

Let's Create Our Frontend with React

Dr.Nuengwong Tuaycharoen

ref: <https://www.udemy.com/course/react-front-to-back-2022/>

You can copy the code from this chapter's slides!

Steps:


1. Stay just outside “vacq” folder
2. `$npx create-react-app vacqfrontend --template redux`
 - a folder “vacqfrontend” created with the node_module folder & react/redux package installed
3. `$cd vacqfrontend`
4. `$npm install react-router-dom react-toastify axios react-icons`
5. To run your frontend & backend: open 2 terminal windows
 1. In vacqfrontend folder: `$npm start`
 2. In vacq backend folder: `$npm run dev`
 3. both frontend & backend will restart automatically every time code is changed.
6. Go to the “src” folder to add the frontend code.

need to be all small
characters

React Redux App

localhost:3000

COVID-19: AI Deco...Automation TestingCP_สรุปตารางสอน 25...journalsCU



-0+

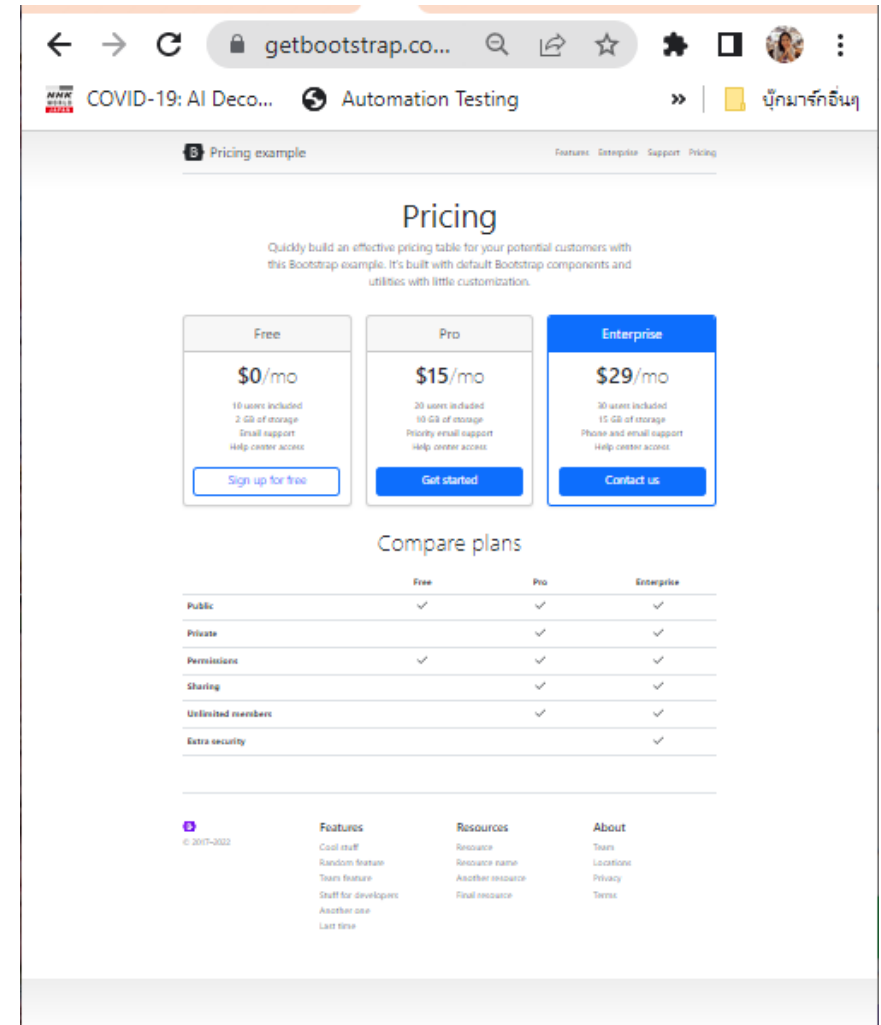
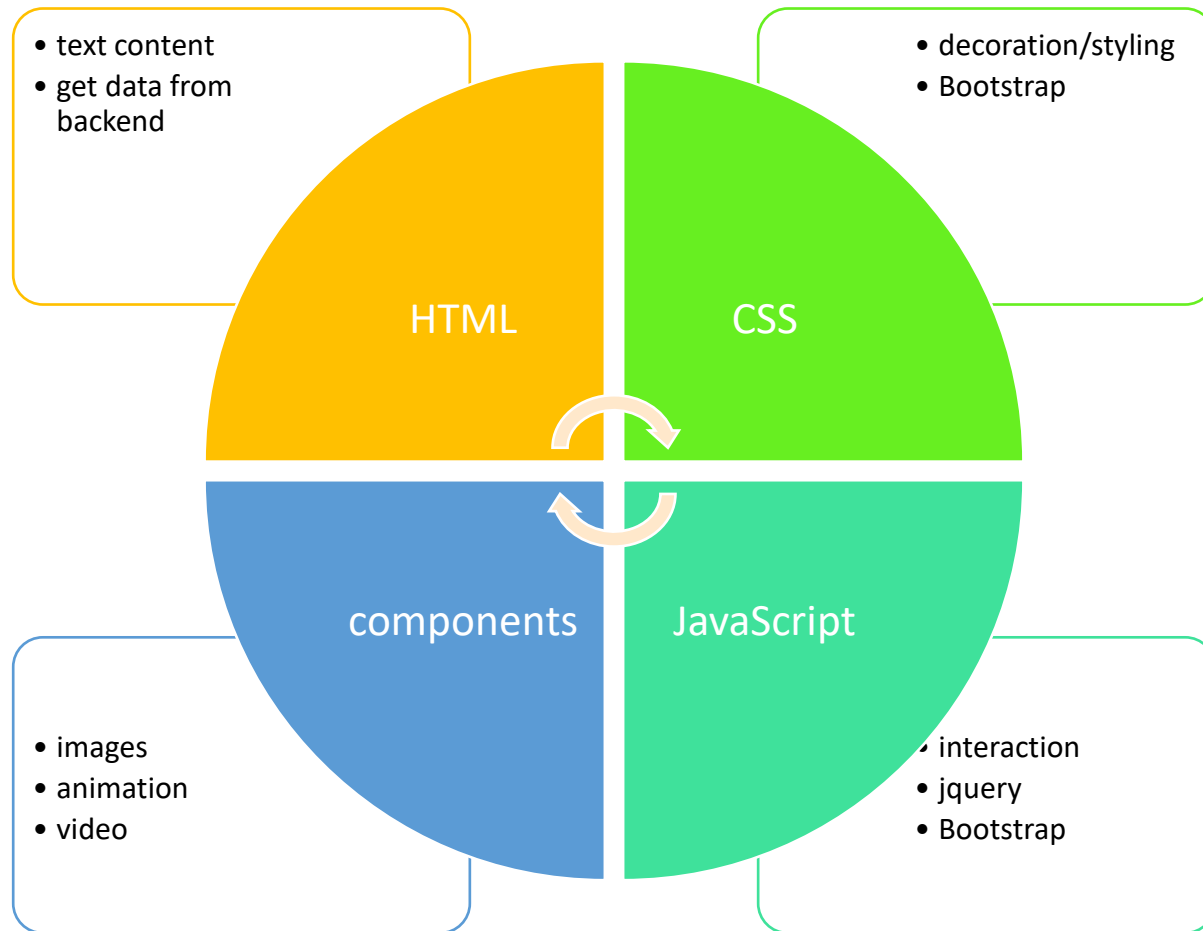
2Add AmountAdd AsyncAdd If Odd

Edit `src/App.js` and save to reload.

