Handling Unmatched Routes in Next.js

Unmatched Routes:

In Next.js, "unmatched routes" refers to routes that do not match any of the defined routes in your application. Next.js provides a way to handle these unmatched routes through its built-in routing system.

When a user tries to access a route that is not explicitly defined in your Next.js application, Next.js provides a way to handle this situation by allowing you to define a custom 404 page or handle the unmatched route in any way you see fit.

default.tsx:

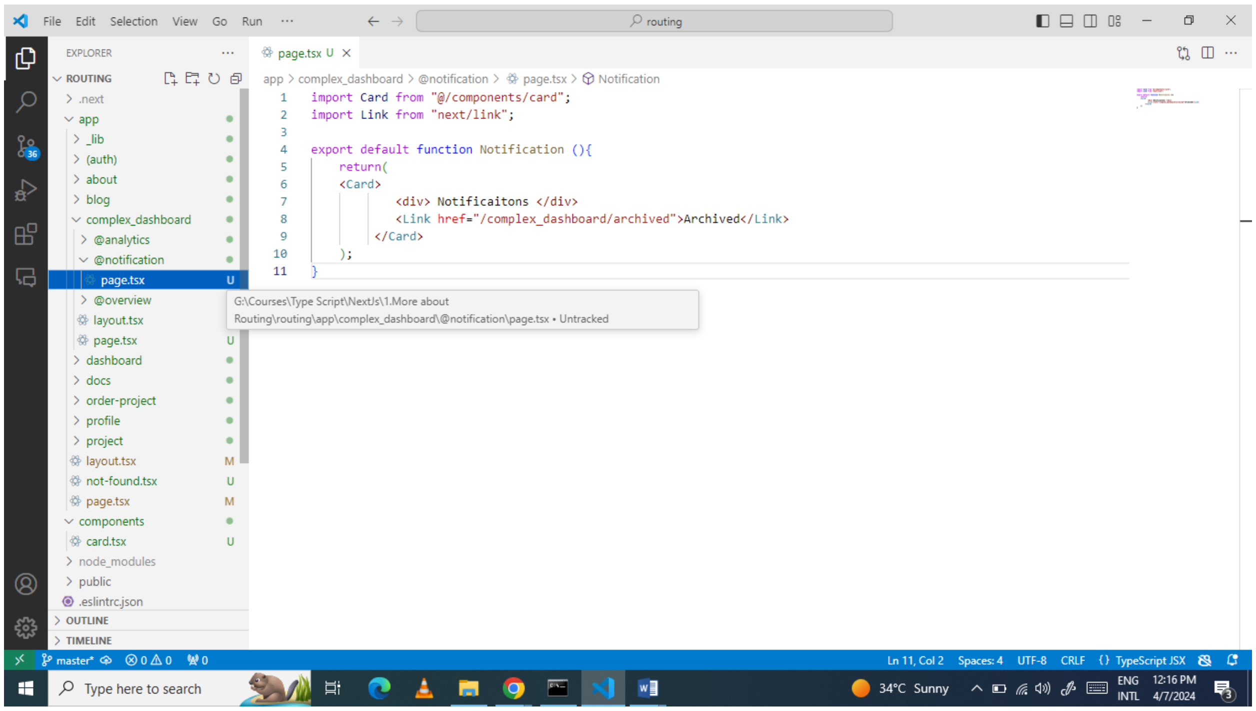
In Next.js, the `default.tsx` file serves as a fallback component that Next.js will render when it cannot find a specific component to render for a given route.

When Next.js encounters an unmatched route, meaning there's no specific component defined for that route, it will look for a `default.tsx` file within the directory corresponding to that route. If the `default.tsx` file exists, Next.js will render its content as the fallback for that route. This allows developers to provide a default UI or content for routes that don't have a specific component assigned to them.

The `default.tsx` file is particularly useful for handling unmatched routes and ensuring that users don't encounter a 404 error when navigating to those routes. It provides flexibility in managing the UI and content for different routes within a Next.js application.

Implementation of “unmatched routes” and “default.tsx file” in Next.js.

Open the `page.tsx` file inside the `@notification` folder and update it as per the screenshots below.



