: white;
            transition: all 0.5s ease-in-out;
            position: relative;
            top: 300px;
        }

        .container.move-up {
            transform: translateY(-100%);
        }

        .container.yellow {
```

```
            background-color: yellow;
        }

        .checkbox-label {
            margin-bottom: 10px;
        }
    </style>
</head>

<body ng-controller="AnimationController">
    <div class="checkbox-label">
        <label>
            <input type="checkbox" ng-model="moveUp"> Move Up
        </label>
    </div>
    <div class="checkbox-label">
        <label>
            <input type="checkbox" ng-model="turnYellow"> Turn Yellow
        </label>
    </div>
    <div class="container" ng-class="{ 'move-up': moveUp, 'yellow': turnYellow
}"></div>

    <script>
        var app = angular.module('myApp', ['ngAnimate']);

        app.controller('AnimationController', function ($scope) {
            $scope.moveUp = false;
            $scope.turnYellow = false;
        });
    </script>
</body>

</html>
```

Move Up

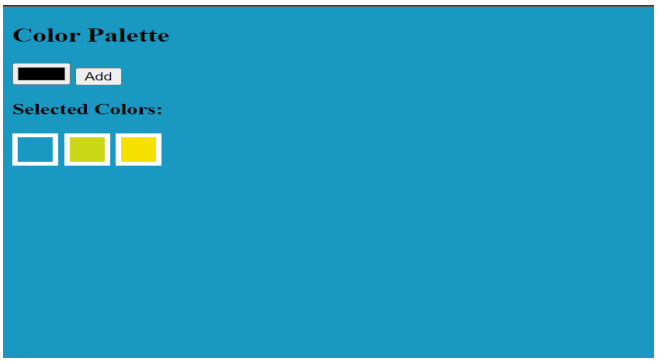
Turn Yellow



9) Using ng repeat create a colour palette as input type="color" select a colour in the palette and click the button 'add' to add it to a list of selected colors.

```
<!DOCTYPE html>
<html ng-app="colorPaletteApp">

<head>
    <title>Color Palette</title>
    <style>
        .color-container {
            display: inline-block;
            width: 30px;
            height: 30px;
            margin-right: 5px;
            border: 5px solid white;
            cursor: pointer;
        }
    </style>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.8.2/angular.min.js"><
/script>
</head>

<body ng-controller="ColorPaletteController" ng-style="{ 'background-color':
selectedBackground }">
    <h2>Color Palette</h2>

    <div>
        <input type="color" ng-model="selectedColor">
        <button ng-click="addColor()">Add</button>
    </div>
```

```
    <h3>Selected Colors:</h3>
    <div>
        <div class="color-container" ng-repeat="color in selectedColors"
ng-style="{ 'background-color': color }"
            ng-click="updateBackground(color)"></div>

    </div>

    <script>
        angular.module('colorPaletteApp', [])
            .controller('ColorPaletteController', ['$scope', function ($scope)
{
                $scope.selectedColor = '';
                $scope.selectedColors = [];
                $scope.selectedBackground = '';

                $scope.addColor = function () {
                    if ($scope.selectedColor) {
                        $scope.selectedColors.push($scope.selectedColor);
                        $scope.selectedColor = '';
                    }
                };

                $scope.updateBackground = function (color) {
                    $scope.selectedBackground = color;
                };
            }]);
    </script>
</body>

</html>
```



10) **create a word game using angular js. the user has to guess 6 words in 12 tries. display the correctly guessed words.**

```
<!DOCTYPE html>
<html ng-app="myApp">


<head>
    <title>Word Game</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <style>
        .correct-word {
            color: green;
            font-weight: bold;
        }


        .incorrect-word {
            color: red;
            font-weight: bold;
        }
    </style>
</head>


<body ng-controller="WordGameController">
    <h2>Word Game</h2>
    <div>
        <p>Guess the following words:</p>
        <ul>
            <li ng-repeat="word in words">
                <span ng-class="{'correct-word': word.guessed,
'incorrect-word': word.failed || gameOver}">
                    {{ word.guessed || gameOver ? word.value : word.hint}}
                </span>
            </li>
        </ul>
    </div>
    <div>
        <p>Guess a word ({{ remainingTries }} tries remaining):</p>
```

