ORIENTAL EDUCATION SOCIETY'S

SANPADA COLLEGE OF COMMERCE & TECHNOLOGY

SECTOR-2, PLOT-3/4/5, ADJACENT SANPADA RAILWAY STATION,

SANPADA (W), NAVI MUMBAI - 400 705.

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This Is To Certify That ________________________________Of **Class:- S.Y.BSC(CS)** Bearing **Roll No:-** ____ Of Semester IV Has Successfully Completed The Practical Work In The Subject Of "________________________________" During The **Academic Year 2024-25** Under The Guidance Of **Prof.** ________________ Being The Partial Requirement For The Fulfillment Of The Curriculum Of Degree Of Bachelor Of Science In Computer Science, University of Mumbai.

Place: Sanpada

Date: / / 2025

Sign Of Subject Teacher	**Sign Of HOD**	**Sign of External**	**Sign of. Principal**

INDEX

Practical No 1 :-

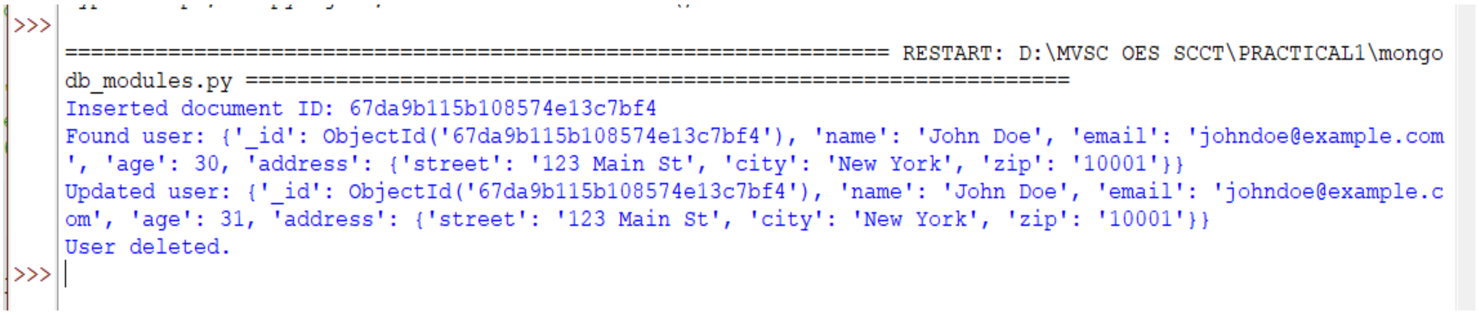
Aim :- Write A Program To Implement MongoDB Data Models.

Code :-

```
# Install pymongo (open cmd and enter this command “pip install
pymongo”)
from pymongo import MongoClient

# Connect to MongoDB
client = MongoClient('mongodb://localhost:27017/')
db = client['my_database']  # Create or connect to a database
users_collection = db['users']  # Create or connect to a
collection

# Define a sample user model
user_model = {
    "name": "John Doe",
    "email": "johndoe@example.com",
    "age": 30,
    "address": {
        "street": "123 Main St",
        "city": "New York",
        "zip": "10001"
    }
}

# Insert data into the collection
result = users_collection.insert_one(user_model)
print(f"Inserted document ID: {result.inserted_id}")

# Read data from the collection
user = users_collection.find_one({"name": "John Doe"})
print(f"Found user: {user}")

# Update data in the collection
users_collection.update_one(
    {"name": "John Doe"},
    {"$set": {"age": 31}}
)
updated_user = users_collection.find_one({"name": "John Doe"})
print(f"Updated user: {updated_user}")

# Delete data from the collection
users_collection.delete_one({"name": "John Doe"})
print("User deleted.")
```

```
# Close the connection
client.close()
```

Output :-



```
>>>
================================================================ RESTART: D:\MVSC OES SCCT\PRACTICAL1\mongo
db_modules.py ================================================================
Inserted document ID: 67da9b115b108574e13c7bf4
Found user: {'_id': ObjectId('67da9b115b108574e13c7bf4'), 'name': 'John Doe', 'email': 'johndoe@example.com
', 'age': 30, 'address': {'street': '123 Main St', 'city': 'New York', 'zip': '10001'}}
Updated user: {'_id': ObjectId('67da9b115b108574e13c7bf4'), 'name': 'John Doe', 'email': 'johndoe@example.c
om', 'age': 31, 'address': {'street': '123 Main St', 'city': 'New York', 'zip': '10001'}}
User deleted.
>>>
```

