DAY 4

Thursday, 8 May 2025 7:59 AM

Day 4

- yarn add react-router-dom toastify axious.

Home.jsx

auth > Login
Register.
property > Home.jsx & properties.jsx.
Add Property.jsx.
Prop Details.jsx
My Property.jsx.
User > profile.jsx.
utils > contact us.jsx
aboutus.jsx

rfce → shortcut Boilerplate.

Q. Configure router
→ In main.jsx → <BrowserRouter>
App
<B — R —>
In App.jsx → write all Routes.

```
<Routes>
    <Route path='/' element={} />
<Route/>
```

Add Bootstrap → index.HTML
Bring / Copy server.
Run server.

Q. prblm → NavBar. Redundancy issue.

soln → parent-child
Nested Routing

Top level / Parent Route

- Navbar home saglyat copy karva lagte.
/home.

Only login needs to be Top-level.
Register
Home.

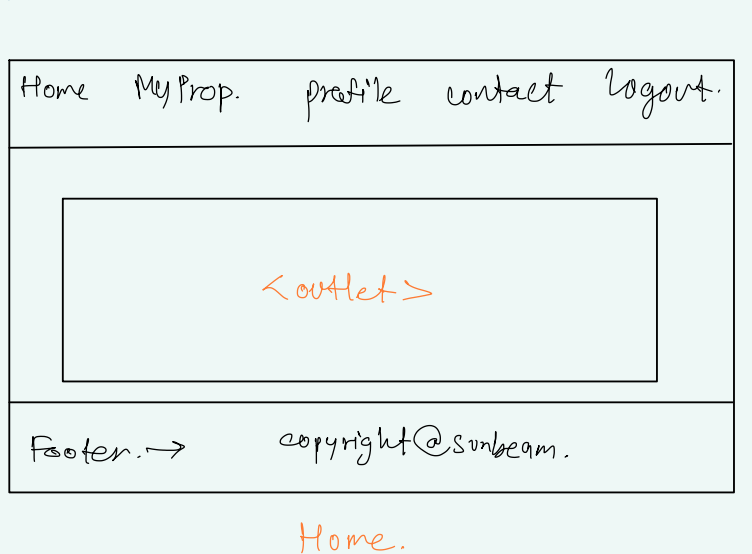
Rest of them are child level routes.

/aboutus Art hoto launch 1a top-level hota.

Now, /home/aboutus → asa lihycha.
↘ child level route.

Eg. gaana → Artist → Top level route.
Arjit singh → child level route.

/home/aboutus
↓ parent ↓ child.


Home MyProp. profile contact Logout.
<outlet>
Footer. → copyright@sunbeam.

Home.

- After outlet, about-us is now loaded inside Home.
Advantage → No need to repeat.

so keep Navbar inside Home.
so no need to repeat Navbar again.

Now change paths.

to = '/home/about-us'

Now, not copying Navbar again multiple Time.

soln - Nested Routing.

HTML 5 → main → element.
- same like div.
- semantic (specific)

- footer added.

index.css. → footer →

position: absolute → will scroll as page scroll
Fixed → fix.

Line Height : 2em ↘ emphasis.
↘ For vertically centre.

Now, Top Bar & Bottom will be fixed.

getAPI →

```
export async fn getProfile() {
try {
    const url =
    const response = await. - - - -   headers : token.
    const token = session storage.

}
```

token necessary ←

* In JS by default if we not return anything it return undefined.

Profile.jsx

```
fn profile() {

const [firstName, set - - -] = useState(' ')

const onGetProfile = async () => {

    const result = await getProfile()
    if (!result) {  }

        else {
            if () {    }
            else () {  }

}
```

→ we need UseEffect Here.

① Mounting
② Demounting.
③ state change
④ state change bcz of member

- check in Query, if it is array ↓ direct objects. → Before 0 th position chya aat hota sagla
Here it was data. ← we made users [ ]

object destructure → so can access separately
- lastname, address

```
const { first - - - - - }
    setFirstName = firstName.
        "
        "
```

- readonly
- Disable Email → As Email should not change.

- update profile →
- url, body, { } ek add honr fakt.

```
const onUpdate = async () => {
    //validations here.

}
```

```
<button onClick = {onupdate}
```

- onchange was missing
*? Add it

* ToastContainer was missing.
Remember. ↓ otherwise nothing runs.

Q. prblm → unless user register login
Protected APIs
Token should not be asked.
Here it was saying Token is missing.

protected Route.

prblm → Even without login user can visit about us profile even if nothing work still he should not be able to get inside website details.

soln → Authorization of reg. & login la token nhi paje.

Q. How to share data among multiple components?
→ Context API
Redux.

- User login successfully chi status store krycha. mg status check krycha Home. nhi App component made login.

soln → context API.

context API
Demo
App. Component.
comp 1
comp 2.

sharing Data in components.

Eg. counter Eg.

```
Increment.
onClick = {
    //logic.
}

<button onclick = { Increment }

return (
    divstyle = { {         } }
```

JS ↗ ↖ object.

comp 1
comp 2 > Not showed.
separate chalt ahe.

prblm → aplyala. common counter pahije.

soln → context API
↓
it is a built in feature of react used to share data b/w multiple components.

redux → external library
context → Built in Feature.