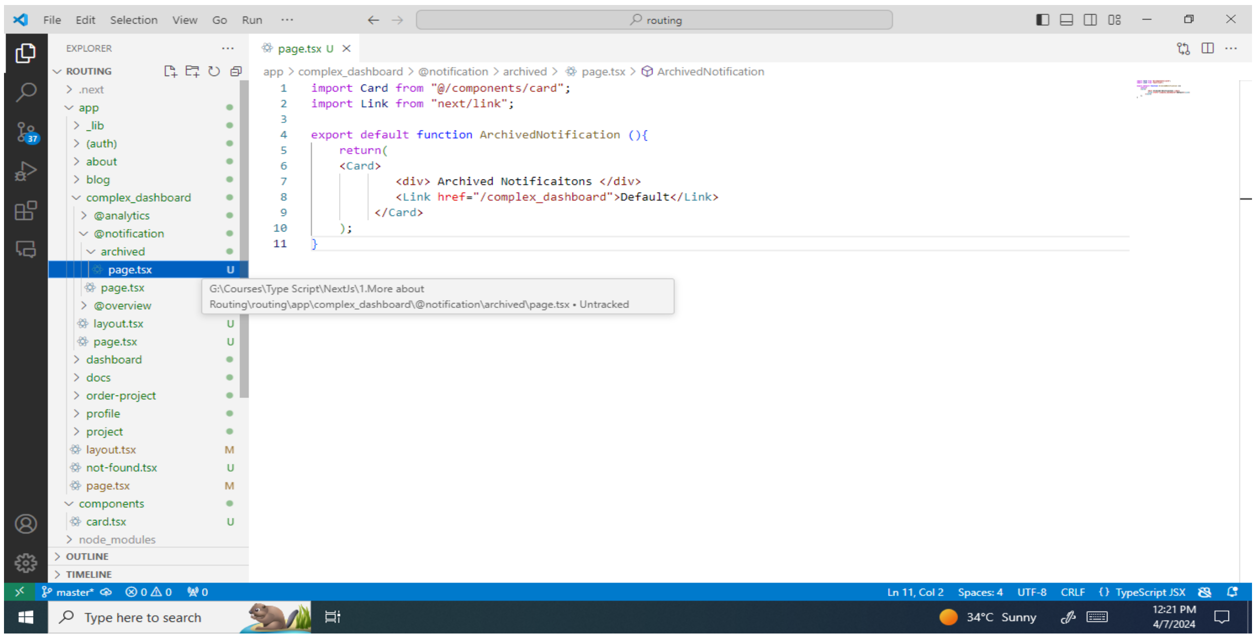
app > complex_dashboard > @notification > page.tsx > Notification

```tsx
import Card from "@/components/card";
import Link from "next/link";

export default function Notification (){
    return(
    <Card>
            <div> Notificaitons </div>
            <Link href="/complex_dashboard/archived">Archived</Link>
        </Card>
    );
}
```

G:\Courses\Type Script\NextJs\1.More about Routing\routing\app\complex_dashboard\@notification\page.tsx • Untracked

Inside the notification slot (@notification) creat a new folder named archived and inside archived folder(this is regular folder not a slot) create page.tsx file and update as per below screen shot.



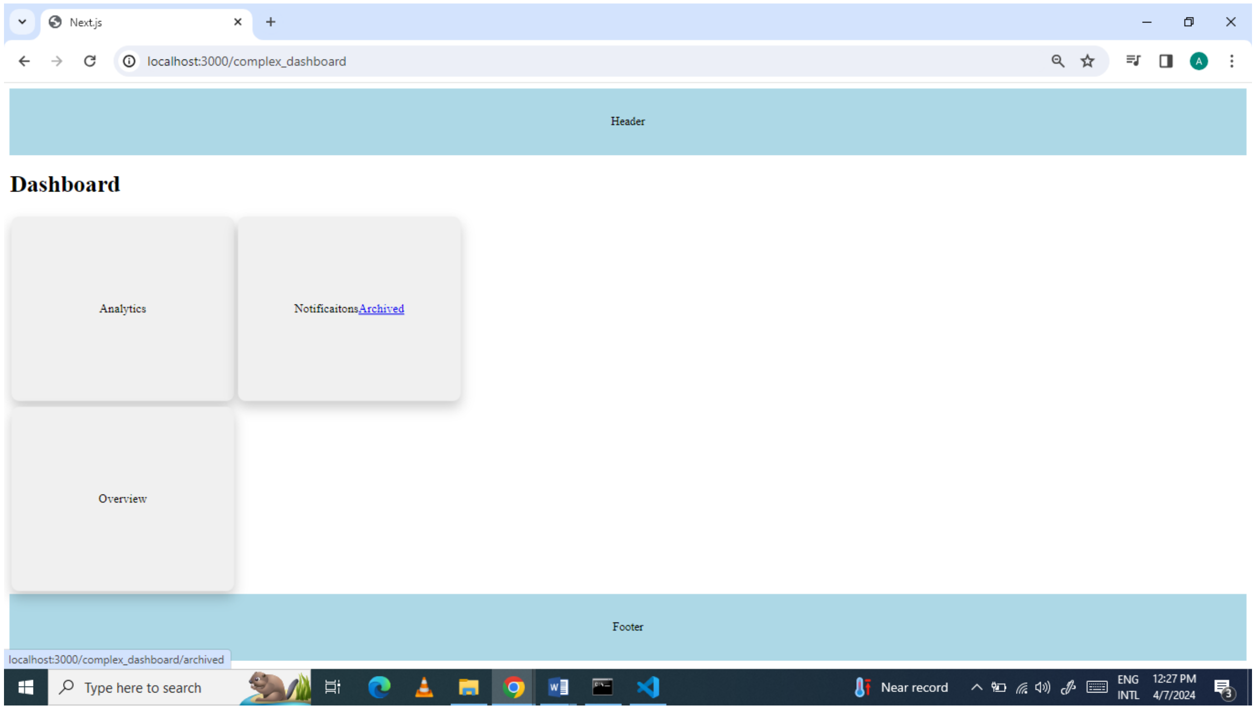
app > complex_dashboard > @notification > archived > page.tsx > ArchivedNotification

```tsx
import Card from "@/components/card";
import Link from "next/link";

export default function ArchivedNotification (){
    return(
    <Card>
            <div> Archived Notificaitons </div>
            <Link href="/complex_dashboard">Default</Link>
        </Card>
    );
}
```

