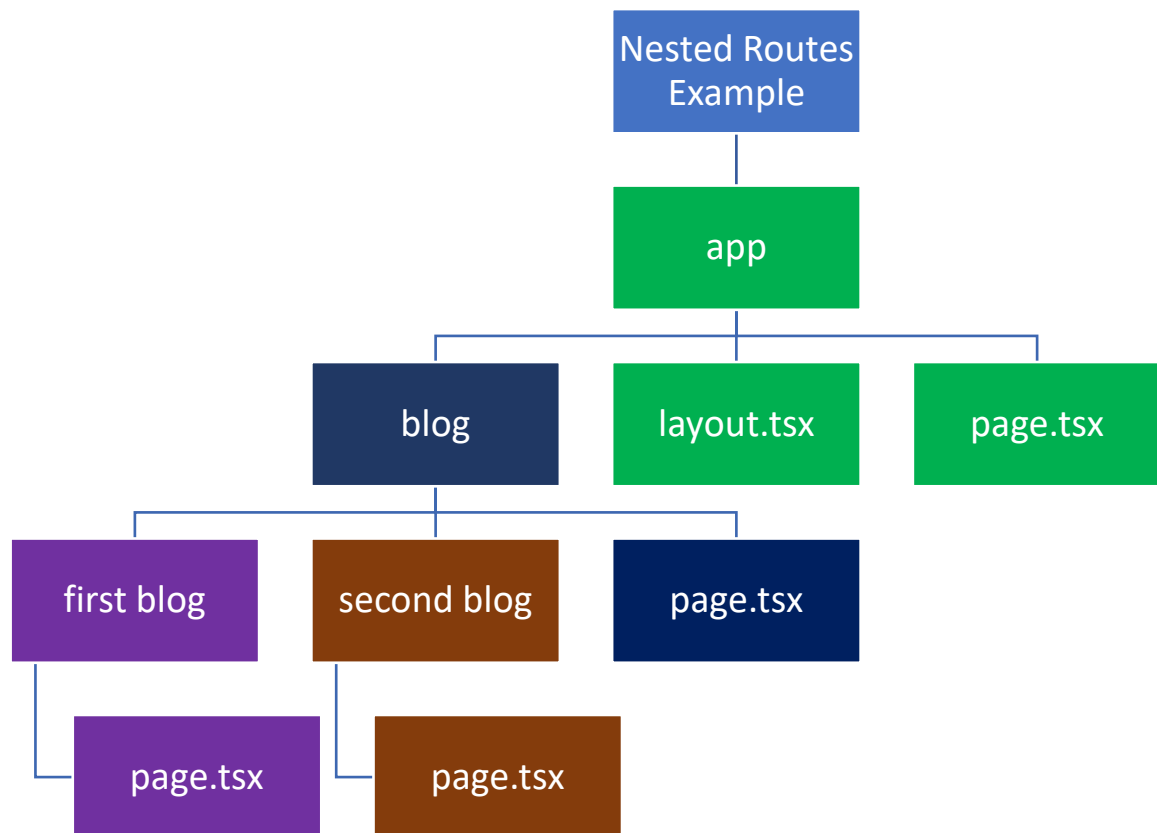


Concept of Nested Routes in Next.js

What is Nested Routes:

Nested routes, refers to the practice of defining routes within routes in a web application. This means that routes are organized hierarchically, with some routes containing or being "nested" within others. Each nested route corresponds to a specific URL path segment, and navigating to a nested route renders the corresponding content or component within the parent route's layout.



Purpose of Nested Routes:

The purpose of nested routing is to create a structured and organized navigation flow within a web application, especially when dealing with complex UIs or multi-level navigation hierarchies. Nested routing allows developers to break down the application into smaller, more manageable sections, making it easier to maintain and extend over time.

In Next.js, nested routing can be achieved by organizing files into subdirectories within the app directory. Each subdirectory represents a nested route, and the files within that subdirectory correspond to the pages or components for that route. When a user navigates to a nested route, Next.js matches the URL to the corresponding file in the subdirectory and renders its content.