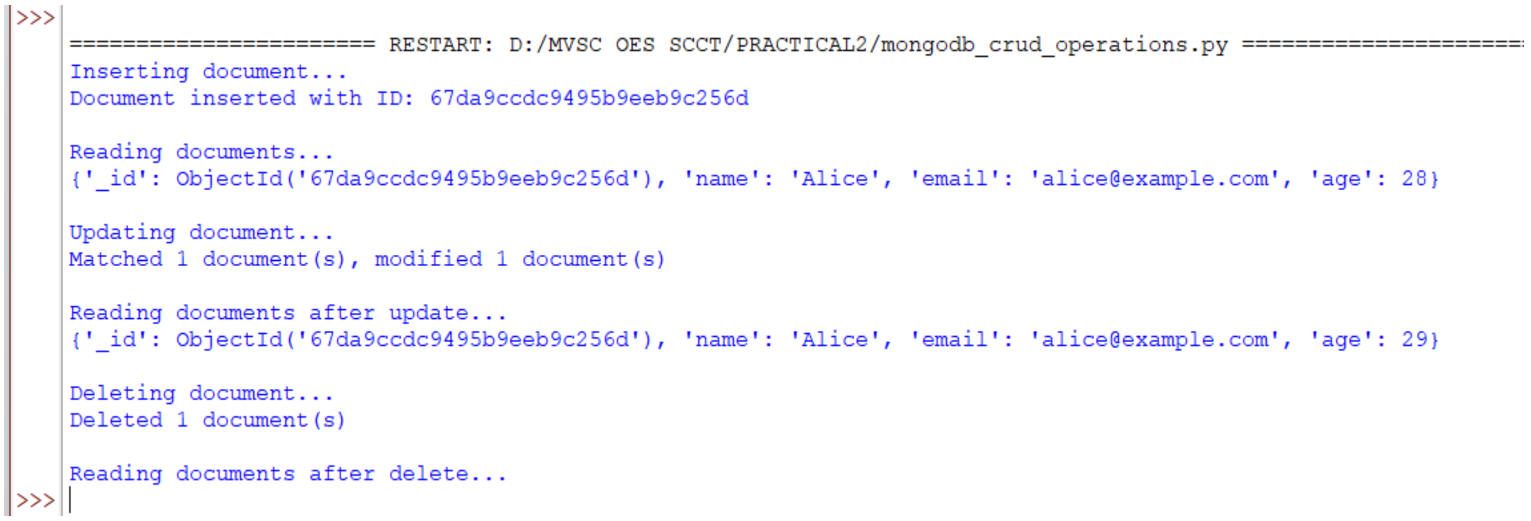
Practical No :- 2

Aim :- Write A Program To Implement CRUD Operations On MongoDB.

Code :-

```
from pymongo import MongoClient

# Connect to MongoDB
client = MongoClient('mongodb://localhost:27017/') # Connect to
the MongoDB server
db = client['test_database']  # Create or connect to a database
collection = db['test_collection']  # Create or connect to a
collection

# CREATE operation
def create_document():
    document = {
        "name": "Alice",
        "email": "alice@example.com",
        "age": 28
    }
    result = collection.insert_one(document)  # Insert the
document into the collection
    print(f"Document inserted with ID: {result.inserted_id}")

# READ operation
def read_document():
    result = collection.find()  # Fetch all documents in the
collection
    for doc in result:
        print(doc)

# UPDATE operation
def update_document():
    query = {"name": "Alice"}  # Filter documents by name
    new_values = {"$set": {"age": 29}}  # Specify the changes
    result = collection.update_one(query, new_values)  # Update
a single document
    print(f"Matched {result.matched_count} document(s),
modified {result.modified_count} document(s)")

# DELETE operation
def delete_document():
    query = {"name": "Alice"}  # Filter documents to delete
    result = collection.delete_one(query)  # Delete a single
document
    print(f"Deleted {result.deleted_count} document(s)")

# Main function to execute operations
if __name__ == "__main__":
    print("Inserting document...")
    create_document()

    print("\nReading documents...")
    read_document()

    print("\nUpdating document...")
    update_document()

    print("\nReading documents after update...")
    read_document()

    print("\nDeleting document...")
    delete_document()

    print("\nReading documents after delete...")
    read_document()

    # Close the connection
    client.close()
```

Output :-



```
>>>
====================== RESTART: D:/MVSC OES SCCT/PRACTICAL2/mongodb_crud_operations.py =====================
Inserting document...
Document inserted with ID: 67da9ccdc9495b9eeb9c256d

Reading documents...
{'_id': ObjectId('67da9ccdc9495b9eeb9c256d'), 'name': 'Alice', 'email': 'alice@example.com', 'age': 28}

Updating document...
Matched 1 document(s), modified 1 document(s)

Reading documents after update...
{'_id': ObjectId('67da9ccdc9495b9eeb9c256d'), 'name': 'Alice', 'email': 'alice@example.com', 'age': 29}

Deleting document...
Deleted 1 document(s)

Reading documents after delete...
>>>
```