G:\Courses\Type Script\NextJs\1.More about Routing\routing\app\complex_dashboard\@notification\archived\page.tsx • Untracked

"Go to the browser and navigate to http://localhost:3000/complex_dashboard to see the dashboard."



Header

Dashboard

Analytics

NotificaitonsArchived

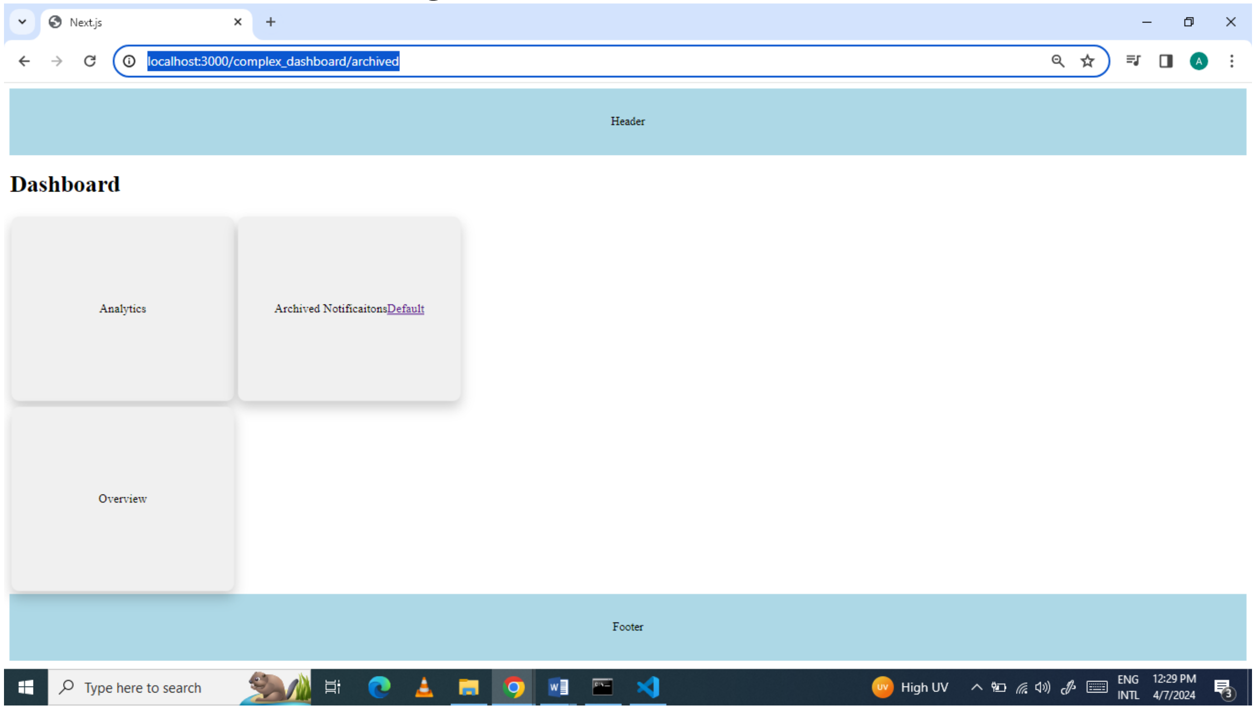
Overview

Footer

localhost:3000/complex_dashboard/archived

If you click on Archived link then it will be navigated to http://localhost:3000/complex_dashboard/archived

"Go to the browser and navigate to see the dashboard."



