CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

1. CONTROLLER:

```
package com.movieplan.controller;

public class checkoutController {

}
```

1.1 BOOKING SEATS

```
package com.movieplan.controller;

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

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
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.movieplan.model.BookSeatsRequest;
import com.movieplan.model.BookedSeats;
import com.movieplan.model.Movie;
import com.movieplan.model.Theater;
import com.movieplan.repository.bookedSeatsRepository;
import com.movieplan.repository.theaterRepository;

@RestController
@CrossOrigin
@RequestMapping("/bookedseats")
public class bookedSeatsController {

    @Autowired
    private bookedSeatsRepository bsRepo;
```

```
    @Autowired
    private theaterRepository tRepo;

    @PostMapping("/getbydatetimetheaterid/{theaterId}")
    public ResponseEntity<Map<String, Object>> getBookSeatsByDateTime(@PathVariable
long theaterId,
                    @RequestBody BookSeatsRequest request) {

            Theater theater = tRepo.getOne(theaterId);
            Date date = request.getDate();
            String time = request.getTime();
            List<BookedSeats> bookedSeats = bsRepo.getBookSeatsByDateTime(theater,
date, time);
            Map<String,Object> map = new HashMap<>();
            map.put("response",bookedSeats);
            return ResponseEntity.status(HttpStatus.OK)
                .body(map);
    }

    @PostMapping("/postbydatetimetheaterid/{theaterId}")
    public ResponseEntity<Map<String, Object>> postBookSeatsByDateTime(@PathVariable
long theaterId,
                    @RequestBody BookSeatsRequest request) {

            Theater theater = tRepo.getOne(theaterId);
            Date date = request.getDate();
            String time = request.getTime();

            for(String seat:request.getSeats()) {
                BookedSeats bookedSeats = new BookedSeats();
                bookedSeats.setId(0);
                bookedSeats.setDate(date);
                bookedSeats.setSeat(seat);
                bookedSeats.setTime(time);
                bookedSeats.setTheater(theater);
                bsRepo.save(bookedSeats);
            }
            Map<String,Object> map = new HashMap<>();
            map.put("text","booked seats successfully");
            return ResponseEntity.status(HttpStatus.OK)
                .body(map);
    }

}
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

1 .2 USER CONTROLLER:

```
package com.movieplan.controller;

import java.util.Optional;

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.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import com.movieplan.model.Movie;
import com.movieplan.model.User;
import com.movieplan.model.authUser;
import com.movieplan.repository.movieRepository;
import com.movieplan.repository.userRepository;

@RestController
@CrossOrigin
@RequestMapping("/user")
public class userController {
	@Autowired
	private userRepository uRepo;
```

```
	@PostMapping("/authenticate")
	public User authenticate(@RequestBody authUser authuser) {

		String email = authuser.getEmail();
		String password = authuser.getPassword();

		User theUser = uRepo.findByEmail(email);
		String Userpassword = theUser.getPassword();

		if(Userpassword.equals(password)) {
			System.out.println("login succesful");
		} else {
			throw new RuntimeException("password or email is incorrect");
		}
		return theUser;
	}

	@PostMapping("/signup")
	public User add(@RequestBody User theUser) {

		theUser.setId(0);

		uRepo.save(theUser);