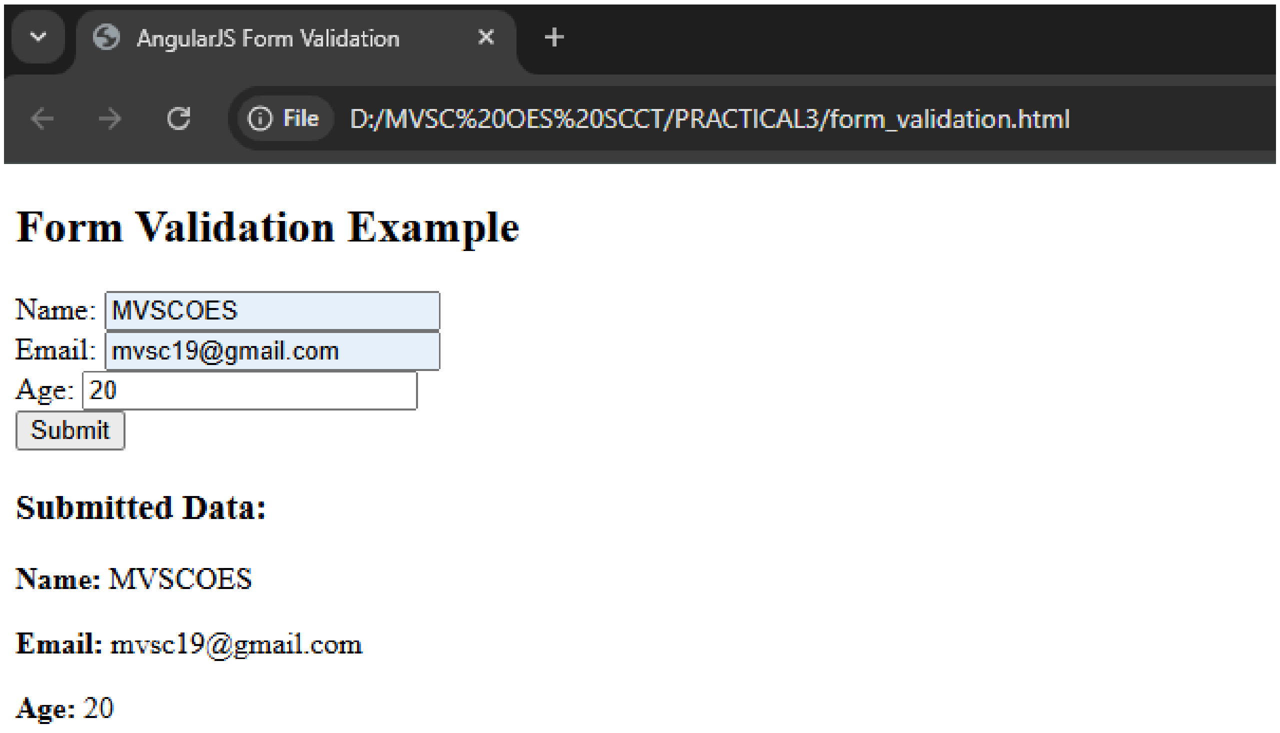
Practical No :- 3

Aim :- Write A Program To Perform Validation Of A Form Using AngularJS.

Code :-

```
<!DOCTYPE html>
<html ng-app="formApp">
<head>
    <title>AngularJS Form Validation</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
    <style>
        .error { color: red; }
    </style>
</head>
<body ng-controller="FormController">

    <h2>Form Validation Example</h2>

    <form name="userForm" ng-submit="submitForm()" novalidate>
        <!-- Name Field -->
        <div>
            <label>Name:</label>
            <input type="text" name="name" ng-model="user.name" required />
            <span class="error" ng-show="userForm.name.$touched &&
userForm.name.$error.required">
                Name is required.
            </span>
        </div>

        <!-- Email Field -->
        <div>
            <label>Email:</label>
            <input type="email" name="email" ng-model="user.email" required />
            <span class="error" ng-show="userForm.email.$touched &&
userForm.email.$error.required">
                Email is required.
            </span>
            <span class="error" ng-show="userForm.email.$touched &&
userForm.email.$error.email">
                Invalid email format.
            </span>
        </div>

        <!-- Age Field -->
        <div>
            <label>Age:</label>
            <input type="number" name="age" ng-model="user.age" min="18"
required />
            <span class="error" ng-show="userForm.age.$touched &&
userForm.age.$error.required">
                Age is required.
            </span>
            <span class="error" ng-show="userForm.age.$touched &&
userForm.age.$error.min">
                Age must be 18 or above.
            </span>
        </div>

        <!-- Submit Button -->
        <button type="submit"
ng-disabled="userForm.$invalid">Submit</button>
    </form>

    <!-- Display Submitted Data -->
    <div ng-show="submitted">
        <h3>Submitted Data:</h3>
        <p><strong>Name:</strong> {{ user.name }}</p>
        <p><strong>Email:</strong> {{ user.email }}</p>
        <p><strong>Age:</strong> {{ user.age }}</p>
    </div>

    <script>
        // Define AngularJS module and controller
        angular.module('formApp', [])
            .controller('FormController', function($scope) {
                $scope.user = {};
                $scope.submitted = false;
                $scope.submitForm = function() {
                    if ($scope.userForm.$valid) {
                        $scope.submitted = true;
                        console.log("Form submitted successfully:", $scope.user);
                    }
                };
            });
    </script>
</body>
</html>
```

Output :-


AngularJS Form Validation
File
D:/MVSC%20OES%20SCCT/PRACTICAL3/form_validation.html

Form Validation Example

Name: MVSCOES
Email: mvsc19@gmail.com
Age: 20
Submit

Submitted Data:

Name: MVSCOES

Email: mvsc19@gmail.com

Age: 20

Practical No :- 4

Aim :- Write A Program To Create And Implement Modules And Controllers In AngularJS.

Code :- a) index.js :- It contains the main HTML structure and references the AngularJS app and controller files.

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <title>AngularJS Module and Controller Example</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angu
lar.min.js"></script>
</head>
<body>

<div ng-controller="UserController">
    <h2>User List</h2>
    <ul>
        <li ng-repeat="user in users">
            {{ user.name }} - {{ user.email }}
        </li>
    </ul>

    <h3>Add User</h3>
    <form ng-submit="addUser()">
        <input type="text" ng-model="newUser.name"
placeholder="Name" required />
        <input type="email" ng-model="newUser.email"
placeholder="Email" required />
        <button type="submit">Add User</button>
    </form>
</div>

<!-- Include AngularJS App and Controller -->
<script src="app.js"></script>
<script src="userController.js"></script>

