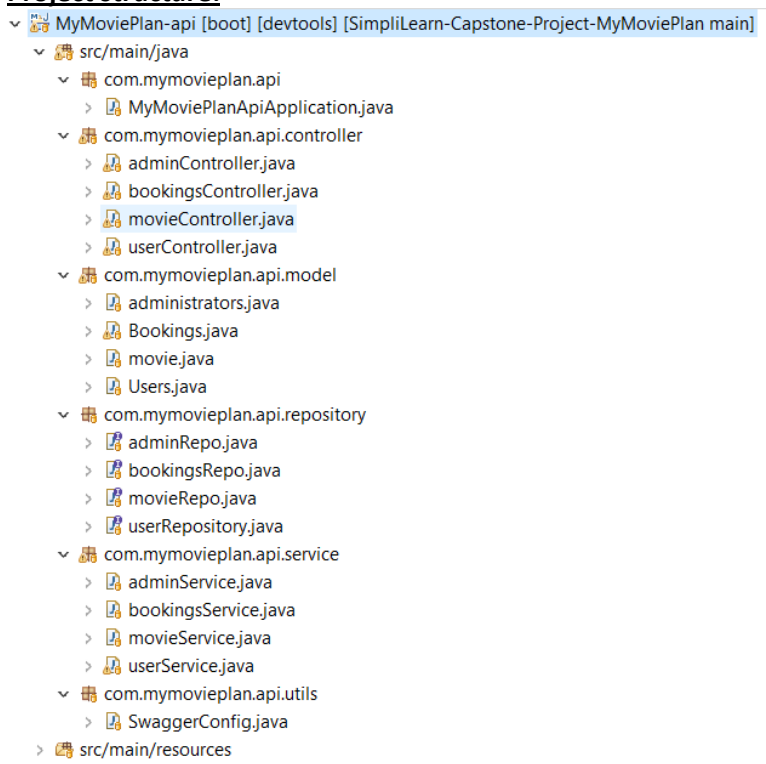
SimpliLearn Capstone Project: MyMoviePlan
Movie ticket booking portal

Submitted by: KAVIN K R

Date of submission : 16-03-2022
GitHub Project Repository URL : GitHub Link

Backend Rest API:

Project structure:



MyMoviePlanApiApplication.java

```
package com.mymovieplan.api;

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

@SpringBootApplication
public class MyMoviePlanApiApplication {

	public static void main(String[] args) {
		SpringApplication.run(MyMoviePlanApiApplication.class, args);
		System.out.println("App works!!");
	}

}
```

adminController.java

```
package com.mymovieplan.api.controller;

import java.util.HashMap;
import java.util.Map;

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.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.api.model.administrators;
import com.mymovieplan.api.service.adminService;

@CrossOrigin(allowedHeaders = "*")
@RestController
@RequestMapping("/api/admin")
public class adminController {
	@Autowired
	adminService service;
	@PostMapping("/addAdminUser")
	public Map<String,String> addAdmin(@RequestBody administrators usr){
		Map<String,String> result = new HashMap<>();
		result.put("status", "0");
		try{
			service.addAdminUser(usr);
			result.replace("status", "1");
		}catch(Exception e) {
			System.out.println(e);

		}
		return result;
	}
	@PutMapping("/changePassword")
	public Map<String,String> changePassword(@RequestBody administrators usr){
		Map<String,String> result = new HashMap<>();
		result.put("status", "0");
		try{
			service.changePassword(usr);
			result.replace("status", "1");
		}catch(Exception e) {
			System.out.println(e);

		}
		return result;
	}

	@PostMapping ("/authenticate")
	public Map<String,String> authentication(@RequestBody administrators usr){
		Map<String,String> result = new HashMap<>();
		String pattern = "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}";
		System.out.println(usr);
		if(usr.getEmail()!= null && (usr.getEmail()).matches(pattern) ) {
			return service.AuthenticateByEmail(usr);
		}else if(usr.getUserName()!=null ) {
			return service.AuthenticateByUsername(usr);
		}
```

```
                result.put("admin", "Bad request");
                result.put("authentication", "Bad request");

                return result;
        }

        @DeleteMapping("/remove")
        public Map<String,String> removeAdmin(@RequestBody administrators usr){
                Map<String,String> result = new HashMap<>();
                result.put("status", "0");
                try {
                        service.removeAdminUser(usr);
                        result.replace("status", "1");
                }catch(Exception e) {
                        System.out.println();
                }
                return result;
        }

}
```

bookingsController.java

```
package com.mymovieplan.api.controller;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
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.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.mymovieplan.api.model.Bookings;
import com.mymovieplan.api.model.Users;
import com.mymovieplan.api.service.bookingsService;

@CrossOrigin(allowedHeaders = "*")
@RestController
@RequestMapping("/api/bookings")
public class bookingsController {
        @Autowired
        bookingsService service;

        @GetMapping("/all")
        public List<Bookings> getAllBookings(){
                return service.getAllBookings();
        }

        @PostMapping("/saveBooking")
        public Map<String, String> saveBooking(@RequestBody Bookings bk){
                Map<String, String> status = new HashMap<>();
```

```
                status.put("status", null);
                try {
                Bookings bkng =         service.saveBooking(bk);
                        status.replace("status", Integer.toString(bkng.getBookingId()));
                }catch(Exception e) {
                        System.out.println(e);
                        status.replace("status", "error");
                }
                return status;
        }
        @PostMapping("/user")
        public List<Bookings> getBookingsByUser(@RequestBody Users usr){
                return service.getAllBookingsByUsername(usr);
        }
}
```

movieController.java

```
package com.mymovieplan.api.controller;

import java.sql.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PatchMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import com.mymovieplan.api.model.movie;
import com.mymovieplan.api.service.movieService;

@CrossOrigin(allowedHeaders = "*")
@RestController
@RequestMapping("/api/movie")
public class movieController {

        @Autowired
        movieService service;

        @GetMapping("/all")
        public List<movie> getAllMovies() {
                return service.getAllMovie();
        }
        @GetMapping("/find/{id}")
        public movie getById(@PathVariable("id")int id ) {
                movie mv = new movie();
                mv.setMovieId(id);
                return service.findById(mv);
        }
```

```
	@PostMapping("/addMovie")
	public Map<String, String> addMovie(@RequestBody movie mv) {
		System.out.println(mv.getReleasedate());
		return service.addMovie(mv);
	}

	@GetMapping("/movieName")
	public movie getMovieByName(@RequestParam(name = "title") String name) {
		movie mv = new movie();
		mv.setTitle(name);
		return service.findByName(mv);
	}

	@PatchMapping("/statusChange")
	public Map<String, String> changeStatus(@RequestBody movie mv) {
		return service.toggleStatus(mv);
	}

	@PatchMapping("/updateMovie")
	public Map<String,String> updateMovieDetails(@RequestBody movie mv){
		return service.updateMovie(mv);
	}
}
```

userController.java

```
package com.mymovieplan.api.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.util.StringUtils;
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.PatchMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import com.mymovieplan.api.model.Users;
import com.mymovieplan.api.service.userService;


@CrossOrigin(allowedHeaders = "*")
@RestController
@RequestMapping("/api/user")
public class userController {
	@Autowired
```

```
	userService usrService;

	@GetMapping("/")
	public String landingGreeting(){
		return "You have landed on the Users page!!!<br><h1>Have a nice day!!</h1>";
	}

	@GetMapping("/all")
	public List<Users> getAllUsers(){
		return usrService.findAllUsers();
	}

	@GetMapping("/finduser")
	public Users getUserByName(@RequestParam(name="name") String name){
		Users usr;
		String pattern = "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}";
		if(name.matches(pattern)) {
			usr = usrService.findByEmail(name);
		}else {
		usr = usrService.findUserByUserName(name);
		}
		return usr;
	}

	@PatchMapping("/updateuser/{id}")
	public Map<String, String> updateUser(@PathVariable("id") int id, @RequestBody Users
usr){
		Map<String, String> status = new HashMap<>();

		try {
			usrService.updateUser(usr);
			status.put("status", "1");
			Users updateduser = new Users();
			updateduser=usrService.findUserById(id);
			status.put("uid", Integer.toString(updateduser.getUid()));
			status.put("firstname", updateduser.getFirstName());
			status.put("middlename", updateduser.getMiddleName());
			status.put("lastname", updateduser.getLastName());
			status.put("username", updateduser.getUserName());
			status.put("email", updateduser.getEmail());
			status.put("password", updateduser.getPassword());
			status.put("country", updateduser.getCountry());
			}
			catch(Exception e) {
				System.out.println(e);
				status.put("status", "0");
			}
			return status;
	}

	@PostMapping("/adduser")
	public Map<String, Integer> createUser(@RequestBody Users[] usr){

		Map<String, Integer> status = new HashMap<>();
		try {
```

```
		usrService.addUser(usr);
		status.put("status", 1);
		}
		catch(Exception e) {
			status.put("status", 0);
		}
		return status;
	}

	@DeleteMapping("/remove/{id}")
	public Map<String, Integer> deleteUser(@PathVariable int id){
		Map<String, Integer> status = new HashMap<>();
		try {
		usrService.deleteUser(usrService.findUserById(id));
		status.put("status", 1);
		}
		catch(Exception e) {
			System.out.println(e);
			status.put("status", 0);
		}
		return status;
	}


	}
```

Administrators.java

```
package com.mymovieplan.api.model;

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

@Entity
public class administrators {
	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int adminId;
	private String userName;
	private String password;
	private String email;
	private String role;

	public administrators() {
		// TODO Auto-generated constructor stub
	}

	public int getAdminId() {
		return adminId;
	}

	public void setAdminId(int adminId) {
		this.adminId = adminId;
	}
```

```
	public String getUserName() {
		return userName;
	}

	public void setUserName(String userName) {
		this.userName = userName;
	}

	public String getPassword() {
		return password;
	}

	public void setPassword(String password) {
		this.password = password;
	}

	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;
	}

	@Override
	public String toString() {
		return "administrators [adminId=" + adminId + ", userName=" + userName + ",
password=" + password + ", email="
				+ email + ", role=" + role + "]";
	}

}
```

Bookings.java

```
package com.mymovieplan.api.model;

import java.time.LocalDateTime;
import java.util.HashMap;
import java.util.List;

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

@Entity
public class Bookings {
	@Id
```

```
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int bookingId;
	private int userId;
	private List<String> bookingDetails;
	private double bookingAmount;
	private LocalDateTime bookingStamp;

	public Bookings() {

	}
	public int getBookingId() {
		return bookingId;
	}
	public void setBookingId(int bookingId) {
		this.bookingId = bookingId;
	}
	public int getUserId() {
		return userId;
	}
	public void setUserId(int userId) {
		this.userId = userId;
	}
	public List<String> getBookingDetails() {
		return bookingDetails;
	}
	public void setBookingDetails(List<String> bookingDetails) {
		this.bookingDetails = bookingDetails;
	}
	public double getBookingAmount() {
		return bookingAmount;
	}
	public void setBookingAmount(double bookingAmount) {
		this.bookingAmount = bookingAmount;
	}
	public LocalDateTime getBookingStamp() {
		return bookingStamp;
	}
	public void setBookingStamp(LocalDateTime bookingStamp) {
		this.bookingStamp = bookingStamp;
	}



}
```

Movie.java

```
package com.mymovieplan.api.model;

import java.sql.Date;
import java.util.List;

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