Learn [React](#), [Redux](#), [Redux Toolkit](#), and [React Redux](#)

Press F12 to open Chrome Developer Tools

Original Web Frontend Structure



index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible"
content="ie=edge">
  <link rel="stylesheet" href="./style.css">
  <script defer type="text/javascript"
src="./script.js"></script>
  <title>Document</title>
</head>
<body>
  <h1>This is a first level heading in HTML. With
CSS, I will turn this into red color</h1>
  <h2>This is a second level heading in HTML. With
CSS, I will turn this into blue color</h2>
  <h3>This is a third level heading in HTML. With
CSS, I will turn this into green color</h3>
  <p>This is a <em>paragraph</em> As you can see, I
placed an emphasis on the word "paragraph". Now, I
will change also
  the background color of the word "paragraph" to
black, and its text color to green, all with just
CSS.</p>
  <p>The main essence of this tutorial is to:</p>
```

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ul>
```

```
<p>Next, I am going to add the following two
numbers and display the result, all with
JavaScript</p>
<p>First number:<span id="firstNum">2</span>
<br></p>
<p>Second number: <span id="secondNum">7</span>
</p>
<p>Therefore, the sum of the two of those numbers
is: <span id="answer">(placeholder for the
answer)</span></p>
<input type="button" id="sumButton" value="Click
to add!">
</body>
</html>
```

script.js

```
function displaySum() {
  let firstNum = Number(document.getElementById('firstNum').innerHTML)
  let secondNum = Number(document.getElementById('secondNum').innerHTML)

  let total = firstNum + secondNum;
  document.getElementById("answer").innerHTML = ` ${firstNum} + ${secondNum}
equals to ${total}` ;
}

document.getElementById('sumButton').addEventListener("click", displaySum);
```

style.css

```
h1 {
  background-color: #ff0000;
}

h2 {
  background-color: #0000FF;
}

h3 {
  background-color: #00FF00;
}

em {
  background-color: #000000;
  color: #ffffff;
}
```

Document x +

127.0.0.1:5500/basicWeb/

COVID-19: AI Deco... Automation Testing CP_ สรุปตารางสอน 25... journals CU CP_ สรุปตารางสอน 25... Other bookmarks

This is a first level heading in HTML. With CSS, I will turn this into red color

This is a second level heading in HTML. With CSS, I will turn this into blue color

This is a third level heading in HTML. With CSS, I will turn this into green color

This is a **paragraph** As you can see, I placed an emphasis on the word "paragraph". Now, I will change also the background color of the word "paragraph" to black, and its text color to green, all with just CSS.

The main essence of this tutorial is to:

- Show you how to format a web document with HTML
- Show you how to design a web page with CSS
- Show you how to program a web document with JavaScript

Next, I am going to add the following two numbers and display the result, all with JavaScript

First number: 2

Second number: 7

Therefore, the sum of the two of those numbers is: $2 + 7$, equals to 9

Click to add!

React Template: index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport"
content="width=device-width, initial-
scale=1" />
    <meta name="theme-color"
content="#000000" />
    <meta
      name="description"
      content="Web site created using
create-react-app"
    />
    <link rel="apple-touch-icon"
href="/logo192.png" />
    <link rel="manifest"
href="/manifest.json" />
    <title>React Redux App</title>
    <script defer
src="/static/js/bundle.js"></script></head>
    <body>
      <noscript>You need to enable JavaScript
to run this app.</noscript>
      <div id="root"></div>
    </body>
  </html>
```

React Template: index.css

```
body {  
  margin: 0;  
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',  
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',  
    sans-serif;  
  -webkit-font-smoothing: antialiased;  
  -moz-osx-font-smoothing: grayscale;  
}  
  
code {  
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',  
    monospace;  
}
```

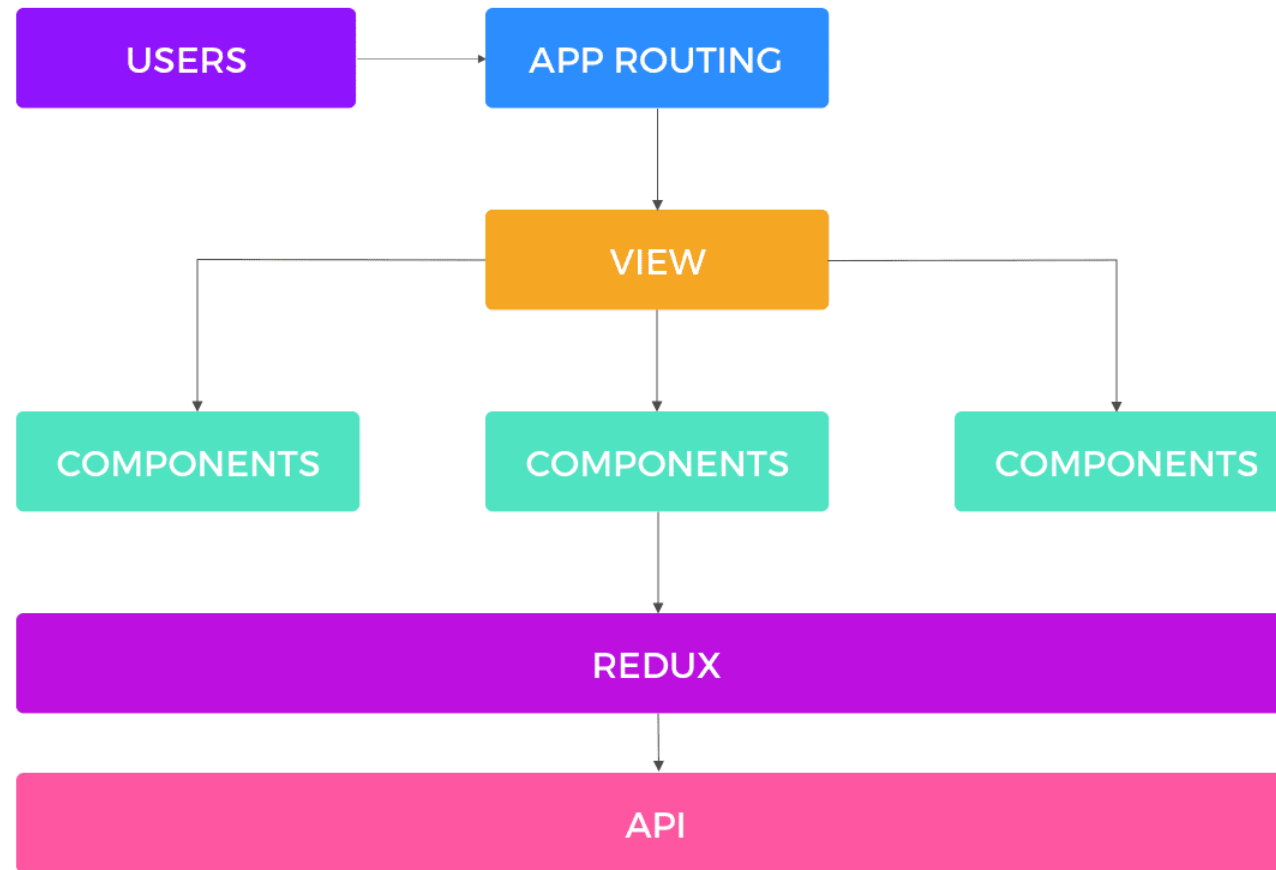

React Template: JavaScript & JSX

