EMPLOYEE

<!DOCTYPE html>

<html lang="en" ng-app="myApp">

<head>

    <title>Employee fire check</title>

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

</head>

<body>

    <div ng-controller="mainCtrl"

        style="height: 100vh; display: flex; flex-direction: column; justify-content: center; align-items: center">

        <h1>Employee fire check</h1>

        <div style="display: flex; flex-direction: column; gap: 20px">

            <div class="input-container" ng-repeat="score in scores">

                <label for="score.id">Score:</label>

                <input type="number" id="score.id" ng-model="score.score" />

            </div>

        </div>

        <button ng-click="addScore(score)">Add Score</button>

        <button ng-click="checkFirePossiblity()">Check Fire Possiblity</button>

        <div ng-show="fire != undefined">

            <div ng-if="fire">

                <h4>Employee has a chance to be fired</h4>

            </div>

            <div ng-if="!fire">

                <h4>Employee will not be fired</h4>

            </div>

        </div>

    </div>

    <script>

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

        app.service('scoreEvaluvation', function () {

            this.checkFire = function (*scoreList*) {

                let s = 0;

                for (let i = 0; i < scoreList.length; i++) {

                    s += scoreList[i].score;

                }

                let avg = s / scoreList.length;

                if (avg > 60) {

                    return false;

                } else {

                    return true;

                }

            };

        });

        app.controller('mainCtrl', function (*$scope*, *scoreEvaluvation*) {

            $scope.scores = [

                { id: 0, score: 0 },

                { id: 1, score: 0 },

                { id: 2, score: 0 },

            ];

            $scope.addScore = function (*score*) {

                $scope.scores.push({ id: $scope.scores.length, score: 0 });

            };

            $scope.checkFirePossiblity = function () {

                $scope.fire = scoreEvaluvation.checkFire($scope.scores);

            };

        });

    </script>

</body>

</html>

GAME OF 15

<!DOCTYPE html>

<html>

<head>

    <title>Game of 15</title>

    <link rel="stylesheet" type="text/css" href="styles.css" />

</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 src="script.js"></script>

</body>

</html>

    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();