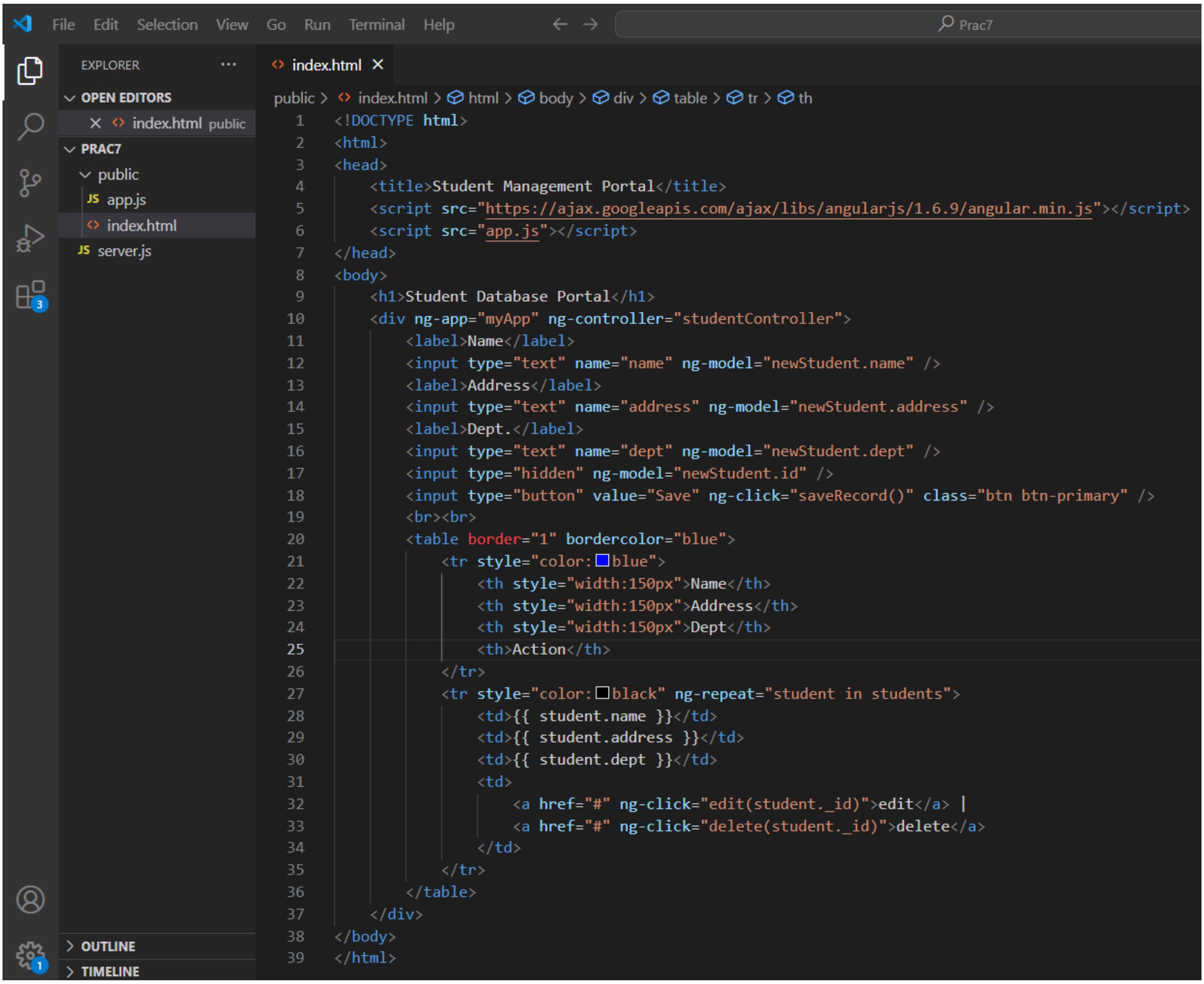
Practical 7

Aim - Write a program to create a simple web application using Express, Node JS and Angular JS