Practical Implementation of Nested Routes in Next.js Application

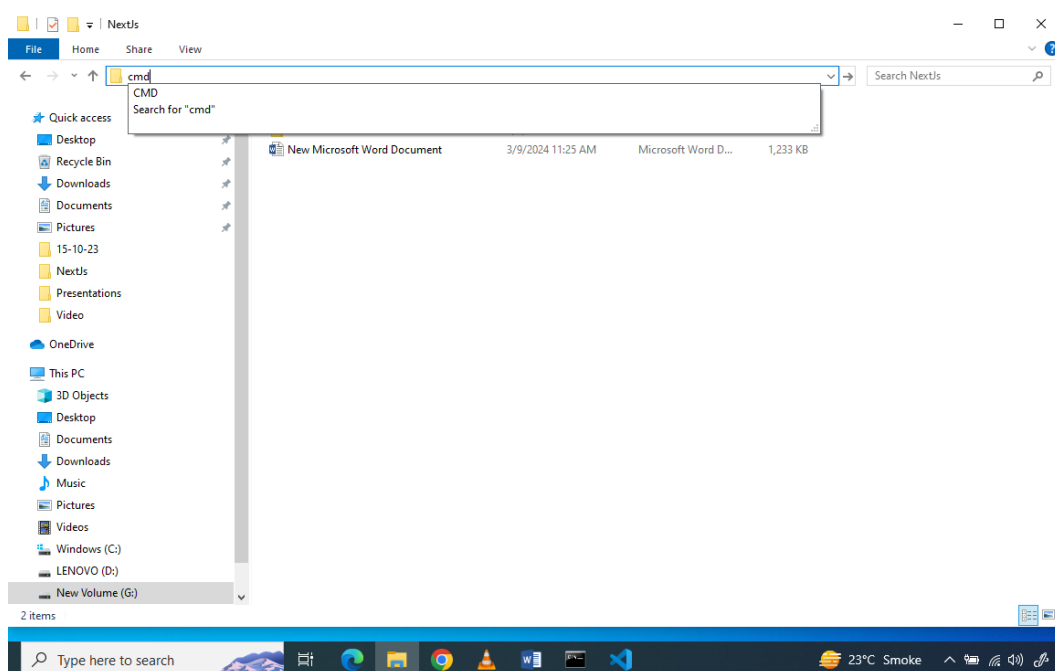
Requirements:

Node.js installed on your system.

A text editor such as Visual Studio Code.

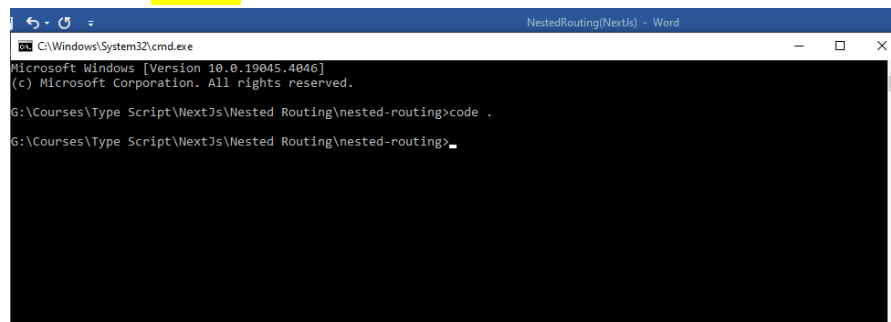
Step 1: Setting Up the Project Environment

Create a new folder for your Next.js project and run the **cmd** command in address bar:



Open Visual Studio Code in this folder:

by run the command **code .**

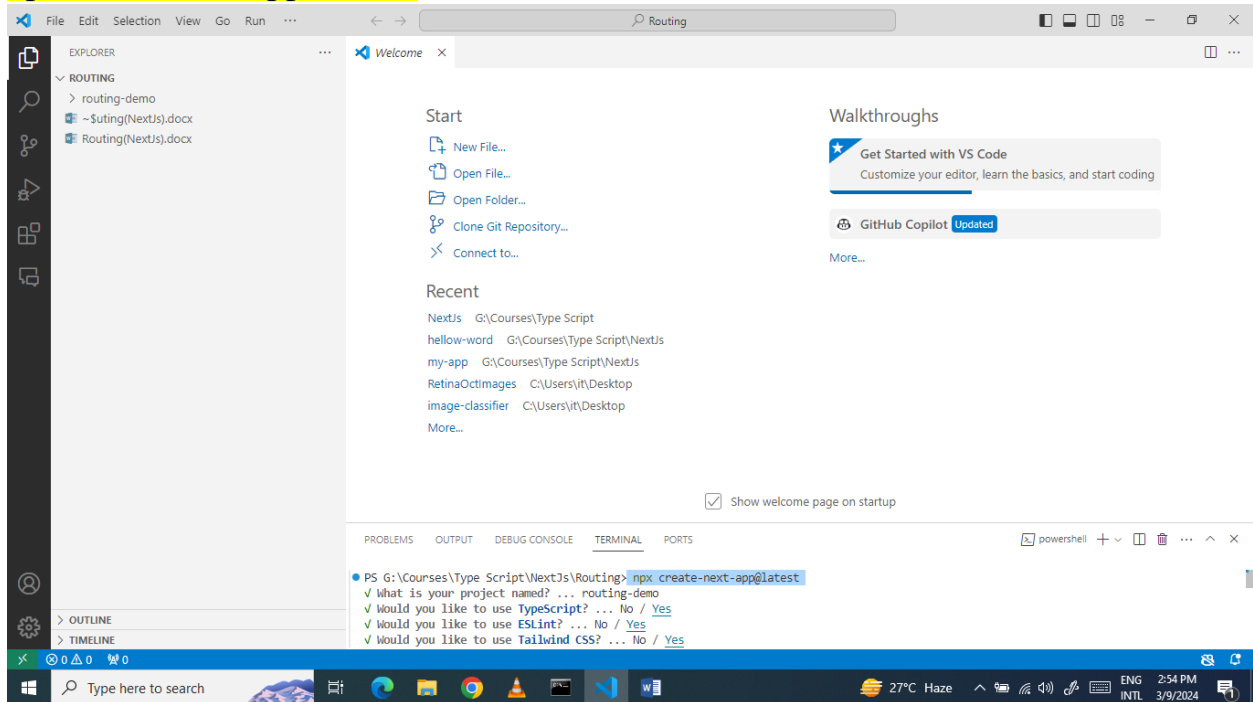


Step 2: Initializing Your Next.js App

With your text editor open and the project folder selected, ensure you have Node.js installed on your system.

In the terminal, run the following command to create a new Next.js app:

npx create-next-app@latest

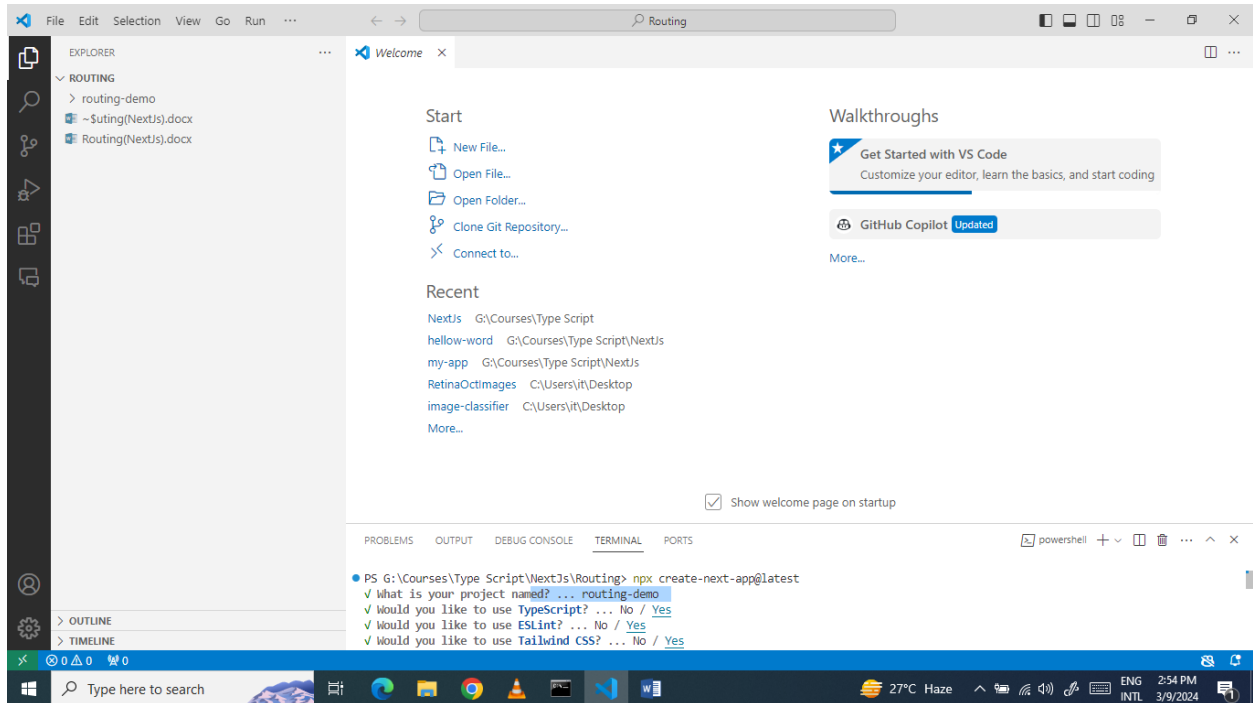


After running the `npx create-next-app@latest` command to set up a new Next.js project, you may encounter a series of prompts to configure your project.