```
public class movie {
		@Id
		@GeneratedValue(strategy = GenerationType.IDENTITY)
		private int movieId;
		private String title;
		private String genre;
		private float rating;
		private Date releaseDate;
		private boolean status;
		private List<String> actors;
		private double runtime;
		private String language;
		private String imageUrl;
		private String director;
		private double moviePrice;

		public movie() {
			// TODO Auto-generated constructor stub
		}

		public int getMovieId() {
			return movieId;
		}

		public void setMovieId(int movieId) {
			this.movieId = movieId;
		}

		public String getTitle() {
			return title;
		}

		public void setTitle(String title) {
			this.title = title;
		}

		public String getGenre() {
			return genre;
		}

		public void setGenre(String genre) {
			this.genre = genre;
		}



		public Date getReleasedate() {
			return releaseDate;
		}

		public void setReleasedate(Date releasedate) {
			this.releaseDate = releasedate;
		}

		public boolean isStatus() {
```

```
		return status;
	}

	public void setStatus(boolean status) {
		this.status = status;
	}

	public List<String> getActors() {
		return actors;
	}

	public void setActors(List<String> actors) {
		this.actors = actors;
	}

	public double getRuntime() {
		return runtime;
	}

	public void setRuntime(double runtime) {
		this.runtime = runtime;
	}

	public String getLanguage() {
		return language;
	}

	public void setLanguage(String language) {
		this.language = language;
	}

	public String getImageUrl() {
		return imageUrl;
	}

	public void setImageUrl(String imageUrl) {
		this.imageUrl = imageUrl;
	}

	public String getDirector() {
		return director;
	}

	public void setDirector(String director) {
		this.director = director;
	}

	public double getMoviePrice() {
		return moviePrice;
	}

	public void setMoviePrice(double moviePrice) {
		this.moviePrice = moviePrice;
	}
	public float getRating() {
```

```
		return rating;
	}

	public void setRating(float rating) {
		this.rating = rating;
	}

	@Override
	public String toString() {
		return "movie [movieId=" + movieId + ", title=" + title + ", genre=" + genre
+ ", rating=" + rating
				+ ", releasedate=" + releaseDate + ", status=" + status + ",
actors=" + actors + ", runtime="
				+ runtime + ", language=" + language + ", imageUrl=" +
imageUrl + ", director=" + director
				+ ", moviePrice=" + moviePrice + "]";
	}




}
```

Users.java

```
package com.mymovieplan.api.model;

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

@Entity
public class Users {
	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int Uid;
	private String firstName;
	private String middleName;
	private String lastName;
	private String userName;
	private String email;
	private String password;
	private String country;


	public Users() {
		// TODO Auto-generated constructor stub
	}

	public int getUid() {
		return Uid;
	}
	public void setUid(int uid) {
		Uid = uid;
	}
	public String getFirstName() {
```

```
		return firstName;
	}
	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}
	public String getMiddleName() {
		return middleName;
	}
	public void setMiddleName(String middleName) {
		this.middleName = middleName;
	}
	public String getLastName() {
		return lastName;
	}
	public void setLastName(String lastName) {
		this.lastName = lastName;
	}
	public String getUserName() {
		return userName;
	}
	public void setUserName(String userName) {
		this.userName = userName;
	}
	public String getEmail() {
		return email;
	}
	public void setEmail(String email) {
		this.email = email;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	public String getCountry() {
		return country;
	}
	public void setCountry(String country) {
		this.country = country;
	}
	@Override
	public String toString() {
		return "Users [Uid=" + Uid + ", firstName=" + firstName + ", middleName=" +
middleName + ", lastName="
				+ lastName + ", userName=" + userName + ", email=" + email + ",
password=" + password + ", country="
				+ country + "]";
	}

}
```

adminRepo.java

```
package com.mymovieplan.api.repository;

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

```
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import com.mymovieplan.api.model.administrators;

@Repository
public interface adminRepo extends JpaRepository<administrators, Integer> {

	@Query("SELECT u FROM administrators u WHERE u.userName=?1")
	public administrators findByUsername(String username);
	@Query("SELECT u FROM administrators u WHERE u.email=?1")
	public administrators findByEmail(String email);
}
```

bookingsRepo.java

```
package com.mymovieplan.api.repository;

import java.util.List;

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

import com.mymovieplan.api.model.Bookings;

@Repository
public interface bookingsRepo extends JpaRepository<Bookings, Integer>{

	@Query("Select b from Bookings b where b.userId=?1")
	public List<Bookings> getAllBookingsByUser(int uid);
}
```

movieRepo.java

```
package com.mymovieplan.api.repository;

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

import com.mymovieplan.api.model.movie;

@Repository
public interface movieRepo extends JpaRepository<movie, Integer> {
	@Query("SELECT m FROM movie m WHERE m.title=?1")
	public movie findByName(String title);
}
```

userRepository.java

```
package com.mymovieplan.api.repository;

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

import org.springframework.stereotype.Repository;

import com.mymovieplan.api.model.Users;
```

```
@Repository
public interface userRepository extends JpaRepository<Users, Integer>{

	@Query("SELECT u from Users u WHERE u.userName=?1")
	Users findByUserName(String name);
	@Query("SELECT u from Users u WHERE u.email=?1")
	Users findByEmail(String email);

}
```

adminService.java

```
package com.mymovieplan.api.service;

import java.util.HashMap;
import java.util.Map;

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

import com.mymovieplan.api.model.administrators;
import com.mymovieplan.api.repository.adminRepo;

@Service
public class adminService {
	@Autowired
	adminRepo repo;

	public void addAdminUser(administrators usr) {
		repo.save(usr);

	}

	public void removeAdminUser(administrators usr) {
		administrators u =null;
		if(usr.getUserName()!=null) {
			u = repo.findByUsername(usr.getUserName());
		}else if(usr.getEmail()!=null) {
			u = repo.findByEmail(usr.getEmail());
		}
		if(u!=null) {
			repo.delete(u);
		}

	}

	public Map<String, String> AuthenticateByUsername(administrators usr) {
		Map<String, String> result = new HashMap<>();
		result.put("admin", "0");
		result.put("authentication", "0");
		administrators u = null;
		u = repo.findByUsername(usr.getUserName());
		if (u == null) {
			return result;
		} else if ((u.getUserName()).equals(usr.getUserName())) {
			result.put("admin", "1");
			if ((u.getPassword()).equals(usr.getPassword())) {
```

```
                                result.put("authentication", "1");
                        }

                }
                System.out.println(result);
                return result;
        }
        public Map<String, String> AuthenticateByEmail(administrators usr) {
                Map<String, String> result = new HashMap<>();
                result.put("admin", "0");
                result.put("authentication", "0");
                administrators u = null;
                u = repo.findByEmail(usr.getEmail());
                System.out.println("passed :"+ usr);
                System.out.println("found :"+ u);
                if (u == null) {
                        return result;
                } else if ((u.getEmail()).equals(usr.getEmail())) {
                        result.put("admin", "1");
                        if ((u.getPassword()).equals(usr.getPassword())) {
                                result.put("authentication", "1");
                        }

                }
                return result;
        }

        public String changePassword(administrators usr) {
                String result;
                administrators u =repo.findByUsername(usr.getUserName());
                u.setPassword(usr.getPassword());
                try {
                        repo.save(u);
                        result="1";
                }
                catch(Exception e) {
                        System.out.println(e);
                        result="0";
                }
                return result;
        }
}
```

bookingsService.java

```
package com.mymovieplan.api.service;

import java.util.List;

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

import com.mymovieplan.api.model.Bookings;
import com.mymovieplan.api.model.Users;
import com.mymovieplan.api.repository.bookingsRepo;

@Service
```

```
public class bookingsService {
	@Autowired
	private bookingsRepo repo;

	public List<Bookings> getAllBookings() {
		return repo.findAll();
	}

	public Bookings saveBooking(Bookings bk) {
		return repo.save(bk);
	}

	public List<Bookings> getAllBookingsByUsername(Users usr) {
		return repo.getAllBookingsByUser(usr.getUid());
	}
}
```

movieService.java:

```
package com.mymovieplan.api.service;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

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

import com.mymovieplan.api.model.movie;
import com.mymovieplan.api.repository.movieRepo;

@Service
public class movieService {

		@Autowired
		movieRepo repo;

		public List<movie> getAllMovie(){
			return repo.findAll();
		}

		public movie findById(movie mv){
			return repo.findById(mv.getMovieId()).get();
		}

		public movie findByName(movie mv) {
			return repo.findByName(mv.getTitle());
		}
		public Map<String,String> addMovie(movie mv){
			Map<String,String> status = new HashMap<>();
			status.put("status", null);
			try {
				repo.save(mv);
				status.replace("status", "1");
			} catch (Exception e) {
				System.out.println(e);
				status.replace("status", "0");
```

```
		}
		return status;
	}

	public Map<String,String> updateMovie(movie mv){
		Map<String,String> status = new HashMap<>();
		movie mv1 = repo.findById(mv.getMovieId()).get();

		if(mv.getTitle()==null) {
			mv.setTitle(mv1.getTitle());
		}
		if(mv.getGenre()==null) {
			mv.setGenre(mv1.getGenre());
		}
		if(mv.getRating()==0) {
			mv.setRating(mv1.getRating());
		}
		if(mv.getReleasedate()==null) {
			mv.setReleasedate(mv1.getReleasedate());
		}
		//mv.setStatus(mv1.isStatus());
		if(mv.getActors()==null) {
			mv.setActors(mv1.getActors());
		}
		if(mv.getRuntime()==0) {
			mv.setRuntime(mv1.getRuntime());
		}
		if(mv.getLanguage()==null) {
			mv.setLanguage(mv1.getLanguage());
		}
		if(mv.getImageUrl()==null) {
			mv.setImageUrl(mv1.getImageUrl());
		}
		if(mv.getDirector()==null) {
			mv.setDirector(mv1.getDirector());
		}
		if(mv.getMoviePrice()==0) {
			mv.setMoviePrice(mv1.getMoviePrice());
		}


		status.put("status", null);
		try {
			repo.save(mv);
			status.replace("status", "1");
		} catch (Exception e) {
			System.out.println(e);
			status.replace("status", "0");
		}
		return status;
	}

	public Map<String,String> deleteMovie(movie mv){
		Map<String,String> status = new HashMap<>();
		status.put("status", null);
```

```
                try {
                        repo.delete(mv);
                        status.replace("status", "1");
                } catch (Exception e) {
                        System.out.println(e);
                        status.replace("status", "0");
                }
                return status;
        }

        public Map<String,String> toggleStatus(movie mv){
                Map<String,String> status =new HashMap<>();
                movie mv1 = repo.findById(mv.getMovieId()).get();
                status.put("status", null);
                mv1.setStatus(mv.isStatus());
                try {
                        repo.save(mv1);
                        status.replace("status", "1");
                } catch (Exception e) {
                        System.out.println();
                        status.replace("status", "0");
                }
                return status;
        }
}
```

userService.java

```
package com.mymovieplan.api.service;

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

import java.util.List;

import com.mymovieplan.api.model.Users;
import com.mymovieplan.api.repository.userRepository;
@Service
public class userService {

@Autowired
userRepository repository;
        //find all users
        public List<Users>findAllUsers() {
                return repository.findAll();
        }

        //find all by id
        public Users findUserById(Integer id) {
                return repository.getById(id);
        }

        //find users by username
        public Users findUserByUserName(String name) {
                System.out.println("finding by name " + name);
                return repository.findByUserName(name);
        }
```