1) NodeJS should be installed in your system

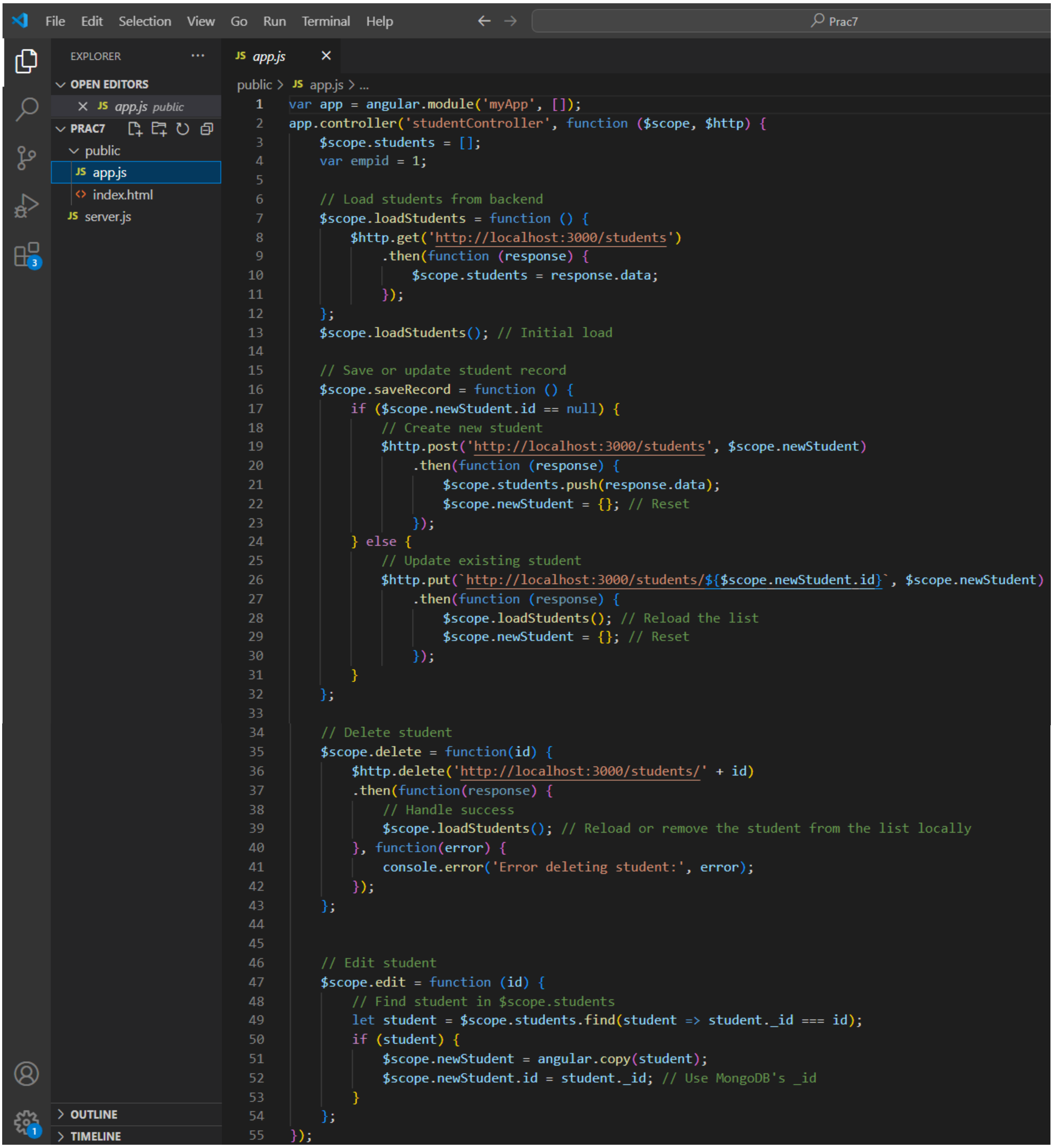
2) Mongo DB should be installed for Database system

1. Create the index.html file which is within public folder to run the home page



```html
<!DOCTYPE html>
<html>
<head>
    <title>Student Management Portal</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
    <script src="app.js"></script>
</head>
<body>
    <h1>Student Database Portal</h1>
    <div ng-app="myApp" ng-controller="studentController">
        <label>Name</label>
        <input type="text" name="name" ng-model="newStudent.name" />
        <label>Address</label>
        <input type="text" name="address" ng-model="newStudent.address" />
        <label>Dept.</label>
        <input type="text" name="dept" ng-model="newStudent.dept" />
        <input type="hidden" ng-model="newStudent.id" />
        <input type="button" value="Save" ng-click="saveRecord()" class="btn btn-primary" />
        <br><br>
        <table border="1" bordercolor="blue">
            <tr style="color:blue">
                <th style="width:150px">Name</th>
                <th style="width:150px">Address</th>
                <th style="width:150px">Dept</th>
                <th>Action</th>
            </tr>
            <tr style="color:black" ng-repeat="student in students">
                <td>{{ student.name }}</td>
                <td>{{ student.address }}</td>
                <td>{{ student.dept }}</td>
                <td>
                    <a href="#" ng-click="edit(student._id)">edit</a> |
                    <a href="#" ng-click="delete(student._id)">delete</a>
                </td>
            </tr>
        </table>
    </div>
</body>
</html>
```