		return theUser;
	}
}
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

1.3 THEATHRE CONTROLLER:

```
package com.movieplan.controller;

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

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

import com.movieplan.model.Movie;
import com.movieplan.model.Theater;
import com.movieplan.repository.movieRepository;
import com.movieplan.repository.theaterRepository;

@RestController
@CrossOrigin
@RequestMapping("/theater")
public class theaterController {

        @Autowired
        private theaterRepository tRepo;

        @Autowired
        private movieRepository mRepo;

        @GetMapping("/{theaterId}")
        public Optional<Theater> getMovie(@PathVariable long theaterId) {

                Optional<Theater> theTheater = tRepo.findById(theaterId);
```

```
		return theTheater;
	}

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

		List<Theater> theTheaters = tRepo.findAll();

		return theTheaters;
	}

	@GetMapping("/bymovie/{movieId}")
	public ResponseEntity<Map<String, Object>>
getAllTheatersByMovieId(@PathVariable long movieId) {

		Movie movie = mRepo.getOne(movieId);
		System.out.println(movie);
		List<Theater> theTheaters = tRepo.getTheatersByMovie(movie);
		System.out.println(theTheaters);
		for(Theater theater:theTheaters){
			System.out.println(theater);
		}
		Map<String,Object> map = new HashMap<>();
		map.put("text",theTheaters);
		return ResponseEntity.status(HttpStatus.OK)
		    .body(map);
	}

	@PostMapping("/{movieId}")
	public ResponseEntity<Map<String, Object>> add(@RequestBody Theater
theTheater, @PathVariable long movieId) {

		theTheater.setId(0);
		Movie movie = mRepo.getOne(movieId);
		theTheater.setMovie(movie);
		tRepo.save(theTheater);
		Map<String,Object> map = new HashMap<>();
		map.put("text","Successfully added");
		return ResponseEntity.status(HttpStatus.OK)
		    .body(map);
	}

	@PutMapping("/{theaterId}/{movieId}")
```

```
	public ResponseEntity<Map<String, Object>> editMovie(@PathVariable long
theaterId,
				@PathVariable long movieId,
				@RequestBody Theater theTheater) {

			Theater theTheater1 = new Theater();
			Movie movie = mRepo.getOne(movieId);
			theTheater1.setTheatreName(theTheater.getTheatreName());
			theTheater1.setTheatreAddress(theTheater.getTheatreAddress());
			theTheater1.setMovie(movie);
			theTheater1.setId(theaterId);

			Theater resTheter = tRepo.save(theTheater1);

			Map<String,Object> map = new HashMap<>();
			map.put("text","Successfully edited");
			return ResponseEntity.status(HttpStatus.OK)
				.body(map);
		}

		@DeleteMapping("/{theaterId}")
		public ResponseEntity<Map<String, Object>> delete(@PathVariable long
theaterId) {

			tRepo.deleteById(theaterId);

			Map<String,Object> map = new HashMap<>();
			map.put("text","Deleted Theater Id"+ theaterId);

			return ResponseEntity.status(HttpStatus.OK)
				.body(map);

		}

	}
```

1.4 SHOW CONTROLLER:

```
	package com.movieplan.controller;

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

	import org.springframework.beans.factory.annotation.Autowired;
	import org.springframework.http.HttpStatus;
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

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

import com.movieplan.model.MovieShow;
import com.movieplan.repository.showRepository;

@RestController
@RequestMapping("/movieshow")
@CrossOrigin(origins = "http://localhost:4200")
public class showController {

	@Autowired
	private showRepository sRepo;

	@GetMapping("/{showId}")
	public MovieShow getMovie(@PathVariable long showId) {

		MovieShow show = sRepo.getOne(showId);
		return show;
	}

/*	@PutMapping("/{showId}")
	public Show editMovie(@PathVariable long movieId,
			@RequestBody Show theMovie) {

		Show movie = sRepo.getOne(movieId);

		if(theMovie.getLanguage() != null) {
			movie.setLanguage(theMovie.getLanguage());
		}
		if(theMovie.getDuration() != null) {
			movie.setDuration(theMovie.getDuration());
		}
		if(theMovie.getName() != null) {
			movie.setName(theMovie.getName());
```

```
		}
		if(theMovie.getReleaseDate() != null) {
			movie.setReleaseDate(theMovie.getReleaseDate());
		}
		if(theMovie.getCensorCertificate() != null) {
			movie.setCensorCertificate(theMovie.getCensorCertificate());
		}
		Show resMovie = movieService.updateMovie(movie);

		return resMovie;
	}

	@PostMapping("/{showId}")
	public Show addShow() {
		Show show = sRepo.getOne(null)
	}*/

	@DeleteMapping("/{showId}")
	public ResponseEntity<Map<String, Object>> delete(@PathVariable long
showId) {

		MovieShow show = sRepo.getOne(showId);

		sRepo.deleteById(showId);

		Map<String,Object> map = new HashMap<>();
		map.put("text","Deleted Movie Id"+ showId);
		map.put("obj", show);

		return ResponseEntity.status(HttpStatus.OK)
		     .body(map);

	}

	@GetMapping("/")
	public List<MovieShow> getAllShows() {

		List<MovieShow> theShows = sRepo.findAll();

		return theShows;
	}

}
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

1.5 MOVIE CONTROLLER:

```
package com.movieplan.controller;

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

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

import com.movieplan.model.Movie;
import com.movieplan.repository.movieRepository;

@RestController
@CrossOrigin
@RequestMapping("/movies")
public class movieController {

	@Autowired
	private movieRepository mRepo;

	@GetMapping("/{movieId}")
	public Optional<Movie> getMovie(@PathVariable long movieId) {

		Optional<Movie> theMovie = mRepo.findById(movieId);

		if (theMovie == null) {
			throw new RuntimeException("Employee id not found - " +
movieId);
		}
```

```
		return theMovie;
	}

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

		List<Movie> theMovies = mRepo.findAll();

		return theMovies;
	}

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

		List<Movie> theMovies = mRepo.getMoviesByTheater();

		return theMovies;
	}

	@PostMapping("/")
	public Movie add(@RequestBody Movie theMovie) {

		theMovie.setId(0);

		Movie theResMovie = mRepo.save(theMovie);



		return theResMovie;
	}

	@PutMapping("/{movieId}")
	public Movie editMovie(@PathVariable long movieId,
			@RequestBody Movie theMovie) {

		theMovie.setId(movieId);

		Movie resMovie = mRepo.save(theMovie);

		return resMovie;
	}

	@DeleteMapping("/{movieId}")
```

```
        public ResponseEntity<Map<String, Object>> delete(@PathVariable long
movieId) {

                mRepo.deleteById(movieId);

                Map<String,Object> map = new HashMap<>();
                map.put("text","Deleted Movie Id"+ movieId);

                return ResponseEntity.status(HttpStatus.OK)
                    .body(map);

        }

}
```

2. MODEL:

2.1 SEATS:

```
    package com.movieplan.model;


import java.util.Date;

import java.util.List;


public class BookSeatsRequest {


    private String time;


    private Date date;


    private List<String> seats;


    public BookSeatsRequest() {

        super();

    }
```

```
public String getTime() {
        return time;
}

public void setTime(String time) {
        this.time = time;
}

public Date getDate() {
        return date;
}

public void setDate(Date date) {
        this.date = date;
}

public List<String> getSeats() {
        return seats;
}

public void setSeats(List<String> seats) {
        this.seats = seats;
}
}
```

2.2 MOVIE PLAN :

```
package com.movieplan.model;

import java.util.Date;
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

```
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;

@Entity
public class BookedSeats {

	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	@Column
	private long id;
	@Column
	private String seat;
	@Column
	private Date date;
	@Column
	private String time;

	@ManyToOne(cascade= {CascadeType.MERGE,
				CascadeType.DETACH, CascadeType.REFRESH}, fetch =
FetchType.EAGER)
	@JoinColumn(name="theater")
	private Theater theater;

	public BookedSeats() {
		super();
	}

	public long getId() {
		return id;
	}

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

	public String getSeat() {
		return seat;
```

```
	}

	public void setSeat(String seat) {
		this.seat = seat;
	}

	public Date getDate() {
		return date;
	}

	public void setDate(Date date) {
		this.date = date;
	}

	public String getTime() {
		return time;
	}

	public void setTime(String time) {
		this.time = time;
	}

	public Theater getTheater() {
		return theater;
	}

	public void setTheater(Theater theater) {
		this.theater = theater;
	}

}
```

2.3 USER:

```
package com.movieplan.model;

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

@Entity
@Table(name="user")
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

```
public class User {

	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	@Column(name="id")
	private Integer id;
	@Column(nullable = false, unique = true, length = 45)
	private String email;
	@Column(nullable = false, length = 64)
	private String password;
	@Column(nullable = false, length = 20)
	private String firstName;
	@Column(nullable = false, length = 20)
	private String lastName;
	@Column(nullable = false, length = 20)
	private String role;

	public User() {
		super();
	}

	public Integer getId() {
		return id;
	}

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

	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 getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public String getRole() {
		return role;
	}