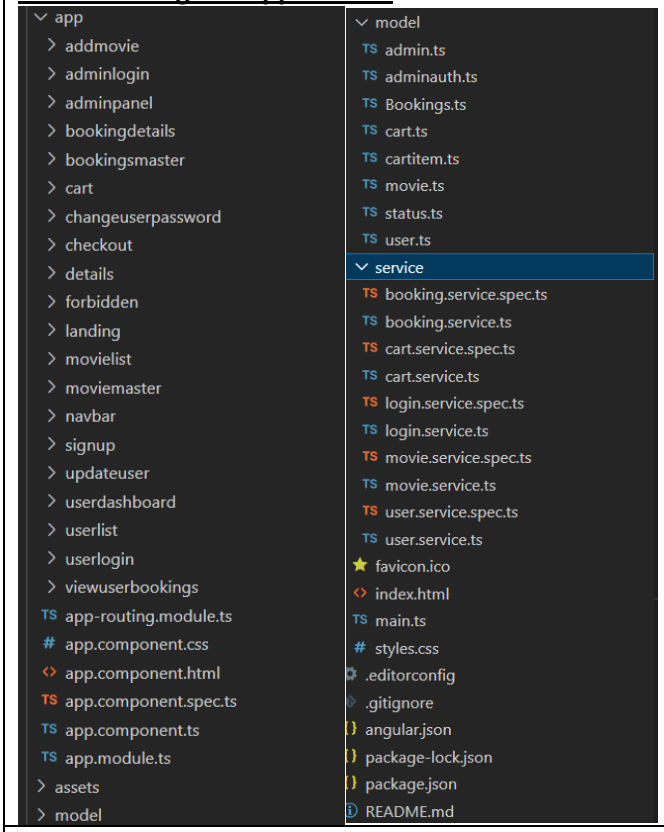
```
	public Users findByEmail(String email) {
		System.out.println("finding by email" +email);
		return repository.findByEmail(email);
	}

	public void addUser(Users[] usr) {
		for(Users u:usr) {
		repository.save(u);
		}
	}

	//update userdetails
	public void updateUser(Users usr) {
		Users oldUserdata = repository.getById(usr.getUid());
		if(usr.getFirstName()==null) {
			usr.setFirstName(oldUserdata.getFirstName());
		}
		if(usr.getLastName()==null) {
			usr.setLastName(oldUserdata.getLastName());
		}
		if (usr.getMiddleName()==null) {
			usr.setMiddleName(oldUserdata.getMiddleName());
		}
		if(usr.getUserName()==null) {
			usr.setUserName(oldUserdata.getUserName());
		}
		if(usr.getEmail()==null) {
			usr.setEmail(oldUserdata.getEmail());
		}
		if(usr.getPassword()==null) {
			usr.setPassword(oldUserdata.getPassword());
		}
		if(usr.getCountry()==null) {
			usr.setCountry(oldUserdata.getCountry());
		}

		repository.save(usr);

	}

	//delete user by userId
	public void deleteUser(Users usr) {
		repository.deleteById(usr.getUid());
	}

}
```

SwaggerConfig.java

```
package com.mymovieplan.api.utils;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import io.swagger.v3.oas.annotations.OpenAPIDefinition;
import io.swagger.v3.oas.models.OpenAPI;
```

```
import io.swagger.v3.oas.models.info.Info;

@OpenAPIDefinition
@Configuration
public class SwaggerConfig {

	@Bean
	public OpenAPI baseOpenAPI() {
		return new OpenAPI().info(new Info().title("Spring
Doc").version("1.0.0").description("Spring Doc"));
	}
}
```

Front End Angular Application:



Addmovie.component.html

```
<div class="container py-5 h-100">
  <div class="row d-flex justify-content-center align-items-center h-100">
    <span class="text-center">
      <h3>Add Movie</h3>
    </span>
    <div class="col-lg-8 col-xl-6">

      <div class="card rounded-3">

        <div class="card-body p-4 p-md-5">
          <form>
            <!-- 2 column grid layout with text inputs for the first and last names -->
            <div class="row">
              <div class="col">
                <div class="form-outline">
                  <input type="text" id="title`" class="form-control" name="title"
[(ngModel)]="movie.title"/>
                  <label class="form-label" for="title">Title</label>
```

```
                    </div>
                  </div>

                </div>
                <div class="row">
                  <div class="col">
                    <div class="form-outline">
                      <input type="text" id="genre" class="form-control" name="genre"
[(ngModel)]="movie.genre"/>
                      <label class="form-label" for="genre">Genre</label>
                    </div>
                  </div>
                  <div class="col">
                    <div class="form-outline">
                      <input type="number" id="rating" min="1" max="5" class="form-control"
name="rating" [(ngModel)]="movie.rating" />
                      <label class="form-label" for="rating">Rating</label>
                    </div>
                  </div>
                </div>
                <div class="row">
                  <div class="col">
                    <div class="form-outline">
                      <input type="date" id="releasedate" class="form-control"
name="releasedate" [(ngModel)]="movie.releasedate"/>
                      <label class="form-label" for="releasedate">Release Date</label>
                    </div>
                  </div>
                  <div class="col">
                    <div class="form-outline">
                      <input type="number" id="runtime" class="form-control" name="runtime"
[(ngModel)]="movie.runtime"/>
                      <label class="form-label" for="runtime">Runtime(mins)</label>
                    </div>
                  </div>
                </div>
                <div class="form-outline">
                  <input type="email" id="language" class="form-control" name="language"
[(ngModel)]="movie.language"/>
                  <label class="form-label" for="language">Language</label>
                </div>
                <div class="row">
                  <div class="col">
                <div class="form-outline">
                  <input type="text" id="actor" name="actor"class="form-control"
[(ngModel)]="actortemp"/>
                  <label class="form-label" for="actor">Add actor</label>
                </div></div>
                <div class="col">
                <button class="btn btn-primary btn-block mb-4" (click)="addActor()">Add
actor</button>
                </div></div>
                <h6>Actors:</h6>
                <li *ngFor="let actor of movie.actors">{{actor}}</li>
                <br><div class="row">
```

```
                    <div class="col">
                      <div class="form-outline">
                        <input type="text" id="director" class="form-control" name="director"
[(ngModel)]="movie.director" />
                        <label class="form-label" for="director">Director</label>
                      </div>
                    </div>
                    <div class="col">
                      <div class="form-outline">
                        <input type="number" id="price" class="form-control" name="price"
[(ngModel)]="movie.moviePrice"/>
                        <label class="form-label" for="price">Price</label>
                      </div>
                    </div>
                  </div>
                  <div class="form-outline">
                    <input type="text" id="imageurl" class="form-control" name="url"
[(ngModel)]="movie.imageUrl" />
                    <label class="form-label" for="imageurl">Image file name</label>
                  </div>
                  <!-- Submit button -->
                  <button class="btn btn-primary btn-block mb-4" (click)="saveMovie()">Add
movie</button>

                </form>
              </div>
            </div>
          </div>
        </div>
```

addmovie.component.ts

```
import { Location } from '@angular/common';
import { HttpClient } from '@angular/common/http';
import { Component } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { movie } from 'src/model/movie';
import { MovieService } from 'src/service/movie.service';

@Component({
 selector: 'app-addmovie',
 templateUrl: './addmovie.component.html',
 styleUrls: ['./addmovie.component.css']
})
export class AddmovieComponent {
 movie:movie = new movie();
 actortemp:string="";
 constructor(private mvservice:MovieService, private location:Location){}
 ngOnInit(){
  this.movie.actors=[];
 }
 addActor(){
  if(this.actortemp !==""){
  this.movie.actors.push(this.actortemp);
  this.actortemp='';
```

```
  }
 }
 saveMovie(){
  this.mvservice.addMovie(this.movie).subscribe(res=>console.log(res));
  this.location.back();
 }
}
```

Adminlogin.component.html

```
<section class="vh-100" style="background-color: #ffc800;">
  <div class="container py-5 h-100">
   <div class="row d-flex justify-content-center align-items-center h-100">
    <div class="col col-xl-10">
     <div class="card" style="border-radius: 1rem;">
      <div class="row g-0">
       <div class="col-md-6 col-lg-5 d-none d-md-block">
        <img src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-login-
form/img1.webp"
          alt="login form" class="img-fluid" style="border-radius: 1rem 0 0 1rem;" />
       </div>
       <div class="col-md-6 col-lg-7 d-flex align-items-center">
        <div class="card-body p-4 p-lg-5 text-black">

         <form>

          <div class="d-flex align-items-center mb-3 pb-1">
           <img src="favicon.ico" width="30" height="30" class="d-inline-block align-top"
alt="">
           <span class="navbar-brand mb-0 h1">MyMoviePlan</span>
          </div>

          <h5 class="fw-normal mb-3 pb-3" style="letter-spacing: 1px;">Sign into your
account</h5>

          <div class="form-outline mb-4">
           <input type="email" id="form2Example17" class="form-control form-control-lg"
name="username" [(ngModel)]="admin.email"/>
           <label class="form-label" for="form2Example17">Email address or Username</label>
          </div>

          <div class="form-outline mb-4">
           <input type="password" id="form2Example27" class="form-control form-control-lg"
name="password" [(ngModel)]="admin.password"/>
           <label class="form-label" for="form2Example27">Password</label>
          </div>

          <div class="pt-1 mb-4">
           <button class="btn btn-dark btn-lg btn-block" type="button"
(click)="adminLogin()">Login</button>
          </div>


         </form>

        </div>
       </div>
```

```
        </div>
       </div>
      </div>
     </div>
   </div>
 </section>
```

Adminlogin.component.ts

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { admin } from 'src/model/admin';
import { LoginService } from 'src/service/login.service';

@Component({
 selector: 'app-adminlogin',
 templateUrl: './adminlogin.component.html',
 styleUrls: ['./adminlogin.component.css']
})
export class AdminloginComponent {
 admin:admin = new admin();
 constructor(private loginservice:LoginService, private router:Router){}

 adminLogin(){
  this.admin.userName=this.admin.email;
  this.loginservice.adminAuth(this.admin).then(res=>{
   if(res){
    this.router.navigateByUrl("adminpanel");
   }
  });

 }

}
```

Adminpanel.component.html

```
<app-navbar></app-navbar>
<div class="d-flex" style="margin-top:1rem;">
  <ul class="list-inline mx-auto justify-content-center">
    <li class="list-inline-item"><button class="btn btn-primary" routerLink="userlist">View Users
Master</button></li>
    <li class="list-inline-item"><button class="btn btn-primary" routerLink="moviemaster">View
Movie Master</button></li>
    <li class="list-inline-item"><button class="btn btn-
primary"routerLink="bookingsmaster">View Movie Bookings</button></li>
  </ul>
</div>
<router-outlet></router-outlet>
```

Adminpanel.component.ts

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { LoginService } from 'src/service/login.service';



@Component({
 selector: 'app-adminpanel',
```

```
  templateUrl: './adminpanel.component.html',
  styleUrls: ['./adminpanel.component.css']
})
export class AdminpanelComponent {

  constructor(private loginservice:LoginService, private router:Router){}
  ngOnInit(){
    if(this.loginservice.adminauthstatus){

    }else{
     // this.router.navigateByUrl('forbidden');
    }
  }

}
```

Bookingdetails.component.html

```
<nav class="navbar navbar-light bg-light">
   <a class="navbar-brand" href="#">
      <img src="favicon.ico" width="30" height="30" class="d-inline-block align-top" alt="">
      <span class="navbar-brand mb-0 h1">MyMoviePlan</span>
   </a>
</nav>

<div class="card" style="width:50rem; margin: 5px auto;">
   <h3 class="text-center">Your Booking Details:</h3>
   <h4 class="text-center">Your Booking ID:{{bookingID}}</h4>
   <ul class="list-group list-group-horizontal" *ngFor="let item of cart.items">
      <li class="list-group-item"><img src="/assets/movie/{{item.movie.imageUrl}}" class="img-
fluid rounded" style="width:10rem;object-fit: cover;" alt="Responsive image"></li>
      <li class="list-group-item text-center" style="width:31%;"><h4 >{{item.movie.title}}</h4></li>
      <li class="list-group-item text-center" style="width:31%;">Total Tickets: {{item.quantity}}</li>
      <li class="list-group-item text-center" style="width:31%;">Price: {{item.price |
currency:"INR"}}</li>

    </ul>
    <h5 style="text-align: right;">Total Booking Price: {{cart.totalprice | currency:"INR"}}</h5>
</div>
<p class="lead text-center">Thank you for choosing MyMoviePlan as your trusted booking
partner. Excited to serve you again!!!</p>
```