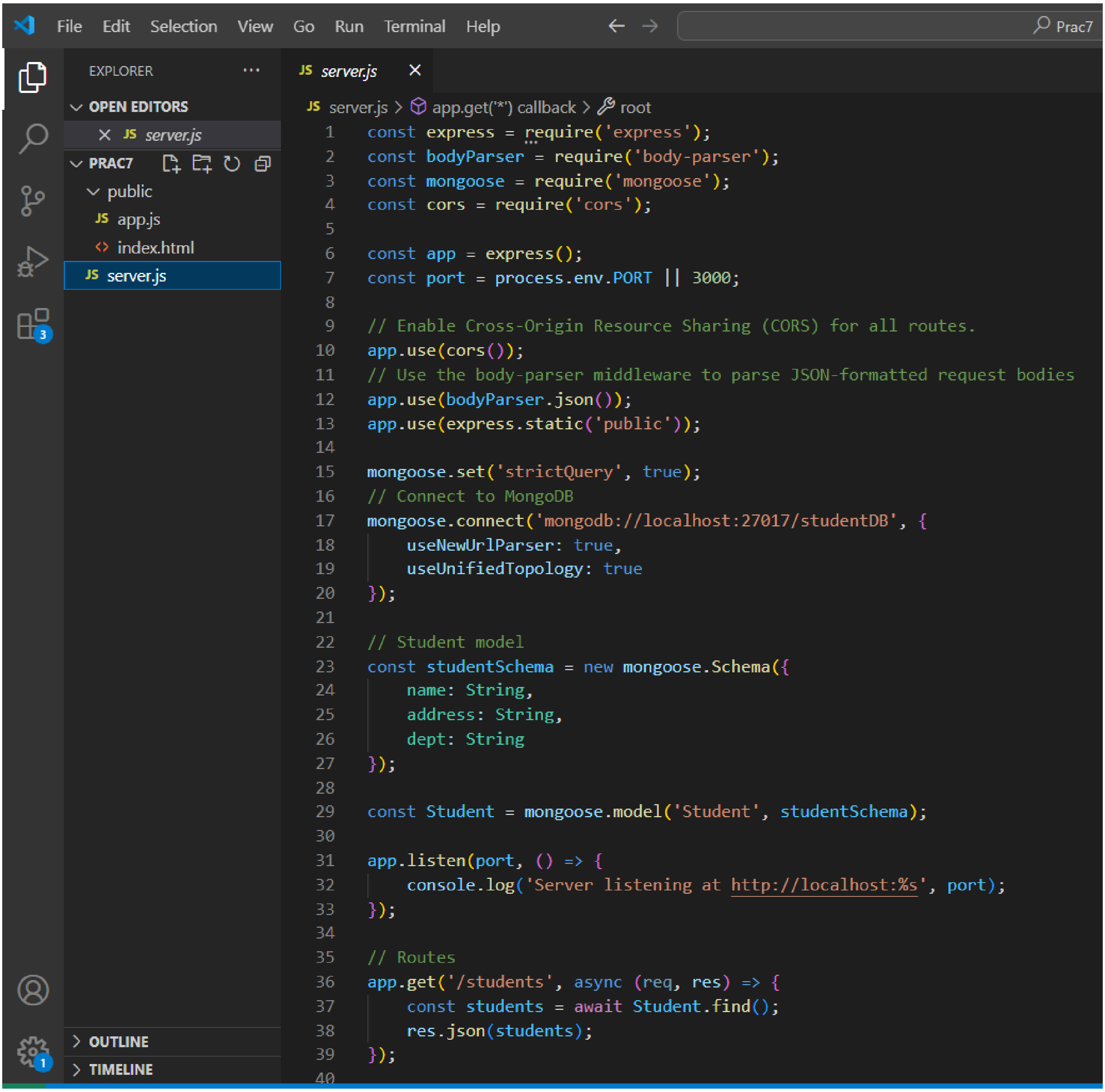
2. Create the app.js with the api calls necessary to interact with Mongo DB DATABASE



