**EX :** 5

**DATE:**

**ToDo List Application using AngularJS with Persistent Storage using Node.js and JSON File**

**AIM:**

To develop a **ToDo application** using **AngularJS** that stores tasks into a **JSON file** using a **simple Node.js server**, and retrieves them during **page reloads** for persistent task tracking.

**Procudure:**

**STEP 1:** Start by creating a basic HTML file and link it with the AngularJS library.

**STEP 2:** Define an AngularJS module using angular.module() and a controller using .controller().

**STEP 3:** Initialize an empty array ($scope.tasks) in the controller to hold todo tasks.

**STEP 4:** Create an input field and bind it to $scope.newTask using ng-model.

**STEP 5:** Design an "Add Task" button and link it to $scope.addTask() using ng-click.

**STEP 6:** Inside addTask(), push the new task to $scope.tasks.

**STEP 7:** Use AngularJS $http service to make a POST request to the server with the updated task list.

**STEP 8:** Set up a Node.js server using the Express.js framework to handle frontend and API.

**STEP 9:** Configure the server to serve static files (like your HTML) using express.static().

**STEP 10:** Create a GET endpoint /tasks that reads the JSON file using fs.readFile() and returns the data.

**STEP 11:** Create a POST endpoint /tasks that writes incoming task data to the JSON file using fs.writeFile().

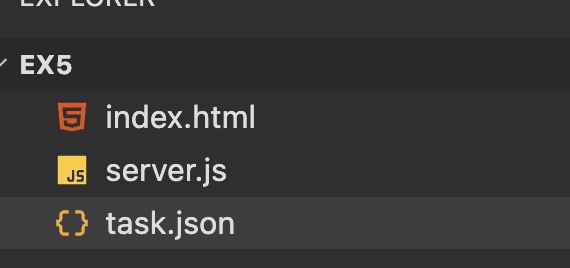
**STEP 12:** When the app loads, use $http.get('/tasks') in AngularJS to fetch the saved tasks from the server.

**STEP 13:** Bind the fetched tasks to $scope.tasks so that they appear in the task list on page reload.

**STEP 14:** Test the application to ensure tasks are saved, retrieved, and displayed correctly even after refresh.

**STEP 15:** End the process.

**Design:**

****

1. **User loads index.html** in browser.
2. AngularJS sends a **GET request** to server.js, which loads data from task.json.
3. User **adds, edits, deletes, or checks off** tasks.
4. AngularJS sends a **POST request** with the new task list.
5. server.js writes the updated tasks to task.json.

**Implementation:**

**Index.html:**

<!DOCTYPE html>

<html ng-app="todoApp">

<head>

<meta charset="UTF-8">

<title>ToDo App</title>

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

<style>

body { font-family: sans-serif; background: #eef2f5; display: flex; justify-content: center; align-items: center; height: 100vh; }

.container { background: #fff; padding: 20px; border-radius: 8px; width: 350px; box-shadow: 0 4px 10px rgba(0,0,0,0.1); }

h2 { text-align: center; }

input[type="text"] { width: 65%; padding: 8px; }

button { padding: 8px; margin-left: 5px; }

ul { padding: 0; list-style: none; }

li { margin: 8px 0; padding: 8px; background: #f4f4f4; border-radius: 5px; display: flex; justify-content: space-between; align-items: center; }

.completed { text-decoration: line-through; color: gray; }

</style>

</head>

<body ng-controller="todoController">

<div class="container">

<h2>My ToDo</h2>

<input ng-model="newTask" placeholder="New task">

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

<ul>

<li ng-repeat="task in tasks">

<div>

<input type="checkbox" ng-model="task.done">

<span ng-class="{completed: task.done}" ng-hide="task.editing">{{ task.name }}</span>

<input ng-model="task.name" ng-show="task.editing">

</div>

<div>

<button ng-click="task.editing = !task.editing">{{ task.editing ? 'Save' : 'Edit' }}</button>

<button ng-click="deleteTask($index)">X</button>

</div>

</li>

</ul>

</div>

<script>

angular.module('todoApp', [])

.controller('todoController', function($scope, $http) {

$scope.tasks = [];

$http.get('/tasks').then(res => $scope.tasks = res.data);

$scope.addTask = function() {

if ($scope.newTask) {

$scope.tasks.push({ name: $scope.newTask, done: false });