Bookingdetails.component.ts

```
import { Component } from '@angular/core';
import { timestamp } from 'rxjs';
import { Bookings } from 'src/model/Bookings';
import { Cart } from 'src/model/cart';
import { BookingService } from 'src/service/booking.service';
import { CartService } from 'src/service/cart.service';
import { LoginService } from 'src/service/login.service';

@Component({
  selector: 'app-bookingdetails',
  templateUrl: './bookingdetails.component.html',
  styleUrls: ['./bookingdetails.component.css']
})
export class BookingdetailsComponent {
  cart: Cart;
```

```
  bkng: Bookings = {
    bookingId: 0,
    userId: 0,
    bookingDetails: [],
    bookingAmount: 0,
    bookingStamp: new Date()
  }
  bookingID: string='0';
  constructor(
    private cartservice: CartService,
    private loginservice: LoginService,
    private bookingservice: BookingService) { }
  ngOnInit() {
    this.cart = this.cartservice.getCart();
    console.log(this.loginservice.getCurrentUsr());
    this.bkng.userId = this.loginservice.getCurrentUsr().uid;
    this.bkng.bookingAmount = this.cart.totalprice;
    this.bkng.bookingStamp = new Date();
    this.cart.items.forEach(x => {
      this.bkng.bookingDetails.push(x.movie.title +'--'+ x.quantity.toString()+' Nos.');
    });
    if(this.bkng.userId !== -1){
    this.bookingservice.saveBooking(this.bkng).subscribe(res => {
      console.log(res);
      this.bookingID = res.status
    });
  }
  }
}
```

Bookingsmaster.component.html

```
<div class="container">
   <table class="table table-bordered table-striped">
      <thead class="table-dark">
         <tr>
            <th class="text-center">Booking ID</th>
            <th class="text-center">User ID</th>
            <th class="text-center">Booking Details</th>
            <th class="text-center">Booking Amount</th>
            <th class="text-center">Booking Time</th>
         </tr>
      </thead>
      <tbody>
         <tr *ngFor="let bkng of bkngs">
            <td>{{bkng.bookingId}}</td>
            <td>{{bkng.userId}}</td>
            <td>{{bkng.bookingDetails}}</td>
            <td>{{bkng.bookingAmount | currency:'INR'}}</td>
            <td>{{bkng.bookingStamp | date:'dd-MM-yyy hh:mm:ss'}}</td>
         </tr>
      </tbody>
   </table>
```

Bookingsmaster.component.ts

```
import { Component } from '@angular/core';
import { Bookings } from 'src/model/Bookings';
import { BookingService } from 'src/service/booking.service';
```

```
@Component({
 selector: 'app-bookingsmaster',
 templateUrl: './bookingsmaster.component.html',
 styleUrls: ['./bookingsmaster.component.css']
})
export class BookingsmasterComponent {
 bkngs:Bookings[];
 constructor(private bookingservice:BookingService){}
 ngOnInit(){
  this.bookingservice.getAllBookings().subscribe(res=>{this.bkngs=res;});
 }
}
```

Cart.component.html

```
<div style="float:left;">
   <span><b>Your Cart Total:</b></span>
   <p>{{cart.totalprice | currency:"INR"}}</p>

</div>
<div style="float:right;">
   <button class="btn btn-success" *ngIf="cart.items.length>0"
routerLink="checkout">Checkout</button>
</div>
<div class="clearfix"></div>
<div class="card" *ngFor="let item of cart.items">
   <div class="card-body">
      <div class="row">
         <table>
            <tr>
               <th style="width:35%;">Title</th>
               <th style="width:25%;">Quantity</th>
               <th style="width:25%;">Price</th>
               <th style="width:15%;">Action</th>
            </tr>
       <tr>
         <td style="width:35%;"> {{item.movie.title}}</td>
         <td style="width:25%;">{{item.quantity}}</td>
          <td style="width:25%;"> {{item.price | currency:"INR"}}</td>
          <td style="width:15%;"><a class="btn btn-danger btn-sm"
(click)="removefromcart(item)"><i class="bi bi-x"></i></a></td>
      </tr>
      </table>
      </div>

   </div>
</div>
```

Cart.component.ts

```
import { Component } from '@angular/core';
import { Cart } from 'src/model/cart';
import { cartitem } from 'src/model/cartitem';
import { CartService } from 'src/service/cart.service';

@Component({
 selector: 'app-cart',
 templateUrl: './cart.component.html',
```

```
  styleUrls: ['./cart.component.css']
})
export class CartComponent {
  cart:Cart = new Cart();
  constructor(private cartservice:CartService){}
  ngOnInit(){
    this.cart = this.cartservice.getCart();
  }
  removefromcart(itm:cartitem){
    this.cartservice.removeFromCart(itm.movie.movieId);
  }
}
```

Changeuserpassword.component.html

```
<div class="container py-5 h-100">
   <div class="row d-flex justify-content-center align-items-center h-100">
      <div class="col col-md-9 col-lg-7 col-xl-5">
         <div class="card" style="border-radius: 15px;">
            <div class="card-body p-4 ">
               <div class="d-flex text-black">

                  <div class="flex-grow-1 ms-3">
                     Enter Old Password:
                     <input type="password" id="form3Example1q1" class="form-control"
name="oldpass"
                        [ngStyle]="{'border-color': oldpass? '' : '#ff8282'}"
                        [(ngModel)]="oldpass" />
                        Enter New Password:
                     <input type="password" id="form3Example1q2" class="form-control"
name="newpass"
                        [ngStyle]="{'border-color': newpass? '' : '#ff8282'}"
                        [(ngModel)]="newpass" />
                        Confirm New Password:
                     <input type="password" id="form3Example1q3" class="form-control"
name="newpassre"
                        [ngStyle]="{'border-color': newpassre==newpass? '' : '#ff8282'}"
                        [(ngModel)]="newpassre" />
                     <button class="btn btn-primary" *ngIf="oldpass && newpass==newpassre"
(click)="changepassword()">Change Password</button>
                  </div>
               </div>
            </div>
         </div>
      </div>
   </div>
</div>
```

Changeuserpassword.component.ts

```
import { Location } from '@angular/common';
import { Component } from '@angular/core';
import { user } from 'src/model/user';
import { LoginService } from 'src/service/login.service';
import { UserService } from 'src/service/user.service';

@Component({
  selector: 'app-changeuserpassword',
  templateUrl: './changeuserpassword.component.html',
```

```
  styleUrls: ['./changeuserpassword.component.css']
})
export class ChangeuserpasswordComponent {
  oldpass:string;
  newpass:string;
  newpassre:string;
  currentuser:user;
  constructor(private loginservice:LoginService, private userservice:UserService, private
location:Location){}
  ngOnInit(){
    this.currentuser=this.loginservice.getCurrentUsr();
  }

  changepassword(){
    if(this.oldpass==this.currentuser.password){
      if(this.newpass==this.newpassre){
        this.currentuser.password=this.newpass;

this.userservice.updateUser(this.currentuser).subscribe(res=>{if(res.status=="1"){alert('Password
Changed Successfully! Please login again to continue.')}});
        this.loginservice.logout();
        this.location.back();
      }
    }else{
      alert('Please check your old password');
      this.oldpass='';
      this.newpass='';
      this.newpassre='';
    }
  }
}
```

Checkout.component.html

```
<nav class="navbar navbar-light bg-light">
    <a class="navbar-brand" routerLink="">
        <img src="favicon.ico" width="30" height="30" class="d-inline-block align-top" alt="">
        <span class="navbar-brand mb-0 h1">MyMoviePlan</span>
    </a>
</nav>
<div class="card" id="cartitems">
    <p style="display:inline;float:left;"><b><i class="bi bi-cart4"></i> Your Cart Items:</b></p>
    <span style="display:inline;float:right;"> <i class="bi bi-cart4"></i><b> Your Cart Total:</b>
        <p class="text-primary"><b>{{cart.totalprice | currency:"INR"}}</b></p> </span>
        <div class="clearfix"></div>
    <div class="card" *ngFor="let item of cart.items">
        <div class="card-body">
            <div class="row">
                <table>
                    <tr>
                        <th style="width:35%;">Title</th>
                        <th style="width:25%;">Quantity</th>
                        <th style="width:25%;">Price</th>
                        <th style="width:15%;">Action</th>
                    </tr>
                    <tr>
                        <td style="width:35%;"> {{item.movie.title}}</td>
```

```
                <td style="width:25%;">{{item.quantity}}</td>
                <td style="width:25%;"> {{item.price | currency:"INR"}}</td>
                <td style="width:15%;"><a class="btn btn-danger btn-sm"
(click)="removefromcart(item)"><i class="bi bi-x"></i></a>
                </td>
              </tr>
            </table>

          </div>

        </div>
      </div>


</div>
<div class="card" id="payment">
  <p><b>Payment Gateway:</b></p>
  <form>
    <div class="row">
      <div class="col">
        <div class="form-outline">
          <label class="form-label" for="cardnumber">Credit/Debit card number</label>
          <input type="number" id="cardnumber" class="form-control" name="cardnumber"
            placeholder="Enter your card number" [(ngModel)]="cardnumber" />
        </div>
      </div>
      <div class="col">
        <div class="form-outline">
          <label class="form-label" for="cvv">CVV number</label>
          <input type="number" id="cvv" class="form-control" name="cvv" min="001"
max="999"
            placeholder="Enter your CVV" [(ngModel)]="cvv" />
        </div>
      </div>

    </div>
    <div class="col">
      <div class="form-outline">
        <label class="form-label" for="name">Name on card</label>
        <input type="text" id="name" class="form-control" name="name" placeholder="Enter
name on card"
          [(ngModel)]="cardname" />
      </div>
    </div>
    <br>
    <div class="text-center">
      <button class="btn btn-sm btn-success" *ngIf="cardnumber && cvv && cardname"
(click)="pay()"> Pay
        Now</button>
    </div>
  </form>
</div>
```

Checkout.component.ts

```
import { Location } from '@angular/common';
import { Component } from '@angular/core';
```

```
import { Router } from '@angular/router';
import { Cart } from 'src/model/cart';
import { cartitem } from 'src/model/cartitem';
import { CartService } from 'src/service/cart.service';
import { LoginService } from 'src/service/login.service';

@Component({
 selector: 'app-checkout',
 templateUrl: './checkout.component.html',
 styleUrls: ['./checkout.component.css']
})
export class CheckoutComponent {
 cart:Cart = new Cart();
 cardnumber:number;
 cvv:number;
 cardname:string;
 constructor(
  private cartservice:CartService,
  private router:Router,
  private loginservice:LoginService,
  private location:Location){}
 ngOnInit(){
  if(this.loginservice.userauthstatus){

  }else{
   alert('Please login as a user to continue!!');
   this.location.back();
  }
  this.cart = this.cartservice.getCart();
 }
 removefromcart(itm:cartitem){
  this.cartservice.removeFromCart(itm.movie.movieId);
 }
 pay(){
  const response = confirm("Confirm your purchase of "+this.cart.totalprice+ " INR for the movie
tickets ?");
  if(response){
   alert("Payment Successful");
   this.router.navigateByUrl("bookingdetails")
  }else{
   alert("Payment Aborted");
  }

 }
}
```

