

# Netflix – GPT

---

The commands you've mentioned are used to create new projects in two different JavaScript frameworks: Vue.js and React. Let me explain the differences between them:

## 1. ``npm create vue@latest``:

- This command uses the ``npm`` package manager to create a new Vue.js project.
- It installs the Vue CLI (Command Line Interface) globally if you haven't already.
- ``vue@latest`` specifies that you want to use the latest version of Vue.js.
- After running this command, you'll be prompted to answer some questions about your project, such as the project name and features you want to include. Once you provide these answers, it will scaffold a new Vue.js project for you.

## 2. ``npx create-react-app my-app``:

- This command uses the ``npx`` tool to create a new React.js project.
- It uses the ``create-react-app`` package, which is a set of tools and configurations for setting up a new React project quickly.
- ``my-app`` is the name you give to your React project. You can replace it with your desired project name.
- After running this command, it will generate a new React application with all the necessary files and dependencies, ready to be developed.

In summary, ``npm create vue@latest`` is used for creating Vue.js projects, whereas ``npx create-react-app my-app`` is used for creating React.js projects. The main difference lies in the framework and tooling used for each project type.

---

We have create npx create-react-app netflix-gpt

Cd netflix-gpt

Npm start

---

We have installed tailwind css and added config.

---

Project structure

---

## # Features

- Login/Sign Up
  - Sign In /Sign up Form
  - redirect to Browse Page
- Browse (after authentication)
  - Header
  - Main Movie
    - Tailer in Background
    - Title & Description
    - MovieSuggestions
      - MovieLists \* N
- NetflixGPT
  - Search Bar
  - Movie Suggestions

---

Now lets start building the project.

Lets first create header , body.

Inside my body

```
import React from 'react'
import Login from './Login'
import Browse from './Browse'
const Body = () => {
  return (
    <div>
      <Login/>
      <Browse/>
    </div>
  )
}
```

```
export default Body
```

I created login and browse comp with rafce.

In my app.js

```
import Body from './components/Body';
import Header from './components/Header';
function App() {
  return (
    <div>
      <Header/>
      <Body/>
    </div>
  );
}
export default App;
```

If you notice , the root was removed but it still works somehow

Lets create router now.

So for this we need to insta react router dom

Npm I -D react-router-dom

```
"devDependencies": {
  "react-router-dom": "^6.16.0",
  "tailwindcss": "^3.3.3"
}
```

You can check in package,json it will be available.

```
import { RouterProvider, createBrowserRouter } from "react-router-dom";
import Login from "./Login";
const Browse = ()=>{
  // create a approuter for providing path
  const appRouter = createBrowserRouter(
    [
      {
        path: '/',
        element: <Login />
      },
      {
        path: '/browse',
        element: <Browse />
      },
    ]
  );
}
```

```

    )
    return(
      <div>
        <RouterProvider router={appRouter}/>
      </div>
    )
  }
export default Browse;