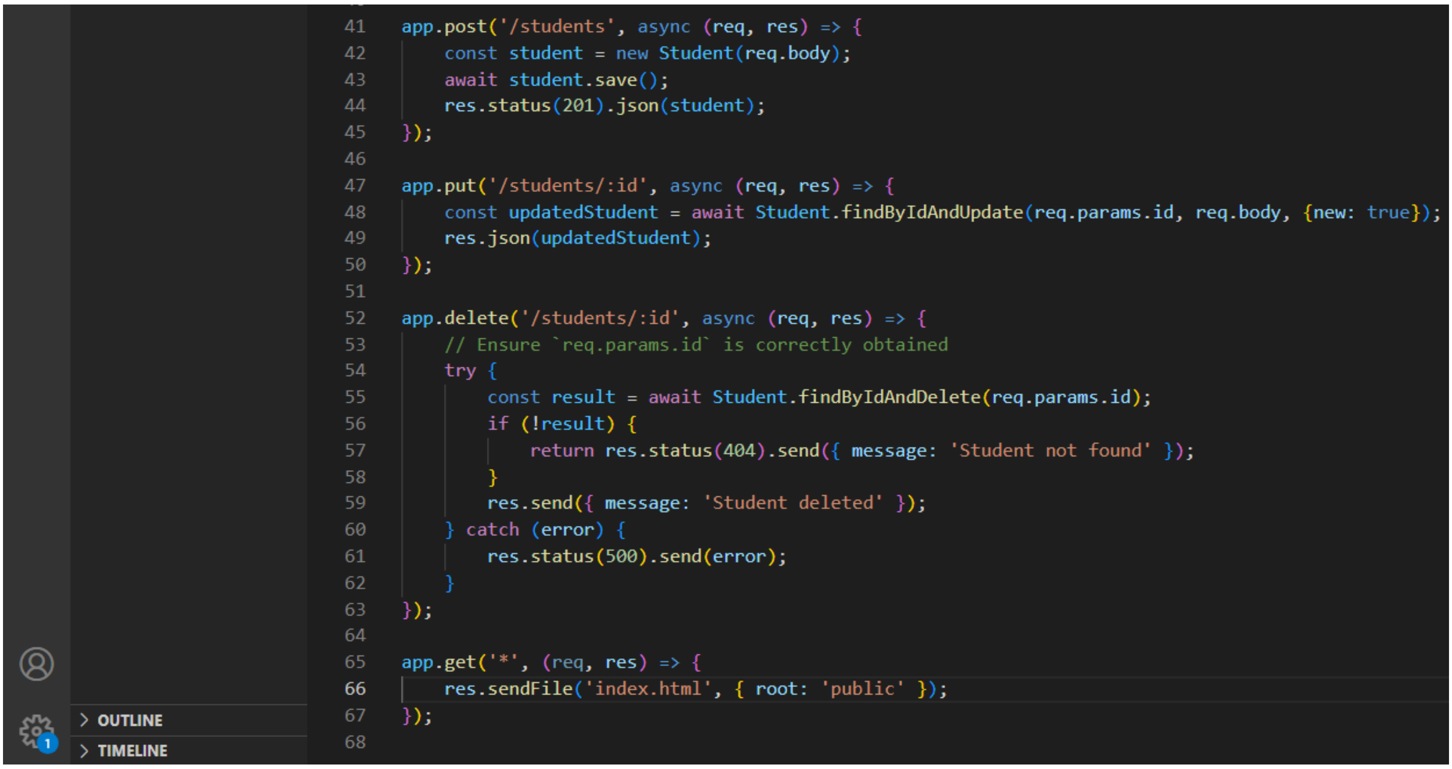
```
var app = angular.module('myApp', []);
app.controller('studentController', function ($scope, $http) {
    $scope.students = [];
    var empid = 1;

    // Load students from backend
    $scope.loadStudents = function () {
        $http.get('http://localhost:3000/students')
            .then(function (response) {
                $scope.students = response.data;
            });
    };
    $scope.loadStudents(); // Initial load

    // Save or update student record
    $scope.saveRecord = function () {
        if ($scope.newStudent.id == null) {
            // Create new student
            $http.post('http://localhost:3000/students', $scope.newStudent)
                .then(function (response) {
                    $scope.students.push(response.data);
                    $scope.newStudent = {}; // Reset
                });
        } else {
            // Update existing student
            $http.put(`http://localhost:3000/students/${$scope.newStudent.id}`, $scope.newStudent)
                .then(function (response) {
                    $scope.loadStudents(); // Reload the list
                    $scope.newStudent = {}; // Reset
                });
        }
    };

    // Delete student
    $scope.delete = function(id) {
        $http.delete('http://localhost:3000/students/' + id)
        .then(function(response) {
            // Handle success
            $scope.loadStudents(); // Reload or remove the student from the list locally
        }, function(error) {
            console.error('Error deleting student:', error);
        });
    };


    // Edit student
    $scope.edit = function (id) {
        // Find student in $scope.students
        let student = $scope.students.find(student => student._id === id);
        if (student) {
            $scope.newStudent = angular.copy(student);
            $scope.newStudent.id = student._id; // Use MongoDB's _id
        }
    };
});
```

Prepared by Prof. Mehdi Rezaei

3. Create the server.js (outside of public folder in the root directory) which will run the nodejs with expressJS middleware