Header

Dashboard

Analytics

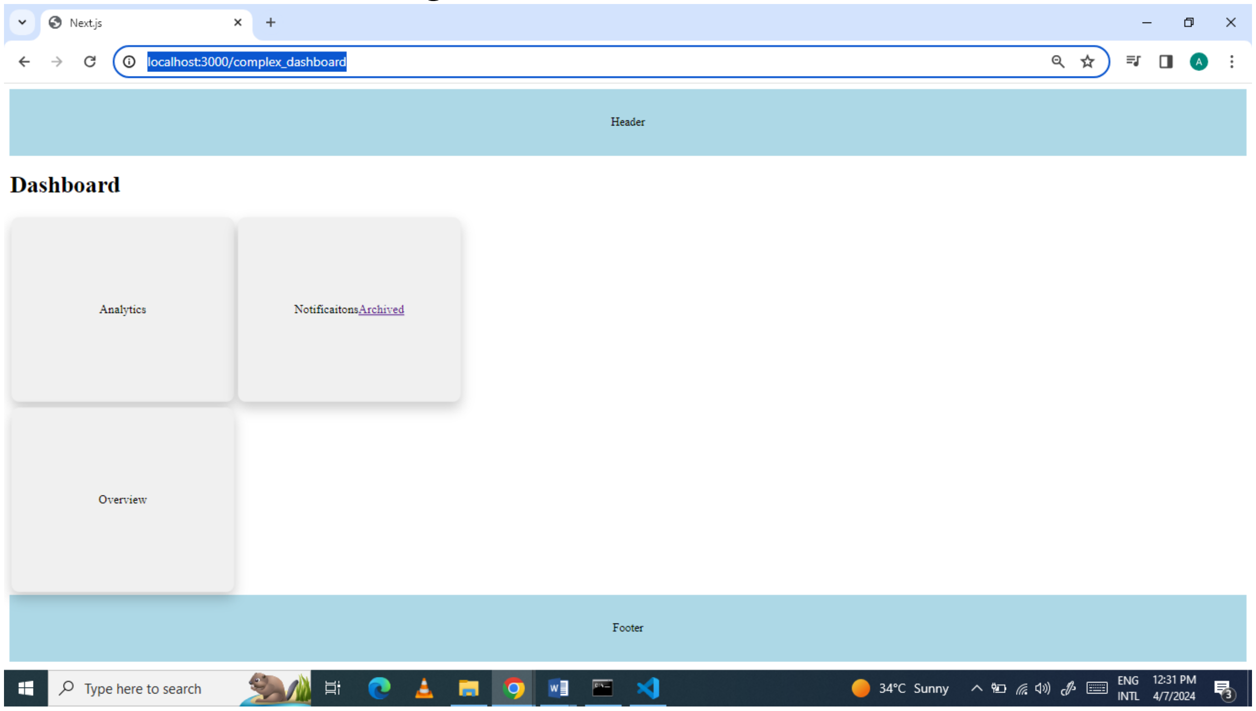
Archived NotificaitonsDefault

Overview

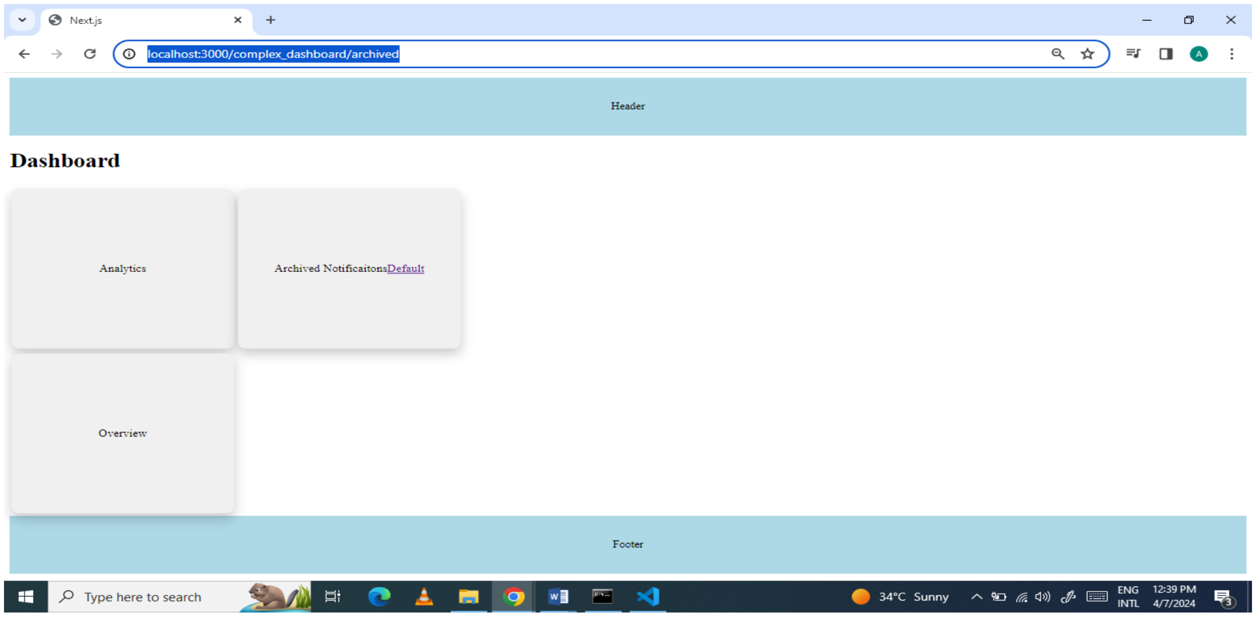
Footer

If you click on Default link then it will be navigated to http://localhost:3000/complex_dashboard