```

```

    <div>
      <RouterProvider router={appRouter}/>
    </div>

```

Now lets create a form.

```

import Header from "../Header";
const Login = () => {
  return (
    <div>
      <Header />
      <div className="absolute">
        
      </div>
      <form
        className="w-3/12 absolute p-12
        bg-black my-36 mx-auto right-0
        left-0 text-white rounded-lg
        bg-opacity-80
        "
      >
        <h1 className="font-bold text-3xl py-4">Sign In</h1>

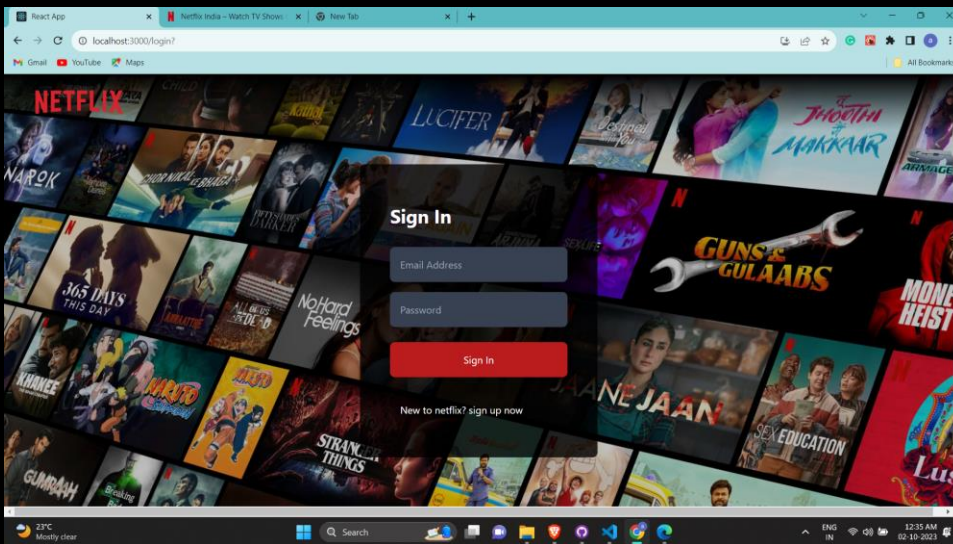
```

```

<input type="text" placeholder="Email Address"
className="p-4 my-4 block w-full bg-gray-700 rounded-lg" />
<input type="password" placeholder="Password"
className="p-4 my-2 block w-full bg-gray-700 rounded-lg" />
<button className="p-4 my-6 block bg-red-700 w-full rounded-lg">Sign In</button>
<p className="p-4">New to netflix? sign up now</p>

</form>
</div>
);
};
export default Login;

```



For this code I got this much

-----

See , now I need to build sign up form , but can I change the same thing to sign up form with a validation check??

```

const [isSignInForm , setIsSignInForm] = useState(true);

const toggleSignInForm = () => {
  setIsSignInForm(!isSignInForm);
};

```

In here I am using state variables . If somebody click on sign up this event will be triggered because , I have given onClick evenet – togglesigninform.

```

<p className="p-4" onClick={toggleSignInForm}>New to netflix? sign up now</p>

```

Here is the code on how I made toggle functionalities.

```

import { useState } from "react";
import Header from "../Header";

const Login = () => {

  const [isSignInForm, setIsSignInForm] = useState(true);

  const toggleSignInForm = () => {
    setIsSignInForm(!isSignInForm);
  };

  return (
    <div>
      <Header />
      <div className="absolute">
        
      </div>
      <form
        className="w-3/12 absolute p-12 bg-black my-36 mx-auto right-0 left-0 text-white rounded-lg bg-opacity-80"
      >

        <h1 className="font-bold text-3xl py-4">
          {isSignInForm ? "Sign In" : "Sign Up"}
        </h1>
        <input type="text"
          placeholder="Name"
          className="p-4 my-4 block w-full bg-gray-700 rounded-lg"
        />

        <input type="password" placeholder="Password"
          className="p-4 my-2 block w-full bg-gray-700 rounded-lg" />

        <button className="p-4 my-6 block bg-red-700 w-full rounded-lg">{isSignInForm
          ? "Sign In" : "Sign Up"}</button>

```

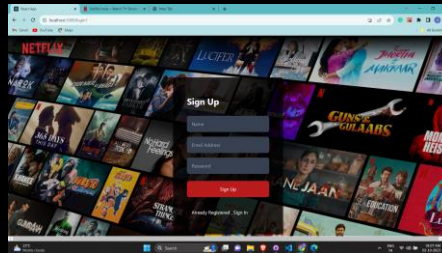
```

        <p className="p-4 cursor-pointer" onClick={toggleSignInForm}> {isSignInForm ?
        "New to Netflix? Sign Up now" : "Already Registered , Sign In"}</p>

        </form>
    </div>
  );
};

export default Login;

```



-----commit 3 done

Now lets see How to handle form Validations.

We will also use useRef hooks.

How to build form validations.

When I click on sign in it should validate and if there is any error it should throw error.

For that , I will use utility

```

const checkValidData = (email , password) =>{
  const isEmailValid = /^[a-zA-Z0-9.%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/ .test(email);
  const isPasswordValid = /^[a-zA-Z0-9.%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/ .test(password);
  if(!isEmailValid){
    return "Email is not valid";
  }
  if(!isPasswordValid){
    return "Password is not valid";
  }

  return "";
}

```

Now I will use this returned value for validation.

```
const handleClick = () => {
  // validate the form Data
  checkValidData(email,password);
}
```

But how do I get email and password . It should come from forms

Either I can use state variables where I use event handlers and update my state variable for name , password , email.