```
import React from 'react';
import { createRoot } from 'react-dom/client';
import { Provider } from 'react-redux';
import { store } from './app/store';
import App from './App';
import reportWebVitals from './reportWebVitals';
import './index.css';

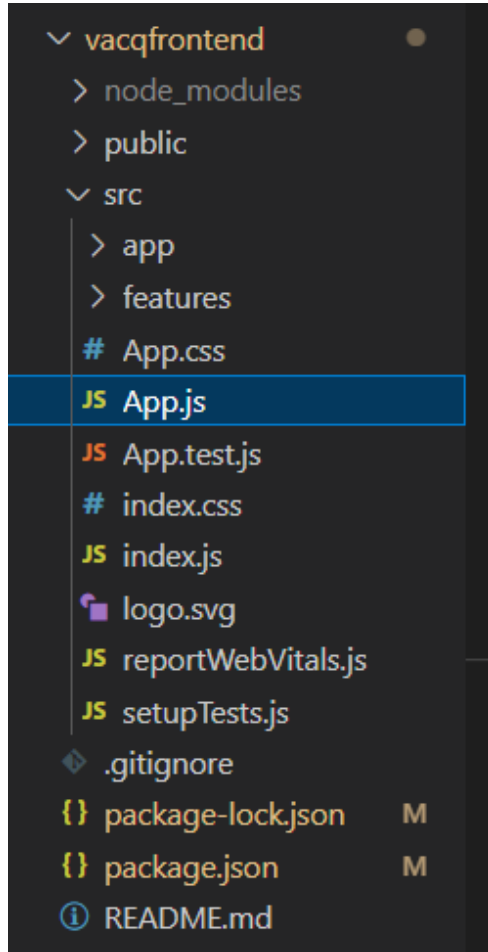
const container = document.getElementById('root');
const root = createRoot(container);

root.render(
  <React.StrictMode>
    <Provider store={store}>
      <App />
    </Provider>
  </React.StrictMode>
);
reportWebVitals();
```

React.js Application Flow



frontend file structure



- index.js: frontend entry point (no modification)
- index.css: main decoration
- **App.js**: Your main code is here, including all components
- App.css: decoration for App.js
- package.json: frontend installation settings

index.css

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600;700&display=swap');* { margin: 0; padding: 0; box-sizing: border-box;}body { font-family: 'Poppins', sans-serif; height: 100vh;}a { text-decoration: none; color: #000;}p { line-height: 1.7;}ul { list-style: none;}li { line-height: 2.2;}h1,h2,h3 { font-weight: 600; margin-bottom: 10px;}container { width: 100%; max-width: 960px; margin: 0 auto; padding: 0 20px; text-align: center;}header { display: flex; justify-content: space-between; align-items: center; padding: 20px 0; border-bottom: 1px solid #e6e6e6; margin-bottom: 60px;}header ul { display: flex; align-items: center; justify-content: space-between;}header ul li { margin-left: 20px;}header ul li a { display: flex; align-items: center;}header ul li a: hover { color: #777;}header ul li a svg { margin-right: 5px;}heading { font-size: 2rem; font-weight: 700; margin-bottom: 50px; padding: 0 20px;}heading p { color: #828282;}boxes { display: grid; grid-template-columns: repeat(3, 1fr); gap: 20px; justify-content: space-between; align-items: center; text-align: center; margin-bottom: 30px;}boxes div { padding: 30px; border: 1px solid #e6e6e6; border-radius: 10px;}boxes h2 { margin-top: 20px;}boxes a: hover { color: #777;}form { width: 70%; margin: 0 auto;}form-group { margin-bottom: 10px;}form-group input, form-group textarea, form-group select { width: 100%; padding: 10px; border: 1px solid #e6e6e6; border-radius: 5px; margin-bottom: 10px; font-family: inherit;}form-group label { text-align: left; display: block; margin: 0 0 5px 3px;}btn { padding: 10px 20px; border: 1px solid #000; border-radius: 5px; background: #000; color: #fff; font-size: 16px; font-weight: 700; cursor: pointer; text-align: center; appearance: button; display: flex; align-items: center; justify-content: center;}btn svg { margin-right: 8px;}btn-reverse { background: #fff; color: #000;}btn-block { width: 100%; margin-bottom: 20px;}btn-sm { padding: 5px 15px; font-size: 13px;}btn-danger { background: darkred; border: none;}btn-back { width: 150px; margin-bottom: 20px;}btn: hover { transform: scale(0.98);}ticket-created { border: 1px solid #e6e6e6; border-radius: 5px; padding: 50px;}ticket-number { margin-bottom: 30px;}ticket-number h2 { font-size: 2.3rem; margin-bottom: 10px;}ticket-number p { font-size: 1.3rem;}ticket-info { font-size: 1.3rem;}ticket, ticket-headings { display: grid; grid-template-columns: repeat(4, 1fr); gap: 20px; justify-content: space-between; align-items: center; margin-bottom: 20px; background: #f4f4f4; padding: 10px 15px; border-radius: 5px; text-align: center;}ticket-headings { font-weight: 700;}status { background-color: #333; color: #fff; width: 100px; padding: 0 20px; justify-self: center; border-radius: 10px; font-size: 16px; text-align: center;}status-new { background-color: green; color: #fff; border-radius: 10px;}status-open { background-color: steelblue; color: #fff; border-radius: 10px;}status-closed { background-color: darkred; color: #fff; border-radius: 10px;}ticket-page { position: relative; text-align: left;}ticket-page h2 { display: flex; align-items: center; justify-content: space-between;}ticket-page .btn { margin-bottom: 30px;}ticket-page .btn-block { width: 100%; margin-top: 30px;}ticket-desc { margin: 20px 0; font-size: 17px; background-color: #f4f4f4; border: 1px #ccc solid; padding: 10px 15px; border-radius: 5px;}note { border: 1px solid #e6e6e6; border-radius: 5px; text-align: left; padding: 20px; margin-bottom: 20px; position: relative;}note-head { background: #f4f4f4; padding: 5px 20px; display: flex; justify-content: space-between; align-items: center; margin-bottom: 20px;}note-date { position: absolute; top: 15px; right: 10px; font-size: 14px;}delete-note { color: red; cursor: pointer; position: absolute; bottom: 10px; right: 20px;}btn-close { background: none; border: none; color: #000; position: absolute; top: 5px; right: 5px; font-size: 16px; cursor: pointer;}btn-close: hover { color: red; transform: scale(0.98);}p.status-in-progress { color: orangered;}p.status-waiting { color: red;}p.status-ready { color: steelblue;}p.status-complete { color: green;}footer { position: sticky; top: 95vh; text-align: center; align-items: center;}loadingSpinner { position: fixed; top: 0; right: 0; bottom: 0; left: 0; background-color: rgba(0, 0, 0, 0.5); z-index: 5000; display: flex; justify-content: center; align-items: center;}loadingSpinner { width: 64px; height: 64px; border: 8px solid; border-color: #000 transparent #555 transparent; border-radius: 50%; animation: spin 1.2s linear infinite;}@keyframes spin { 0% { transform: rotate(0deg); } 100% { transform: rotate(360deg); }}@media (max-width: 600px) { .boxes { grid-template-columns: 1fr; } .form { width: 90%; } .ticket-created h2, .heading h1 { font-size: 2rem; } .heading p { font-size: 1.5rem; }}
```