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



}
```

2.4 THEATHE:

```
package com.movieplan.model;

import java.util.List;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

```
@Entity
public class Theater {

	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	@Column(name="id")
	private long id;

	@Column
	private String theatreName;

	@Column
	private String theatreAddress;

	@ManyToOne(cascade= {CascadeType.MERGE,
				CascadeType.DETACH, CascadeType.REFRESH}, fetch =
FetchType.EAGER)
	@JoinColumn(name="movie")
	private Movie movie;

	@OneToMany(mappedBy = "theater")
   private List<BookedSeats> seats;

	@Column
	private Integer theatreSeatCapacity;
	public Theater() {
		super();
	}

	public long getId() {
		return id;
	}

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

	public Movie getMovie() {
		return movie;
	}

	public void setMovie(Movie movie) {
		this.movie = movie;
```

```
	}

	public String getTheatreName() {
		return theatreName;
	}

	public void setTheatreName(String theatreName) {
		this.theatreName = theatreName;
	}

	public String getTheatreAddress() {
		return theatreAddress;
	}

	public void setTheatreAddress(String theatreAddress) {
		this.theatreAddress = theatreAddress;
	}

	public Integer getTheatreSeatCapacity() {
		return theatreSeatCapacity;
	}

	public void setTheatreSeatCapacity(Integer theatreSeatCapacity) {
		this.theatreSeatCapacity = theatreSeatCapacity;
	}


}
```

2.5 MOVIE PLAN:

```
		package com.movieplan.model;


import java.util.List;


import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

```
import javax.persistence.Id;

import javax.persistence.ManyToOne;

import javax.persistence.OneToMany;


import org.hibernate.annotations.Fetch;

import org.hibernate.annotations.FetchMode;


@Entity

public class Movie {


	@Id

	@GeneratedValue(strategy=GenerationType.IDENTITY)

	@Column

	private long id;

	@Column

	private String name;

	@Column

	private String duration;

	@Column

	private String language;

	@Column

	private String genre;

	@Column

	private String banner;

	@OneToMany(mappedBy = "movie")

  private List<Theater> theater;


	public String getBanner() {

		return banner;
```

```
}

public void setBanner(String banner) {
        this.banner = banner;
}

public Movie() {
        super();
}

public long getId() {
        return id;
}

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

public String getName() {
        return name;
}

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

public String getDuration() {
        return duration;
}
```

```
	public void setDuration(String duration) {
		this.duration = duration;
	}

	public String getLanguage() {
		return language;
	}

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

	public String getGenre() {
		return genre;
	}

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

}
```

3. REPOSITORY:

```
package com.movieplan.repository;

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

import com.movieplan.model.MovieShow;

public interface showRepository extends JpaRepository<MovieShow, Long>{
```

```
}
package com.movieplan.repository;

import java.util.List;

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

import com.movieplan.model.Movie;

public interface movieRepository extends JpaRepository<Movie, Long>{

	@Query("SELECT distinct m FROM Movie m join Theater t on t.movie = m")
	List<Movie> getMoviesByTheater();

}

package com.movieplan.repository;

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

import com.movieplan.model.BookingHistory;

public interface bookingHistoryRepository extends JpaRepository<BookingHistory, Long>{

}
package com.movieplan.repository;

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

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

import com.movieplan.model.BookedSeats;
import com.movieplan.model.Theater;

public interface bookedSeatsRepository extends JpaRepository<BookedSeats, Long>{

	@Query("SELECT distinct bs FROM BookedSeats bs where bs.theater=?1 and bs.date=?2
and bs.time=?3")
	List<BookedSeats> getBookSeatsByDateTime(Theater theater, Date date, String time);

}
```

CAPSTONE PROJECT
SOURCE CODE FOR MY MOVIE PLAN

4. MOVIE PLANNER

```
package com.movieplan;

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

@SpringBootApplication
public class MyMoviePlannerApplication {

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

}
```