**Weather Information App (AngularJS + API Integration)**

* Create a simple UI with HTML5 and CSS3.
* Fetch weather data from a public API using AngularJS $http service with dependency injection.
* Display data using filters and custom directives.

**Employee Management System (AngularJS Controllers + MongoDB)**

* + Build a CRUD application for employees.
  + Use AngularJS controllers for Add/Edit/Delete the employee details.
  + Store employee details in MongoDB using Node.js.

**Online Movie Ticket Booking (AngularJS + Node.js)**

* + Create a responsive seat booking layout using HTML5 + CSS3.
  + Use AngularJS data binding to select seats.
  + Store booking info in MongoDB.

**Student Marks Dashboard (AngularJS + Filters + Custom Directive)**

* Create a dashboard showing student marks using angularJs services.
* Use AngularJS filters for grade calculation (e.g., pass/fail, distinction).
* Display using a custom directive for reusability.

**Chat Application (Node.js + MongoDB + AngularJS)**

* Implement a simple chat UI with AngularJS controllers and services.
* Use Node.js backend with MongoDB to store chat messages.
* Display real-time updates using AngularJS data binding.

CODE:

<!DOCTYPE html>

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

<head>

<meta charset="UTF-8">

<title>Weather Information App</title>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<style>

body {

font-family: Arial;

text-align: center;

background: linear-gradient(to right, #74ebd5, #ACB6E5);

color: #333;

padding-top: 50px;

}

.card {

background: white;

width: 300px;

margin: auto;

padding: 20px;

border-radius: 15px;

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

input {

padding: 8px;

border: none;

border-radius: 5px;

margin-bottom: 10px;

}

button {

padding: 8px 15px;

border: none;

border-radius: 5px;

background: #4CAF50;

color: white;

cursor: pointer;

}

button:hover {

background: #45a049;

}

</style>

</head>

<body ng-controller="WeatherController">

<div class="card">

<h2>🌤 Weather Information</h2>

<input type="text" ng-model="city" placeholder="Enter city name">

<br>

<button ng-click="getWeather()">Get Weather</button>

<div ng-if="weather">

<hr>

<h3>{{ weather.name | uppercase }}</h3>

<p>Temperature: {{ weather.main.temp - 273.15 | number:1 }} °C</p>

<p>Weather: {{ weather.weather[0].description | capitalize }}</p>

<weather-icon icon="weather.weather[0].icon"></weather-icon>

</div>

</div>

<script>

// Step 1: Create AngularJS app

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

// Step 2: Custom filter to capitalize

app.filter('capitalize', function() {

return function(input) {

return input ? input.charAt(0).toUpperCase() + input.slice(1) : '';

};

});

// Step 3: Custom directive to display weather icon

app.directive('weatherIcon', function() {

return {

restrict: 'E',

scope: { icon: '=' },

template: '<img ng-src="https://openweathermap.org/img/wn/{{icon}}@2x.png" alt="Weather Icon">'

};

});

// Step 4: Controller with dependency injection

app.controller('WeatherController', function($scope, $http) {

$scope.getWeather = function() {

var apiKey = 'b6907d289e10d714a6e88b30761fae22'; // Sample API key

var url = `https://api.openweathermap.org/data/2.5/weather?q=${$scope.city}&appid=${apiKey}`;

$http.get(url).then(function(response) {

$scope.weather = response.data;

}, function(error) {

alert('City not found! Please try again.');

});

};

});

</script>

</body>

</html>

2)html

<!DOCTYPE html>

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

<head>

  <meta charset="UTF-8">

  <title>Employee Management System</title>

  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

  <style>

    body { font-family: Arial; text-align:center; padding:30px; }

    input { padding:5px; margin:5px; }

    button { padding:5px 10px; margin:5px; }

    table { margin:auto; border-collapse: collapse; }

    th, td { padding:8px 12px; border:1px solid #ccc; }

  </style>

</head>

<body ng-controller="EmpController">

  <h2>Employee Management System</h2>

  <div>

    <input type="text" ng-model="emp.name" placeholder="Name">

    <input type="text" ng-model="emp.position" placeholder="Position">

    <input type="number" ng-model="emp.salary" placeholder="Salary">

    <button ng-click="addEmployee()">Add Employee</button>

    <button ng-click="updateEmployee()" ng-show="editMode">Update Employee</button>

  </div>

  <h3>Employee List</h3>

  <table>

    <tr>

      <th>Name</th>

      <th>Position</th>

      <th>Salary</th>

      <th>Actions</th>

    </tr>

    <tr ng-repeat="e in employees">

      <td>{{ e.name }}</td>

      <td>{{ e.position }}</td>

      <td>{{ e.salary }}</td>

      <td>

        <button ng-click="editEmployee(e)">Edit</button>

        <button ng-click="deleteEmployee(e.\_id)">Delete</button>

      </td>

    </tr>

  </table>

  <script>

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

    app.controller('EmpController', function($scope, $http) {

      $scope.employees = [];

      $scope.emp = {};

      $scope.editMode = false;

      // Load employees

      function loadEmployees() {

        $http.get('http://localhost:3000/employees').then(res => {

          $scope.employees = res.data;

        });

      }

      loadEmployees();

      // Add employee

      $scope.addEmployee = function() {

        $http.post('http://localhost:3000/employees', $scope.emp).then(res => {

          $scope.emp = {};

          loadEmployees();

        });

      };

      // Edit employee (load data into input fields)

      $scope.editEmployee = function(employee) {

        $scope.emp = angular.copy(employee);

        $scope.editMode = true;

      };

      // Update employee

      $scope.updateEmployee = function() {

        $http.put('http://localhost:3000/employees/' + $scope.emp.\_id, $scope.emp).then(res => {