App.js

```
import {BrowserRouter as Router, Routes, Route} from
'react-router-dom'
import {ToastContainer} from 'react-toastify'
import 'react-toastify/dist/ReactToastify.css'
import Header from './components/Header'
import Home from './pages/Home'
import Login from './pages/Login'
import Register from './pages/Register'

function App() {
  return (
    <>
    <Router>
    <div className="container">
      <Header />
      <Routes>
        <Route path="/" element={<Home/>} />
        <Route path="/login" element={<Login/>} />
        <Route path="/register" element={<Register/>} />
      </Routes>
    </div>
    </Router>
    <ToastContainer />
  </>
);
}

export default App;
```

create 4 components

1. create a folder /src/pages
2. create 3 files for 3 components
 1. Home.jsx
 2. Login.jsx
 3. Register.jsx
3. create a folder /src/components
4. create a file: Header.jsx

pages/Home.jsx

```
import {Link} from 'react-router-dom'
import {FaQuestionCircle, FaTicketAlt}
from 'react-icons/fa'

function Home(){
  return(
    <>
      <section className='heading'>
        <h1>Vac Q: A Vaccine
Booking System</h1>
        <p>Please choose from an
option below</p>
      </section>
      <Link to='/new-ticket'
className='btn btn-reverse btn-block'>
        <FaQuestionCircle />Create New Appointment
      </Link>
      <Link to='/tickets'
className='btn btn-block'>
        <FaTicketAlt />View My
        Appointments
      </Link>
    </>
  )
}

export default Home;
```

pages/Register.jsx

```
import {useState} from 'react'
import {toast} from 'react-toastify'
import {FaUser} from 'react-icons/fa'

function Register(){
  const [formData,setFormData]=useState({
    name:'',
    email: '',
    password: '',
    password2:''
  });

  const {name,email,password,password2}=formData;

  const onChange = (e) =>{
    setFormData((prevState)=>({
      ...prevState,
      [e.target.name]: e.target.value
    }));
  }

  const onSubmit =(e) => {
    e.preventDefault()

    if(password!==password2){
      toast.error('Passwords do not match')
    }
  }

  return(
    <>
      <section className="heading">
        <h1>
          <FaUser /> Register
        </h1>
        <p>Please create an account</p>
      </section>

      <section className='form'>
        <form onSubmit={onSubmit}>
          <div className="form-group">
            <input type="text" className='form-control'
              id='name' name='name' value={name} onChange={onChange}
              placeholder='Enter Your name' required/>
          </div>
          <div className="form-group">
            <input type="email" className='form-control'
              id='email' name='email' value={email} onChange={onChange}
              placeholder='Enter Your email' required/>
          </div>
          <div className="form-group">
            <input type="password" className='form-control'
              id='password' name='password' value={password}
              onChange={onChange} placeholder='Enter Your password' required/>
          </div>
          <div className="form-group">
            <input type="password" className='form-control'
              id='password2' name='password2' value={password2}
              onChange={onChange} placeholder='Confirm Your password' required/>
          </div>
          <div className='form-group'>
            <button className='btn btn-block'>Submit</button>
          </div>
        </form>
      </section>
    </>
  )
}

export default Register;
```