Or

UseRef - I get the reference of the input text fields and get the value present in it

```
const email = useRef(null);
const password = useRef(null);
```

-----

```
<input
  ref={email}
  type="text" placeholder="Email Address"
  className="p-4 my-4 block w-full bg-gray-700 rounded-lg" />
```

```
<input
  ref={password}
  type="password" placeholder="Password"
  className="p-4 my-2 block w-full bg-gray-700 rounded-lg" />
```

```
<button className="p-4 my-6 block bg-red-700 w-full rounded-lg"
  onClick={handleButtonClick}>
  {isSignInForm ? "Sign In" : "Sign Up"}</button>
```

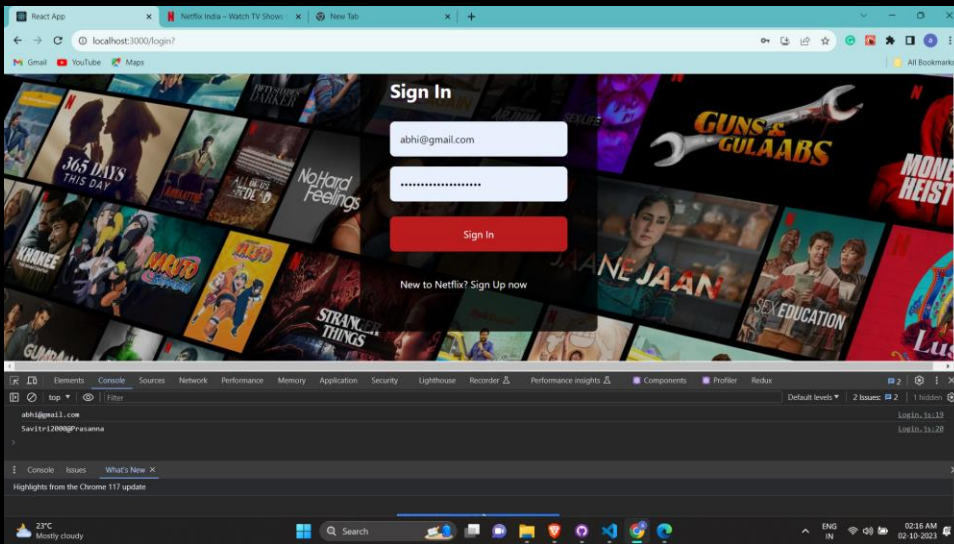
```
<p className="p-4 cursor-pointer" onClick={toggleSignInForm}> {isSignInForm ?
  "New to Netflix? Sign Up now" : "Already Registered , Sign In"}</p>
```

With the help of useRef, I referenced the input text areas of email and password

```
<form onSubmit={(e) => e.preventDefault()}>
```

I have done this because when I click on button its trying to submit and it will throw an error since it gets refreshed.





As you can see with useRef I was able to get that.

I will show you what exactly I have written.

```
const handleButtonClick = () => {
  // validate the form Data
  // checkValidData(email,password);
  console.log(email);
  console.log(password)
}
```

