Explanation

The folder is organised in a MVC (Model, View, Controller) pattern.

**Backend**

Inside server, contains

* Auth, which contains verifyToken that checks whether the JWT token is valid, and whether the *JWT token is for a staff member or a customer (additional)*
* Config, which contains a secretkey
* Controller, which contains app.js where all the different endpoints are stored.
* generateImage.js a file used to generate 100 different images based on random strings in order to give users their profile pictures
* Model, which is split into several different files.
  + Actors
    - insertActor: allows an admin to insert actor by first name and last name
  + Customer
    - insertCustomer: allows an admin to insert customer as well as the customer’s address information
    - loginCustomer: *allows a customer to login in the login page with email and password (additional)* => look at login page for more info
  + Film
    - getAllCategory: get all the categories from the films by ascending order in order to dynamically generate a dropdown to filter by categories later on
    - searchFilm: allows user to search for summarised film by category, substring, or maximum defined rental rate (required) *or by a any combination of the above (additional)*
    - getFilm: allows user to get the more detailed information on the film by filmid.
  + Review
    - getReviewsByFilmId: *allows logged in customers to view all the reviews and ratings (out of 5) made by other customers for a certain film (additional)*
    - submitReview: *allows logged in customers to submit a review and a rating (out of 5)*
  + Staff
    - loginStaff: similar to loginCustomer but instead allows a staff to login instead with email and password.
  + Store
    - getStoreAddress: get all the store address in order to dynamically generate the dropdown of the stores needed for insertCustomer

**Frontend**

In the frontend there are a total of 9 html files and one index.js

* Index.js: This is the server that serves all the web pages
* Index.html: This is where the user starts. The user can either login (goes to login.html) or search for a specific film (search.html).
* Login.html: This is where the user can either log in as a customer or as a staff. This can be done by switching via the toggle button present. After a successful login, the staff will be redirected to admin.html *while the customers will be redirected to customer.html*.*(additional)* If the user is already logged in either as a customer or as a staff and they click the log in button from any page, they will immediately head to their respective page, admin.html for staff and customer.html for customers (persistent login). The JWT token and the userData (whether they are a customer or staff, their name and their email etc.) are stored in localstorage. Their passwords have been made sure not to be stored there as it contains sensitive information.
* Customer.html: Logged in customers will see a *profile picture (additional) => stored in the backend as a number. Users can choose a random profile (from 1 to 100),* their name and their email address. They can either choose to search for a film (where they can then review a film as a logged in customer) or log out. This would clear their local storage and bring them back to index.html. *If the customer is not logged in and manually keys in the url, they will be redirected to index.html. If an admin keys in the url, they will be redirected to admin.html (additional)*
* Admin.html: Logged in staff will also see their *profile picture (additional) => stored in the backend as a number. Users can choose a random profile (from 1 to 100),* name and their email address. They can then choose to add a customer (addCustomer.html) or add an actor (addActor.html) on the navbar. They can also log out which will similarly clear their local storage. *If a user is not logged and keys in the url, they will be redirected to index.html. If they are logged in as a customer, they will be redirected to customer.html (additional)*
* addActor.html: logged in staff can add an actor by their first name and last name. If the form is not filled up (missing data on either field) an error message will appear below. Just like in admin, *if a user is not logged in they will be redirected to login.html (additional)*
* addCustomer.html: logged in staff can add a customer by their address, personal information and the store they are at. Just like in admin, *if a user is not logged in they will be redirected to login.html (additional)*
* search.html: By default, all the films are shown. The user (does not have to be logged in) can choose to search a film by category, subtitle of the film, or the maximum price *or a combination of any of the above (additional)*. => If the user does not want to search by price, uncheck the “Filter by Price” check box. Each film has summarised information which includes:
  + title
  + year released
  + rating (as in G, PG, NC-17 etc. not the rating out of 5)
  + rental rate (amount not more than 5.99) and duration (not more than 7 days)
  + A view details button which allows user to check out more on the film (users will head to film.html)

If the server is down, none of the cards will show and there will just be a message saying that the server is down.

* Film.html: Users can view a more detailed information on the film. The url contains the filmid of the film which is used to dynamically generate the content for each page. It includes information such as
  + Title
  + Year released
  + rating (as in G, PG, NC-17 etc. not the rating out of 5)
  + Category of the film
  + Description of the film
  + Cast members
  + Special Features
  + Original Language of the film

*There is also a green link at the bottom where users can “Leave a Review”. This can only be done if users are logged in. This will lead them to review.html. If they are not, they will be redirected back to login.html. (additional)*

* *Review.html: Users have to be logged in to view this. If they manually key in the url, they will be redirected back to login.html. Admins are not allowed to leave reviews (if they are logged in as admins). Each review contains the name of the customer who left the review, the review itself and the rating out of 5 stars. Customers can also choose to leave their own reviews by selecting a rating out of 5 (static dropdown) and leaving their review in the textarea. (additional)*