Details.component.html

```
<section class="vh-100" style="background-color: #efefef;">
 <div class="container py-5 h-100">
  <div class="row d-flex justify-content-center align-items-center h-100">
   <div class="col col-md-9 col-lg-7 col-xl-5">
    <div class="card" style="border-radius: 15px;">
     <div class="card-body p-4 "  >
      <div class="d-flex text-black">
       <div class="flex-shrink-0" >
```

```
            <img src="/assets/user_images/{{user.userName}}.jpg"
              alt="Generic placeholder image" class="img-fluid "
              style="width: 180px; border-radius: 10px;" onerror =
"this.onerror=null;this.src='/assets/user_images/placeholder.jpg'">
          </div>
          <div class="flex-grow-1 ms-3">
            <h5 class="mb-1">{{user.firstName +' '+user.lastName}}</h5>
            <p class="mb-2 pb-1" style="color: #2b2a2a;">@username:{{user.userName}}</p>
            <div class="d-flex justify-content-start rounded-3 p-2 mb-2"
              style="background-color: #efefef;">
              <div>
                e-mail: <p class="small text-muted mb-1">{{user.email}}</p>

              </div>
              <div class="px-3">
                Country: <p class="small text-muted mb-1">{{user.country}}</p>

              </div>

            </div>
            <div class="d-flex pt-1">
              <button type="button" class="btn btn-outline-primary me-1 flex-grow-1"
(click)="navigateToUserlist()">Go to Userlist</button>
              <button type="button" class="btn btn-primary flex-grow-1"
(click)="navigateToUpdate(user.userName)">Edit</button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
 </div>
</section>
```

Details.component.ts

```
import { Component } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { user } from 'src/model/user';
import { UserService } from 'src/service/user.service';

@Component({
 selector: 'app-details',
 templateUrl: './details.component.html',
 styleUrls: ['./details.component.css']
})
export class DetailsComponent {
 user:user;
 constructor(private usrService:UserService, private route:ActivatedRoute, private
router:Router){}
 ngOnInit(){

  const username = this.route.snapshot.paramMap.get('usrname');
  this.usrService.getUserByName(String(username)).subscribe(data=>this.user=data);
 }
 navigateToUserlist(){
```

```
   this.router.navigate(['/adminpanel/userlist']);
  }
  navigateToUpdate(param:string){
   this.router.navigate(['/adminpanel/update',param])
  }
}
```

Forbidden.component.html

```
<section class="vh-100" style="background-color: #ffc800;">

   <div class="container py-5 h-100">
    <div class="row d-flex justify-content-center align-items-center h-100">
     <div class="col col-xl-10">

       <div class="row">
          <div class="col-md-2 text-center">
           <h1 style="font-size:90px;"><i class="bi bi-exclamation-triangle"></i></h1>
             <p><b>Status Code: 403</b></p>
          </div>
          <div class="col-md-10">
             <h3>OPPSSS!!!! Sorry...</h3>
             <p>Sorry, your access is refused due to security reasons of our server and also our
sensitive data.<br/>Please go back to the previous page to continue browsing.</p>
             <a class="btn btn-danger" (click)="backClicked()">Go Back</a>
          </div>
       </div>
    </div>
</div>
</div>
</section>
```

Forbidden.component.ts

```
import { Component } from '@angular/core';
import {Location} from '@angular/common';
import { Router } from '@angular/router';

@Component({
 selector: 'app-forbidden',
 templateUrl: './forbidden.component.html',
 styleUrls: ['./forbidden.component.css']
})
export class ForbiddenComponent {

 constructor(private location: Location, private router:Router){}
 backClicked() {
  this.router.navigateByUrl('');
 }
}
```

Landing.component.html

```
<app-navbar></app-navbar>
 <h3 class="text-center">Welcome to MyMoviePlan Ticket Booking Application</h3>
<div class="movielist" [ngStyle]="{ 'width': (cart.items.length==0)? '100%' : '70%' }">
 <app-movielist></app-movielist>
</div>
<div class="cart" *ngIf="cart.items.length>0" >
<app-cart></app-cart>
```

```
</div>
<div class="clearfix"></div>
```

Landing.component.ts

```
import { Component,OnInit } from '@angular/core';
import { Cart } from 'src/model/cart';
import { user } from 'src/model/user';
import { CartService } from 'src/service/cart.service';
import { LoginService } from 'src/service/login.service';
import { UserService } from 'src/service/user.service';

@Component({
  selector: 'app-landing',
  templateUrl: './landing.component.html',
  styleUrls: ['./landing.component.css']
})
export class LandingComponent implements OnInit {
  cart:Cart;
  currentuser:string;
  constructor(private cartservice:CartService, private loginservice:LoginService){

    }
    ngOnInit():void{
     this.cart=this.cartservice.getCart();
    this.currentuser= this.loginservice.getCurrentUser();
    if(this.currentuser=="0"){
      this.currentuser='Guest'
    }
    console.log(this.currentuser);
    }


}
```

Movielist.component.html

```
<h4>Available Movies:</h4>
<div class="card-deck">
  <div class="row">
    <div class="card" style="width: 18rem;padding-right:0;padding-left:0;margin:3px" *ngFor="let
movie of movies">
      <img class="card-img-top" style="height:10rem;object-fit: cover;"
src="/assets/movie/{{movie.imageUrl}}"
        alt="Card image cap">
      <div class="card-body">
        <h5 class="card-title">{{movie.title}}</h5>
        <small class="text-muted">{{movie.language}}</small> | <small class="text-
muted">{{movie.director}}</small>
        <p class="card-text">
          <a class="text-decoration-none text-primary" data-bs-toggle="collapse"
href="#coll{{movie.movieId}}"
            role="button" aria-expanded="false" aria-controls="coll1">
            Show Details
          </a>

        </p>
        <div class="collapse" id="coll{{movie.movieId}}">
```

```
        <p class="card-text"><b>Genre:</b> {{movie.genre}} <br><b>Actors:</b>
          <li *ngFor="let actor of movie.actors">{{actor}}</li>
        </p>
       </div>

       <a href="#" class="btn btn-primary" (click)="click(movie.movieId)">Book Ticket</a>
       <span class="price" style="float:right;"><h5 class="font-weight-bold text-primary"
style="text-decoration: none;">{{movie.moviePrice | currency:"INR"}}</h5></span>

     </div>
     <div class="card-footer">
       <small class="text-muted">Released: {{(movie.releasedate | date:"dd-MMM-yyy")}}
</small><br>
       <small class="text-muted">Runtime: {{movie.runtime}} mins </small>
     </div>
    </div>


  </div>
</div>
```

Movielist.component.ts

```
import { Component } from '@angular/core';
import { movie } from 'src/model/movie';
import { CartService } from 'src/service/cart.service';
import { MovieService } from 'src/service/movie.service';

@Component({
  selector: 'app-movielist',
  templateUrl: './movielist.component.html',
  styleUrls: ['./movielist.component.css']
})
export class MovielistComponent {
  movies:movie[];


  constructor(private movieService:MovieService,private cartservice:CartService){}
  ngOnInit(){
    this.movieService.getAllMovies().subscribe(data=>{
      this.movies = data;
      this.movies=this.movies.filter(x=>x.status);
    });


  }

  click(movieId:number){
    let  mv1:movie;
    this.movieService.getMovieById(movieId).subscribe(data=>{
      mv1 = data;
     this.cartservice.addToCart(mv1);
    });
  }

}
```

Moviemaster.component.html

```
<div class="container">
   <button class="btn btn-success" (click)="navigateToAddMovie()">Add Movie</button>
   <table class="table table-bordered table-striped" >
      <thead class="table-dark">
         <tr>
            <th>Movie ID</th>
            <th>Movie Poster</th>
            <th>Movie Title</th>
            <th>Director</th>
            <th>Language</th>
            <th>Genre</th>
            <th>Actors</th>
            <th>Runtime</th>
            <th>Release Date</th>
            <th>Rating</th>
            <th>Movie Price</th>
            <th>Movie Status</th>
         </tr>
      </thead>
      <tbody>
         <tr *ngFor="let movie of movies">
            <td>{{movie.movieId}}</td>
            <td><img class="card-img-top" style="height:10rem;object-fit: cover;"
src="/assets/movie/{{movie.imageUrl}}"
               alt="Card image cap"></td>
               <td>{{movie.title}}</td>
               <td>{{movie.director}}</td>
               <td>{{movie.language}}</td>
               <td>{{movie.genre}}</td>
               <td><li *ngFor="let actor of movie.actors">{{actor}}</li></td>
               <td>{{movie.runtime}} mins</td>
               <td>{{movie.releasedate | date:"dd-MMM-yyy"}}</td>
               <td>{{movie.rating}}</td>
               <td>{{movie.moviePrice | currency : "INR"}}</td>
               <td><mat-slide-toggle color="accent" [checked]="movie.status"
(change)="toggleChanged(movie.movieId, $event)"></mat-slide-toggle></td>

         </tr>
      </tbody>
   </table>
</div>
```

Moviemaster.component.ts

```
import { Component } from '@angular/core';
import { MatSlideToggleChange } from '@angular/material/slide-toggle';
import { Router } from '@angular/router';
import { movie } from 'src/model/movie';
import { MovieService } from 'src/service/movie.service';

@Component({
 selector: 'app-moviemaster',
 templateUrl: './moviemaster.component.html',
 styleUrls: ['./moviemaster.component.css']
})
export class MoviemasterComponent {
```

```
 movies:movie[];
 constructor(private movieservice:MovieService, private router:Router){}
 ngOnInit(){
  this.movieservice.getAllMovies().subscribe(res=>{this.movies=res;});
 }
 toggleChanged(id:number,event:MatSlideToggleChange){
  let mv:movie={
   "movieId":id,
   "status":event.checked,
   "title":"",
  "genre":"",
  "rating":0,
  "releasedate":new Date,
  "actors":[],
  "runtime":0,
  "language":"",
  "imageUrl":"",
  "director":"",
  "moviePrice":0,
  }
  this.movieservice.changestatus(mv).subscribe(res=>{});
 }
 navigateToAddMovie(){
  this.router.navigate(["adminpanel/addmovie"]);
 }
}
```

Navbar.component.html

```
<!-- Image and text -->
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand" routerLink="">
    <img src="favicon.ico" width="30" height="30" class="d-inline-block align-top" alt="">
    <span class="navbar-brand mb-0 h1">MyMoviePlan</span>
  </a>
  <span>Welcome {{currentuser}} !!</span>
  <div id="navbarSupportedContent">
     <form class="form-inline my-2 my-lg-0">

       <span style="margin:5px;" *ngIf="!login">New User?</span>
       <button style="margin:5px;" *ngIf="!login"class="btn btn-outline-primary my-2 my-sm-0"
type="submit" routerLink="signup">Sign Up</button>

        <button style="margin:5px;" *ngIf="!login" class="btn btn-outline-success my-2 my-sm-0"
type="submit" routerLink="userlogin">User Login</button>
        <button style="margin:5px;" *ngIf="!login" class="btn btn-outline-warning my-2 my-sm-0"
type="submit" routerLink="adminlogin">Admin Login</button>
        <button style="margin:5px;" *ngIf="login" class="btn btn-outline-danger my-2 my-sm-0"
type="submit" (click)="logout()">Logout</button>

       </form>
  </div>
 </nav>
```

