**Front End – Back End Integration.**

In order to pass and verify login credentials for a user; data needs to be transferred in safe manner.

So, From the React Application Side:

1. Observe Header.js file under common folder.

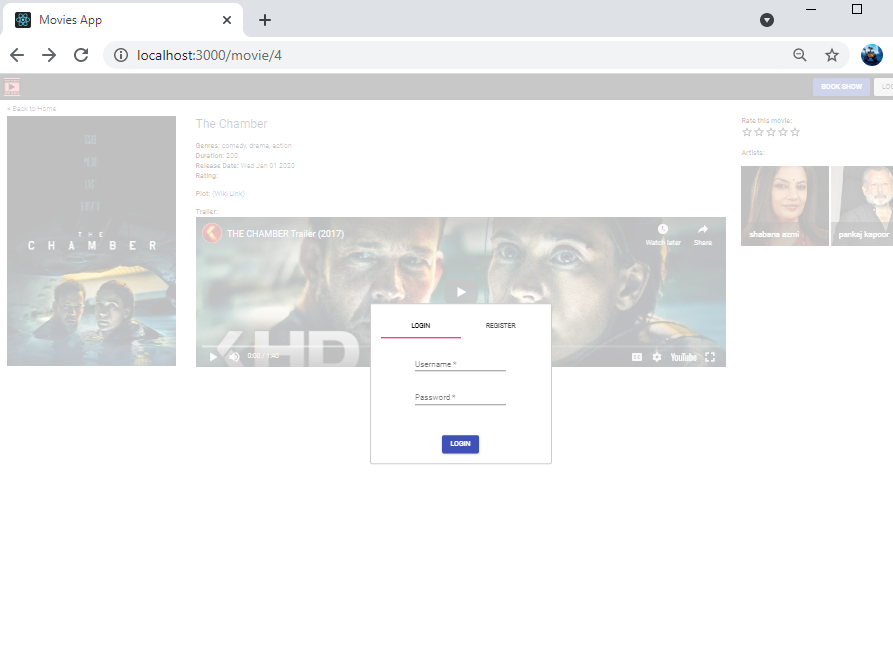
* UserName & Password Details are collected from the UI
* Further POST will put data in body section of HTTP Packet and pass it to the URL mentioned i.e ***auth/login***
* These credentials are further converted into non readable format using **window.btoa()** function
* Keep in mind to reverse the process we need to use atob() but since we will be doing this on NODE side & Node does not have access to Window Object or DOM in general; we will be using third party node module package to get it done.
* Request Header can be seen modified by passing “Authorization” header with it.

Now, from the Node Application Side:

1. Observe ***User.Routes.JS*** file

* Route /auth/login is handled via express and a call will be routed to users controller’s login method.
* Now, if you open ***“../controllers/user.controller.js”***
* Observe header details fetched from the request object and usage of “b2a” node module to get the data in reverse.
* In order to continue to further safe & secure pages; one need not pass username and password again; so what we are doing here is we are creating a UNIQUE access-token and UUID using again third party node – modules namely ***uuid-token-generator*** and ***uuidv4***
* This token will be given back to the browser / react after we verify username and password.
* This token will also be stored in Mongo DB under user’s collection.
* User need not pass credentials for next subsequent calls as passing this token itself is sufficient. This token is to tell server side that the client has already gone through the process of login.