After running the command, you will be prompted to provide the **name of the app**. For the subsequent prompts, you can select the default option for the remaining settings.

Name of the app: You should provide a name for your Next.js application. This name will be used as the directory name for your project. You can choose any name you like, but it's recommended to use a descriptive and relevant name for your project.

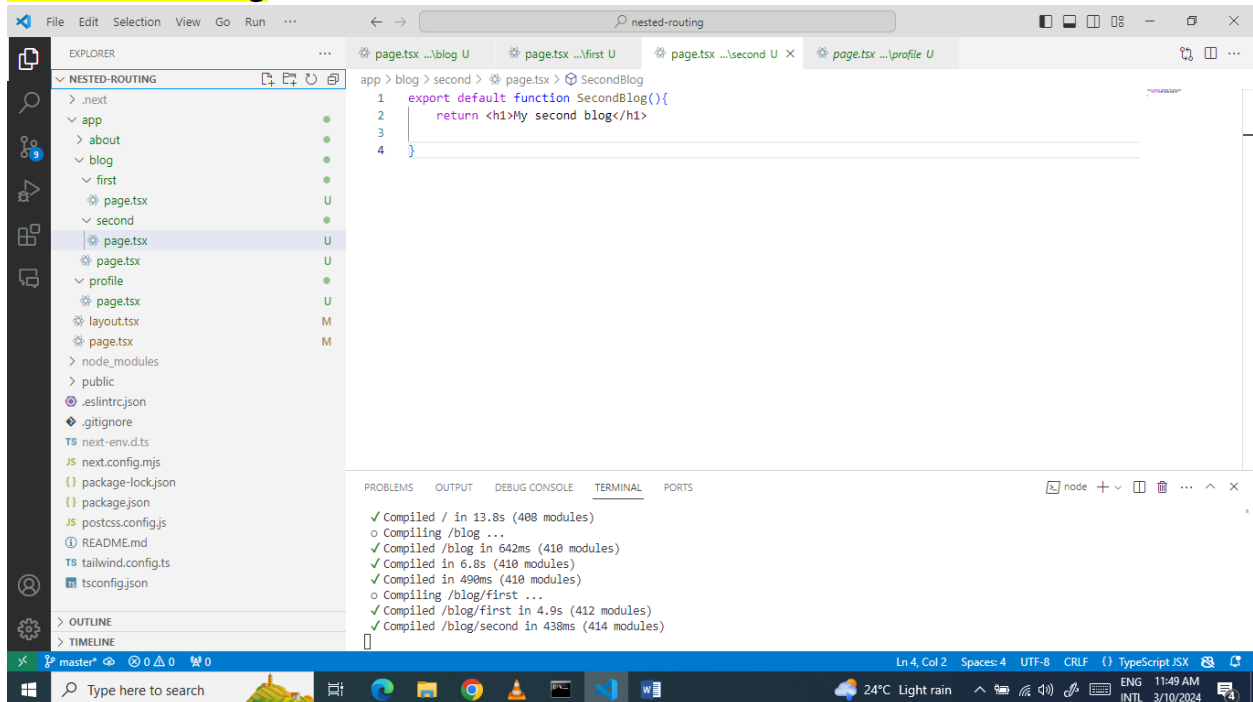
For the subsequent prompts, you can select the default option for the remaining settings.