```
        <div ng-repeat="word in words">
            Word: {{ $index+1 }}
            <input type="text" ng-model="word.guess" ng-disabled="word.guessed
|| gameOver">
            <button ng-click="checkGuess(word)" ng-disabled="word.guessed ||
gameOver">Submit</button>
        </div>
    </div>
    <div>
        <p>Correctly guessed words:</p>
        <ul>
            <li ng-repeat="word in correctlyGuessedWords">{{ word }}</li>
        </ul>
    </div>


    <script>
        var app = angular.module('myApp', []);


        app.controller('WordGameController', function ($scope) {
            $scope.guessCount = 0;
            $scope.remainingTries = 12;

            $scope.gameOver = false;
            $scope.correctlyGuessedWords = [];

            $scope.words = [
                { value: 'apple', guessed: false, failed: false, hint: 'A
fruit' },
                { value: 'banana', guessed: false, failed: false, hint: 'A
yellow fruit' },
                { value: 'orange', guessed: false, failed: false, hint: 'A
citrus fruit' },
                { value: 'grape', guessed: false, failed: false, hint: 'A
small fruit in bunches' },
                { value: 'kiwi', guessed: false, failed: false, hint: 'A green
fruit with brown skin' },
                { value: 'melon', guessed: false, failed: false, hint: 'A
large fruit with juicy flesh' }
            ];
```

```
            $scope.checkGuess = function (word) {

                var guessedWord = word.guess.toLowerCase();

                if (guessedWord === word.value) {
                    word.guessed = true;
                    word.failed = false;
                    $scope.correctlyGuessedWords.push(guessedWord);
                    word.guess = '';
                    $scope.guessCount++;
                }
                else {
                    word.guess = '';
                    word.failed = true;
                }


                $scope.remainingTries--;

                if ($scope.remainingTries <= 0 || $scope.guessCount >= 6) {
                    $scope.gameOver = true;
                    alert('Game Over');
                    return;
                }

            };
        });
    </script>
</body>
</html>
```

11) **Get past 3 months employee performance score out of 100 and see if there is a chance of the employee being fired. If the average is greater than 60, the employee won't be fired, or else there is a chance of firing. Do this using services in angularjs.**

```
<!DOCTYPE html>
<html ng-app="employeeApp">
<head>
  <title>Employee Performance</title>
```

```
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.8.2/angular.min.js"><
/script>
</head>
<body ng-controller="EmployeeController">

  <div>
    <label>Performance Score (Past 3 months):</label>
    <input type="number" ng-model="scores[0]">
    <input type="number" ng-model="scores[1]">
    <input type="number" ng-model="scores[2]">
    <button ng-click="calculateAverage()">Calculate</button>
  </div>

  <div ng-if="averageScore !== null">
    <p>Average Score: {{ averageScore }}</p>
    <p ng-if="firingPossible">There is a chance of firing.</p>
    <p ng-if="!firingPossible">The employee won't be fired.</p>
  </div>

  <script>
    angular.module('employeeApp', [])
  .service('EmployeeService', function() {
    this.calculateAverageScore = function(scores) {
      var sum = 0;
      for (var i = 0; i < scores.length; i++) {
        sum += scores[i];
      }
      return sum / scores.length;
    };

    this.isFiringPossible = function(scores) {
      var averageScore = this.calculateAverageScore(scores);
      return averageScore <= 60;
    };
  });
  angular.module('employeeApp')
  .controller('EmployeeController', ['$scope', 'EmployeeService',
function($scope, EmployeeService) {
    $scope.scores = [];
    $scope.averageScore = null;
    $scope.firingPossible = false;
```

```
    $scope.calculateAverage = function() {
      $scope.averageScore =
EmployeeService.calculateAverageScore($scope.scores);
      $scope.firingPossible = EmployeeService.isFiringPossible($scope.scores);
    };
  }]);
  </script>
</body>
</html>
```

12) display the skyline of a city. a sky line is the silhouette made by the skyscrapers of a city. get the position and height of each skyscraper from the user.