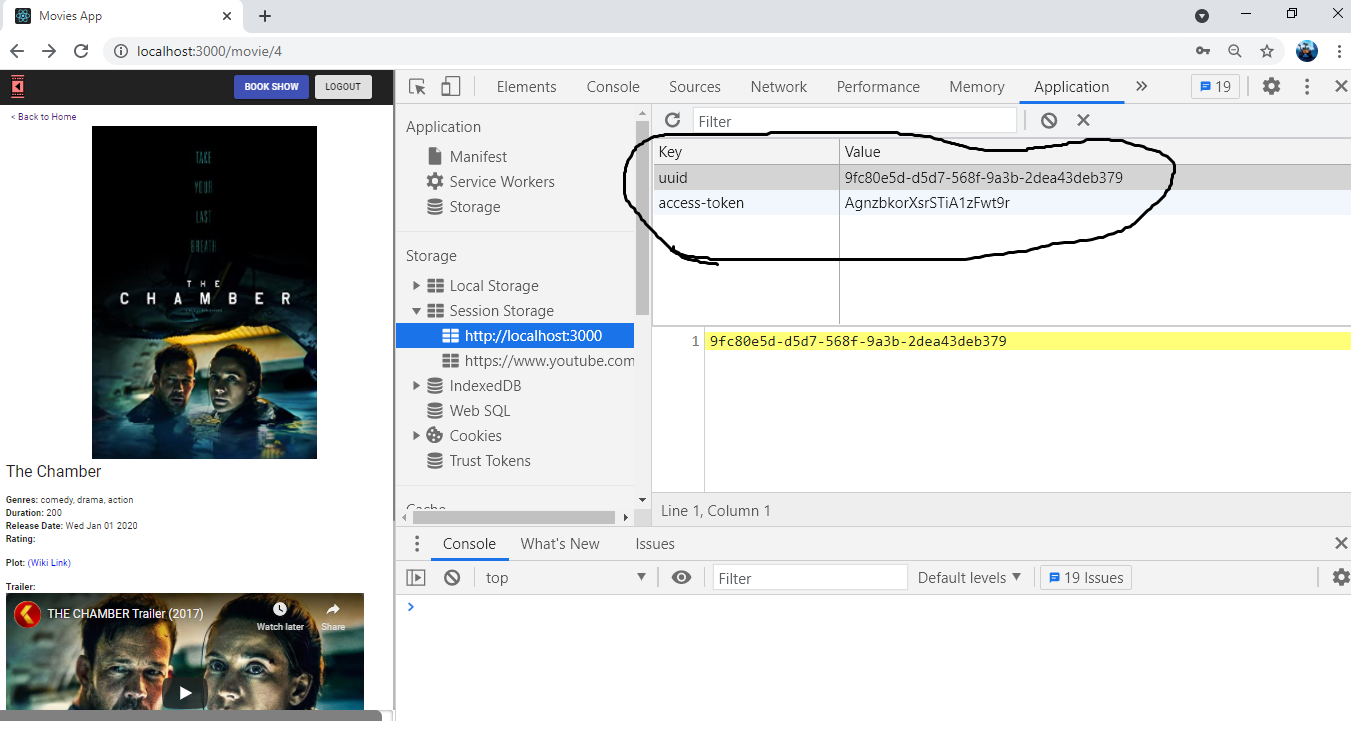
1. Also observe calls to ***signup and logout*** which are self explanatory
2. Once you click on “Login” button on the header; a pop up driven by “material UI” CSS library appears like:



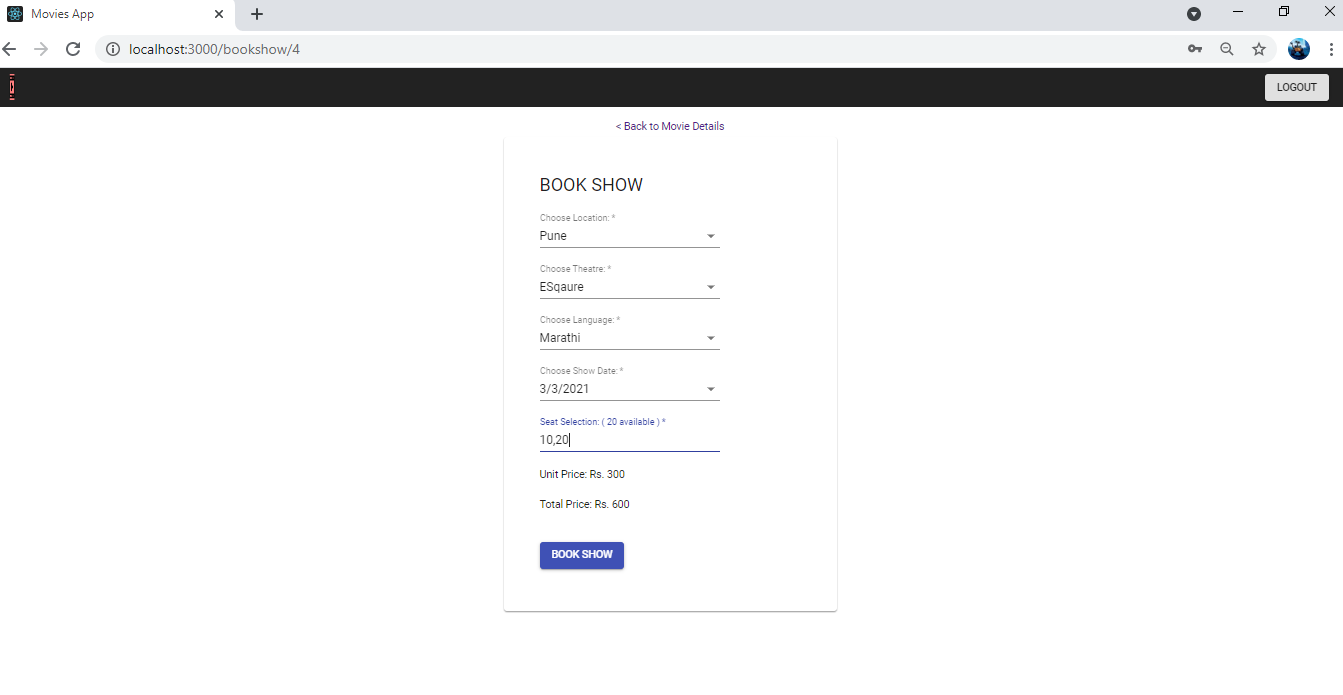
1. Once we enter correct credentials and click on Login button – on popup itself;