Step 3: Running Your Next.js App

Once the installation is complete, navigate into your project directory if you're not already there:

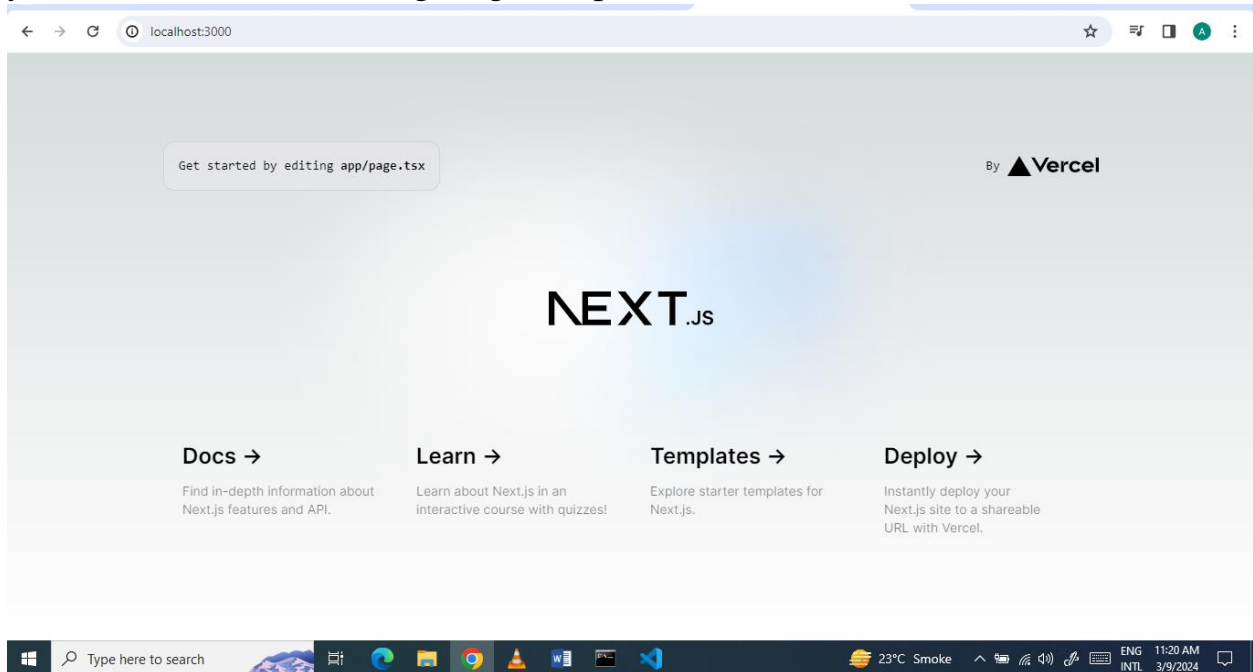
cd nested-routing



Start the development server by running the following command:

```
npm run dev
```

This will launch your Next.js application locally. You can access it by opening your web browser and navigating to <http://localhost:3000>.



Step 4: Exploring Your Next.js App

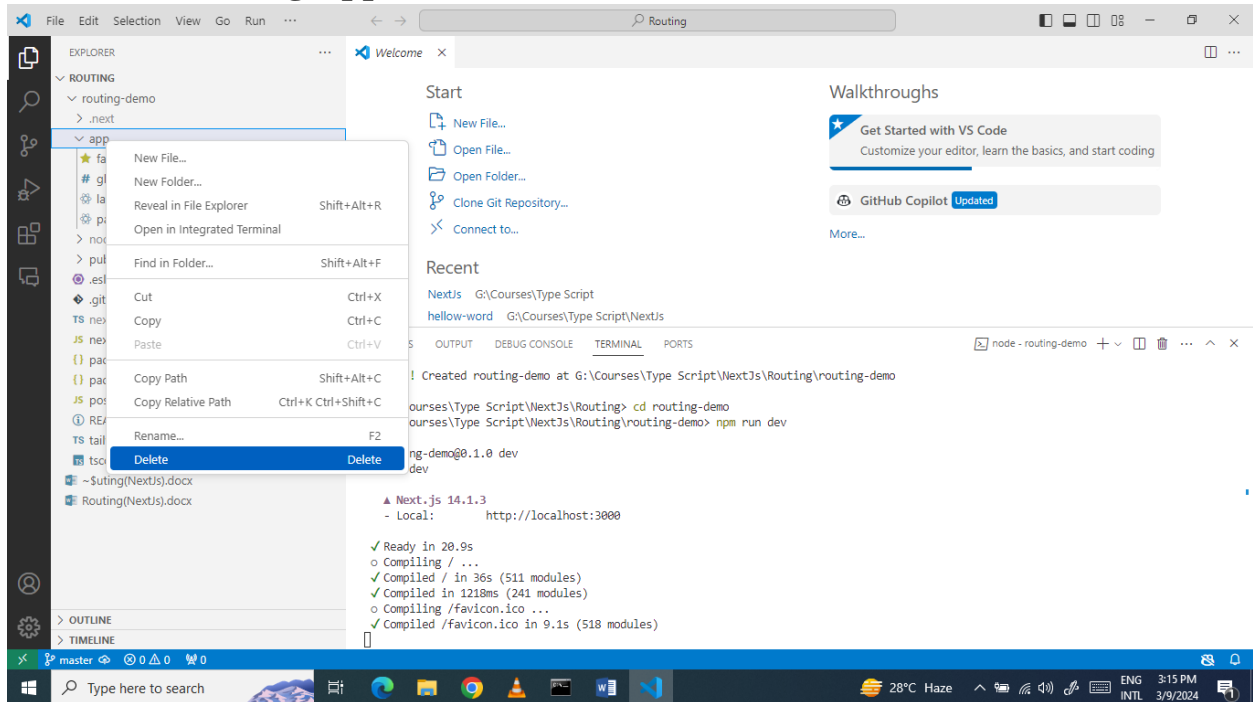
With your Next.js app running, you can start exploring the project structure and making changes to the code.