```
<!DOCTYPE html>
<html>

<head>
    <title>City Skyline</title>
    <style>
        #skyline {
            width: 100%;
            height: 400px;
            background-color: lightblue;
            position: relative;
        }

        .building {
            position: absolute;
            bottom: 0;
            width: 50px;
            background-color: black;
        }
    </style>
</head>

<body>
    <h2>City Skyline</h2>
    <div id="skyline"></div>

    <script>
```

```
        function addBuilding() {
            var positionInput = document.getElementById('position');
            var heightInput = document.getElementById('height');
            var position = parseInt(positionInput.value);
            var height = parseInt(heightInput.value);

            if (isNaN(position) || isNaN(height) || position < 0 || height <
0) {
                alert('Please enter valid position and height values.');
                return;
            }

            var skyline = document.getElementById('skyline');
            var building = document.createElement('div');
            building.className = 'building';
            building.style.left = position + 'px';
            building.style.height = height + 'px';
            skyline.appendChild(building);

            positionInput.value = '';
            heightInput.value = '';
        }
    </script>

    <h3>Add Skyscraper</h3>
    <label for="position">Position:</label>
    <input type="number" id="position" placeholder="Enter position" required>
    <br>
    <label for="height">Height:</label>
    <input type="number" id="height" placeholder="Enter height" required>
    <br>
    <button onclick="addBuilding()">Add</button>
</body>

</html>
```

City Skyline

Add Skyscraper

Position: Enter position
Height: Enter height
Add

13) Create routing between a main page, a page about solar eclipse and a page about water using AngularJS in such a way that all pages are accessible from other pages.

```
//index.html
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular-route.min.j
s"></script>
  <link rel="stylesheet" href="style.css">
</head>
<body ng-app="navApp">
  <header>
```

```
    <nav class="nav">
      <a href="#!/main">Main</a>
      <a href="#!/solar">Solar Eclipse</a>
      <a href="#!/water">Water</a>
    </nav>
  </header>
  <div ng-view></div>
  <script>
    var app = angular.module('navApp', ['ngRoute']);
    app.config(function($routeProvider) {
      $routeProvider
        .when('/main', {
          templateUrl: 'main.html'
        })
        .when('/solar', {
          templateUrl: 'solar.html'
        })
        .when('/water', {
          templateUrl: 'water.html'
        })
        .otherwise({
          redirectTo: '/main'
        });
    });
  </script>

</body>
</html>
```

```
//main.html
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-aweso
me.min.css">
        <link rel="stylesheet" href="style.css">
    </head>
```

```
    <body>
        <p>This is the main page</p>
    </body>
</html>
```

```
//solar.html
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-aweso
me.min.css">
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <p>This page is about solar eclipse.</p>
    </body>
</html>
```

```
//water.html
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-aweso
me.min.css">
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
            <p>This page is about water</p>
    </body>
</html>
```

```
//style.css
```

```css
*{
    margin: 0;
    padding: 0;
}

html{
    background-color: #f2f2f2;
    color: black;
}

.nav{
    background-color: black;
    color: #f2f2f2;
    padding: 1rem;
    margin-bottom: 1rem;
}

.nav a{
    padding: 1rem;
    text-decoration: none;
    color: #f2f2f2;
}
```

14) Create an EMI calculator using plain HTML,CSS and JavaScript. Later use angularJS to filter the rows in the table that are after a certain date.

```html
<!DOCTYPE html>
<html>

<head>
    <title>EMI Calculator</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        .container {
            max-width: 500px;
            margin: 0 auto;
        }
```

```
        h1 {
            text-align: center;
            margin-top: 20px;
        }

        .form-group {
            margin-bottom: 10px;
        }

        .form-group label {
            display: inline-block;
            width: 100px;
            font-weight: bold;
        }

        .form-group input[type="number"] {
            width: 200px;
            padding: 5px;
        }

        .form-group .result {
            font-weight: bold;
            margin-left: 10px;
        }

        .table-container {
            margin-top: 20px;
        }

        .table-container table {
            width: 100%;
            border-collapse: collapse;
        }

        .table-container table th,
        .table-container table td {
            padding: 8px;
            border-bottom: 1px solid #ddd;
        }

        .table-container table th {
            background-color: #f5f5f5;
```

```
            font-weight: bold;
        }

        .filter-container {
            margin-top: 20px;
        }

        .filter-container label {
            margin-right: 10px;
        }
    </style>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <script>
        var app = angular.module('EMICalculator', []);

        app.controller('EMICalculatorController', function ($scope, $filter) {
            $scope.emis = [];
            $scope.principal = 0;
            $scope.rate = 0;
            $scope.months = 0;
            $scope.filterDate = '';

            $scope.calculateEMI = function () {
                $scope.emis = [];

                if ($scope.principal <= 0 || $scope.rate <= 0 || $scope.months
<= 0) {
                    return;
                }

                var emiAmount = calculateEMIAmount($scope.principal,
$scope.rate, $scope.months);

                var currentDate = new Date();
                var currentMonth = currentDate.getMonth() + 1;
                var currentYear = currentDate.getFullYear();

                for (var i = 1; i <= $scope.months; i++) {
                    var date = new Date(currentYear, currentMonth - 1 + i, 1);
                    var formattedDate = $filter('date')(date, 'MMM yyyy');
```