pages/Login.jsx

```
import {useState} from 'react'
import {toast} from 'react-toastify'
import {FaSignInAlt} from 'react-icons/fa'

function Login(){
  const [formData,setFormData]=useState({
    email: '',
    password: '',
  });

  const {name,email,password,password2}=formData;

  const onChange = (e) =>{
    setFormData((prevState)=>({
      ...prevState,
      [e.target.name]: e.target.value
    }));
  }

  const onSubmit =(e) => {
    e.preventDefault()
  }

  return(
    <>
      <section className="heading">
        <h1>
          <FaSignInAlt /> Login
        </h1>
        <p>Please login to get support</p>
      </section>

      <section className='form'>
        <form onSubmit={onSubmit}>
          <div className="form-group">
            <input type="email" className='form-control' id='email' name='email'
              value={email} onChange={onChange} placeholder= 'Enter Your email' required/>
          </div>
          <div className="form-group">
            <input type="password" className='form-control' id='password' name='password'
              value={password} onChange={onChange} placeholder= 'Enter Your password' required/>
          </div>
          <div className='form-group'>
            <button className='btn btn-block'>Submit</button>
          </div>
        </form>
      </section>
    </>
  )
}

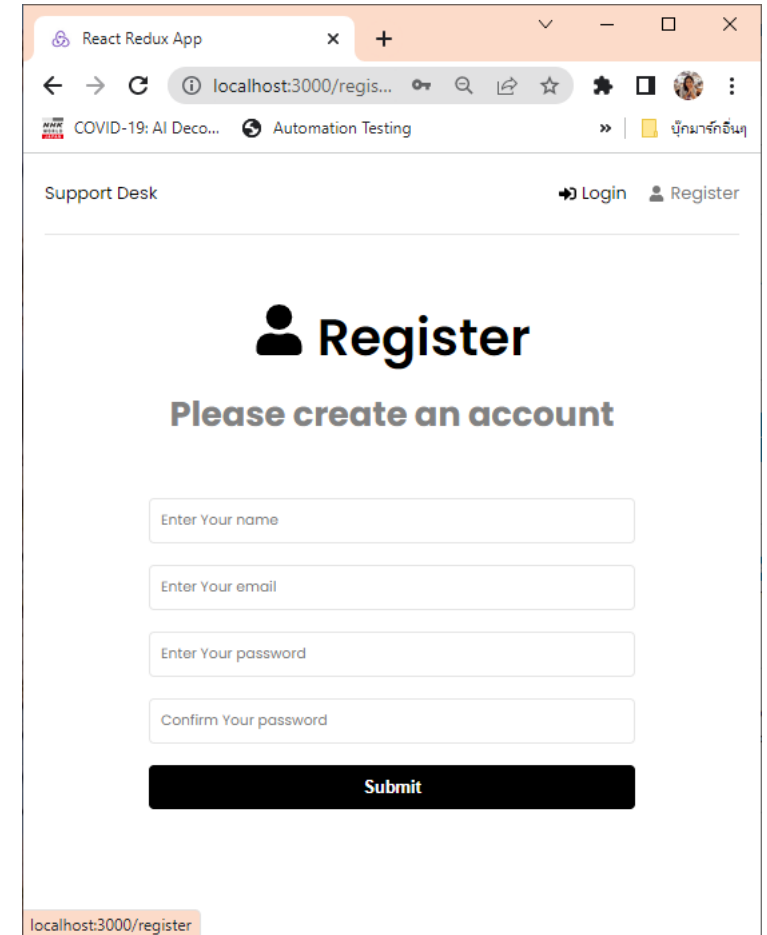
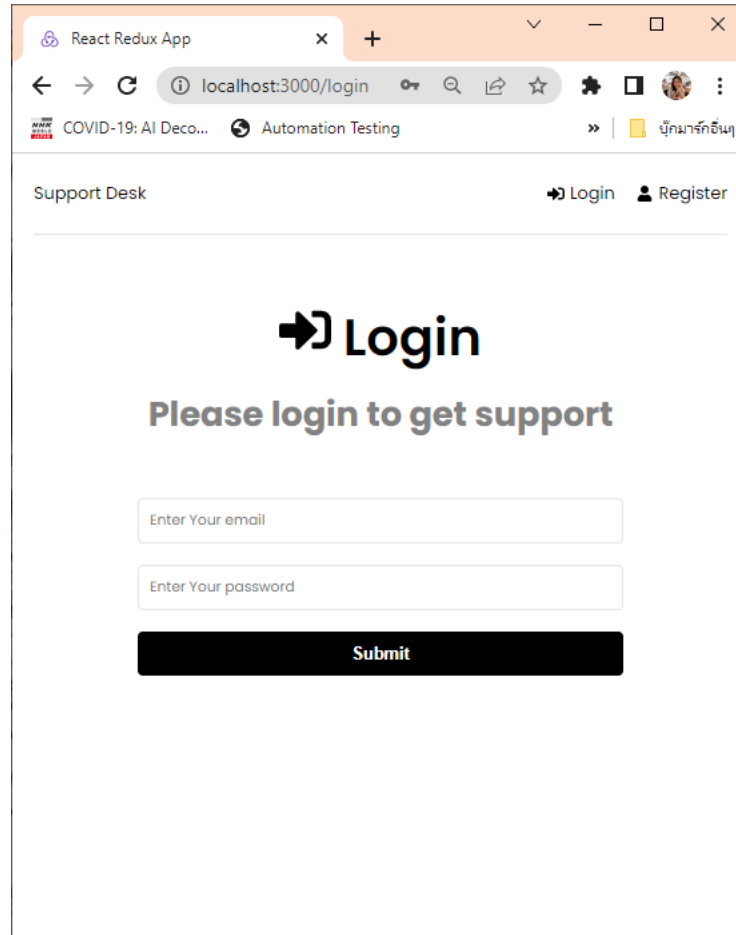
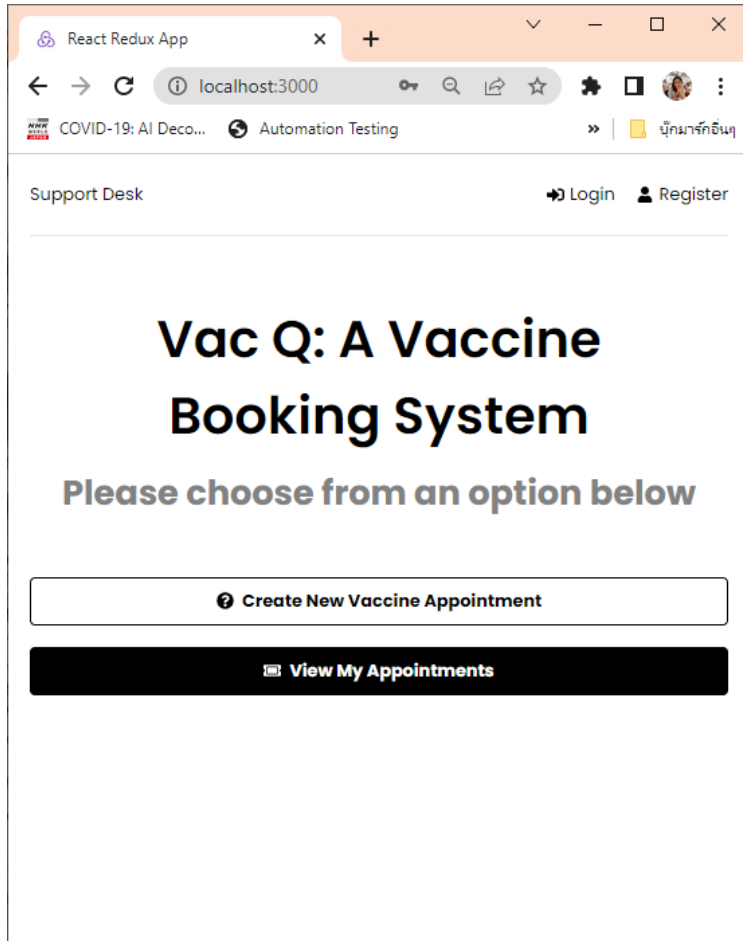
export default Login;
```

components/Header.jsx

```
import {FaSignInAlt, FaSignOutAlt, FaUser} from 'react-icons/fa'
import {Link} from 'react-router-dom'

function Header(){
  return (
    <header className='header'>
      <div className='logo'>
        <Link to='/'>Support Desk</Link>
      </div>
      <ul>
        <li>
          <Link to='/login'>
            <FaSignInAlt />Login
          </Link>
        </li>
        <li>
          <Link to='/register'>
            <FaUser />Register
          </Link>
        </li>
      </ul>
    </header>
  )
}

export default Header
```