</body> </html>
```

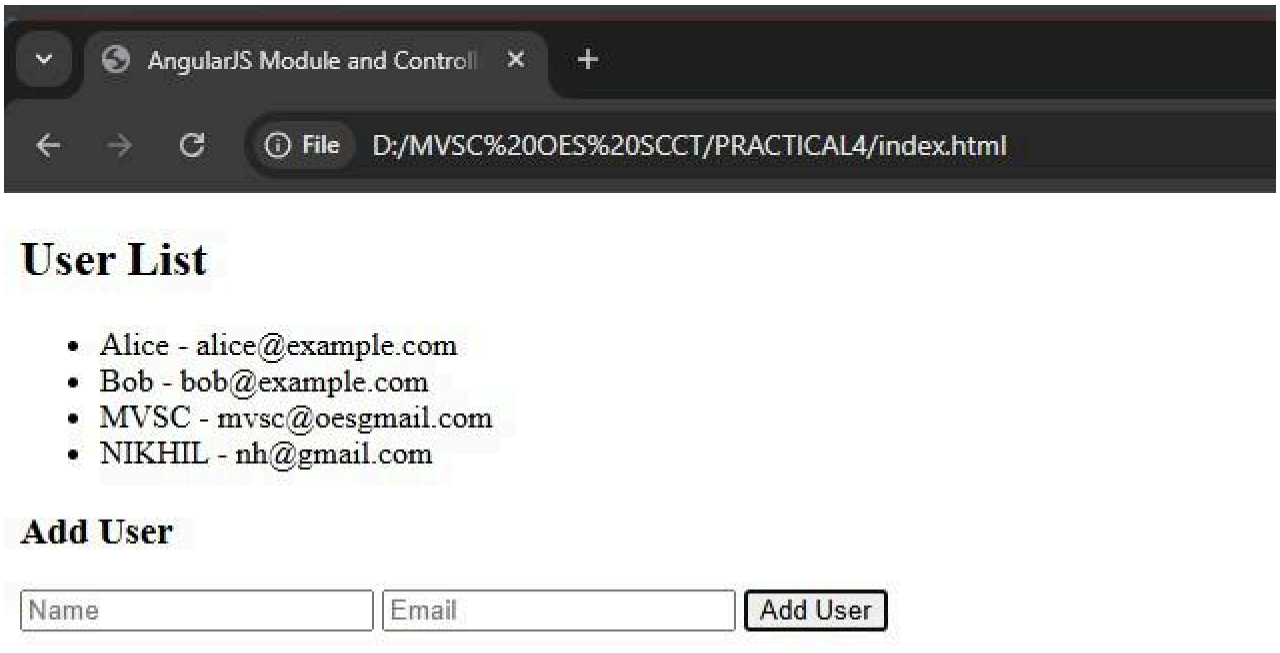
b) app.js :- It creates an AngularJS module named myApp.

```
// Create AngularJS module
var app = angular.module('myApp', []);
```

c) userController.js :- It defines the UserController to manage user data.

```
// Create a controller within the 'myApp' module
app.controller('UserController', function($scope) {
    // Initial list of users
    $scope.users = [
        { name: 'Alice', email: 'alice@example.com' },
        { name: 'Bob', email: 'bob@example.com' }
    ];

    // Object to store new user input
    $scope.newUser = {};

    // Function to add a new user
    $scope.addUser = function() {
        if ($scope.newUser.name && $scope.newUser.email) {
            $scope.users.push({
                name: $scope.newUser.name,
                email: $scope.newUser.email
            });
            $scope.newUser = {}; // Reset form fields after
adding
        }
    };
});
```

Output :-


AngularJS Module and Controll
File
D:/MVSC%20OES%20SCCT/PRACTICAL4/index.html

User List

- Alice - alice@example.com
- Bob - bob@example.com
- MVSC - mvsc@oesgmail.com
- NIKHIL - nh@gmail.com

Add User

where
alice - alice@example.com & Bob - bob@example.com are old users.

MVSC - mvsc@oesgmail.com & NIKHIL - nh@gmail.com are new(inserted) users.

Practical No :- 5

Aim :- Write A Program To Implement Error Handling In AngularJS.

Code :- a) index.html :- This file acts as the main entry point for your AngularJS application.

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <title>AngularJS Error Handling Example</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angu
lar.min.js"></script>
</head>
<body>

<div ng-controller="ErrorHandlingController">
    <h2>User List</h2>

    <!-- Display Loading Status -->
    <div ng-if="loading">Loading...</div>

    <!-- Display Error Message -->
    <div ng-if="error" style="color: red;">
        Error: {{ error }}
    </div>

    <!-- Display User Data -->
    <ul>
        <li ng-repeat="user in users">
            {{ user.name }} - {{ user.email }}
        </li>
    </ul>

    <button ng-click="fetchUsers()">Reload Users</button>
</div>

<!-- Include AngularJS App and Controller -->
<script src="app.js"></script>
```