"Go to the browser and navigate to see the dashboard."

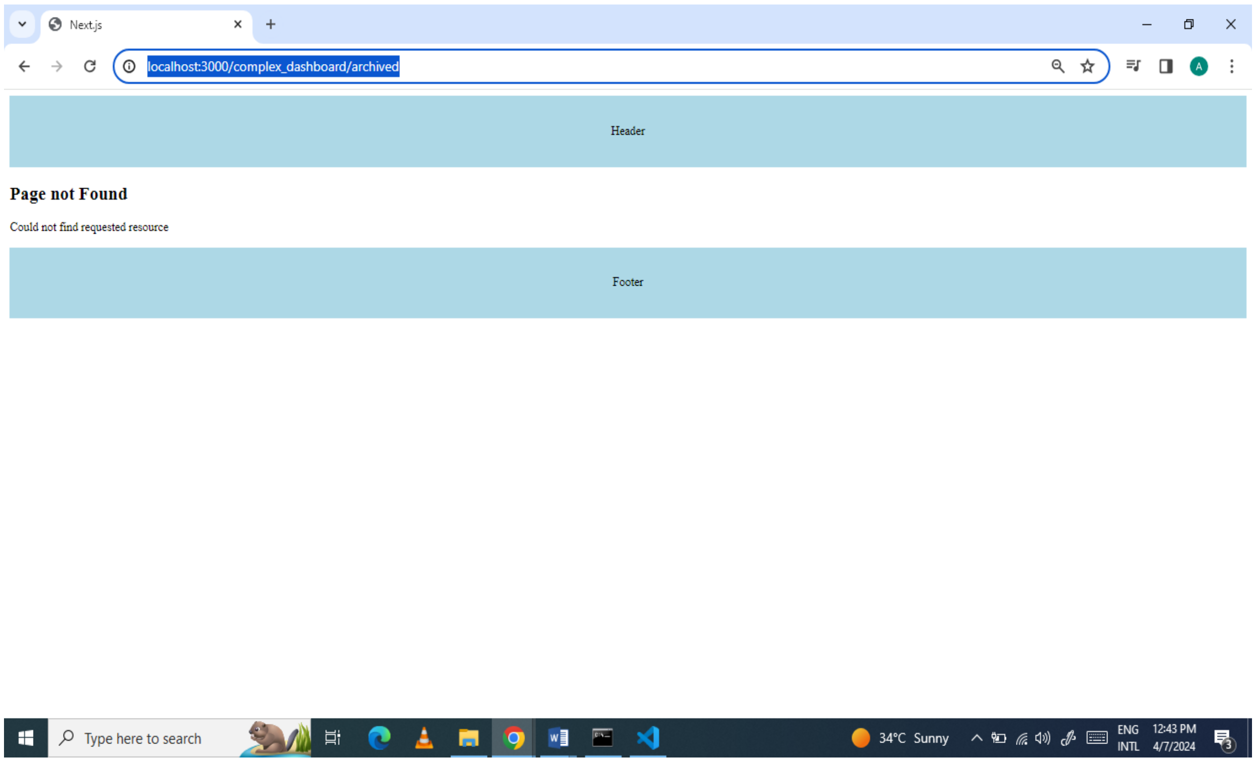


Usually, what you see in a slot matches the web address you're on. In our dashboard, we've got four slots: "children", "analytics", "overview", and "notifications". They all show their stuff when you go to `localhost:3000/complex_dashboard`. But if you go to `localhost:3000/complex_dashboard/archived`, only the "notifications" slot shows something. The other three slots, "children", "overview", and "analytics", don't have anything for that address.



These slots will keep showing the same content they had before, regardless of any changes in the URL path, such as moving from `/complex_dashboard` to `/complex_dashboard/archived`, or vice versa. When you reload the page, Next.js looks for a `default.tsx` file in each slot that doesn't have a matching route. This file is crucial because it provides the default content that Next.js will display in the user interface. If there's no `default.tsx` file in any of these slots for the current route, Next.js will show a 404 error. For instance, if you navigate to `/archived` and refresh the page, you'll see a "Page Not Found" error because there's no `default.tsx` file in the "children", "users", or "revenue" slots. Without this file, Next.js can't determine what content to display in these slots when the page initially loads.