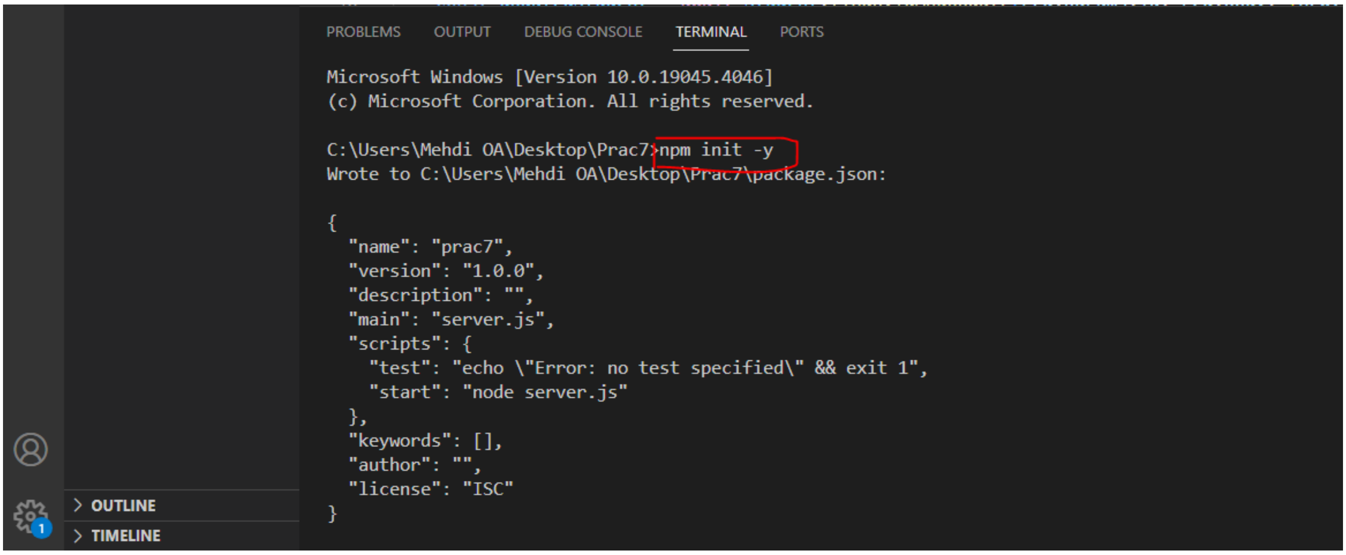
```javascript
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const cors = require('cors');

const app = express();
const port = process.env.PORT || 3000;

// Enable Cross-Origin Resource Sharing (CORS) for all routes.
app.use(cors());
// Use the body-parser middleware to parse JSON-formatted request bodies
app.use(bodyParser.json());
app.use(express.static('public'));

mongoose.set('strictQuery', true);
// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/studentDB', {
    useNewUrlParser: true,
    useUnifiedTopology: true
});

// Student model
const studentSchema = new mongoose.Schema({
    name: String,
    address: String,
    dept: String
});

const Student = mongoose.model('Student', studentSchema);

app.listen(port, () => {
    console.log('Server listening at http://localhost:%s', port);
});

// Routes
app.get('/students', async (req, res) => {
    const students = await Student.find();
    res.json(students);
});
```



```javascript
app.post('/students', async (req, res) => {
    const student = new Student(req.body);
    await student.save();
    res.status(201).json(student);
});

app.put('/students/:id', async (req, res) => {
    const updatedStudent = await Student.findByIdAndUpdate(req.params.id, req.body, {new: true});
    res.json(updatedStudent);
});

app.delete('/students/:id', async (req, res) => {
    // Ensure `req.params.id` is correctly obtained
    try {
        const result = await Student.findByIdAndDelete(req.params.id);
        if (!result) {
            return res.status(404).send({ message: 'Student not found' });
        }
        res.send({ message: 'Student deleted' });
    } catch (error) {
        res.status(500).send(error);
    }
});

app.get('*', (req, res) => {
    res.sendFile('index.html', { root: 'public' });
});