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

</body>
</html>
```

b) app.js :- This file initializes the AngularJS application module.

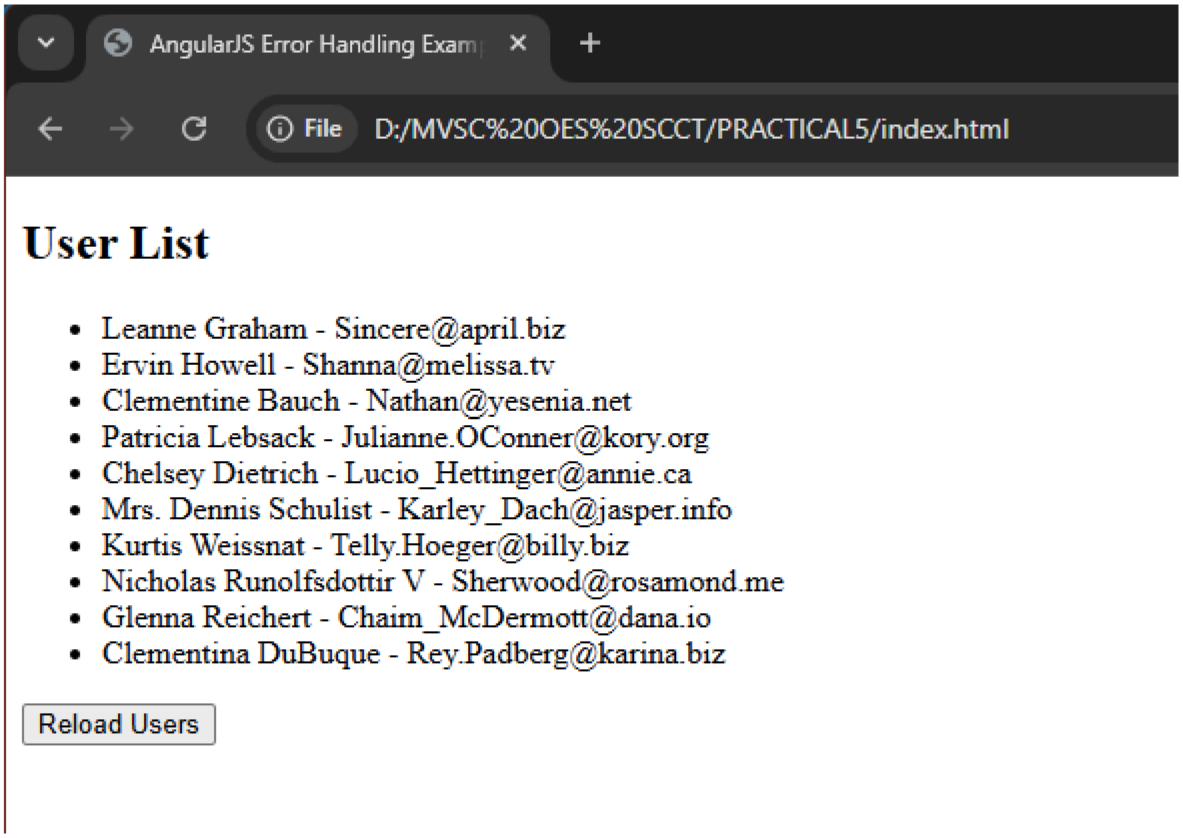
```
// Create AngularJS module
var app = angular.module('myApp', []);
```

c) errorHandlingController.js :- This file defines the ErrorHandlingController to manage error handling and user interactions.

```
// Define the controller within the 'myApp' module
app.controller('ErrorHandlingController', function($scope,
$http) {
    $scope.users = [];
    $scope.loading = false;
    $scope.error = '';

    // Function to fetch user data
    $scope.fetchUsers = function() {
        $scope.loading = true;
        $scope.error = '';

        $http.get('https://jsonplaceholder.typicode.com/users')
// Simulate API call
            .then(function(response) {
                // Success: Load data into scope
                $scope.users = response.data;
            })
            .catch(function(error) {
                // Error handling
                if (error.status === 404) {
                    $scope.error = 'Data not found (404).';
                } else if (error.status === 500) {
                    $scope.error = 'Internal Server Error (500).';
                } else if (error.status === -1) {
                    $scope.error = 'Network Error: Unable to
connect to server.';
```

```
                } else {
                    $scope.error = `Error: ${error.status} -
${error.statusText}`;
                }
            })
            .finally(function() {
                $scope.loading = false;
            });
    };

    // Load data initially
    $scope.fetchUsers();
});
```

Output :-



AngularJS Error Handling Examp
File
D:/MVSC%20OES%20SCCT/PRACTICAL5/index.html

User List

- Leanne Graham - Sincere@april.biz
- Ervin Howell - Shanna@melissa.tv
- Clementine Bauch - Nathan@yesenia.net
- Patricia Lebsack - Julianne.OConner@kory.org
- Chelsey Dietrich - Lucio_Hettinger@annie.ca
- Mrs. Dennis Schulist - Karley_Dach@jasper.info
- Kurtis Weissnat - Telly.Hoeger@billy.biz
- Nicholas Runolfsdottir V - Sherwood@rosamond.me
- Glenna Reichert - Chaim_McDermott@dana.io
- Clementina DuBuque - Rey.Padberg@karina.biz

Practical No :- 6

Aim :- Create An Application For Customer / Students Records Using AngularJS.

Code :-

a) index.html

```
<!DOCTYPE html>
<html ng-app="recordApp">
<head>
    <title>Customer/Student Records</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular
.min.js"></script>
    <style>
        body { font-family: Arial, sans-serif; }
        .error { color: red; }
        table { width: 100%; border-collapse: collapse;
margin-bottom: 20px; }
        th, td { padding: 10px; border: 1px solid #ccc;
text-align: left; }
        th { background-color: #f4f4f4; }
        button { margin-right: 5px; }
    </style>
</head>
<body ng-controller="RecordController">

    <h2>Customer/Student Records</h2>

    <!-- Add Record Form -->
    <form ng-submit="addRecord()" novalidate>
        <input type="text" ng-model="newRecord.name"
placeholder="Name" required />
        <input type="email" ng-model="newRecord.email"
placeholder="Email" required />
        <input type="number" ng-model="newRecord.age"
placeholder="Age" required />
        <button type="submit">{{ isEdit ? 'Update' : 'Add'
}}</button>
```

```
        <button type="button"
ng-click="resetForm()">Cancel</button>
    </form>