GIT

- `git init`
- `git add .`
- `git commit -m 'add pages: Home, Login, Register, Header'`

Connect Register with Database

- delete the folder “features/counter” and all files inside
 - they are sample code to use redux.
- create a folder features/auth
- create 2 files:
 - authService.js
 - authSlice.js

features/auth/authService.js

```
import axios from 'axios'

const
API_URL='http://localhost:5000/api/v1/auth/register'

//Register user
const register = async (userData) => {
  try{
    const response = await
    axios.post(API_URL,userData)
    console.log(JSON.stringify(response.data))
    console.log(response.data.name)
    if(response.data) {
      //localStorage.setItem('user',JSON.stringify(
response.data))
      localStorage.setItem('user',response.data.name)
    }
  } catch(error) {
    console.log('authService: register')
    console.log(error);
  }
  return response.data.name//response.data
}

const authService ={
  register,
}

export default authService;
```

features/auth/authSlice.js

```
import {createSlice, createAsyncThunk} from '@reduxjs/toolkit'
import authService from './authService'

//Get user from localStorage
//const user = JSON.parse(localStorage.getItem('user'))

const user=localStorage.getItem('user')

const initialState={
  user: user? user: null,
  //user: null,
  isError: false,
  isSuccess:false,
  isLoading: false,
  message: ''
}

//Register new user
export const register =
createAsyncThunk('auth/register',async(user,thunkAPI)=>{
  console.log(user);
  try {
    return await authService.register(user)
  } catch (error) {
    const message=(error.response && error.response.data &&
error.response.data.message)||error.message||error.toString();
    return thunkAPI.rejectWithValue(message)
  }
})

export const login =
createAsyncThunk('auth/login',async(user,thunkAPI)=>{
  console.log(user);
})

export const authSlice = createSlice({
  name: 'auth',
  initialState,
  reducers: {
    reset: (state)=> {
      state.isLoading = false
      state.isError = false
      state.isSuccess = false
      state.message = ''
    },
  },
  extraReducers: (builder)=> {
    builder.addCase(register.pending, (state)=>{
      state.isLoading=true
    })
    .addCase(register.fulfilled,(state, action)=>{
      state.isLoading = false
      state.isSuccess=true
      state.user=action.payload
    })
    .addCase(register.rejected,(state, action)=>{
      state.isLoading = false
      state.isError=true
      state.message=action.payload
      state.user=null
    })
  },
})

export const {reset} = authSlice.actions
export default authSlice.reducer
```

app/store.js

```
import { configureStore } from '@reduxjs/toolkit';  
import authReducer from '../features/auth/authSlice';  
  
export const store = configureStore({  
  reducer: {  
    auth: authReducer,  
  },  
});
```


pages/Register.jsx

```
import {useState, useEffect} from 'react'
import {useNavigate} from 'react-router-dom'
import {toast} from 'react-toastify'
import {FaUser} from 'react-icons/fa'
import {useSelector, useDispatch} from 'react-redux'
import {register, reset} from '../features/auth/authSlice'

function Register(){

  const [formData,setFormData]=useState({

    name:'',

    email: '',

    password: '',

    password2:'',

    role:'user'

  });

  const {name,email,password,password2,role}=formData;

  const dispatch = useDispatch()
  const navigate=useNavigate()

  const {user,isLoading, isError, isSuccess, message}=useSelector((state)=>{

    return state.auth

  })

  useEffect(()=>{

    if(isError){

      toast.error(message)

    }

    //redirect when logged in

    if(isSuccess || user){

      navigate('/')

    }

    dispatch(reset())

  },[isError, isSuccess, user, message, navigate, dispatch])

  const onChange = (e) =>{

    setFormData((prevState)=>({

      ...prevState,

      [e.target.name]: e.target.value

    }));

  }

  const onSubmit =(e) => {

    e.preventDefault()

    if(password!==password2){

      toast.error('Passwords do not match')

    } else {

      const userData ={

        name,

        email,

        password,

        role

      }

      dispatch(register(userData))

    }

  }

  return(

    <>

      <section className="heading">

        <h1>

          <FaUser /> Register

        </h1>

        <p>Please create an account</p>

      </section>

    </>

  )

}

export default Register;
```

backend/controllers/auth.js >> sendTokenResponse()

```
//Get token from model, create cookie and send response
const sendTokenResponse=(user,statusCode,res)=>{
  //Create token
  const token=user.getSignedJwtToken();

  const options={
    expires:new Date(Date.now()+process.env.JWT_COOKIE_EXPIRE*24*60*60*1000),
    httpOnly:true
  };

  if(process.env.NODE_ENV==='production'){
    options.secure=true;
  }

  res.status(statusCode)/*.cookie('token',token,options)*/.json({
    success:true,
    //add for frontend
    _id:user._id,
    name: user.name,
    email:user.email,
    //end for frontend
    token
  })
}
```

The screenshot shows a web browser window with the address bar displaying 'localhost:30...'. The page title is 'React Redux App'. The main content area features a 'Register' form with the following fields:

- Username: 'user8'
- Email: 'user8@gmail.com'
- Password: '.....'
- Confirm Password: '.....'
- First Name: 'user'

A 'Submit' button is located at the bottom of the form. Below the form, the Chrome DevTools interface is visible, showing the 'Sources' panel with a file named 'top'. The 'Console' panel shows a log entry:

```
{email: 'user3@gmail.com', password: '12345678'} authSlice.js:29
null authSlice.js:61
```

The screenshot shows a web browser with the address bar displaying 'localhost:3000'. The page title is 'React Redux App'. The main content area has a heading 'Vac Q: A Vaccine Booking System' and a subheading 'Please choose from an option below'. There are two buttons: 'Create New Vaccine Appointment' and 'View My Appointments'. Below the buttons, the browser's developer console is open, showing a log of a successful login for user8 with email user8@gmail.com and password 12345678. The log includes the request body, the response, and the success message.