* Credentials will be passed to server
* Access-Token will be given back to browser
* We are storing the token in SESSION\_STORAGE for further usage

This storage can be seen under development window like:

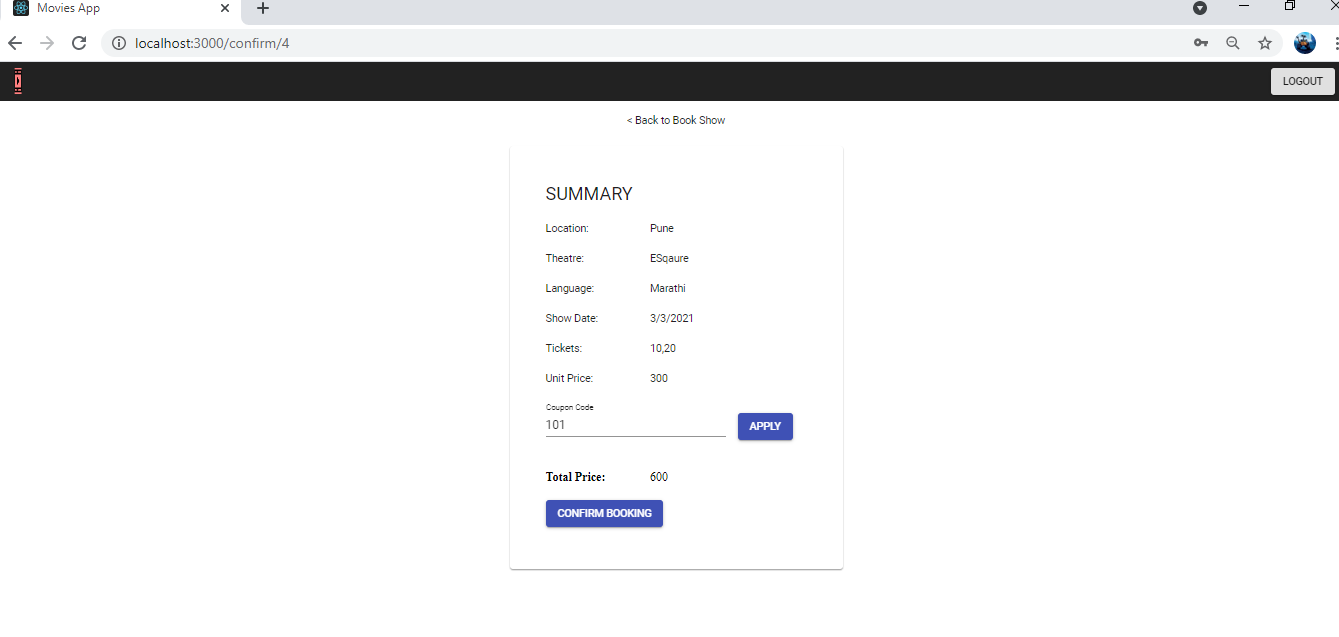


1. ***“ Logout”*** button will remove the entry from DB via AJAX call first and then will remove entry from browser.
2. Register TAB on the earlier “login” POPUP is self explanatory.
3. **However since there was no provision to accept preferred username; combination of first\_name and last\_name will act as a username.**