    <p class="error" ng-show="errorMessage">{{ errorMessage }}</p>

    <!-- Display Records -->
    <table>
        <thead>
            <tr>
                <th>Name</th>
                <th>Email</th>
                <th>Age</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            <tr ng-repeat="record in records">
                <td>{{ record.name }}</td>
                <td>{{ record.email }}</td>
                <td>{{ record.age }}</td>
                <td>
                    <button
ng-click="editRecord(record)">Edit</button>
                    <button
ng-click="deleteRecord(record)">Delete</button>
                </td>
            </tr>
        </tbody>
    </table>

    <!-- Include AngularJS App and Controller -->
    <script src="app.js"></script>
    <script src="customerController.js"></script>

</body>
</html>
```

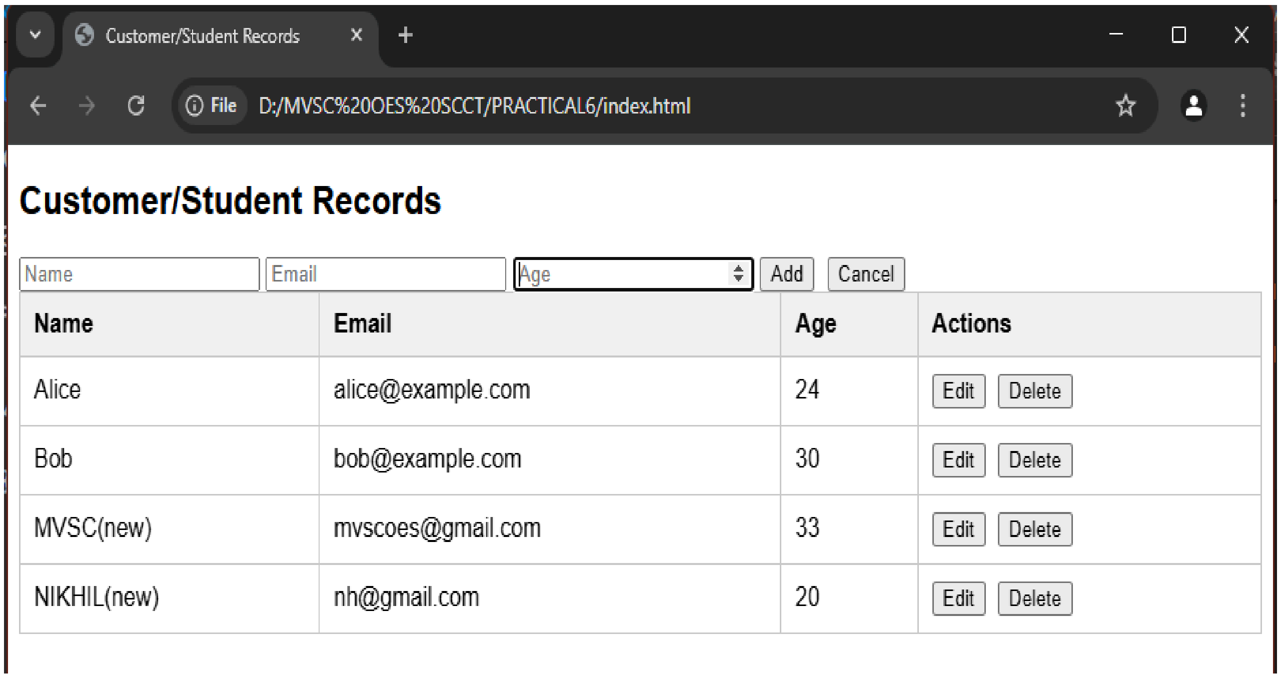
b) app.js :-

```
// Create AngularJS module
var app = angular.module('recordApp', []);
```

c) customerController.js :-

```
// Create the controller within the 'recordApp' module
app.controller('RecordController', function($scope) {
    // Sample records
    $scope.records = [
        { name: 'Alice', email: 'alice@example.com', age: 24 },
        { name: 'Bob', email: 'bob@example.com', age: 30 }
    ];

    $scope.newRecord = {};
    $scope.isEdit = false;
    $scope.errorMessage = '';

    // Add new record
    $scope.addRecord = function() {
        if ($scope.newRecord.name && $scope.newRecord.email &&
$scope.newRecord.age) {
            if ($scope.isEdit) {
                // Update existing record
                const index =
$scope.records.indexOf($scope.editingRecord);
                $scope.records[index] =
angular.copy($scope.newRecord);
                $scope.isEdit = false;
            } else {
                // Add new record

$scope.records.push(angular.copy($scope.newRecord));
            }
            $scope.resetForm();
        } else {
            $scope.errorMessage = 'All fields are required!';
        }
    };

    // Edit record
    $scope.editRecord = function(record) {
        $scope.newRecord = angular.copy(record);
        $scope.editingRecord = record;
        $scope.isEdit = true;
        $scope.errorMessage = '';
    };

    // Delete record
    $scope.deleteRecord = function(record) {
        const index = $scope.records.indexOf(record);
        if (index !== -1) {
            $scope.records.splice(index, 1);
        }
    };