```
                    var emi = {
                        month: i,
                        date: formattedDate,
                        amount: emiAmount
                    };
                    $scope.emis.push(emi);
                }
            };

            $scope.reset = function () {
                $scope.emis = [];
                $scope.principal = 0;
                $scope.rate = 0;
                $scope.months = 0;
            };

            $scope.filterByDate = function (emi) {
                if (!$scope.filterDate) {
                    return true;
                }

                var emiDate = new Date(emi.date);
                var filterDate = new Date($scope.filterDate);

                return emiDate >= filterDate;
            };

            function calculateEMIAmount(principal, rate, months) {
                var monthlyRate = rate / 1200; // rate is in percentage per
annum
                var emi = principal * monthlyRate * Math.pow(1 + monthlyRate,
months) / (Math.pow(1 + monthlyRate, months) - 1);
                return emi.toFixed(2);
            }
        });
    </script>
</head>

<body ng-app="EMICalculator">
    <div class="container" ng-controller="EMICalculatorController">
        <h1>EMI Calculator</h1>
```

```
        <div class="form-group">
            <label for="principal">Principal Amount:</label>
            <input type="number" id="principal" ng-model="principal">
        </div>

        <div class="form-group">
            <label for="rate">Rate of Interest:</label>
            <input type="number" id="rate" ng-model="rate">
        </div>

        <div class="form-group">
            <label for="months">Loan Term (Months):</label>
            <input type="number" id="months" ng-model="months">
        </div>

        <div class="form-group">
            <button ng-click="calculateEMI()">Calculate</button>
            <button ng-click="reset()">Reset</button>
        </div>

        <div class="result" ng-show="emis.length > 0">
            <p ng-repeat="emi in emis" ng-show="filterByDate(emi)">{{ emi.date
}} - EMI Amount: {{ emi.amount }}</p>
        </div>

        <div class="filter-container">
            <label for="filterDate">Filter by Date:</label>
            <input type="date" id="filterDate" ng-model="filterDate">
        </div>
    </div>
</body>

</html>
```

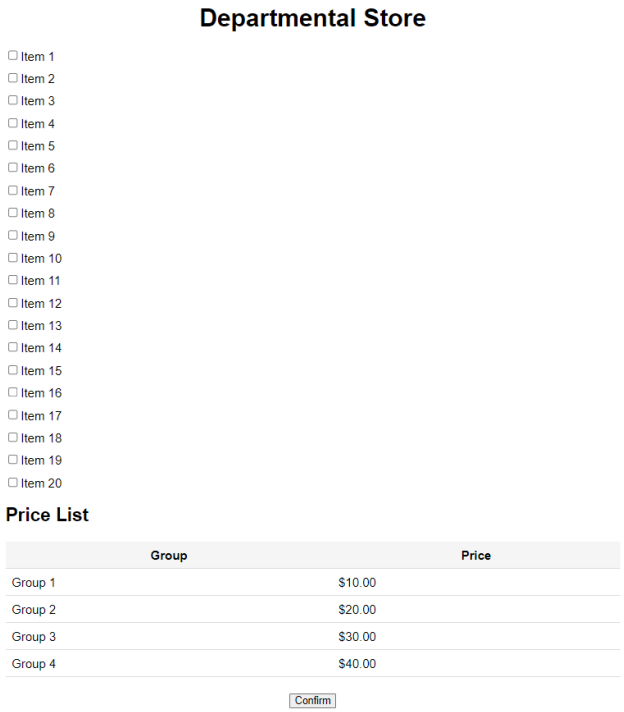
15) Create a website for a departmental store. Display 20 items as a list, these items are grouped into four groups. Items in a group have the same price. The groups the items belong to along with prices are displayed way below link the items from the list to this price list. Below the price list, there should be a confirm button on clicking this button, prices of selected items should be displayed. Use html5, css3, javascript.