pages/Login.jsx

```
import {useState, useEffect} from 'react'
import {toast} from 'react-toastify'
import {FaSignInAlt} from 'react-icons/fa'
import {useSelector, useDispatch} from 'react-redux'
import {login, reset} from '../features/auth/authSlice'
import { useNavigate } from 'react-router-dom'

function Login(){
  const [formData, setFormData]=useState({
    email: '',
    password: '',
  });

  const {name, email, password, password2}=formData;

  const dispatch = useDispatch()
  const navigate =useNavigate()

  const {user, isLoading, isError, isSuccess, message}=useSelector(state=>state.auth)

  useEffect(()=>{
    if(isError){
      toast.error(message)
    }

    //redirect when logged in
    if(isSuccess || user){
      navigate('/')
    }
  }, [isError, message, isSuccess, user])

  const [userData, setEmail, setPassword] = useState({
    email: '',
    password: ''
  })

  const onChange = (e) =>{
    setFormData((prevState)=>({
      ...prevState,
      [e.target.name]: e.target.value
    }));
  }

  const onSubmit =(e) => {
    e.preventDefault()

    const userData={
      email,
      password
    }

    dispatch(login(userData))
  }

  return(
    <>
      <section className="heading">
        <h1>
          <FaSignInAlt /> Login
        </h1>
        <p>Please login to get support</p>
      </section>

      <section className='form'>
        <form onSubmit={onSubmit}>
          <div className="form-group">
            <input type="email" className='form-control'
              id='email' name='email' value={email} onChange={onChange}
              placeholder='Enter Your email' required/>
          </div>
          <div className="form-group">
            <input type="password" className='form-control' id='password' name='password' value={password}
              onChange={onChange} placeholder='Enter Your password' required/>
          </div>
          <div className='form-group'>
            <button className='btn btn-block'>Submit</button>
          </div>
        </form>
      </section>
    </>
  )
}

export default Login;
```

features/auth/authService.js

```
import axios from 'axios'

const API_URL='http://localhost:5000/api/v1/auth/'

//Register user
const register = async (userData) => {
  const response = await axios.post(API_URL+'register/',userData)
  if(response.data) {
    //localStorage.setItem('user',JSON.stringify(response.data))
    localStorage.setItem('user',response.data.name)
  }
  console.log(response.data);
  return response.data
}

//Login user
const login = async (userData) => {
  const response = await axios.post(API_URL + 'login',userData)

  if(response.data) {
    //localStorage.setItem('user',JSON.stringify(response.data))
    localStorage.setItem('user',response.data.name)
  }
  console.log(response.data);
  return response.data
}

//Logout user
const logout = () => {
  localStorage.setItem('user', null);
}

const authService ={
  register,
  logout,
  login,
}

export default authService;
```

features/auth/authSlice.js

```
...
//Register new user
export const register =
createAsyncThunk('auth/register', async(user, thunkAPI) =>{
  ...
})

//Login user
export const login =
createAsyncThunk('auth/login', async(user, thunkAPI) =>{
  try {
    return await authService.login(user)
  } catch (error) {
    const message=(error.response && error.response.data &&
error.response.data.message) || error.message || error.toString();
    return thunkAPI.rejectWithValue(message)
  }
})

//Logout user
export const logout = createAsyncThunk('auth/logout', async()=>{
  await authService.logout()
})

export const authSlice = createSlice({
  ...
  .addCase(register.rejected,(state, action)=>{
```

```
...
  })
  .addCase(login.pending, (state)=>{
    state.isLoading=true
  })
  .addCase(login.fulfilled,(state, action)=>{
    state.isLoading = false
    state.isSuccess=true
    state.user=action.payload
  })
  .addCase(login.rejected,(state, action)=>{
    state.isLoading = false
    state.isError=true
    state.message=action.payload
    state.user=null
  })
  .addCase(logout.fulfilled,(state)=>{
    state.user=null
  })
},
})

export const {reset} = authSlice.actions
export default authSlice.reducer
```

components/Header.jsx

```
import {FaSignInAlt, FaSignOutAlt, FaUser} from 'react-icons/fa'
import {Link, useNavigate} from 'react-router-dom'
import {useSelector, useDispatch} from 'react-redux'
import {logout, reset} from '../features/auth/authSlice'

function Header(){
  const navigate = useNavigate();
  const dispatch = useDispatch()
  const {user} = useSelector((state)=>state.auth)

  const onLogout =()=>{
    dispatch(logout())
    dispatch(reset())
    navigate('/')
  }

  return (
    <header className='header'>
      <div className='logo'>
        <Link to='/'>Support Desk</Link>
      </div>
      <ul>
        {user?(
          <li>
            <button className='btn' onClick={onLogout}><FaSignOutAlt />
            Logout</button>
          </li>
          <li>
            <Link to='/login'>
              <FaSignInAlt />Login
            </Link>
          </li>
          <li>
            <Link to='/register'>
              <FaUser />Register
            </Link>
          </li>
        )}
      </ul>
    </header>
  )
}

export default Header
```