"Go to the browser and navigate to http://localhost:3000/complex_dashboard/archived and refresh the browser and you'll see a "Page Not Found" error."



Header

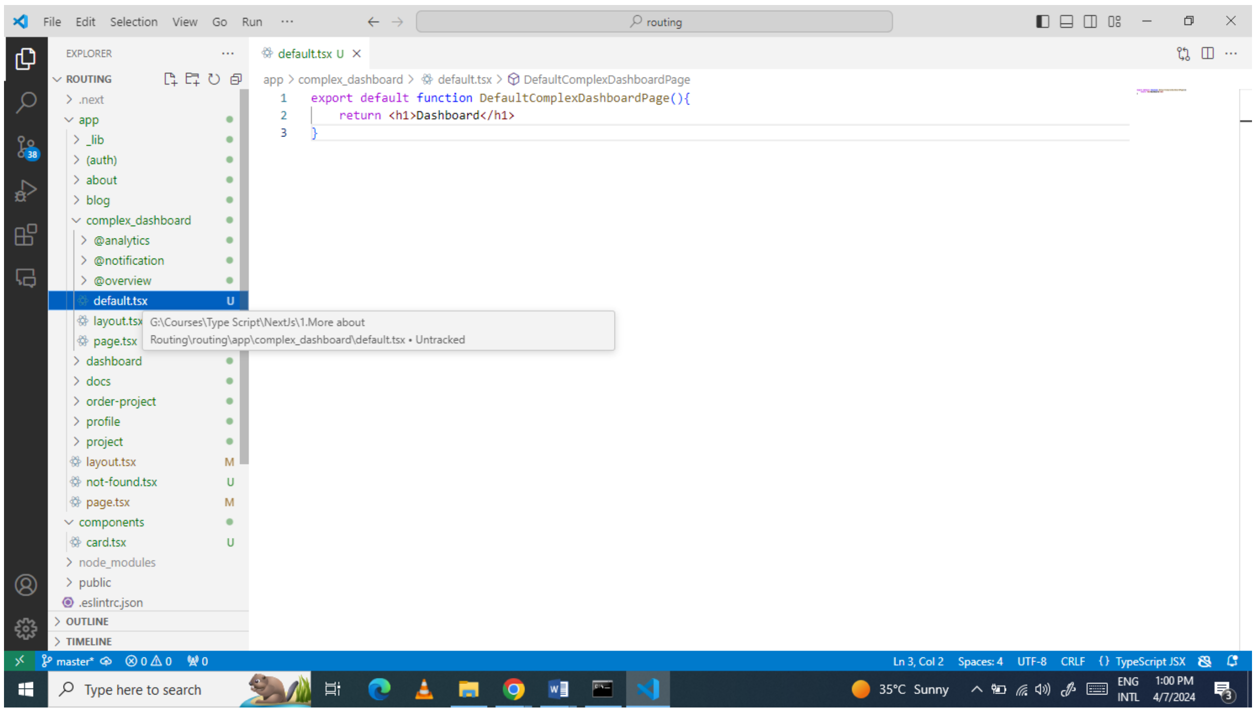
Page not Found

Could not find requested resource

Footer

In our complex dashboard layout, we have four sections: "children", "analytics", "overview", and "notifications". Among these, only the "notifications" section has a specific component for the archived route. To avoid 404 errors for the other sections when accessing this route, we must create fallback views. These fallback views should be placed at the same level in the directory structure as `page.tsx`.