$scope.newTask = '';

save();

}

};

$scope.deleteTask = i => { $scope.tasks.splice(i, 1); save(); };

$scope.$watch('tasks', save, true);

function save() {

$http.post('/tasks', $scope.tasks);

}

});

</script>

</body>

</html>

**Server.js:**

const express = require('express');

const fs = require('fs');

const app = express();

const bodyParser = require('body-parser');

app.use(express.static('.'));

app.use(bodyParser.json());

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

fs.readFile('tasks.json', (err, data) => {

if (err) return res.json([]);

res.json(JSON.parse(data));

});

});

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

fs.writeFile('tasks.json', JSON.stringify(req.body), err => {

if (err) return res.status(500).send('Error saving tasks');

res.send('Tasks saved');

});

});

app.listen(3000, () => {

console.log('Server running at http://localhost:3000');

});

**Task.json:**

[

{

"name": *"Learn AngularJS"*,

"done": false

},

{

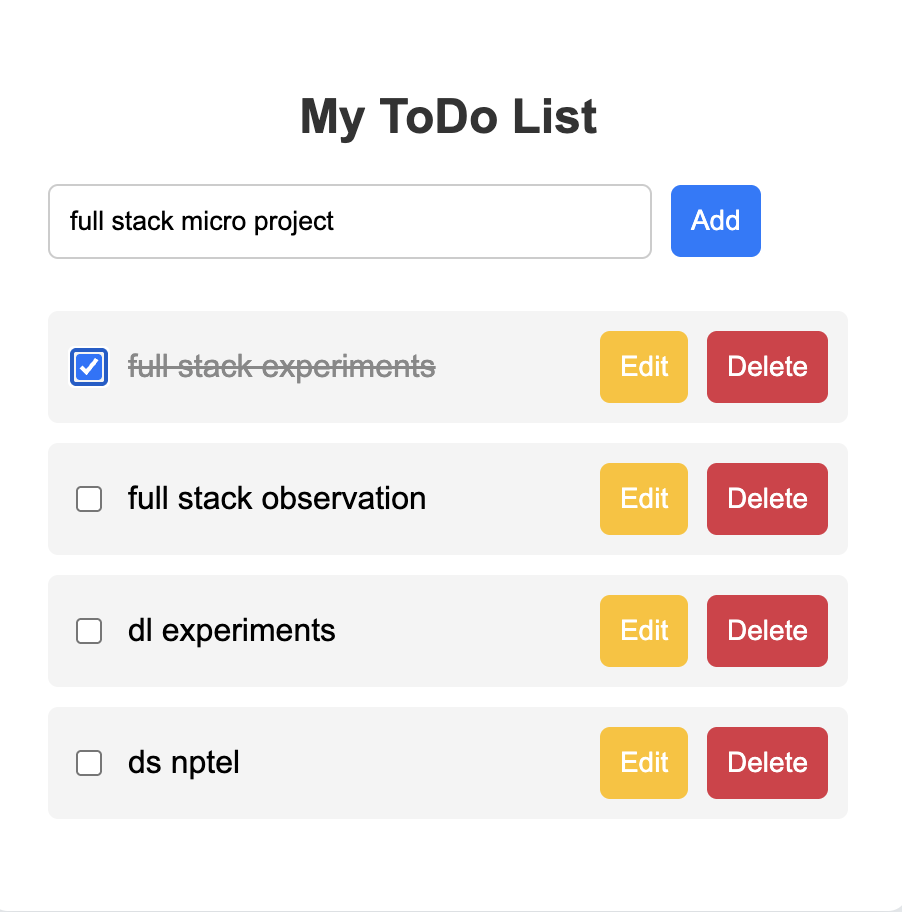
"name": *"Build ToDo App"*,

"done": true

}

]

**Output:**

****

|  |  |
| --- | --- |
| **COE(30)** |  |
| **RECORD(10)** |  |
| **OBSERVATION(10)** |  |
| **VIVA(10)** |  |
| **TOTAL(60)** |  |

**Result:**

The ToDo application allows users to add, edit, mark complete, and delete tasks using a clean AngularJS interface. All tasks are saved to a local server (task.json) and persist even after page reloads. It provides a simple, user-friendly way to manage tasks efficiently in real time.