```
<!DOCTYPE html>
<html>

<head>
    <title>Departmental Store</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        h1 {
            text-align: center;
            margin-top: 20px;
        }

        .container {
            max-width: 800px;
            margin: 0 auto;
            padding: 20px;
        }

        .item-list {
            list-style-type: none;
            padding: 0;
        }

        .item-list li {
            margin-bottom: 10px;
        }

        .price-list {
            margin-top: 20px;
        }

        .price-list table {
            width: 100%;
            border-collapse: collapse;
        }
```

```
        .price-list table th,
        .price-list table td {
            padding: 8px;
            border-bottom: 1px solid #ddd;
        }

        .price-list table th {
            background-color: #f5f5f5;
        }

        .confirm-btn {
            margin-top: 20px;
            text-align: center;
        }
    </style>
    <script>
        function calculatePrices() {
            var selectedItems =
document.querySelectorAll('.item-checkbox:checked');
            var totalPrice = 0;

            for (var i = 0; i < selectedItems.length; i++) {
                var itemPrice =
parseFloat(selectedItems[i].getAttribute('data-price'));
                totalPrice += itemPrice;
            }

            alert('Total Price: $' + totalPrice.toFixed(2));
        }
    </script>
</head>

<body>
    <div class="container">
        <h1>Departmental Store</h1>

        <ul class="item-list">
            <li><input type="checkbox" class="item-checkbox"
data-price="10.00">Item 1</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="10.00">Item 2</li>
```

```
            <li><input type="checkbox" class="item-checkbox"
data-price="10.00">Item 3</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="10.00">Item 4</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="10.00">Item 5</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="20.00">Item 6</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="20.00">Item 7</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="20.00">Item 8</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="20.00">Item 9</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="20.00">Item 10</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="30.00">Item 11</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="30.00">Item 12</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="30.00">Item 13</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="30.00">Item 14</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="30.00">Item 15</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="40.00">Item 16</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="40.00">Item 17</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="40.00">Item 18</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="40.00">Item 19</li>
            <li><input type="checkbox" class="item-checkbox"
data-price="40.00">Item 20</li>
        </ul>

        <div class="price-list">
            <h2>Price List</h2>
            <table>
                <thead>
```

```
                    <tr>
                        <th>Group</th>
                        <th>Price</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>Group 1</td>
                        <td>$10.00</td>
                    </tr>
                    <tr>
                        <td>Group 2</td>
                        <td>$20.00</td>
                    </tr>
                    <tr>
                        <td>Group 3</td>
                        <td>$30.00</td>
                    </tr>
                    <tr>
                        <td>Group 4</td>
                        <td>$40.00</td>
                    </tr>
                </tbody>
            </table>
        </div>

        <div class="confirm-btn">
            <button onclick="calculatePrices()">Confirm</button>
        </div>
    </div>
</body>

</html>
```



16) Game of 15

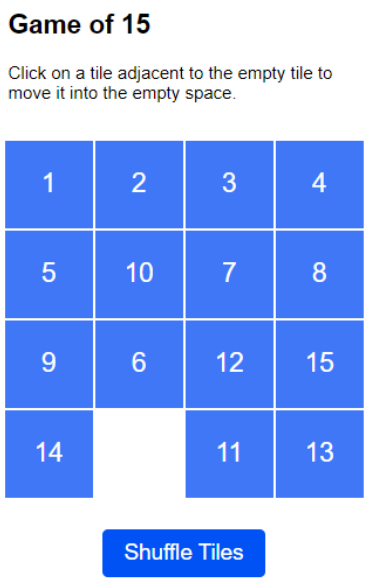
```
<!DOCTYPE html>
<html>

<head>
    <title>Game of 15</title>
    <style>
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            flex-direction: column;
            font-family: sans-serif;
            height: 100vh;
        }

        #gameboard {
            display: grid;
            grid-template-columns: repeat(4, 1fr);
        }

        .tile {
            width: 80px;
```

```
            height: 80px;
            border: 1px solid white;
            display: flex;
            justify-content: center;
            align-items: center;
            font-size: 24px;
            cursor: pointer;
            background: #4077f7;
            color: white;
        }

        #empty {
            background: white;
        }

        button {
            margin-top: 10px;
            padding: 10px 20px;
            font-size: 20px;
            cursor: pointer;
            background: #0052f5;
            color: white;
            border: none;
            border-radius: 5px;
        }
    </style>
</head>

<body>
    <div style="width: 320px">
        <h2>Game of 15</h2>
        <p>Click on a tile adjacent to the empty tile to move it into the
empty space.</p>
    </div>
    <br />
    <div id="gameboard"></div>
    <br />
    <button onclick="shuffleTiles()">Shuffle Tiles</button>
    <script>
        var gameboard = document.getElementById('gameboard');
        var tiles = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0];
```