```

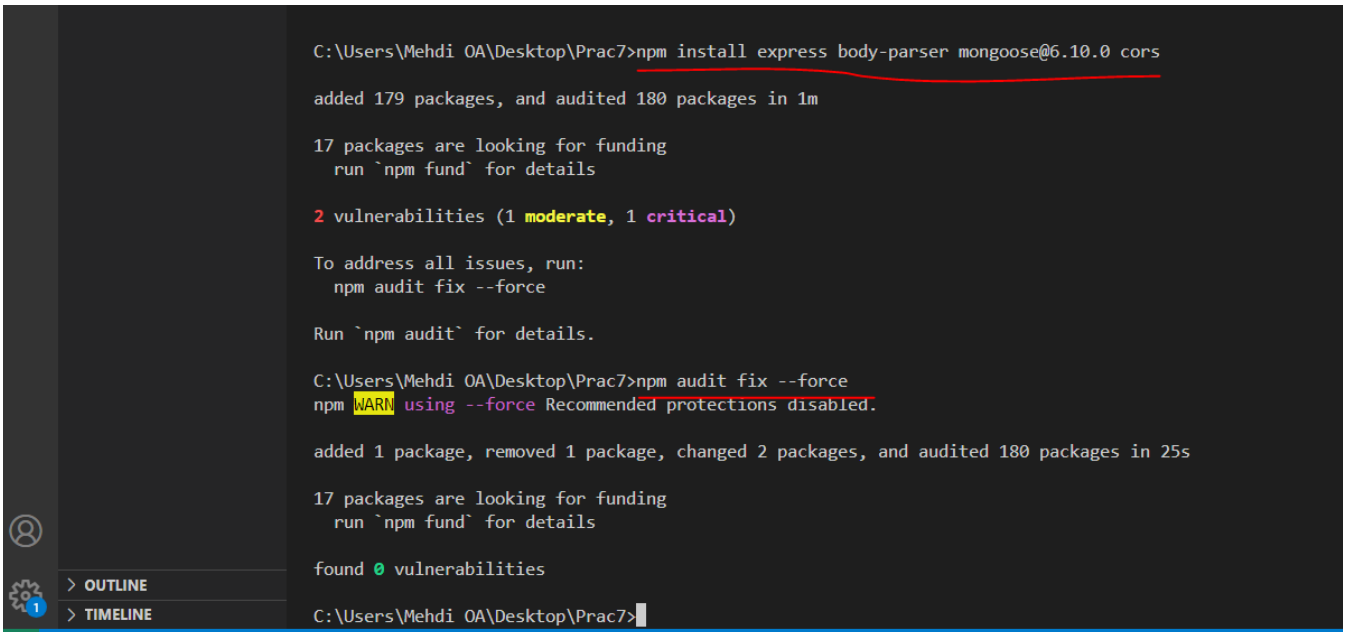
Output

Initialize the package.json file



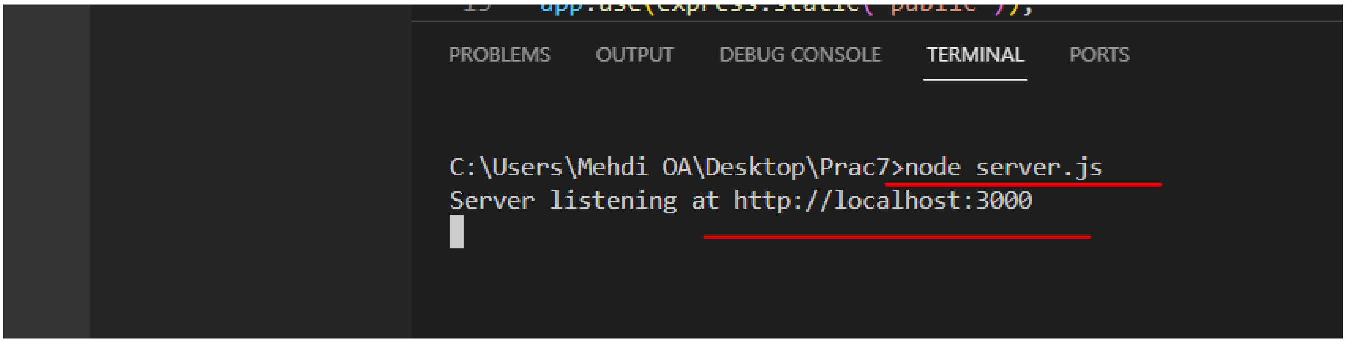
```
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Mehdi OA\Desktop\Prac7>npm init -y
Wrote to C:\Users\Mehdi OA\Desktop\Prac7\package.json:

{
  "name": "prac7",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Install the necessary packages needed to run the full stack application

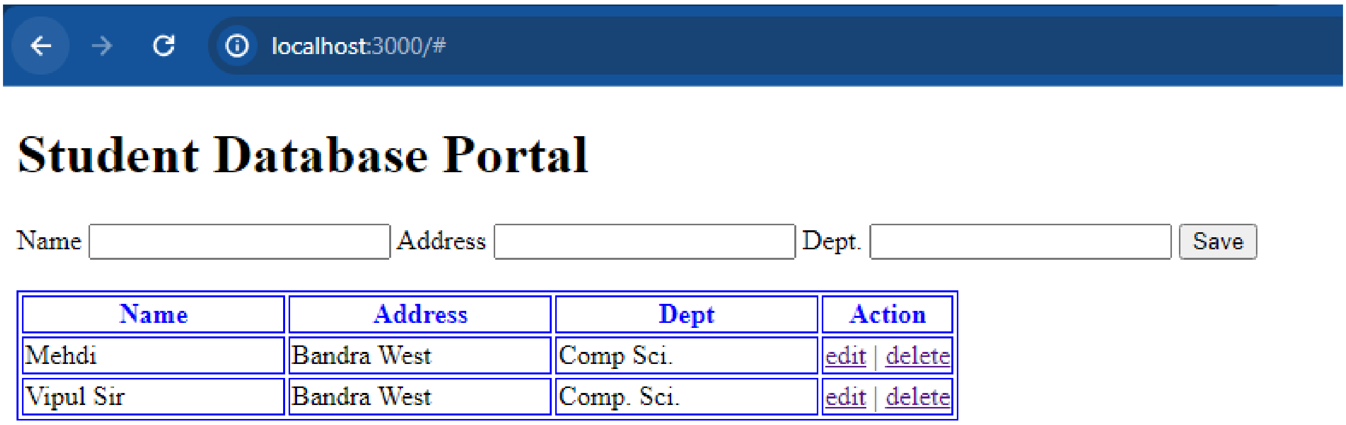