          $scope.emp = {};

          $scope.editMode = false;

          loadEmployees();

        });

      };

      // Delete employee

      $scope.deleteEmployee = function(id) {

        $http.delete('http://localhost:3000/employees/' + id).then(res => {

          loadEmployees();

        });

      };

    });

  </script>

</body>

</html>

SERVER.JS:

const express = require('express');

const mongoose = require('mongoose');

const cors = require('cors');

const app = express();

app.use(cors());

app.use(express.json());

// Step 1: Connect to MongoDB

mongoose.connect('mongodb://127.0.0.1:27017/employeeDB', {

  useNewUrlParser: true,

  useUnifiedTopology: true

}).then(() => console.log('MongoDB connected'))

  .catch(err => console.log(err));

// Step 2: Create Employee Schema

const employeeSchema = new mongoose.Schema({

  name: String,

  position: String,

  salary: Number

});

const Employee = mongoose.model('Employee', employeeSchema);

// Step 3: API Routes

// Get all employees

app.get('/employees', async (req, res) => {

  const employees = await Employee.find();

  res.json(employees);

});

// Add new employee

app.post('/employees', async (req, res) => {

  const emp = new Employee(req.body);

  await emp.save();

  res.json(emp);

});

// Update employee

app.put('/employees/:id', async (req, res) => {

  const emp = await Employee.findByIdAndUpdate(req.params.id, req.body, { new: true });

  res.json(emp);

});

// Delete employee

app.delete('/employees/:id', async (req, res) => {

  await Employee.findByIdAndDelete(req.params.id);

  res.json({ message: 'Employee deleted' });

});

// Start server

app.listen(3000, () => console.log('Server running on http://localhost:3000'));

3)html online booking seat:

<!DOCTYPE html>

<html ng-app="movieApp">

<head>

  <meta charset="UTF-8">

  <title>Movie Ticket Booking</title>

  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

  <style>

    body { font-family: Arial; text-align: center; background: #f0f8ff; padding: 30px; }

    h1 { color: #333; }

    .seat-container { display: inline-block; text-align: left; margin-top: 20px; }

    label { display: inline-block; width: 60px; margin: 5px; }

    input[type="checkbox"]:disabled + span { color: red; font-weight: bold; }

    button { padding: 8px 15px; border: none; border-radius: 5px; background: #4CAF50; color: white; cursor: pointer; }

    button:hover { background: #45a049; }

  </style>

</head>

<body ng-controller="MovieController">

  <h1>🎬 Online Movie Ticket Booking</h1>

  <div class="seat-container">

    <div ng-repeat="seat in seats">

      <label>

        <input type="checkbox" ng-model="seat.selected" ng-disabled="seat.booked">

        <span>{{seat.name}}</span>

      </label>

    </div>

  </div>

  <br><br>

  <button ng-click="bookSeats()">Book Selected Seats</button>

  <h3>🎟️ Booked Seats:</h3>

  <p ng-if="bookedSeats.length === 0">No seats booked yet.</p>

  <ul>

    <li ng-repeat="b in bookedSeats">{{b.name}}</li>

  </ul>

  <script>

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

    app.controller('MovieController', function($scope, $http) {

      // Seat layout

      $scope.seats = [

        {name: 'A1'}, {name: 'A2'}, {name: 'A3'}, {name: 'A4'},

        {name: 'B1'}, {name: 'B2'}, {name: 'B3'}, {name: 'B4'}

      ];

      $scope.bookedSeats = [];

      // Load already booked seats from backend

      function loadBookedSeats() {

        $http.get("http://localhost:3000/booked").then(res => {

          $scope.bookedSeats = res.data;

          // Mark booked seats as disabled

          $scope.seats.forEach(seat => {

            if ($scope.bookedSeats.some(b => b.name === seat.name)) {

              seat.booked = true;

            }

          });

        });

      }

      loadBookedSeats();

      // Book selected seats

      $scope.bookSeats = function() {

        let selected = $scope.seats.filter(s => s.selected && !s.booked);

        if (selected.length === 0) {

          alert("Please select at least one available seat!");

          return;

        }

        $http.post("http://localhost:3000/book", selected).then(res => {

          $scope.bookedSeats = res.data;

          // Reset seat selection

          $scope.seats.forEach(s => {

            s.selected = false;

            if ($scope.bookedSeats.some(b => b.name === s.name)) s.booked = true;

          });

          alert("Seats booked successfully!");

        });

      };

    });

  </script>

</body>

</html>

SEVER.JS:

// Import required modules

const express = require("express");

const mongoose = require("mongoose");

const cors = require("cors");

// Initialize app

const app = express();

app.use(cors());

app.use(express.json());

// Connect to MongoDB

mongoose

  .connect("mongodb://127.0.0.1:27017/movieBooking", { useNewUrlParser: true, useUnifiedTopology: true })

  .then(() => console.log("✅ MongoDB Connected"))

  .catch(err => console.log("❌ DB Connection Error:", err));

// Schema for booked seats

const seatSchema = new mongoose.Schema({

  name: String

});

// Model

const Seat = mongoose.model("Seat", seatSchema);

// 🟢 Get all booked seats

app.get("/booked", async (req, res) => {

  const booked = await Seat.find();

  res.json(booked);

});

// 🟢 Book new seats

app.post("/book", async (req, res) => {

  try {

    const seats = req.body; // Array of seats

    await Seat.insertMany(seats);

    const all = await Seat.find();

    res.json(all);

  } catch (err) {

    res.status(500).json({ error: "Failed to save seats" });

  }

});

// Start the server

app.listen(3000, () => console.log("🚀 Server running on http://localhost:3000"));