4. After signing in to the application; if you click on the movie image to see more details; you get another button beside Login / Logout as **“BookShow”**
5. This click will simple bring a popup populated with data – to be selected – based on movie’s ID.

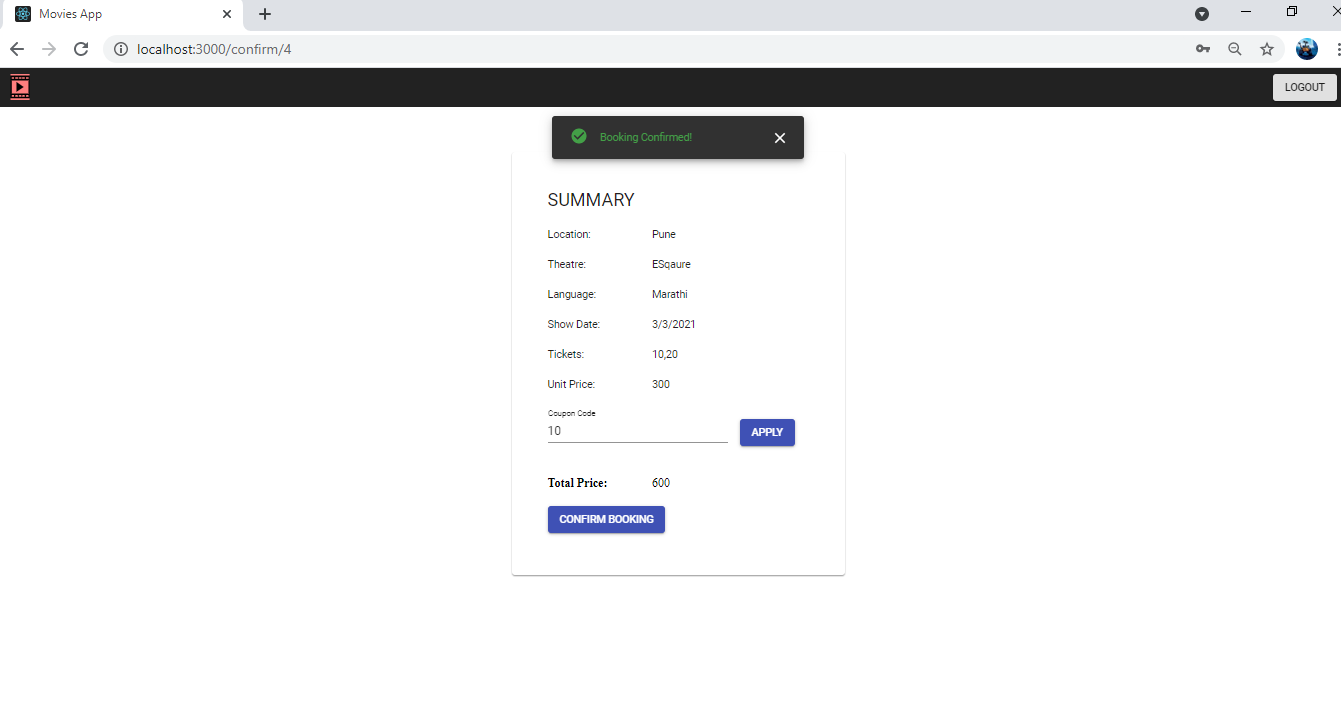


1. Clicking on “Book Show” button takes you to Summary Page where a coupon code can be applied.

For simplicity; Coupon code with details like how much amount of discount needs to be given is already added in sample database shared in checkpoint 4.



1. Clicking on “Confirm Booking” brings note at top saying “Booking Confirmed” like:



1. Finally, logout will bring user back to home page.