Navbar.component.ts

```
import { Component } from '@angular/core';
import { Route, Router } from '@angular/router';
import { LoginService } from 'src/service/login.service';
```

```
@Component({
 selector: 'app-navbar',
 templateUrl: './navbar.component.html',
 styleUrls: ['./navbar.component.css']
})
export class NavbarComponent {
 currentuser:string;
 login:boolean=true;
 constructor(private loginservice:LoginService,private router:Router){

  }
  ngOnInit():void{

   this.currentuser= this.loginservice.getCurrentUser();
   if(this.currentuser=="0"){
    this.currentuser='Guest'
    this.login=false;
   }

   }
   logout(){
    this.loginservice.logout();
    this.currentuser= this.loginservice.getCurrentUser();
    if(this.currentuser=="0"){
     this.currentuser='Guest'
     this.login=false;
    }
    this.router.navigateByUrl("");
   }
}
```

Signup.component.html

```
<section class="h-100 h-custom" style="background-color: #8fc4b7;">
  <div class="container py-5 h-100">
    <div class="row d-flex justify-content-center align-items-center h-100">
      <div class="col-lg-8 col-xl-6">
        <div class="card rounded-3">
          <img src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-
registration/img3.webp"
            class="w-100" style="border-top-left-radius: .3rem; border-top-right-radius: .3rem;"
            alt="Sample photo">
          <div class="card-body p-4 p-md-5">
            <h3 class="mb-4 pb-2 pb-md-0 mb-md-5 px-md-2">User Registration</h3>
            <form class="px-md-2" (ngSubmit)="onSubmit()">
              <div class="row">
                <div class="form-outline mb-4 col-md-6 mb-4">
                  <label class="form-label" for="form3Example1q">First Name</label>
                  <input type="text" id="form3Example1q1" class="form-control"
name="firstName"
                    [ngStyle]="{'border-color': user.firstName? '' : '#ff8282'}"
                    [(ngModel)]="user.firstName" />
                </div>
                <div class="form-outline mb-4 col-md-6 mb-4">
                  <label class="form-label" for="form3Example1q">Middle Name</label>
```

```
                    <input type="text" id="form3Example1q2" class="form-control"
name="middleName"
                        [(ngModel)]="user.middleName" />
                  </div>
                  <div class="form-outline mb-4 col-md-6 mb-4">
                    <label class="form-label" for="form3Example1q">Last Name</label>
                    <input type="text" id="form3Example1q3" class="form-control"
name="lastName"
                        [ngStyle]="{'border-color': user.lastName? '' : '#ff8282'}"
                        [(ngModel)]="user.lastName" />
                  </div>
                  <div class="form-outline mb-4 col-md-6 mb-4">
                    <label class="form-label" for="form3Example1q">Username</label>
                    <input type="text" id="form3Example1q3" class="form-control"
                      placeholder="choose your username" name="userName"
                      [ngStyle]="{'border-color': user.userName? '' : '#ff8282'}"
                      [(ngModel)]="user.userName" />
                  </div>
                  <div class="form-outline mb-4 col-md-6 mb-4">
                    <label class="form-label" for="form3Example1q">E-mail</label>
                    <input type="text" id="form3Example1q3" class="form-control"
                      placeholder="your.email@example.com" name="email"
                      [ngStyle]="{'border-color': user.email? '' : '#ff8282'}"
                      [(ngModel)]="user.email" />
                  </div>
                  <div class="form-outline mb-4 col-md-6 mb-4">
                    <label class="form-label" for="form3Example1q">Country</label>
                    <div>
                      <select id="form3Example1q3" class="select" name="country"
                        [ngStyle]="{'border-color': user.country? '' : '#ff8282'}"
                        [(ngModel)]="user.country">
                        <option *ngFor="let country of country_list"
value="{{country}}">{{country}}
                        </option>
                      </select>
                    </div>
                  </div>
                  <div class="form-outline mb-4 col-md-6 mb-4">
                    <label class="form-label" for="form3Example1q">Password</label>
                    <input type="password" id="form3Example1q3" class="form-control"
                      [ngStyle]="{'border-color': password? '' : '#ff8282'}"
                      placeholder="Enter a password" name="password"
[(ngModel)]="password" />
                  </div>
                  <div class="form-outline mb-4 col-md-6 mb-4">
                    <label class="form-label" for="form3Example1q">Confirm Password</label>
                    <input type="password" id="form3Example1q3" class="form-control"
                      [ngStyle]="{'border-color': confirmpassword? '' : '#ff8282'}"
                      placeholder="Re-Enter the password" name="confirmpassword"
                      [(ngModel)]="confirmpassword" />
                  </div>
                </div>
                <br>
                <button type="submit" class="btn btn-success mb-1">Submit</button>
              </form>
```

```
            </div>
          </div>
        </div>
      </div>
    </div>
</section>
```

Signup.component.ts

```
import { HttpClient } from '@angular/common/http';
import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { user } from 'src/model/user';
import { UserService } from 'src/service/user.service';

@Component({
  selector: 'app-signup',
  templateUrl: './signup.component.html',
  styleUrls: ['./signup.component.css']
})
export class SignupComponent {
 country_list:string[] = ["Afghanistan","Albania","Algeria","Andorra","Angola","Anguilla","Antigua
&amp;
Barbuda","Argentina","Armenia","Aruba","Australia","Austria","Azerbaijan","Bahamas","Bahrain",
"Bangladesh","Barbados","Belarus","Belgium","Belize","Benin","Bermuda","Bhutan","Bolivia","Bo
snia &amp; Herzegovina","Botswana","Brazil","British Virgin Islands","Brunei","Bulgaria","Burkina
Faso","Burundi","Cambodia","Cameroon","Cape Verde","Cayman
Islands","Chad","Chile","China","Colombia","Congo","Cook Islands","Costa Rica","Cote D
Ivoire","Croatia","Cruise Ship","Cuba","Cyprus","Czech
Republic","Denmark","Djibouti","Dominica","Dominican Republic","Ecuador","Egypt","El
Salvador","Equatorial Guinea","Estonia","Ethiopia","Falkland Islands","Faroe
Islands","Fiji","Finland","France","French Polynesia","French West
Indies","Gabon","Gambia","Georgia","Germany","Ghana","Gibraltar","Greece","Greenland","Gre
nada","Guam","Guatemala","Guernsey","Guinea","Guinea
Bissau","Guyana","Haiti","Honduras","Hong
Kong","Hungary","Iceland","India","Indonesia","Iran","Iraq","Ireland","Isle of
Man","Israel","Italy","Jamaica","Japan","Jersey","Jordan","Kazakhstan","Kenya","Kuwait","Kyrgyz
Republic","Laos","Latvia","Lebanon","Lesotho","Liberia","Libya","Liechtenstein","Lithuania","Luxe
mbourg","Macau","Macedonia","Madagascar","Malawi","Malaysia","Maldives","Mali","Malta","
Mauritania","Mauritius","Mexico","Moldova","Monaco","Mongolia","Montenegro","Montserrat",
"Morocco","Mozambique","Namibia","Nepal","Netherlands","Netherlands Antilles","New
Caledonia","New
Zealand","Nicaragua","Niger","Nigeria","Norway","Oman","Pakistan","Palestine","Panama","Papu
a New Guinea","Paraguay","Peru","Philippines","Poland","Portugal","Puerto
Rico","Qatar","Reunion","Romania","Russia","Rwanda","Saint Pierre &amp;
Miquelon","Samoa","San Marino","Satellite","Saudi
Arabia","Senegal","Serbia","Seychelles","Sierra Leone","Singapore","Slovakia","Slovenia","South
Africa","South Korea","Spain","Sri Lanka","St Kitts &amp; Nevis","St Lucia","St Vincent","St.
Lucia","Sudan","Suriname","Swaziland","Sweden","Switzerland","Syria","Taiwan","Tajikistan","Ta
nzania","Thailand","Timor L'Este","Togo","Tonga","Trinidad &amp;
Tobago","Tunisia","Turkey","Turkmenistan","Turks &amp; Caicos","Uganda","Ukraine","United
Arab Emirates","United Kingdom","Uruguay","Uzbekistan","Venezuela","Vietnam","Virgin Islands
(US)","Yemen","Zambia","Zimbabwe"];
  user:user =new user();
  password:string;
  confirmpassword:string;
```

```
 constructor(private userService:UserService,private router:Router){

}


onSubmit(){
  let saved:boolean=false;
  if(this.password==this.confirmpassword){
    this.user.password=this.password;
    if(this.user.firstName){
      if(this.user.lastName){
        if(this.user.userName){
          if(this.user.email){
            if(this.user.password){
              if(this.user.country){
                this.userService.addUser(this.user).subscribe(data=>{console.log(data);})
                saved=true;
              }
            }
          }
        }
      }
    }

  }else{
    alert('Password Does not match');
  }
  if(saved){
    alert("Registration completed Successfully");
    this.router.navigateByUrl("/");
  }else{
    alert("Kindly enter missing fields");
  }


}


}
```

Updateuser.component.html

```
<section class="vh-100" style="background-color: #efefef;">
   <div class="container py-5 h-100">
     <div class="row d-flex justify-content-center align-items-center h-100">
       <div class="col col-md-9 col-lg-7 col-xl-5">
         <div class="card" style="border-radius: 15px;">
           <div class="card-body p-4 "  >
             <div class="d-flex text-black">

               <div class="flex-grow-1 ms-3">
                 First Name:
                 <input type="text" id="form3Example1q3" class="form-control"
name="firstName"[ngStyle]="{'border-color': user.firstName? '' : '#ff8282'}"
[(ngModel)]="user.firstName" />
                 Middle Name:
```

```
            <input type="text" id="form3Example1q3" class="form-control" name="middleName"
[(ngModel)]="user.middleName" />
            Last Name:
            <input type="text" id="form3Example1q3" class="form-control"
name="lastName"[ngStyle]="{'border-color': user.lastName? '' : '#ff8282'}"
[(ngModel)]="user.lastName" />

            Username:
            <input type="text" id="form3Example1q3" class="form-control"
name="userName"[ngStyle]="{'border-color': user.userName? '' : '#ff8282'}"
[(ngModel)]="user.userName" />
            <div class="d-flex justify-content-start rounded-3 p-2 mb-2"
              style="background-color: #efefef;">
              <div>
                e-mail: <p class="small text-muted mb-1">{{user.email}}</p>

              </div>
              <div class="px-3">
                <div>
                  Country:
                  <select id="form3Example1q3" class="select" name="country" [ngStyle]="{'border-
color': user.country? '' : '#ff8282'}" [(ngModel)]="user.country">
                    <option *ngFor="let country of country_list"
value="{{country}}">{{country}}</option>
                  </select></div>

              </div>

            </div>
            <div class="d-flex pt-1">
              <button type="button" class="btn btn-outline-primary me-1 flex-grow-1"
(click)="cancel()">cancel</button>
              <button type="button" class="btn btn-primary flex-grow-1"
(click)="onSave()">Save</button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
 </section>
```