CORS

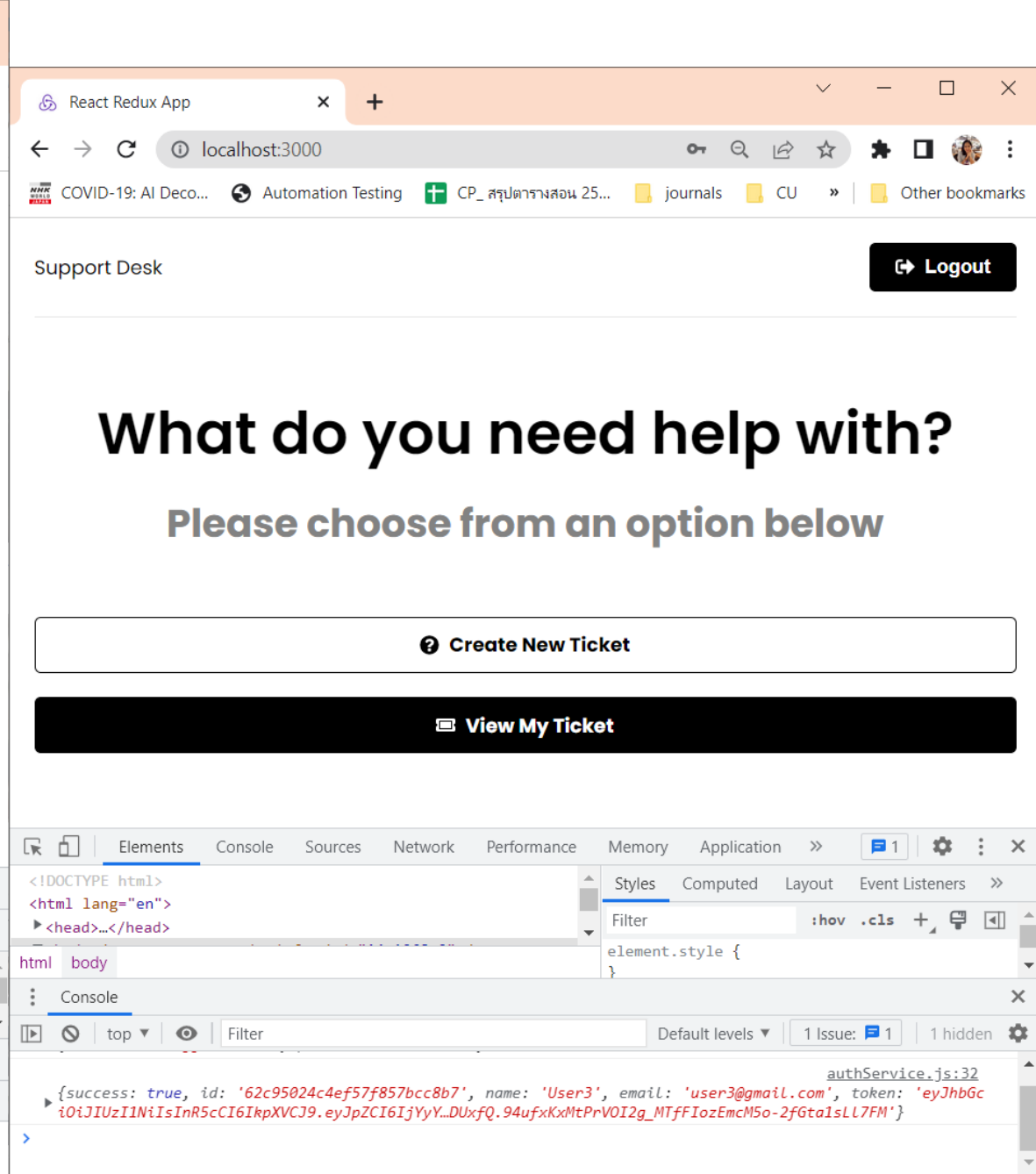
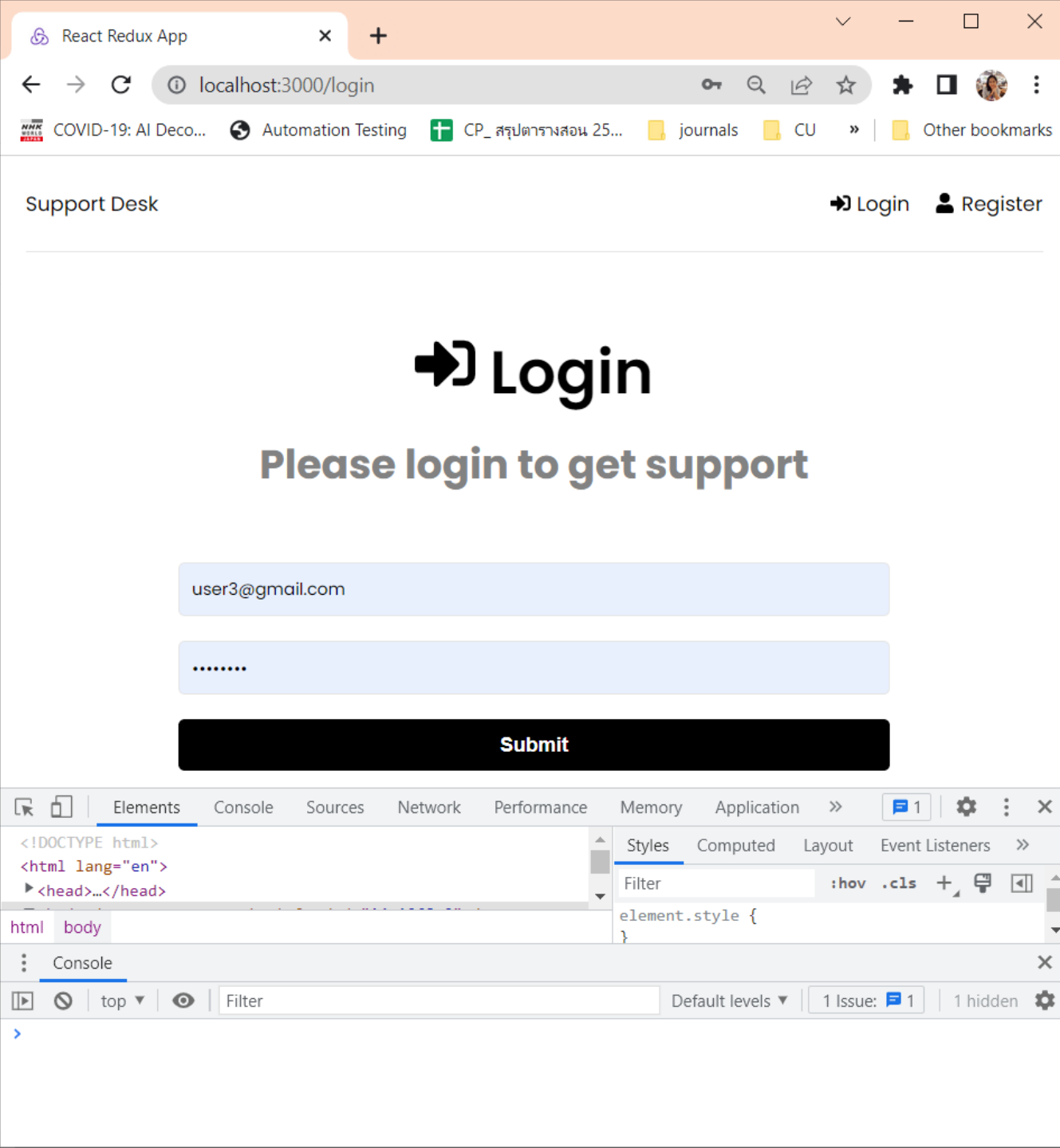
- if you get this error:
 - Access to XMLHttpRequest at 'http://localhost:5000/api/v1/auth/register' from origin 'http://localhost:3000' has been blocked by **CORS** policy: Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource.
- stop your backend server & go to your backend folder.
- `$npm install cors --save`
- add theses 2 lines in your server.js

```
const cors = require('cors');
```

```
...
```

```
const app = express();
```

```
app.use(cors());
```

Conclusions

- HTML, CSS, JavaScript
- React-JSX
- Redux
 - authSlice
 - authService
- Register
- Login