```
        var win_tiles = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
0];
        var emptyTile = { row: 4, col: 4 }; // position of empty tile

        function renderTiles() {
            gameboard.innerHTML = '';
            for (var i = 0; i < tiles.length; i++) {
                var tile = document.createElement('div');
                tile.className = 'tile';

                if (tiles[i] === 0) {
                    tile.id = 'empty';
                    tile.innerHTML = '';
                } else {
                    tile.addEventListener('click', moveTile);
                    tile.innerHTML = tiles[i];
                }
                gameboard.appendChild(tile);
            }
        }

        function shuffleTiles() {
            // randomize tiles
            for (var i = 0; i < tiles.length; i++) {
                var randomIndex = Math.floor(Math.random() * tiles.length);
                var temp = tiles[i];
                tiles[i] = tiles[randomIndex];
                tiles[randomIndex] = temp;
            }
            // look for 0 in tiles and set emptyTile
            for (var i = 0; i < tiles.length; i++) {
                if (tiles[i] === 0) {
                    emptyTile = { row: Math.floor(i / 4) + 1, col: (i % 4) + 1
};
                    break;
                }
            }

            // check if solvable
            var inversions = 0;
            for (var i = 0; i < tiles.length; i++) {
                for (var j = i + 1; j < tiles.length; j++) {
```

```
                    if (tiles[i] > tiles[j]) {
                        inversions++;
                    }
                }
            }
            if (inversions % 2 !== 0) {
                shuffleTiles();
            }
            renderTiles();
        }

        function moveTile() {
            var tile = this;
            var tileIndex = tiles.indexOf(parseInt(tile.innerHTML));
            var emptyTileIndex = tiles.indexOf(0);
            var tilePos = { row: Math.floor(tileIndex / 4) + 1, col:
(tileIndex % 4) + 1 };
            if (isAdjacent(tilePos, emptyTile)) {
                swapTiles(tileIndex, emptyTileIndex);
                emptyTile = tilePos;
            }
            renderTiles();
            if (checkWin()) {
                alert('You win!');
            }
        }

        function isAdjacent(tilePos1, tilePos2) {
            var rowDiff = Math.abs(tilePos1.row - tilePos2.row);
            var colDiff = Math.abs(tilePos1.col - tilePos2.col);
            return (rowDiff === 1 && colDiff === 0) || (rowDiff === 0 &&
colDiff === 1);
        }

        function swapTiles(tile1, tile2) {
            // change the order of tiles array
            var temp = tiles[tile1];
            tiles[tile1] = tiles[tile2];
            tiles[tile2] = temp;
        }

        function checkWin() {
```

```
            // check if tiles is equal to win_tiles
            for (var i = 0; i < tiles.length; i++) {
                if (tiles[i] !== win_tiles[i]) {
                    return false;
                }
            }
            return true;
        }

        renderTiles();
    </script>
</body>

</html>
```



17) Form Updation using PHP

```
//signup.php
<!DOCTYPE html>
<html>

<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.3.0/css/all.min.cs
s" />
    <link rel="stylesheet" href="style.css">
```

```
</head>

<body ng-app="signupApp">
    <div class="container1">
        <div class="image">
            <img src="signup.jpg">
        </div>
        <form class="form1" action="signup1.php" method="post"
ng-controller="signupCtrl">
            <h3>SIGN UP</h3>
            <div class="input-area">
                <i class="fas fa-user-alt"></i>
                <input type="text" placeholder="Name" id="name" name="name"
ng-model="name"><br>
            </div>
            <div class="input-area">
                <i class="icon fas fa-envelope"></i>
                <input type="email" placeholder="Email ID" id="email"
name="email" ng-model="email"><br>
            </div>
            <div class="input-area">
                <i class="fa fa-phone"></i>
                <input type="number" placeholder="Phone number" id="num"
name="phno" ng-model="phno"><br>
            </div>
            <div class="btn-grp1">
                <button class="btn1" type="submit"
value="submit">Continue</button>
            </div>
        </form>
</div>
</body>

</html>
```

```
//signup1.php
<?php
<?php
$con = mysqli_connect('localhost', 'root', '','exam');
$txtName = $_POST['name'];
$txtEmail = $_POST['email'];
$txtMessage = $_POST['msg'];
```

```
$sql = "INSERT INTO `contact` (`name`, `email`,`message`) VALUES
('$txtName','$txtEmail' , '$txtMessage')";
$rs = mysqli_query($con, $sql);
```

18)