Updateuser.component.ts

```
import { Component } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { user } from 'src/model/user';
import { UserService } from 'src/service/user.service';
import { Location } from '@angular/common';


@Component({
 selector: 'app-updateuser',
 templateUrl: './updateuser.component.html',
 styleUrls: ['./updateuser.component.css']
```

```
})
export class UpdateuserComponent {
  country_list:string[] = ["Afghanistan","Albania","Algeria","Andorra","Angola","Anguilla","Antigua
&amp;amp;
Barbuda","Argentina","Armenia","Aruba","Australia","Austria","Azerbaijan","Bahamas","Bahrain",
"Bangladesh","Barbados","Belarus","Belgium","Belize","Benin","Bermuda","Bhutan","Bolivia","Bo
snia &amp;amp; Herzegovina","Botswana","Brazil","British Virgin Islands","Brunei","Bulgaria","Burkina
Faso","Burundi","Cambodia","Cameroon","Cape Verde","Cayman
Islands","Chad","Chile","China","Colombia","Congo","Cook Islands","Costa Rica","Cote D
Ivoire","Croatia","Cruise Ship","Cuba","Cyprus","Czech
Republic","Denmark","Djibouti","Dominica","Dominican Republic","Ecuador","Egypt","El
Salvador","Equatorial Guinea","Estonia","Ethiopia","Falkland Islands","Faroe
Islands","Fiji","Finland","France","French Polynesia","French West
Indies","Gabon","Gambia","Georgia","Germany","Ghana","Gibraltar","Greece","Greenland","Gre
nada","Guam","Guatemala","Guernsey","Guinea","Guinea
Bissau","Guyana","Haiti","Honduras","Hong
Kong","Hungary","Iceland","India","Indonesia","Iran","Iraq","Ireland","Isle of
Man","Israel","Italy","Jamaica","Japan","Jersey","Jordan","Kazakhstan","Kenya","Kuwait","Kyrgyz
Republic","Laos","Latvia","Lebanon","Lesotho","Liberia","Libya","Liechtenstein","Lithuania","Luxe
mbourg","Macau","Macedonia","Madagascar","Malawi","Malaysia","Maldives","Mali","Malta","
Mauritania","Mauritius","Mexico","Moldova","Monaco","Mongolia","Montenegro","Montserrat",
"Morocco","Mozambique","Namibia","Nepal","Netherlands","Netherlands Antilles","New
Caledonia","New
Zealand","Nicaragua","Niger","Nigeria","Norway","Oman","Pakistan","Palestine","Panama","Papu
a New Guinea","Paraguay","Peru","Philippines","Poland","Portugal","Puerto
Rico","Qatar","Reunion","Romania","Russia","Rwanda","Saint Pierre &amp;amp;
Miquelon","Samoa","San Marino","Satellite","Saudi
Arabia","Senegal","Serbia","Seychelles","Sierra Leone","Singapore","Slovakia","Slovenia","South
Africa","South Korea","Spain","Sri Lanka","St Kitts &amp;amp; Nevis","St Lucia","St Vincent","St.
Lucia","Sudan","Suriname","Swaziland","Sweden","Switzerland","Syria","Taiwan","Tajikistan","Ta
nzania","Thailand","Timor L'Este","Togo","Tonga","Trinidad &amp;amp;
Tobago","Tunisia","Turkey","Turkmenistan","Turks &amp;amp; Caicos","Uganda","Ukraine","United
Arab Emirates","United Kingdom","Uruguay","Uzbekistan","Venezuela","Vietnam","Virgin Islands
(US)","Yemen","Zambia","Zimbabwe"];

 user:user=new user();
 constructor(
   private usrService:UserService,
   private route:ActivatedRoute,
   private router:Router,
   private location: Location){

 }
   ngOnInit(){
    const username = this.route.snapshot.paramMap.get('usrname');
     this.usrService.getUserByName(String(username)).subscribe(data=>{this.user=data;});
   }
  onSave(){

   if(this.user.firstName){
     if(this.user.lastName){
       if(this.user.userName){
         if(this.user.email){
           if(this.user.password){
             if(this.user.country){
```

```
            this.usrService.updateUser(this.user).subscribe(data=>{if(data.status=="1"){
              alert("Details updated successfully");
            }else {
              alert("Error in updating details")
            }
           })

            this.location.back();
          }
         }
        }
       }
     }
   }
  cancel(){
   this.location.back();
  }
}
```

Userdashboard.component.html

```
<app-navbar></app-navbar>
<div class="profile" style="display:inline;float:left;width:20%;background-color:#efefef;">
  <div class="row d-flex justify-content-center align-items-center h-100">
    <div class="col col-md-9 col-lg-7 col-xl-5">
      <div class="flex-shrink-0">
        <img src="/assets/user_images/{{currntuser.userName}}.jpg" alt="Generic placeholder
image"
          class="img-fluid " style="width: 180px; border-radius: 10px;"
          onerror="this.onerror=null;this.src='/assets/user_images/placeholder.jpg'">
      </div>
      <div class="flex-grow-1 ms-3">
        <h5 class="mb-1">{{currntuser.firstName +' '+currntuser.lastName}}</h5>
        <p class="mb-2 pb-1" style="color: #2b2a2a;">@username:{{currntuser.userName}}</p>

          <div>
            e-mail: <p class="small text-muted mb-1">{{currntuser.email}}</p>

          </div>
          <div>
            Country: <p class="small text-muted mb-1">{{currntuser.country}}</p>

          </div>

      </div>
    </div>
  </div>
</div>
<div class="content" style="display:inline;float:right;width:80%"><button class="btn btn-success"
routerLink="">Book
    Ticket Now</button>
  <div class="d-flex" style="margin-top:1rem;">
    <ul class="list-inline mx-auto justify-content-center">
      <li class="list-inline-item"><button class="btn btn-primary"
          routerLink="update/{{currntuser.userName}}">Edit Profile Details</button></li>
```

```
        <li class="list-inline-item"><button class="btn btn-primary"
routerLink="changepassword">Change
              Password</button></li>
        <li class="list-inline-item"><button class="btn btn-primary" routerLink="bookings">View
Movie
              Bookings</button></li>
      </ul>
   </div>
   <router-outlet></router-outlet>
</div>
```

Userdashboard.component.ts

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { user } from 'src/model/user';
import { LoginService } from 'src/service/login.service';
import { UserService } from 'src/service/user.service';

@Component({
  selector: 'app-userdashboard',
  templateUrl: './userdashboard.component.html',
  styleUrls: ['./userdashboard.component.css']
})
export class UserdashboardComponent {
currentuser:string;
currntuser:user;
constructor(private loginservice:LoginService, private userservice:UserService, private
router:Router){}
ngOnInit(){
 this.currentuser= this.loginservice.getCurrentUser();
 this.currntuser=this.loginservice.getCurrentUsr();
  if(this.currentuser=="0" && this.currntuser.uid ==-1){
  this.currentuser='Guest';
 this.router.navigateByUrl('forbidden');
 }
}
}
```

Userlist.component.html

```
<div class="container">
   <button class="btn btn-success" routerLink="/signup">Add User</button>
   <table class="table table-bordered table-striped" >
   <thead class="table-dark">
      <tr>
      <th class="text-center">User ID</th>
      <th class="text-center">FULL NAME</th>
      <th class="text-center">USERNAME</th>
      <th class="text-center">E-MAIL</th>
      <th class="text-center">COUNTRY</th>
      <th class="text-center">ACTIONS</th>

   </tr>
</thead>
   <tbody>
   <tr *ngFor="let user of users">
      <td>{{user.uid}}</td>
      <td>{{user.firstName +' '+ user.lastName}}</td>
```

```
      <td>{{user.userName}}</td>
      <td>{{user.email}}</td>
      <td>{{user.country}}</td>
      <td class="text-center">
        <div class="btn-group" role="group">
        <button class="btn btn-danger" (click)="deleteUser(user.uid)"><i class="bi bi-
trash"></i></button>
        <button class="btn btn-warning" (click)="navigateToDetails(user.userName)"><i class="bi
bi-eye"></i></button>
        <button class="btn btn-success" (click)="navigateToUpdate(user.userName)"> <i class="bi
bi-pencil-square"></i></button>
      </div>
      </td>
   </tr>
</tbody>
</table>

</div>
```

Userlist.component.ts

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { status } from 'src/model/status';
import { user } from 'src/model/user';
import { UserService } from 'src/service/user.service';

@Component({
 selector: 'app-userlist',
 templateUrl: './userlist.component.html',
 styleUrls: ['./userlist.component.css']
})
export class UserlistComponent {
 users: user[];
 constructor(private usrService: UserService, private router:Router) {

 }

 ngOnInit() {
  this.usrService.getAllUsers().subscribe(data => { this.users = data });
 }
 deleteUser(id: number) {

  this.usrService.deleteUser(id).subscribe(data => {
   if (data.status == '1') {
    this.users = this.users.filter(c => c.uid !== id);
   }
   console.log(data);
  });


 }
 navigateToDetails(param:string){
  this.router.navigate(['/adminpanel/details',param]);
 }
 navigateToUpdate(param:string){
  this.router.navigate(['adminpanel/update',param])
```

```
  }
}
```

Userlogin.component.html

```
<section class="vh-100" style="background-color: #ffc800;">
   <div class="container py-5 h-100">
    <div class="row d-flex justify-content-center align-items-center h-100">
     <div class="col col-xl-10">
      <div class="card" style="border-radius: 1rem;">
       <div class="row g-0">
        <div class="col-md-6 col-lg-5 d-none d-md-block">
         <img src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-login-
form/img1.webp"
           alt="login form" class="img-fluid" style="border-radius: 1rem 0 0 1rem;" />
        </div>
        <div class="col-md-6 col-lg-7 d-flex align-items-center">
         <div class="card-body p-4 p-lg-5 text-black">

          <form>

           <div class="d-flex align-items-center mb-3 pb-1">
             <img src="favicon.ico" width="30" height="30" class="d-inline-block align-top"
alt="">
             <span class="navbar-brand mb-0 h1">MyMoviePlan</span>
           </div>

           <h5 class="fw-normal mb-3 pb-3" style="letter-spacing: 1px;">Sign into your
account</h5>

           <div class="form-outline mb-4">
            <input type="email" id="form2Example17" class="form-control form-control-lg"
name="username" [(ngModel)]="usr.email"/>
            <label class="form-label" for="form2Example17">Email address or Username</label>
           </div>

           <div class="form-outline mb-4">
            <input type="password" id="form2Example27" class="form-control form-control-lg"
name="password" [(ngModel)]="usr.password"/>
            <label class="form-label" for="form2Example27">Password</label>
           </div>

           <div class="pt-1 mb-4">
            <button class="btn btn-dark btn-lg btn-block" type="button"
(click)="userLogin()">Login</button>
           </div>



          </form>

         </div>
        </div>
       </div>
      </div>
     </div>
```

```
    </div>
   </div>
  </section>
```

Userlogin.component.ts

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { user } from 'src/model/user';
import { LoginService } from 'src/service/login.service';
import { UserService } from 'src/service/user.service';

@Component({
 selector: 'app-userlogin',
 templateUrl: './userlogin.component.html',
 styleUrls: ['./userlogin.component.css']
})
export class UserloginComponent {
 usr:user=new user();
 constructor(private loginservice:LoginService, private router:Router){
 }
 userLogin(){
  this.loginservice.authUser(this.usr).then(res=>{
   if(res.authentication=="1"){
    this.router.navigateByUrl("userdashboard");
   }else if(res.authentication=="0"){
    alert('Username or Password incorrect!!');
    this.usr.email='';
    this.usr.userName='';
    this.usr.password='';
   }
  });
 }
}
```

Viewuserbookings.component.html

```
<table class="table table-bordered table-striped">
  <thead class="table-dark">
    <tr>
      <th class="text-center">Booking ID</th>
      <th class="text-center">Booking Details</th>
      <th class="text-center">Booking Amount</th>
      <th class="text-center">Booked on</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let bkng of userbookings">
      <td class="text-center">{{bkng.bookingId}}</td>
      <td class="text-center">{{bkng.bookingDetails}}</td>
      <td class="text-center">{{bkng.bookingAmount | currency:'INR'}}</td>
      <td class="text-center">{{bkng.bookingStamp | date:'dd-MM-yyy hh:mm:ss'}}</td>
    </tr>
  </tbody>
</table>
```

Viewuserbookings.component.ts

