In this lab, you will be building a Movie buying application using MERN. The application has two panels; the admin panel and the customer panel.

Admin Panel:
The admin panel is responsible for managing movies and managing orders.

Movie management has the following set of features:
- Add a new movie to the database.
- Delete a movie from the database.
- Browse movies (See all the movies available in the database).
- Edit details of a movie.

The movie has the following attributes (id, title, poster, year, rating, genre).

Order management has the following set of features:
- Confirm Orders.
- View Orders.

You can have the following attributes in the order schema (order id, order date, and movie id (as a foreign key from the movie's schema).

Customer Panel:
The customer panel is responsible for the following features:
- Browse movies (See all the movies available in the database).
- Place an order.
- View details of a movie (title, poster, year, rating, genre).

The following set of sub-tasks needs to be done to perform the above tasks.
- Clone backend and frontend template from https://github.com/tareqhassan2014/movie-share
- Install node_modules by going to the frontend and backend folders and running npm install frontend and npm install backend respectively.
- Open the "index.js" file and connect your code to MongoDB.
- Create a folder called models.
- Create a file "movie_models.js" inside that folder. In this file, you will write your database code (Creation of movie schema).
- Create another file "Ordrers_models.js" inside that folder. In this file, you will write your database code (Creation of orders schema).
- Go to the routes folder and create two files "Customer_rotes.js" and "Admin_routes.js".
- In the "Customer_rotes.js" file, you will write routes for browsing movies, placing an order, and viewing details of a film.
- In the "Admin_rotes.js" file, you will write routes for adding a movie to the database, deleting a movie from the database, browsing movies, editing movie details, confirming orders, and viewing orders.
- Now, open the frontend folder and create a folder called "components".
- In the components folder, create components for adding a movie, deleting a movie, editing movie details, browsing movies, confirming orders, viewing orders, and placing orders. To perform this task, you can use the cards component from the react-bootstrap.
- Connect your frontend code with the backend code using the fetch library.