

```

//User Entity
package com.MyMoviePlan.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name="users")
public class User {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long id;
    private String name;
    @Column(unique = true)
    private String email;
    private String password;

}

//Movie Entity
package com.MyMoviePlan.entity;

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

import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@NoArgsConstructor
@AllArgsConstructor
@Data
@Table(name="movie_details")
public class Movie {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long id;
    private String name;
    private String genre;
    private String language;
    private String url;
    private String showTime;
    private int ticketPrice;
    private int noOfSeats;

}

//User Repository
package com.MyMoviePlan.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.MyMoviePlan.entity.User;

@Repository
public interface UserRepository extends JpaRepository<User, Long> {

}

//Movie Repository
package com.MyMoviePlan.repository;

```

```

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import com.MyMoviePlan.entity.Movie;

public interface MovieRepository extends JpaRepository<Movie, Long> {

    public List<Movie> findMovieByLanguage(String lang);

}

//User DTO
package com.MyMoviePlan.dto;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class UserDTO {

    private Long id;
    private String name;
    private String email;
    private String password;

}

//Movie DTO
package com.MyMoviePlan.dto;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class MovieDTO {

    private Long id;
    private String name;
    private String genre;
    private String language;
    private String url;
    private String showTime;
    private int ticketPrice;
    private int noOfSeats;

}

//User Service
package com.MyMoviePlan.service;

import java.util.List;

import com.MyMoviePlan.dto.UserDTO;
import com.MyMoviePlan.entity.User;

public interface UserService {

    public String addUser(UserDTO userDto);

    public List<User> getAllUser();

    public String updateUser(UserDTO userDto);

    public User getUserById(Long id);

    public String deleteUser(Long id);

}

```

```

//Movie Service
package com.MyMoviePlan.service;

import java.util.List;

import com.MyMoviePlan.dto.UserDTO;
import com.MyMoviePlan.entity.User;

public interface UserService {

    public String addUser(UserDTO userDto);

    public List<User> getAllUser();

    public String updateUser(UserDTO userDto);

    public User getUserById(Long id);

    public String deleteUser(Long id);

}

//User Service Implementation
package com.MyMoviePlan.serviceImplement;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.MyMoviePlan.dto.UserDTO;
import com.MyMoviePlan.entity.User;
import com.MyMoviePlan.repository.UserRepository;
import com.MyMoviePlan.service.UserService;

@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserRepository userRepository;

    @Override
    public String addUser(UserDTO userDto) {

        User user = new User();
        user.setEmail(userDto.getEmail());
        user.setName(userDto.getName());
        user.setPassword(userDto.getPassword());
        userRepository.save(user);
        return "New User Registered";
    }

    @Override
    public List<User> getAllUser() {

        return userRepository.findAll();
    }

    @Override
    public String updateUser(UserDTO userDto) {

        User userToUpdate = new User();
        userToUpdate.setId(userDto.getId());
        userToUpdate.setEmail(userDto.getEmail());
        userToUpdate.setName(userDto.getName());
        userToUpdate.setPassword(userDto.getPassword());

        userRepository.save(userToUpdate);

        return "User Updated Successfully";
    }

    @Override
    public User getUserById(Long id) {

        return userRepository.findById(id).get();
    }

}