    // Reset form
    $scope.resetForm = function() {
        $scope.newRecord = {};
        $scope.isEdit = false;
        $scope.errorMessage = '';
    };
});
```

Output :-



Customer/Student Records

Name	Email	Age	Actions
Alice	alice@example.com	24	Edit Delete
Bob	bob@example.com	30	Edit Delete
MVSC(new)	mvscoes@gmail.com	33	Edit Delete
NIKHIL(new)	nh@gmail.com	20	Edit Delete

Practical No :- 7

Aim :- Write A Program To Create A Simple Web Application Using Express, Node Js, And AngularJS.

Code :- Install required backend dependencies :-

“npm install express body-parser cors”

a) backend :- server.js

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

const app = express();
const PORT = 3000;

app.use(bodyParser.json());
app.use(cors());

let records = [
    { id: 1, name: 'Alice', email: 'alice@example.com', age: 25 },
    { id: 2, name: 'Bob', email: 'bob@example.com', age: 30 }
];

// Get all records
app.get('/api/records', (req, res) => {
    res.json(records);
});

// Add new record
app.post('/api/records', (req, res) => {
    const newRecord = {
        id: records.length + 1,
        ...req.body
    };
    records.push(newRecord);
    res.json(newRecord);
});

// Update record
```

```
app.put('/api/records/:id', (req, res) => {
    const id = parseInt(req.params.id);
    const index = records.findIndex(record => record.id === id);
    if (index !== -1) {
        records[index] = { id, ...req.body };
        res.json(records[index]);
    } else {
        res.status(404).json({ error: 'Record not found' });
    }
});

// Delete record
app.delete('/api/records/:id', (req, res) => {
    const id = parseInt(req.params.id);
    records = records.filter(record => record.id !== id);
    res.json({ message: 'Record deleted' });
});

// Start server
app.listen(PORT, () => {
    console.log(`Server is running on http://localhost:${PORT}`);
});
```

b) frontend :-

i) app.js :-

```
// Create AngularJS module
var app = angular.module('recordApp', []);
```

ii) index.html :-

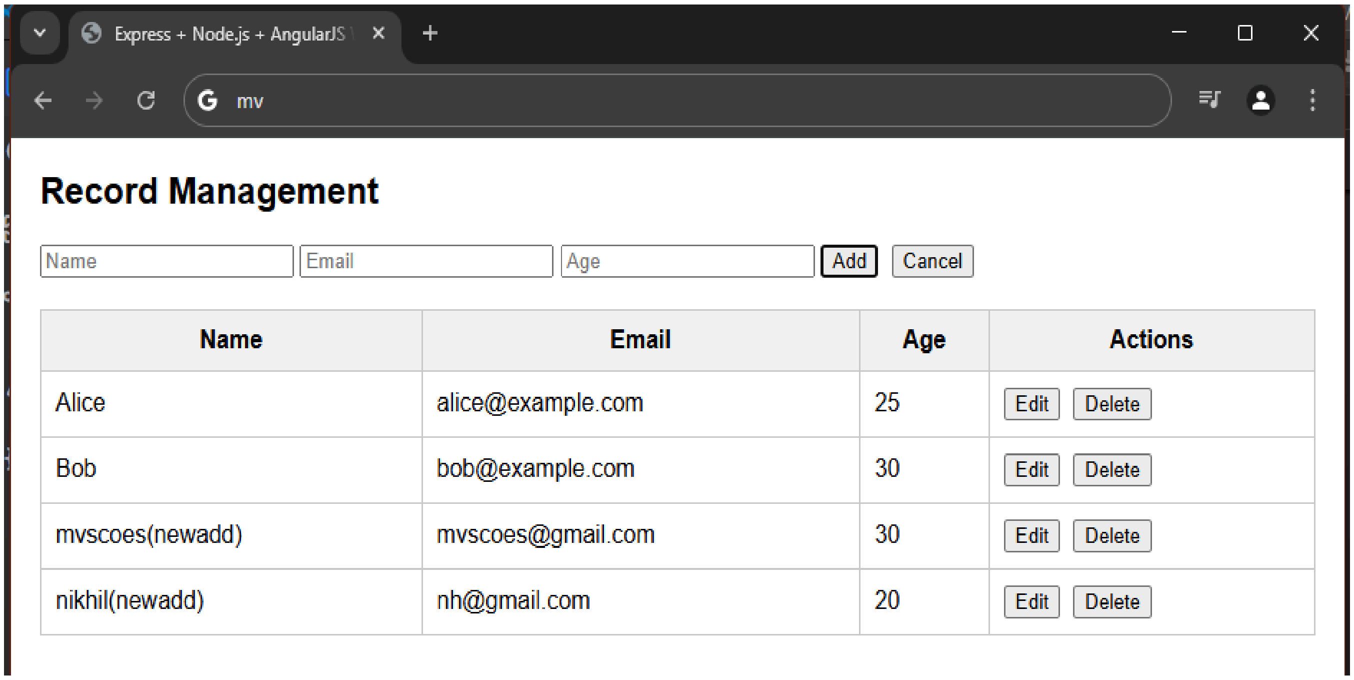
```
<!DOCTYPE html>
<html ng-app="recordApp">
<head>
    <title>Express + Node.js + AngularJS Web App</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular
.min.js"></script>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; }
        .error { color: red; }
        table { width: 100%; border-collapse: collapse;
margin-top: 20px; }
        th, td { padding: 10px; border: 1px solid #ccc; }
        th { background-color: #f4f4f4; }
        button { margin-right: 5px; }
    </style>
</head>
<body ng-controller="RecordController">

    <h2>Record Management</h2>

    <!-- Add/Edit Record Form -->
    <form ng-submit="addOrUpdateRecord()">
        <input type="text" ng-model="newRecord.name"
placeholder="Name" required>
        <input type="email" ng-model="newRecord.email"
placeholder="Email" required>
        <input type="number" ng-model="newRecord.age"
placeholder="Age" required>
        <button type="submit">{{ isEdit ? 'Update' : 'Add'
}}</button>
        <button type="button"
ng-click="resetForm()">Cancel</button>
    </form>

    <!-- Display Error Message -->
    <p class="error" ng-show="errorMessage">{{ errorMessage }}</p>

    <!-- Display Records -->
    <table>
        <thead>
            <tr>
                <th>Name</th>
                <th>Email</th>
                <th>Age</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            <tr ng-repeat="record in records">
```

```
                <td>{{ record.name }}</td>
                <td>{{ record.email }}</td>
                <td>{{ record.age }}</td>
                <td>
                    <button
ng-click="editRecord(record)">Edit</button>
                    <button
ng-click="deleteRecord(record.id)">Delete</button>
                </td>
            </tr>
        </tbody>
    </table>

    <script src="app.js"></script>
    <script src="recordController.js"></script>
</body>
</html>
```