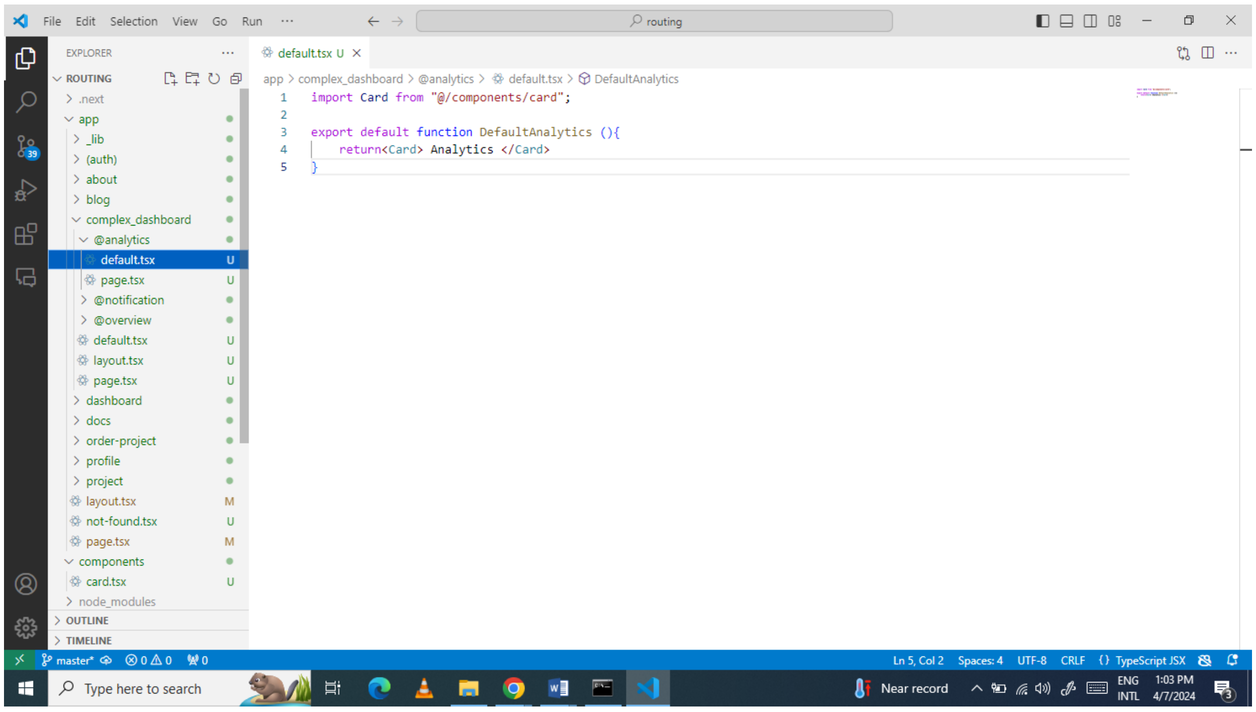
Create a `default.tsx` file within the `complex_dashboard` folder. This file will serve as a backup view for the "children" slot. Please update the component name accordingly, as shown in the screenshots below.


default.tsx U
app > complex_dashboard > default.tsx > DefaultComplexDashboardPage
Ln 3, Col 2 Spaces: 4 UTF-8 CRLF TypeScript JSX

```tsx
export default function DefaultComplexDashboardPage(){
    return <h1>Dashboard</h1>
}
```

G:\Courses\Type Script\NextJs\1.More about Routing\routing\app\complex_dashboard\default.tsx • Untracked

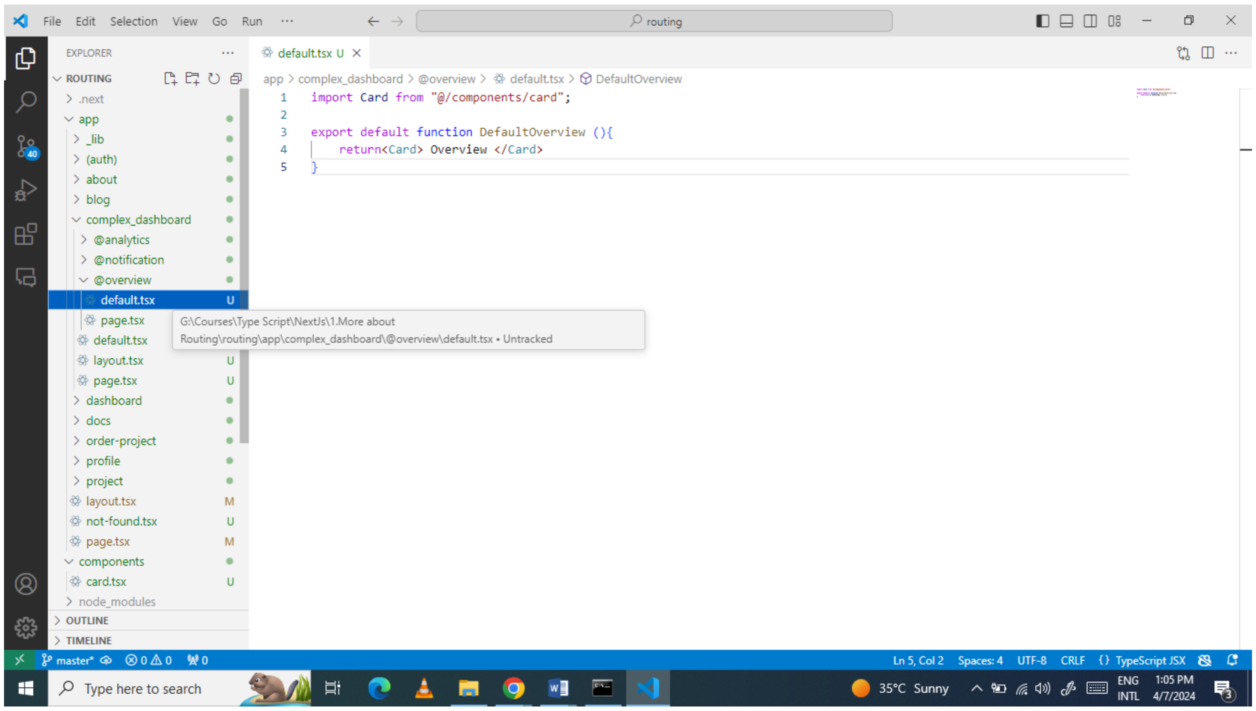
Next, in the "analytics" slot, we'll also create a `default.tsx` file. Please update the component name accordingly, as shown in the screenshots below.