```

```

@Override
public String deleteUser(Long id) {

    userRepository.deleteById(id);
    return "User deleted";
}

}

//Movie Service Implementation

package com.MyMoviePlan.serviceImplement;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.MyMoviePlan.dto.MovieDTO;
import com.MyMoviePlan.entity.Movie;
import com.MyMoviePlan.repository.MovieRepository;
import com.MyMoviePlan.service.MovieService;

@Service
public class MovieServiceImpl implements MovieService {

    @Autowired
    private MovieRepository movieRepository;

    @Override
    public String addMovie(MovieDTO movieDto) {

        Movie movie = new Movie();

        movie.setName(movieDto.getName());
        movie.setGenre(movieDto.getGenre());
        movie.setLanguage(movieDto.getLanguage());
        movie.setUrl(movieDto.getUrl());
        movie.setUrl(movieDto.getUrl());
        movieRepository.save(movie);
        return "New Movie Added Successfully";
    }

    @Override
    public List<Movie> getAllMovies() {
        return movieRepository.findAll();
    }

    @Override
    public String updateMovie(Long id,MovieDTO movieDto) {

        Movie movie = movieRepository.findById(id).get();

        movie.setName(movieDto.getName());
        movie.setGenre(movieDto.getGenre());
        movie.setLanguage(movieDto.getLanguage());
        movie.setUrl(movieDto.getUrl());
        movie.setUrl(movieDto.getUrl());
        movieRepository.save(movie);
        return "Updated Successfully";
    }

    @Override
    public Movie getMovieById(Long id) {

        return movieRepository.findById(id).get();
    }

    @Override
    public String deleteMovie(Long id) {
        movieRepository.deleteById(id);
        return "Movie deleted";
    }

    @Override
    public String updateSeatsById(Long id, int newSeats) {

```

```

        Movie movie = movieRepository.findById(id).get();
        movie.setNoOfSeats(newSeats);
        movieRepository.save(movie);
        return "Seats updated";
    }

    @Override
    public List<Movie> getMovieByLang(String lang){

        return movieRepository.findMovieByLanguage(lang);
    }
}

//User Controller

package com.MyMoviePlan.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.MyMoviePlan.dto.UserDTO;
import com.MyMoviePlan.entity.User;
import com.MyMoviePlan.service.UserService;

@RestController
@RequestMapping("/user")
@CrossOrigin("http://localhost:4200/")
public class UserController {

    @Autowired
    private UserService service;

    @PostMapping("/add-user")
    public String AddUser(@RequestBody UserDTO dto) {
        return service.addUser(dto);
    }

    @GetMapping("/getAll")
    public List<User> getAllUser(){
        return service.getAllUser();
    }

    @PutMapping("/updateUser")
    public String updateUser(@RequestBody UserDTO dto) {
        return service.updateUser(dto);
    }

    @DeleteMapping("/deleteUser/{id}")
    public String deleteUser(@PathVariable Long id) {
        return service.deleteUser(id);
    }

    @GetMapping("/get-user/{id}")
    public User getUserById(@PathVariable Long id) {
        return service.getUserById(id);
    }
}

//Movie Controller

package com.MyMoviePlan.controller;

```

```

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.MyMoviePlan.dto.MovieDTO;
import com.MyMoviePlan.entity.Movie;
import com.MyMoviePlan.service.MovieService;

@RestController
@RequestMapping("/movie")
@CrossOrigin("http://localhost:4200/")
public class MovieController {

    @Autowired
    private MovieService movieService;

    @PostMapping("/addMovie")
    public String addMovie(@RequestBody MovieDTO dto) {

        return movieService.addMovie(dto);
    }

    @GetMapping("/getAllMovies")
    public List<Movie> getAllMovies(){

        return movieService.getAllMovies();
    }

    @PutMapping("/updateMovie/{id}")
    public String updateEmployee(@PathVariable Long id, @RequestBody MovieDTO newMovie){
        System.out.println(newMovie);
        return movieService.updateMovie(id,newMovie);
    }

    @DeleteMapping("/deleteMovie/{id}")
    public String deleteMovie(@PathVariable Long id) {
        return movieService.deleteMovie(id);
    }

    @GetMapping("/getmovie/{id}")
    public Movie getMovieById(@PathVariable Long id) {
        return movieService.getMovieById(id);
    }

    @PutMapping("/updateSeats/{id}")
    public String updateSeats(@PathVariable Long id, int seats){
        return movieService.updateSeatsById(id, seats);
    }

    @GetMapping("/getbylang/{lang}")
    public List<Movie> getMovieByLang(@PathVariable String lang) {
        return movieService.getMovieByLang(lang);
    }

}

//Spring Boot Application

package com.MyMoviePlan;

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

@SpringBootApplication
public class MyMoviePlan3Application {