iii) recordController.js :-

```
// Define controller
app.controller('RecordController', function($scope, $http) {
    const API_URL = 'http://localhost:3000/api/records';

    $scope.records = [];
    $scope.newRecord = {};
    $scope.isEdit = false;
    $scope.errorMessage = '';

    // Fetch records from server
    function loadRecords() {
        $http.get(API_URL)
            .then(response => {
                $scope.records = response.data;
            })
            .catch(error => {
                $scope.errorMessage = 'Error fetching records.';
            });
    }

    // Add or Update Record
    $scope.addOrUpdateRecord = function() {
        if ($scope.isEdit) {
            $http.put(`${API_URL}/${$scope.newRecord.id}`,
$scope.newRecord)
                .then(response => {
                    loadRecords();
                    $scope.resetForm();
                })
                .catch(error => {
                    $scope.errorMessage = 'Error updating
record.';
                });
        } else {
            $http.post(API_URL, $scope.newRecord)
                .then(response => {
                    loadRecords();
                    $scope.resetForm();
                })
                .catch(error => {
                    $scope.errorMessage = 'Error adding record.';
                });
        }
    };

    // Edit Record
    $scope.editRecord = function(record) {
        $scope.newRecord = angular.copy(record);
        $scope.isEdit = true;
    };

    // Delete Record
    $scope.deleteRecord = function(id) {
        $http.delete(`${API_URL}/${id}`)
            .then(response => {
                loadRecords();
            })
            .catch(error => {
                $scope.errorMessage = 'Error deleting record.';
            });
    };
    // Reset Form
    $scope.resetForm = function() {
        $scope.newRecord = {};
        $scope.isEdit = false;
        $scope.errorMessage = '';
    };

    // Initial Load
    loadRecords();
});
```

Output :-



Record Management

Name	Email	Age	Actions
Alice	alice@example.com	25	Edit Delete
Bob	bob@example.com	30	Edit Delete
mvscoes(newadd)	mvscoes@gmail.com	30	Edit Delete
nikhil(newadd)	nh@gmail.com	20	Edit Delete

Practical No :- 8

Aim :- Create a simple HTML "Hello World" Project using AngularJS Framework and apply ng-controller, ng-model, and expressions.

Code :-

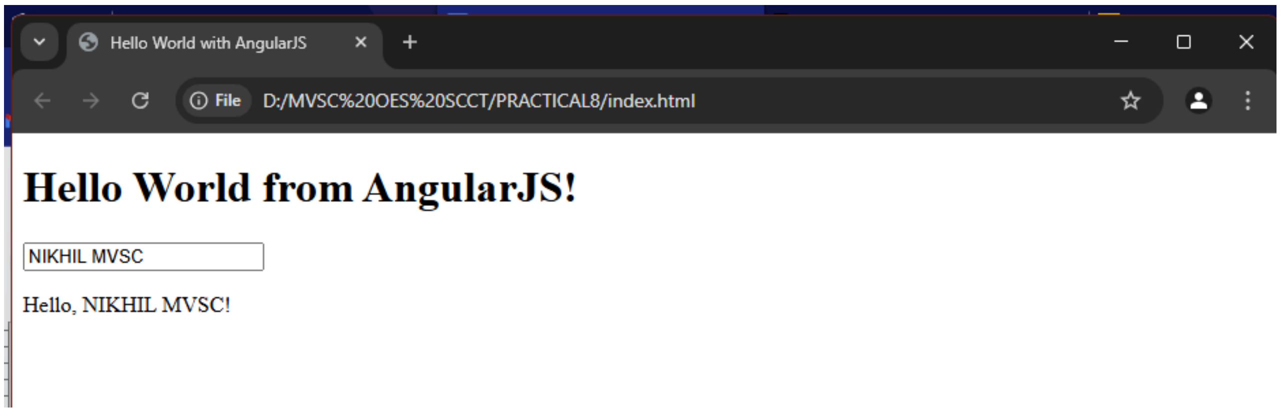
a) index.html :-

```
<!DOCTYPE html>
<html ng-app="helloApp">
<head>
    <title>Hello World with AngularJS</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angu
lar.min.js"></script>
</head>
<body>

<div ng-controller="HelloController">
    <h1>{{ message }}</h1>

    <!-- ng-model Example -->
    <input type="text" ng-model="name" placeholder="Enter your
name" />

    <!-- Display Expression -->
    <p>Hello, {{ name }}!</p>
</div>

<!-- Include AngularJS App -->
<script src="app.js"></script>

</body>
</html>
```

b) app.js :-

```
// Create AngularJS module
var app = angular.module('helloApp', []);

// Define controller
app.controller('HelloController', function($scope) {
    // Initialize message
    $scope.message = "Hello World from AngularJS!";

    // Define ng-model variable
    $scope.name = '';
});
```

Output :-


Hello World with AngularJS
File
D:/MVSC%20OES%20SCCT/PRACTICAL8/index.html

Hello World from AngularJS!

NIKHIL MVSC

Hello, NIKHIL MVSC!