```
import { Component } from '@angular/core';
import { Bookings } from 'src/model/Bookings';
import { user } from 'src/model/user';
```

```
import { BookingService } from 'src/service/booking.service';
import { LoginService } from 'src/service/login.service';

@Component({
  selector: 'app-viewuserbookings',
  templateUrl: './viewuserbookings.component.html',
  styleUrls: ['./viewuserbookings.component.css']
})
export class ViewuserbookingsComponent {
  currntusr:user;
  userbookings:Bookings[];
  constructor(
    private loginservice:LoginService,
    private bookingservice:BookingService
  ){}
  ngOnInit(){
    this.currntusr = this.loginservice.getCurrentUsr();

this.bookingservice.getBookingsForUser(this.currntusr).subscribe(res=>{this.userbookings=res;});
  }
}
```

App-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { Bookings } from 'src/model/Bookings';
import { AddmovieComponent } from './addmovie/addmovie.component';
import { AdminloginComponent } from './adminlogin/adminlogin.component';
import { AdminpanelComponent } from './adminpanel/adminpanel.component';
import { BookingdetailsComponent } from './bookingdetails/bookingdetails.component';
import { BookingsmasterComponent } from './bookingsmaster/bookingsmaster.component';
import { ChangeuserpasswordComponent } from
'./changeuserpassword/changeuserpassword.component';
import { CheckoutComponent } from './checkout/checkout.component';
import { DetailsComponent } from './details/details.component';
import { ForbiddenComponent } from './forbidden/forbidden.component';
import { LandingComponent } from './landing/landing.component';
import { MoviemasterComponent } from './moviemaster/moviemaster.component';
import { SignupComponent } from './signup/signup.component';
import { UpdateuserComponent } from './updateuser/updateuser.component';
import { UserdashboardComponent } from './userdashboard/userdashboard.component';
import { UserlistComponent } from './userlist/userlist.component';
import { UserloginComponent } from './userlogin/userlogin.component';
import { ViewuserbookingsComponent } from
'./viewuserbookings/viewuserbookings.component';

const routes: Routes = [
  {path:'',component:LandingComponent},
  {path:'signup',component:SignupComponent},
  {path:'checkout',component:CheckoutComponent},
  {path:'bookingdetails',component:BookingdetailsComponent},
  {path:'adminlogin',component:AdminloginComponent},
  {path:'userlogin',component:UserloginComponent},
  {path:'adminpanel',component:AdminpanelComponent, children:[
    {path:'userlist', component:UserlistComponent},
    {path:'details/:usrname',component:DetailsComponent},
```

```
   {path:'update/:usrname', component:UpdateuserComponent},
   {path:'addmovie', component:AddmovieComponent},
   {path:'moviemaster', component:MoviemasterComponent},
   {path:'bookingsmaster',component:BookingsmasterComponent}
  ]},
  {path:'forbidden',component:ForbiddenComponent},
  {path:'userdashboard',component:UserdashboardComponent, children:[
   {path:'changepassword',component:ChangeuserpasswordComponent},
   {path:'bookings',component:ViewuserbookingsComponent},
   {path:'update/:usrname',component:UpdateuserComponent}
  ]}
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

App.component.html

```
<router-outlet></router-outlet>
```

Admin.ts

```
export class admin{
   adminId:number;
   userName:string;
   password:string;
   email:string;
   role:string;
}
```

Adminauth.ts

```
export class adminauth{
   admin:string;
   authentication:string;
}
```

Bookings.ts

```
export class Bookings{
   bookingId:number;
   userId:number;
   bookingDetails:string[];
   bookingAmount:number;
   bookingStamp:Date;
}
```

Cart.ts

```
import { cartitem } from "./cartitem";

export class Cart{
   items:cartitem[]=[];

   get totalprice():number{
   let totalprice:number = 0;
      this.items.forEach(item=>{totalprice+=item.price});
      return totalprice;

   }
```

```
}
```

Cartitem.ts

```
import { movie } from "./movie";

export class cartitem{
   quantity:number = 1;
   movie:movie;
   constructor(movie:movie){
      this.movie=movie;
   }
   get price():number{
      return this.movie.moviePrice*this.quantity;
   }
}
```

Movie.ts

```
export class movie{
   movieId:number;
   title:string;
   genre:string;
   rating:number;
   releasedate:Date;
   status:boolean;
   actors:string[];
   runtime:number;
   language:string;
   imageUrl:string;
   director:string;
   moviePrice:number;

}
```

Status.ts

```
export class status{
   status:string;
   }
```

User.ts

```
export class user{
   uid:number;
   firstName:string;
   middleName:string;
   lastName:string;
   userName:string;
   email:string;
   password:string;
   country:string;


}
```

Booking.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable, ObservedValueOf } from 'rxjs';
import { Bookings } from 'src/model/Bookings';
import { status } from 'src/model/status';
```

```
import { user } from 'src/model/user';

@Injectable({
  providedIn: 'root'
})
export class BookingService {
  baseUrl:string='http://localhost:8080/api/bookings';
  constructor(private http:HttpClient) { }

  getAllBookings():Observable<Bookings[]> {
    return this.http.get<Bookings[]>(this.baseUrl+"/all");
  }

  getBookingsForUser(Usr:user):Observable<Bookings[]>{
    return this.http.post<Bookings[]>(this.baseUrl+"/user",Usr);
  }
  saveBooking(bk:Bookings):Observable<status>{
    return this.http.post<status>(this.baseUrl+"/saveBooking",bk);
  }
}
```

Cart.service.ts

```
import { Injectable } from '@angular/core';
import { Cart } from 'src/model/cart';
import { cartitem } from 'src/model/cartitem';
import { movie } from 'src/model/movie';

@Injectable({
  providedIn: 'root'
})
export class CartService {
  private cart:Cart = new Cart();
  constructor() { }

  addToCart(movie: movie):void{
    let cartItem = this.cart.items.find(item => item.movie.movieId === movie.movieId);
    if(cartItem){
      this.changeQuantity(movie.movieId, cartItem.quantity + 1);
      return;
    }
    this.cart.items.push(new cartitem(movie));
  }

  removeFromCart(movieId:number): void{
    this.cart.items =
    this.cart.items.filter(item => item.movie.movieId != movieId);
  }

  changeQuantity(movieId:number, quantity:number){
    let cartItem = this.cart.items.find(item => item.movie.movieId === movieId);
    if(!cartItem) return;
    cartItem.quantity = quantity;
  }

  getCart():Cart{
    return this.cart;
```

```
  }
  resetCart(){
    this.cart=new Cart();
  }

}
```

Login.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { firstValueFrom, Observable } from 'rxjs';
import { admin } from 'src/model/admin';
import { adminauth } from 'src/model/adminauth';
import { user } from 'src/model/user';
import { UserService } from './user.service';

@Injectable({
  providedIn: 'root'
})
export class LoginService {

  currentuser:string="0";
  curntuser:user={
    "uid":-1,
     "firstName":"",
     "middleName":"",
     "lastName":"",
     "userName":"",
     "email":"",
     "password":"",
     "country":""
  };
  baseUrl:string='http://localhost:8080/api/admin';
  userauthstatus:boolean = false;
  adminauthstatus:boolean=false;
  constructor(private http:HttpClient, private usrservice:UserService) { }

  async adminAuthentication(admin:admin):Promise<adminauth>{
    var adminauth:adminauth;
    adminauth={
      "admin":"",
      "authentication":""
    }
  const res = await
firstValueFrom(this.http.post<adminauth>(this.baseUrl+"/authenticate",admin));
  adminauth=res;
  return adminauth;
  }

 async adminAuth(ad:admin){
    var adminauth1:adminauth;
   await this.adminAuthentication(ad).then(res=>{
      adminauth1=res;
    if(adminauth1.authentication=="1"){
      this.adminauthstatus=true;
    }else if(adminauth1.authentication=="0"){
```

```
    this.adminauthstatus=false;
   }
  });
   this.currentuser="admin"
   return this.adminauthstatus;
  }

  async authUser(user:user):Promise<{"user":string,"authentication":string}>{
   const re=/^(([^<>()[\]\\.,;:\s@"]+(\.[^<>()[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-
9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;
   var usr:user = {
    "uid":-1,
    "firstName":"",
    "middleName":"",
    "lastName":"",
    "userName":"",
    "email":"",
    "password":"",
    "country":""
   }
   var userauthstatus={
    "user":"",
    "authentication":"not found"
   }
   if(user.email.match(re)){
    await firstValueFrom(this.usrservice.getUserByName(user.email)).then(res=>usr=res);
   }else{
    await firstValueFrom(this.usrservice.getUserByName(user.email)).then(res=>usr=res);
   }
   if(usr){
    userauthstatus.user="1";
    if(usr.password==user.password){
     userauthstatus.authentication="1"
     this.currentuser=usr.firstName+' '+usr.lastName;
     this.userauthstatus=true;
     this.curntuser=usr;

    }else{
     userauthstatus.authentication="0"
    }
   }else{
    userauthstatus.user="0";
   }
   return userauthstatus;
  }
  getCurrentUser():string{
   return this.currentuser;
  }
  getCurrentUsr():user{
   return this.curntuser;
  }
  logout(){
   this.userauthstatus=false;
   this.adminauthstatus=false;
   this.currentuser="0";
```

```
    this.curntuser ={
      "uid":-1,
      "firstName":"",
      "middleName":"",
      "lastName":"",
      "userName":"",
      "email":"",
      "password":"",
      "country":""
    };
    alert('Logout Successful')
  }

}
```

Movie.service.ts

```
import { HttpClient, HttpClientModule } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { map, Observable } from 'rxjs';
import { movie } from 'src/model/movie';
import { status } from 'src/model/status';

@Injectable({
  providedIn: 'root'
})
export class MovieService {
  baseURL:string='http://localhost:8080/api/movie';
  constructor(private http:HttpClient) { }

  getAllMovies():Observable<movie[]>{
    return this.http.get<movie[]>(this.baseURL+"/all");
  }
  addMovie(movie:movie):Observable<status>{
    return this.http.post<status>(this.baseURL+"/addMovie",movie);
  }
  getMovieById(id:number):Observable<movie>{
    return this.http.get<movie>(this.baseURL+"/find/"+id);
  }
  changestatus(mv:movie):Observable<status>{
    return this.http.patch<status>(this.baseURL+"/statusChange",mv);
  }
}
```

User.service.ts

```
import { HttpClient, HttpClientModule, HttpParams } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { status } from 'src/model/status';
import { user } from 'src/model/user';

@Injectable({
  providedIn: 'root'
})
export class UserService {

  baseURL:string = 'http://localhost:8080/api/user';
```

```
  constructor(private http:HttpClient) { }

  getAllUsers():Observable<user[]>{
    return this.http.get<user[]>(this.baseURL+'/all')
  }
  getUserByName(name:string):Observable<user>{
    return this.http.get<user>(this.baseURL+'/finduser?name='+name);
  }
  addUser(usr:user):Observable<status>{
    console.log(usr);
    let usrarr:user[] = new Array;
    usrarr.push(usr);
    return this.http.post<status>(this.baseURL+'/adduser',usrarr);
  }
  deleteUser(id:number):Observable<status>{
    return this.http.delete<status>(this.baseURL+'/remove/'+id);
  }
  updateUser(usr:user):Observable<status>{
    return this.http.patch<status>(this.baseURL+'/updateuser/'+usr.uid,usr);
  }


}
```