```
C:\Users\Mehdi OA\Desktop\Prac7>npm install express body-parser mongoose@6.10.0 cors

added 179 packages, and audited 180 packages in 1m

17 packages are looking for funding
  run `npm fund` for details

2 vulnerabilities (1 moderate, 1 critical)

To address all issues, run:
  npm audit fix --force

Run `npm audit` for details.

C:\Users\Mehdi OA\Desktop\Prac7>npm audit fix --force
npm WARN using --force Recommended protections disabled.

added 1 package, removed 1 package, changed 2 packages, and audited 180 packages in 25s

17 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\Mehdi OA\Desktop\Prac7>
```

Run the server, which will provide the link



```
C:\Users\Mehdi OA\Desktop\Prac7>node server.js
Server listening at http://localhost:3000
```

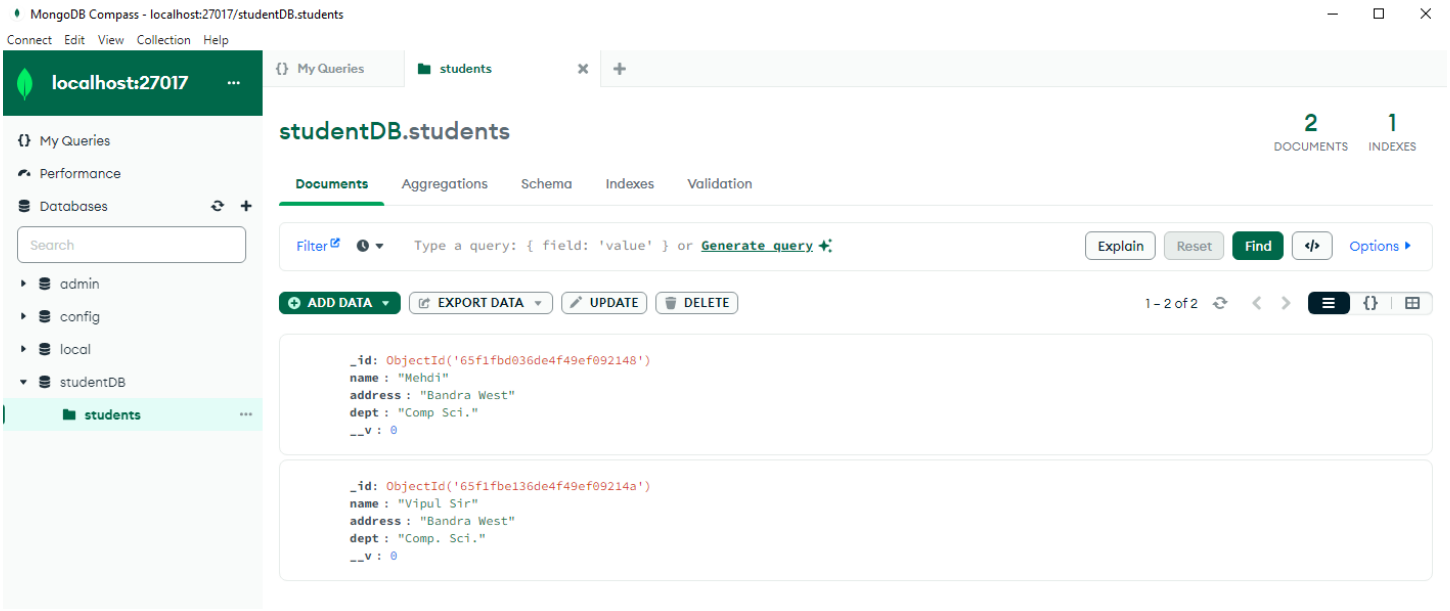


Student Database Portal

Name [ ] Address [ ] Dept. [ ] Save

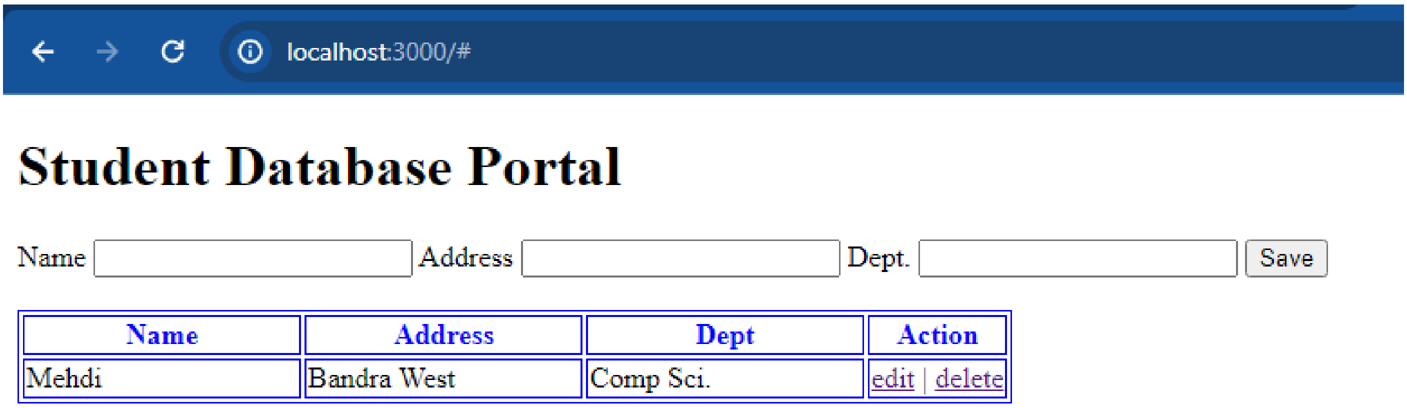
Name	Address	Dept	Action
Mehdi	Bandra West	Comp Sci.	edit | delete
Vipul Sir	Bandra West	Comp. Sci.	edit | delete

Prepared by Prof. Mehdi Rezaei