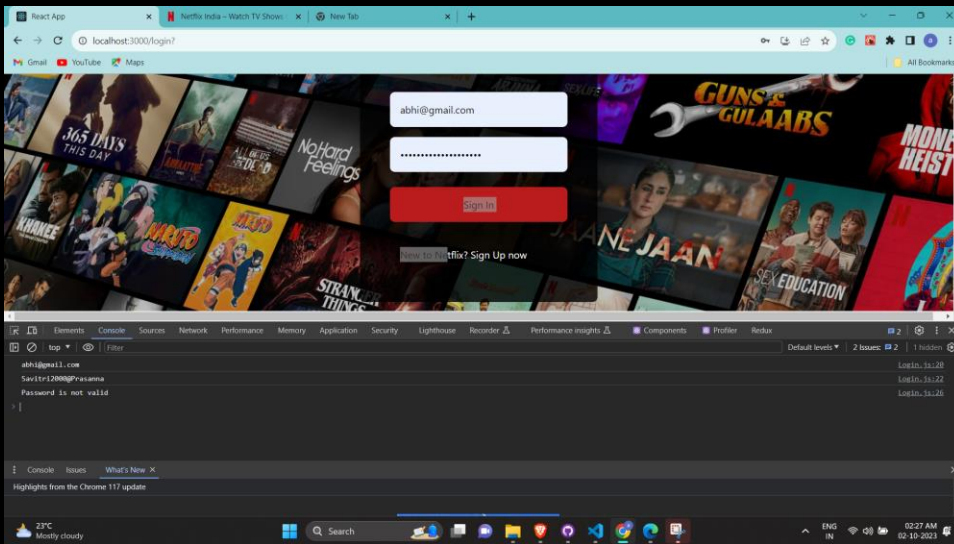
If I write this , Im console.log->

```
{
  "current": "input.p-4.my-4.block.w-full.bg-gray-700.rounded-lg",
  "current": "input.p-4.my-4.block.w-full.bg-gray-700.rounded-lg",
  "value": "abhi@gmail.com",
  "reactEvents$79nfmx7u1g": Set(1) {"Invalid_bubble"},
  "reactFiber$79nfmx7u1g": FiberNode {tag: 5, key: null, elementType: 'input', type: 'input', stateNode: input.p-4.my-4.block.w-full.bg-gray-700.rounded-lg},
  "reactProps$79nfmx7u1g": {type: 'text', placeholder: 'Email Address', className: 'p-4 my-4 block w-full bg-gray-700 rounded-lg'},
  "valueTracker": {getValue: f, setValue: f, stopTracking: f},
  "wrapperState": {initialChecked: undefined, initialValue: '', controlled: false},
  "accept": "",
  "accessKey": "",
  "align": "",
  "alt": "",
  "ariaAtomic": null,
  "ariaAutoComplete": null,
  "ariaBrailleLabel": null,
  "ariaBrailleRoleDescription": null,
  "ariaBusy": null
}
```

That's why we need to put curren.value

```
const handleButtonClick = () => {
  // validate the form Data
  // checkValidData(email,password);
  console.log(email)
  console.log(password)
}
```

To get name and password.



See the validation functionality is working.

Lets try to show the error message in the screen.

We need to store the error message and we can show them in ui.

`{errorMessage && <p className="text-red-500 font-bold p-2">{errorMessage}</p>}`  
 Just create a p tag and put this and it will render if there is a error message

Validate.js -

```
export const checkSignUpData = (name, email, password) => {
  const isNameValid = /^[a-zA-Z0-9]+$/.test(name);
  const isEmailValid = /^[a-zA-Z0-9. %+-]+@gmail\.com$/.test(email);
  const isPasswordValid = /^[a-zA-Z0-9]+$/.test(password);

  if (!isNameValid) {
    return "Name is not valid. It should be alphanumeric";
  }
  if (!isEmailValid) {
    return "Email is not valid. It should be alphanumeric and must have @gmail.com";
  }
  if (!isPasswordValid) {
    return "Password is not valid. It should be alphanumeric";
  }

  return null;
};

export const checkSignInData = (email, password) => {
  const isEmailValid = /^[a-zA-Z0-9. %+-]+@gmail\.com$/.test(email);
  const isPasswordValid = /^[a-zA-Z0-9]+$/.test(password);
  if (!isEmailValid) {
    return "Email is not valid. It should be alphanumeric and must have @gmail.com";
  }
}
```

```

    }
    if (!isPasswordValid) {
        return "Password is not valid. It should be alphanumeric";
    }
    return null;
};

```

The reason I have written 2 checks is because I need to see if its sign in or sign our.

```

const handleClick = () => {
    if (isSignInForm) {
        const message = checkSignInData(email.current.value, password.current.value);
        setErrorMessage(message);
    } else {
        const message = checkSignUpData(name.current.value, email.current.value,
password.current.value);
        setErrorMessage(message);
    }
};

```

Just used a handleClick which was already present to check if its sign in or sign out?

Then called appropriate functions on it.

-----Commit - 4