default.tsx U
app > complex_dashboard > @analytics > default.tsx > DefaultAnalytics
Ln 5, Col 2 Spaces: 4 UTF-8 CRLF TypeScript JSX

```tsx
import Card from "@/components/card";

export default function DefaultAnalytics (){
    return<Card> Analytics </Card>
}
```

Similarly, within the "overview" slot folder, let's define `default.tsx`. Please update the component name accordingly, as shown in the screenshots below.



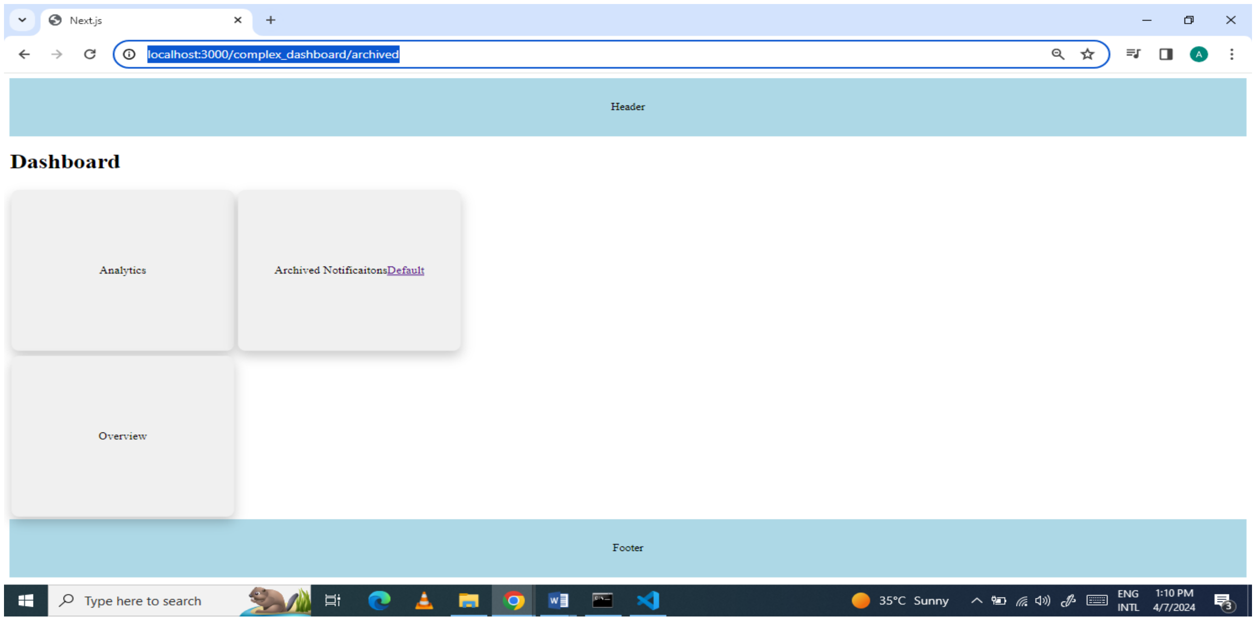
```
import Card from "@/components/card";

export default function DefaultOverview (){
    return<Card> Overview </Card>
}
```

G:\Courses\Type Script\NextJs\1.More about Routing\routing\app\complex_dashboard\@overview\default.tsx • Untracked

In your web browser, if you visit the archived route and refresh the page, it will load properly instead of displaying a 404 error. The notification slot will display its unique content from the "archived" sub-folder because it's the only slot with a component defined for the /archived route. However, the other three slots ("children", "analytics", and "overview") will show content from their respective `default.tsx` files. These files serve as backups for routes without specific content, preventing unintended rendering of routes that shouldn't be parallel-rendered.

"Go to the browser and navigate to http://localhost:3000/complex_dashboard/archived and refresh the browser and it will load properly instead of displaying a 404 error"



In parallel routing, each slot usually shows content related to the current URL. But if a slot doesn't have specific content for that URL, what happens next depends on how you got there. If you clicked a link or button to navigate, the slot will show the last content you saw. But if you refresh the page, Next.js checks for a `default.tsx` file in the slot. If it can't find one, you'll see a 404 error.