Terminologies

- context → instance of Context component used to provide (share) data.
↘ to create a context use create Context() Function.

// create a context to share data
↓
outside App as we want to share it to all.

- 1st time variable we use uppercase. bcz afterwards will use as tag.

```
export const CounterContext = createContext()
```

- Wrap components inside counter Context

```
<CounterContext>
    // Component wrap
<CounterContext>

fn. App() {
    //state to maintain counter.
    const [ counter, setCounter ] = UseState[0]

<CounterContext value = { { counter, setcounter} } >
```

JS ↙ ↘ obj.

- import.

```
const { counter, set Counter } = UseContext
                                 (CounterContext)
```

↓ object deference white. value. krla lagle aste.

↓ ata hyne prblm solve zala.
common counter create zala.

- To use context in child components,
use Context() react Hook.

- Syntax in Notes

Use cases : ① To share simple values like
- login status.
- themes used in application.

Disadvantage → need to pay more attention to components.
↓
comp 3 → if accidentally placed outside
↓ code will not run.
since comp. is not added inside CounterContext.
↓ it is undefined.

redux → global so no issue there.

Airbnb prblm back → even before login. it is allowing us /profile' /aboutus.
- token is Missing

soln → Using context API.
will come to know if user is logged in or not.

- sharing data with App & login component

> contexts. > user.js (theme.js)
↓ can also do this.
auth.context.js.
↘ Convention.

// create context to sharing auth details.
- export const AuthContext = Create context()

- import this in App.

```
const [ user, setUser ] = useState
                          (null).
```
↘ as no user rn so null

```
return (
    <AuthContext value = {{ user, setUser}}
        //All routes inside
    < AuthContext >
```

↓ he saglya routes na bhetnr.

In login,
// set the user details in AuthContext.
setUser({ first, Last.
})

// get the setuser from Auth.
→ const {setUser} = useContext (AuthContext)

protected route

element = { User. ? Home/ : Navigate to Login.

→ component from react router.

In NavBar, Resetting
reset userdetails in AuthContext.
↘ setUser (null)

get the set user
→ const { setUser } =
useContext (AuthContext)

Nested Route Advantages
① No Need to Duplicate
② parent level - protected Route.

* we can give toast by creating component & giving warning.
- we can do fine Tuning.

Shub doubt

Redux → wishlist / Favorite.

Dropdown - categories.

```
fn AddProperty () {
const [ categories, setCategories ]

const getcategories =
            await getCategories
```

→ same name
so use alias.
get categories from server.

↓
// get categories as this component loads
↓
UseEffect.

```
UseEffect() {

}

return (
    // column & rows

)
```

Add prop. jsx

prop. details → 10000

textarea → ▭
↙ multilines.
↓ like. Address

margin bottom → space sodun disala.

// Here we are creating form to fill to add property

checkboxes.

itenary can. → 1 row — 5 columns. ✗
☐ ☐ ☐ ☐ ☐
1 col → 5 rows ✓

Now Image
- Type - File.
width - 100 %.
margin Top →
↓ image preview.

* Form control
↓
Horizontal Form.
↓
type- checkbox

not class ✗
className ✓

Below this form, <hr>

2 buttons
↙ save ↘ cancel.

ms-2
↓ margin extra.

Here, 18 variable.
will not create 18 state members. ✗
↓ 1 state mem. inside 18 ✓ & another state mem for img.
will put in one single object.

```
const [info, setInfo] = use state ({ //object chy aat.

    categoryId : -1,
    |
    |
    18

})
```

rest operator ↑

```
select
onChange = { e ⇒ setInfo({ ...info,
            categoryid : e.target.value }) }
```

↓ overrided.

```
isLakeview : e.target.checked ? 1 : 0
```

↓ as this is checkBox.

For Image :
separate state member. & 3rd state mem. → For use effect.

```
onChange = {(e) ⇒ setImage
            e.target.files.
```

```
UseEffect () ⇒ {

    set PHOTO. = URL.createObjectURL (image))   → this for preview.

        , [image]}
    src = {photo}
```

conditional rendering
→ { image &&

// code

}

```
on Cancel ⇒ navigate(-1)
```

footer → add padding so buttons come up.

* you can't send image using json.
✗ raw → JSON. For image.
✓ form data ↓

icon → File

> property.js →

```
export async fn. upload property.(

    18 - - -

    ) {

    try {   //sending Data using FormData.
        const body = new FormData ()
        body.append ("categoryid", categoryid)
                     ↓ Key           ↓ value
                  = 'photo',     image )

        const token = sessionstorage.
        const response - - - - -

    }
    catch (ex) {

    }
```

To get Token. ←

putting all 18 → into info.

card → Bootstrap.

- we created property.js → using same for All property My property

will just add delete button. in my property.

For property Details
↓
go to prop. details screen along with selected property (state)

```
navigate ('/home/property-details', {state: property })
```

* Dynamic Navigation with Data

see in sir Notes.

As soon as component gets loaded it should happen
so UseEffect()

UseLocation. → react router DOM.

location.state

```
{ details && (
    // rendering.

    )
}
```

→ only if details are there then render.

details = null before. so →

Bootstrap- icons → external library.
↘ CSS
npm i bootstrap-icons.

React bootstrap icons. → React wala.

npm i react bootstrap-icons.