```
--- pom.xml ---
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.1.4</version> <!-- Ensure this matches your desired Spring Boot version -->
<relativePath/> <!-- Lookup parent from repository -->
</parent>

<groupId>com.lms</groupId>
<artifactId>online-learning-management-system</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>Online Learning Management System</name>
<description>Backend for Online Learning Management System</description>
<packaging>jar</packaging>

<properties>
<java.version>17</java.version> <!-- Ensure this matches your Java version -->
</properties>

<dependencies>
<!-- Spring Boot Web Starter -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>

<!-- Spring Boot JPA Starter -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<!-- MySQL Connector -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
</dependency>

<!-- Spring Boot Test Starter -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Spring Boot Maven Plugin -->
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

```
--- OnlineLearningManagementSystemApplication.java ---
package com.lms;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class OnlineLearningManagementSystemApplication {

public static void main(String[] args) {
SpringApplication.run(OnlineLearningManagementSystemApplication.class, args);
}
}
```

```
--- UserController.java ---
package com.lms.controller;

import com.lms.model.User;
import com.lms.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/users")
public class UserController {

@Autowired
private UserService userService;

@PostMapping
public ResponseEntity<User> createUser(@RequestBody User user) {
User createdUser = userService.registerUser(user);
return ResponseEntity.ok(createdUser);
}

@GetMapping("/{id}")
public ResponseEntity<User> getUser(@PathVariable Long id) {
User user = userService.findUserById(id);
return ResponseEntity.ok(user);
}

@PutMapping("/{id}")
public ResponseEntity<User> updateUser(@PathVariable Long id, @RequestBody User user) {
User updatedUser = userService.updateUser(id, user);
return ResponseEntity.ok(updatedUser);
}

@DeleteMapping("/{id}")
public ResponseEntity<Void> deleteUser(@PathVariable Long id) {
userService.deleteUser(id);
return ResponseEntity.noContent().build();
}

@GetMapping
public ResponseEntity<List<User>> getAllUsers() {
List<User> users = userService.findAllUsers();
return ResponseEntity.ok(users);
}
}
```

```
--- User.java ---
package com.lms.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class User {

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;
private String name;
private String email;
private String role;

public User() {
}

public User(String name, String email, String role) {
this.name = name;
this.email = email;
this.role = role;
}

public Long getId() {
return id;
}

public void setId(Long id) {
this.id = id;
}

public String getName() {
return name;
}

public void setName(String name) {
this.name = name;
}

public String getEmail() {
return email;
}

public void setEmail(String email) {
this.email = email;
}

public String getRole() {
return role;
}

public void setRole(String role) {
this.role = role;
}
}
```

```
--- UserRepository.java ---
package com.lms.repository;

import com.lms.model.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface UserRepository extends JpaRepository<User, Long> {
// Additional query methods can be defined here if needed
}
```

```
--- UserService.java ---
package com.lms.service;

import com.lms.model.User;
import com.lms.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.Optional;

@Service
public class UserService {

@Autowired
private UserRepository userRepository;

public User registerUser(User user) {
return userRepository.save(user);
}

public Optional<User> findUserById(Long id) {
return userRepository.findById(id);
}

public void deleteUser(Long id) {
userRepository.deleteById(id);
}
}
```

```
--- application.properties ---
spring.datasource.url=jdbc:mysql://localhost:3306/lms_db
spring.datasource.username=root
spring.datasource.password=your_password
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
server.port=8080
logging.level.org.springframework=INFO
logging.level.com.lms=DEBUG
```

```
--- data.sql ---
-- This file is used to initialize the database with sample data when the application starts.

INSERT INTO users (id, name, email, role) VALUES
(1, 'John Doe', 'john.doe@example.com', 'STUDENT'),
(2, 'Jane Smith', 'jane.smith@example.com', 'INSTRUCTOR'),
(3, 'Alice Johnson', 'alice.johnson@example.com', 'STUDENT'),
(4, 'Bob Brown', 'bob.brown@example.com', 'INSTRUCTOR');
```

```
--- OnlineLearningManagementSystemApplicationTests.java ---
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class OnlineLearningManagementSystemApplicationTests {

@Test
void contextLoads() {
}

}
```

```
--- index.html ---
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Online Learning Management System</title>
<link rel="stylesheet" href="%PUBLIC_URL%/styles.css">
</head>
<body>
<div id="root"></div>
<script src="%PUBLIC_URL%/index.js"></script>
</body>
</html>
```

```
--- index.js ---
import React from 'react';
import ReactDOM from 'react-dom';
import App from './components/App';
import './index.css';

ReactDOM.render(
<React.StrictMode>
<App />
</React.StrictMode>,
document.getElementById('root')
);
```

```
--- App.js ---
import React from 'react';
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';
import Home from './Home';
import UserManagement from './UserManagement';
import NotFound from './NotFound';

const App = () => {
return (
<Router>
<div>
<h1>Online Learning Management System</h1>
<Switch>
<Route path="/" exact component={Home} />
<Route path="/users" component={UserManagement} />
<Route component={NotFound} />
</Switch>
</div>
</Router>
);
};

export default App;
```

```
--- api.js ---
import axios from 'axios';

const API_URL = 'http://localhost:8080/api/users';

export const createUser = async (userData) => {
try {
const response = await axios.post(API_URL, userData);
return response.data;
} catch (error) {
throw error.response.data;
}
};

export const getUser = async (userId) => {
try {
const response = await axios.get(`${API_URL}/${userId}`);
return response.data;
} catch (error) {
throw error.response.data;
}
};

export const updateUser = async (userId, userData) => {
try {
const response = await axios.put(`${API_URL}/${userId}`, userData);
return response.data;
} catch (error) {
throw error.response.data;
}
};

export const deleteUser = async (userId) => {
try {
await axios.delete(`${API_URL}/${userId}`);
} catch (error) {
throw error.response.data;
}
};
```