* **npx create-react-app Netflix-gpt**

**# adding / Installing tailwindcss in project**

* + tailwindcss website> get started > framework guides > create react app
  + **Install Tailwind CSS :-** Install **tailwindcss** via npm, and then run the init command to generate your **tailwind.config.js** file.
    - **npm install -D tailwindcss**
    - **npx tailwindcss init**
  + **Configure your template paths :-** Add the paths to all of your template files in your **tailwind.config.js** file.
  + **Add the Tailwind directives to your CSS :-** Add the **@tailwind** directives for each of Tailwind’s layers to your **./src/index.css** file.
  + <https://tailwindcss.com/docs/guides/create-react-app>

#Router Configuration

* **npm i -D react-router-dom**
* appRouter = createBrowserRouter > path and element
* import RouterProvider 🡪 <RouterProvider router={appRouter}/>

**#Form Validation**

* **Check Formik form validation also for big data validation**
* Regex for email & password validation
* How can I validate an email address using a regular expression?
* <https://saturncloud.io/blog/how-can-i-validate-an-email-address-using-a-regular-expression/> 🡨 Email
* <https://regexr.com/3bfsi> 🡨 Password
* There is test function over regex and we want to test our email and password

**export *const* checkValidData = (email, password) => {**

***// if this regex is pass then it will return TRUE inside it or FALSE inside "isEmailValid"***

***const* isEmailValid = /^([a-zA-Z0-9.\_%-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,})$/.test(email);**

***// if this regex is pass then it will return TRUE inside it or FALSE inside "isPasswordValid"***

***const* isPasswordValid =/^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[a-zA-Z]).{8,}$/.test(password);**

***// return's Error Message***

**if (!isEmailValid) return "Email ID is not valid";**

**if (!isPasswordValid) return "Password is not valid";**

***//if both are valid then return null i.e. No Error***

**return null;**

**};**

* Validate User Full name
* 🡺 <https://regexr.com/3f8cm>
* Authentication
* Google Fire base for Back-End
* Adding Firebase to your web
  + Register app
  + Add Firebase SDK
  + Install Firebase CLI
    - To host your site with Firebase Hosting, you need the Firebase CLI (a command line tool).
    - Run the following [npm](https://www.npmjs.com/) command to install the CLI or update to the latest CLI version.
    - **npm install -g firebase-tools**
  + Deploy to firebase Hosting
    - You can deploy now or [later](https://firebase.google.com/docs/hosting/quickstart?hl=en&authuser=0). To deploy now, open a terminal window, then navigate to or create a root directory for your web app.
    - **Sign in to Google**
      * **firebase login**
    - **initiate your project**

Run this command from your app's root directory:

* + - * **firebase init**
    - **when you’re ready deploy your web app**

Put your static files (e.g., HTML, CSS, JS) in your app's deploy directory (the default is "public").

Then, run this command from your app's root directory:

* + - * **firebase deploy**
* After deploying, view your app at [netflixgpt-d91a2.web.app](https://netflixgpt-d91a2.web.app/)
* Need help? Check out the [Hosting docs](https://firebase.google.com/docs/hosting/quickstart?hl=en&authuser=0)

**#Redux Store for Sign in & Sign up**

* If the user sign up or a sign in we will got this user object and we will have to keep the user object with us because we need this user object anywhere in our app.
* So what we will do is as soon as the user sign in or sign up We will just add all that data to our redux store i.e. once user sign in or sign up we will add user to our redux store.
* **npm i -D @reduxjs/toolkit**
* **npm i react-redux**
  + utils > appStore.js 🡪 configureStore – reducer collection of different reducers
  + utils > userSlice.js 🡪 createSlice – name, initialState – reducers{add or remove} function
  + **userSlice**
* **import { createSlice } from "@reduxjs/toolkit";**
* ***const* userSlice = createSlice({**
* **name: "user",**
* **initialState: null,**
* **reducers: {**
* **addUser: (state, action) => {**
* **return action.payload**
* **},**
* **removeUser: (state, action) => {**
* **return null**
* **}**
* **}**
* **})**
* **export *const* { addUser, removeUser } = userSlice.actions**
* **export default userSlice.reducer**
  + **appStore**
* import { configureStore } from "@reduxjs/toolkit";
* import userReducer from "./userSlice"
* *const* appStore = configureStore({
* reducer: {
* user: userReducer
* }
* })
* **export default appStore**
  + **now providing Store to our App**
* *function* App() {
* return (
* <Provider *store*={appStore}>
* <Body />
* </Provider>
* );
* }
* **Manage Users in Firebase** (not using redux dispatch) - **onAuthStageChange**
* <https://firebase.google.com/docs/auth/web/manage-users?hl=en>
* Firebase gives us an amazing API that is know as onAuthStageChange this API is called whenever the user sign in whenever the user sign up whenever the user sign out and whenever there is an authentication state change/ Authentication happens.
* So where we add this code?
* You can write wherever you want to but mostly use Root level to write this code so I am writing it on Body component