

PRACTICAL 4

Practical 4.1

AIM :- Create a basic book Directory Project using NodeJs and APIs.

Code :-

APP.Js :-

```
const express = require('express');
const port = 3000;
const books = require('./books');
const app = express();
app.use(express.json());
app.use('/api/v1/books', books);
app.listen(port, () => {
  console.log(`Server listening at http://localhost:${port}`);
});
```

BOOKS.Js :-

```
const express = require('express');
const router = express.Router();
const books = require('./books.json');
// Get all the books
router.get('/', (req, res) => {
  res.json(books);
});
// Get a specific book
router.get('/:id', (req, res) => {
  const { id } = req.params;
  res.json(books.filter((ele) => ele.id === parseInt(id)));
});
router.post('/', (req, res) => {
  const body = req.body;
  console.log(body);
  books.push(body);
  res.json({ message: 'The book has been added' });
});
```

```
router.put('/:id', (req, res) => {
  const { id } = req.params;
  const body = req.body;
  books.forEach((book, index) => {
    if (book.id === parseInt(id)) {
      books[index] = body;
    }
  });
  res.json({ message: `The book with ID ${id} has been updated` });
  // res.json(books);
});
router.delete('/:id', (req, res) => {
  const { id } = req.params;
  books.forEach((book, index) => {
    if (book.id === parseInt(id)) {
      books.splice(index);
    }
  });
  res.json({ message: `Book with id #${id} has been deleted` });
});
module.exports = router;
```

BOOKS.JSON :-

```
[
  { "id": 1, "name": "Alice in Wonderland", "author": "Lewis Carroll"
  },
  {
    "id": 2,
    "name": "Around the World in Eighty Days",
    "author": "Jules Verne"
  },
  { "id": 3, "name": "David Copperfield", "author": "Charles Dickens"
  },
]
```

```
{ "id": 4, "name": "The Adventures of Tom Sawyer", "author": "Mark Twain" },  
{  
  "id": 5,  
  "name": "Harry Potter and The Philosophers Stone",  
  "author": "J.K. Rowley"  
}
```

PACKAGE.JSON :-

```
{  
  "name": "books-directory",  
  "version": "1.0.0",  
  "description": "A practice project for REST APIs",  
  "main": "app.js",  
  "dependencies": {  
    "express": "^4.17.1"  
  },  
  "devDependencies": {  
    "nodemon": "^2.0.4"  
  },  
  "scripts": {  
    "start": "node app.js",  
    "start:dev": "nodemon --watch app.js --watch books.js",  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "repository": {  
    "type": "git",  
    "url": "git+https://github.com/chandrangreat/books-directory-nodejs.git"  
  },  
  "keywords": [  
    "books-directory"  
  ],  
}
```

```
"author": "Ravi Chandran",  
"license": "ISC",  
"bugs": {  
  "url": "https://github.com/chandrangreat/books-directory-  
nodejs/issues"  
},  
"homepage": "https://github.com/chandrangreat/books-directory-  
nodejs#readme"  
}
```

Output :-

The most basic project you can create using ‘Node.js’ and ‘Express.js’ or ‘Nest.js’ is a simple REST API. For this purpose, I’d like to suggest you build a book directory, where you would need to create endpoints, using the four most basic methods: GET, POST, PUT and DELETE. You’d use GET for getting all books or getting only one book by id. With the POST method, you can add a new book to the list. You’d need the PUT method for updating the existing book, and it’s evident that with the DELETE method, you will remove the book from the list.

Practical 4.2

AIM :- Using Angular Js features make a shopping list/To – do list where you can add or remove items.

Code :-

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min
.js"></script>
<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">

<body>

    <style>
        body {
            left: 35pc;
            top: 8pc;
            position: absolute;
        }
    </style>

    <script>
        var app = angular.module("myToDoList", []);
        app.controller("myCtrl", function ($scope) {
            $scope.products = ["Playing Cricket", "Lecture", "Coding"];
            $scope.addItem = function () {

                $scope.errortext = "";
                if (!$scope.addMe) { return; }
                if ($scope.products.indexOf($scope.addMe) == -1) {
                    $scope.products.push($scope.addMe);
                } else {
                    $scope.errortext = "The Task is already in your To-Do
list.";
                } }
            $scope.removeItem = function (x) {
```

```
        $scope.errortext = "";
        $scope.products.splice(x, 1);
    }
});
</script>
<div ng-app="myToDoList" ng-cloak ng-controller="myCtrl"
class="w3-card-2 w3-margin" style="max-width:400px;">
    <header class="w3-container w3-blue w3-padding-16">
        <h3>My To-Do List</h3>
    </header>
    <ul class="w3-ul">
        <li ng-repeat="x in products" class="w3-padding-16">{{x}}<span ng-click="removeItem($index)"
            style="cursor:pointer;" class="w3-right w3-margin-right">x</span></li>
    </ul>
    <div class="w3-container w3-white w3-padding-16">
        <div class="w3-row w3-margin-top">
            <div class="w3-col s10">
                <input placeholder="Add shopping items here" ng-model="addMe" class="w3-input w3-border w3-padding">
            </div>
            <div class="w3-col s2">
                <button ng-click="addItem()" class="w3-btn w3-padding w3-blue">Add</button>
            </div>
        </div>
        <p class="w3-text-red">{{errortext}}</p>
    </div>
</div>
</body>

</html>
```

Output :-

My To-Do List

Playing Cricket	×
Lecture	×
Coding	×
wake up at 6	×
sleeping	×

Add

The Task is already in your To-Do list.