```
_id: ObjectId('65f1fbd036de4f49ef092148')
name : "Mehdi"
address : "Bandra West"
dept : "Comp Sci."
__v : 0
```

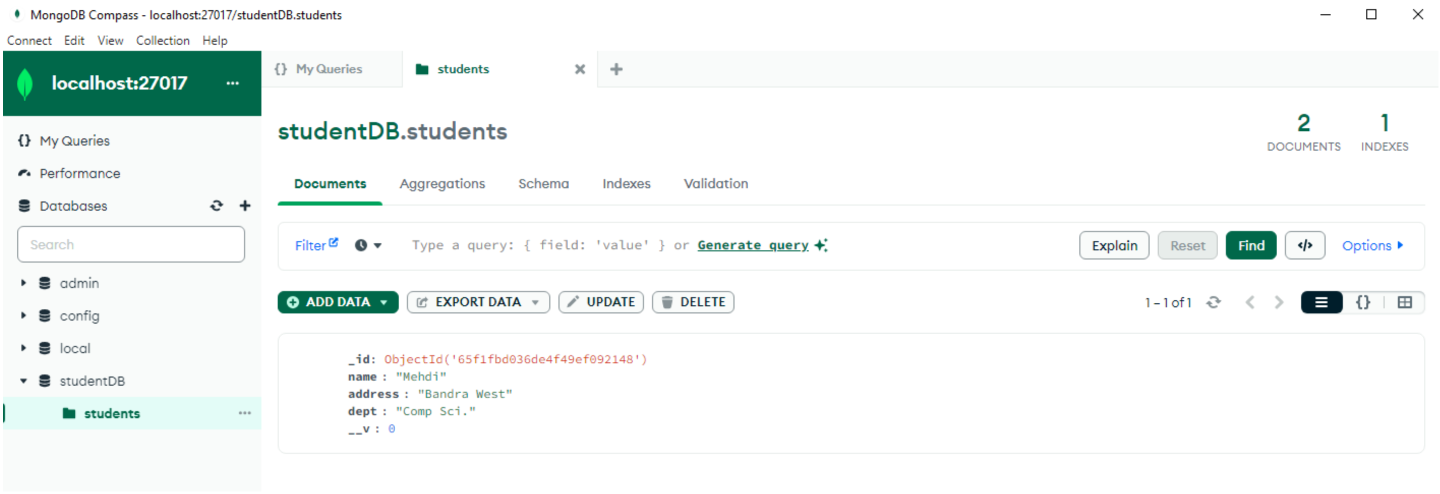
```
_id: ObjectId('65f1fbe136de4f49ef09214a')
name : "Vipul Sir"
address : "Bandra West"
dept : "Comp. Sci."
__v : 0
```



Student Database Portal

Name [ ] Address [ ] Dept. [ ] Save

Name	Address	Dept	Action
Mehdi	Bandra West	Comp Sci.	edit | delete



```
_id: ObjectId('65f1fbd036de4f49ef092148')
name : "Mehdi"
address : "Bandra West"
dept : "Comp Sci."
__v : 0
```