```

```

    public static void main(String[] args) {
        SpringApplication.run(MyMoviePlan3Application.class, args);
    }
}

//User Tests
package com.MyMoviePlan.serviceImplement;

import static org.assertj.core.api.Assertions.assertThat;
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;
import static org.junit.jupiter.api.Assertions.assertNotNull;

import java.util.List;

import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

import com.MyMoviePlan.dto.UserDTO;
import com.MyMoviePlan.entity.User;
import com.MyMoviePlan.service.UserService;

@SpringBootTest
public class UserServiceTest {

    @Autowired
    private UserService userService;

    @Test
    public void addUserTest() {

        UserDTO user = new UserDTO(45L, "Abc Xyz", "abc@xyz.com", "12345");
        String addedUser= userService.addUser(user);
        assertNotNull(addedUser);

    }

    @Test
    public void getAllUsersTest() {
        List list = userService.getAllUser();
        assertThat(list).size().isGreaterThan(0);
    }

    @Test
    public void updateUserTest() {

        User newuser = new User();
        newuser.setId(45L);
        newuser.setName("Abc Xyz");
        newuser.setEmail("abc@xyz.com");
        newuser.setPassword("12345");

        UserDTO newuser2 = new UserDTO(55L, "Pqr Stu", "pqr@stu.com", "67890");

        String response = userService.updateUser(newuser2);
        assertEquals("User Updated Successfully",response);

    }

}

//Movie Test
package com.MyMoviePlan.serviceImplement;

import static org.assertj.core.api.Assertions.assertThat;
import static org.junit.Assert.assertEquals;
import static org.junit.jupiter.api.Assertions.*;

import java.util.List;

import org.junit.jupiter.api.Test;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

```

```

import org.springframework.test.annotation.DirtiesContext;

import com.MyMoviePlan.dto.MovieDTO;
import com.MyMoviePlan.entity.Movie;
import com.MyMoviePlan.service.MovieService;

import jakarta.transaction.Transactional;

@SpringBootTest
@Transactional
class MovieServiceTest {

    @Autowired
    private MovieService movieService;

    private Logger logger = LoggerFactory.getLogger(this.getClass());

    @Test
    @DirtiesContext
    public void addMovieTest() {

        MovieDTO movie = new MovieDTO(10L,"Titanic","Drama","English", "titanic.jpg","12:30PM", 100, 200);

        String add = movieService.addMovie(movie);
        assertNotNull(add);

    }

    @Test
    @DirtiesContext
    public void getAllMoviesTest() {
        List list = movieService.getAllMovies();
        assertThat(list).size().isGreaterThan(0);
    }

    public void updateMovieTest() {

        Movie movie = new Movie();

        movie.setId(10L);
        movie.setLanguage("English");
        movie.setName("Titanic");
        movie.setGenre("Drama");
        movie.setNoOfSeats(100);
        movie.setShowTime("12:30PM");
        movie.setTicketPrice(200);
        movie.setUrl("titanic.jpg");

        MovieDTO movie2 = new MovieDTO(10L,"Interstellar","si-fi","English", "Interstellar.jpg","12:30PM", 100, 200);

        String response = movieService.updateMovie(10L,movie2);
        assertTrue(response.isEmpty());
        assertEquals("User Updated Successfully",response);

    }

    @Test
    @DirtiesContext
    public void getMovieByLangTest() {

        MovieDTO movie = new MovieDTO(10L,"Titanic","Drama","English", "titanic.jpg","12:30PM", 100, 200);

        String mymovie = movieService.addMovie(movie);

        assertEquals("English", movie.getLanguage());

    }

}

```