Open the project folder in Visual Studio Code and begin editing files within the pages directory to create your app's pages and components.

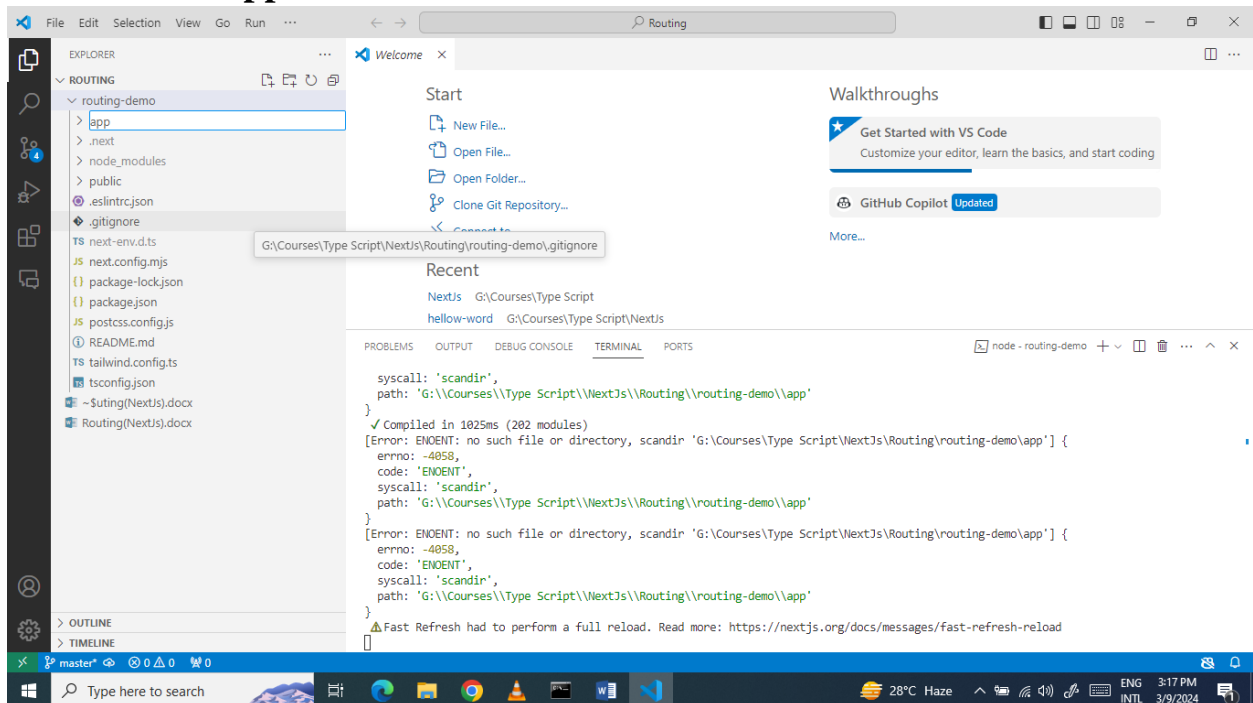
For example

If you need to recreate the app folder from scratch and then create a new page.tsx file, you can follow these steps:

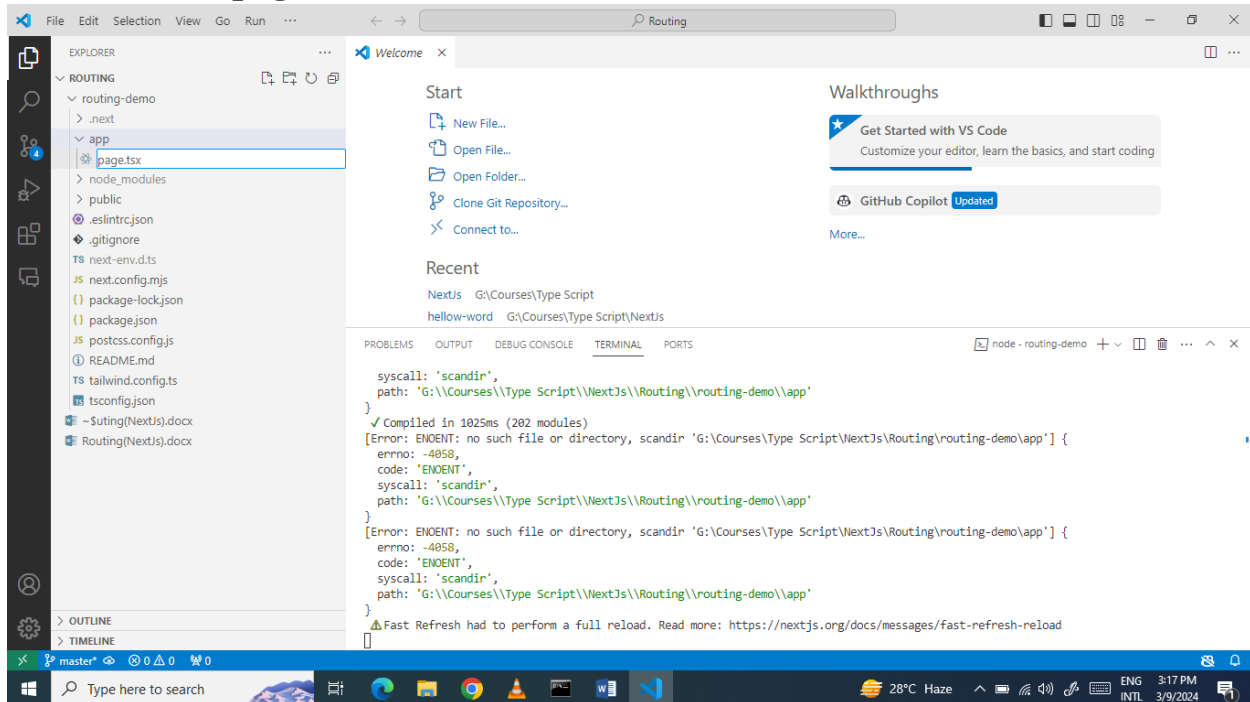
Delete the Existing App Folder:



Recreate the App Folder:



Create a New page.tsx File:

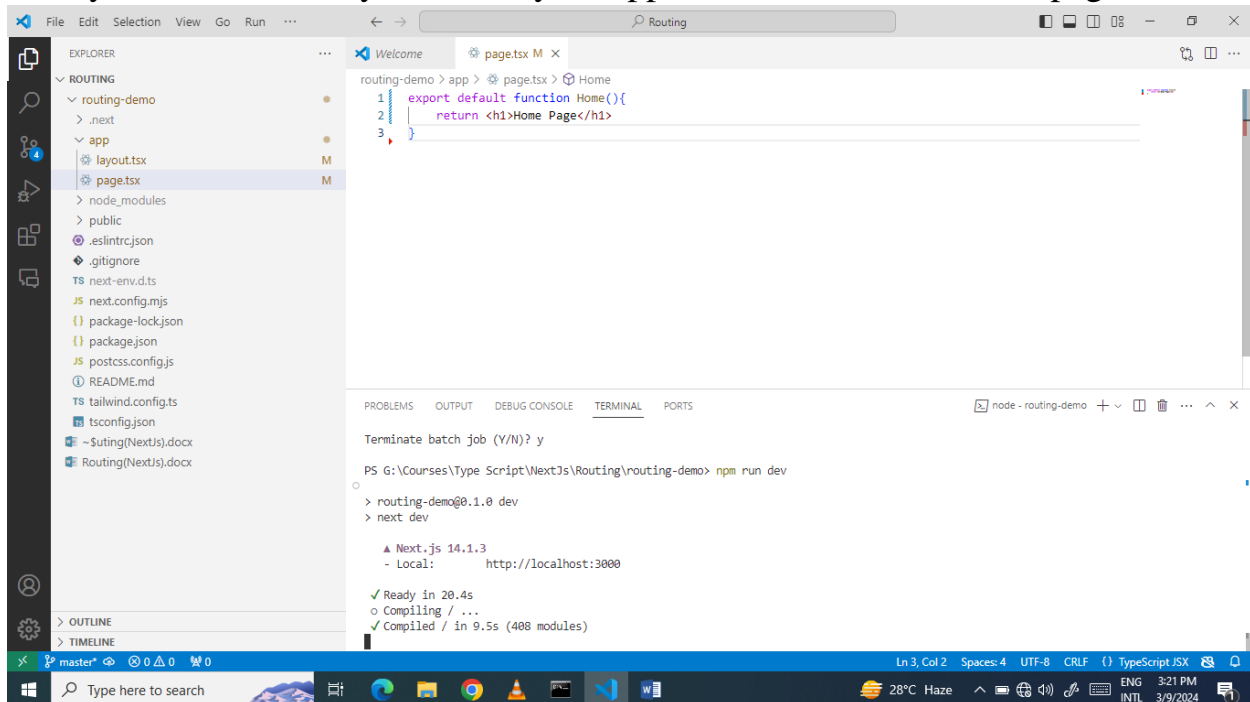


Edit the page.tsx File:

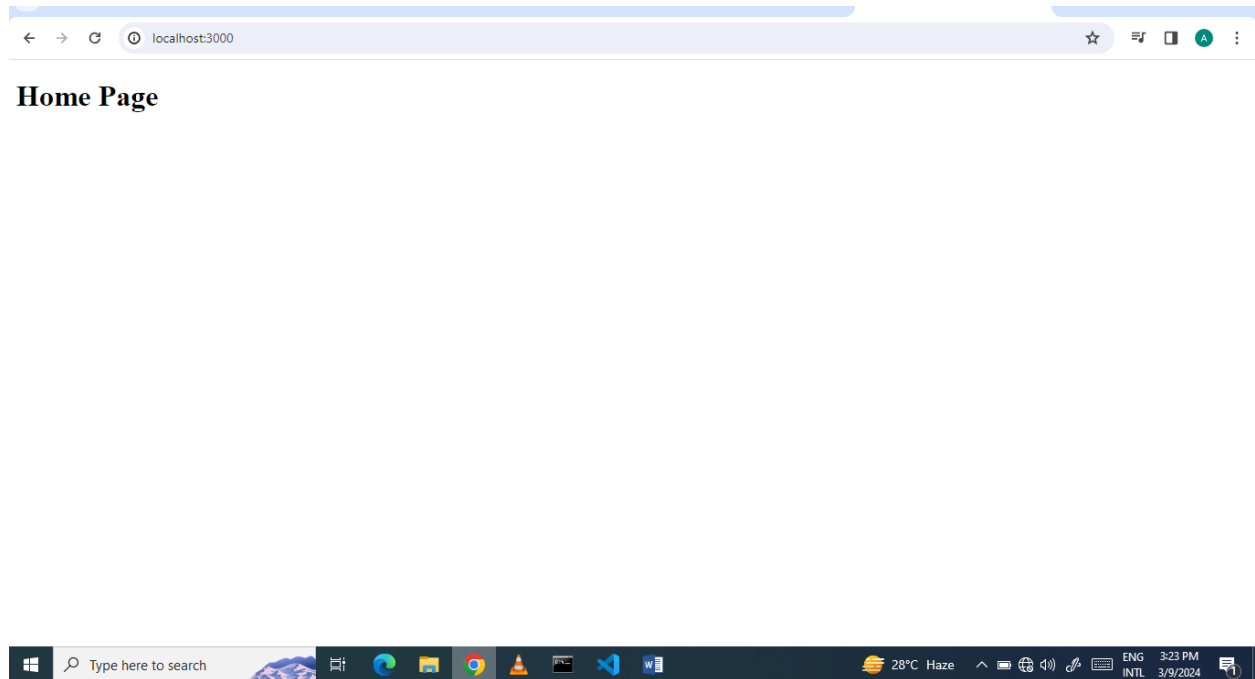
Save Changes:

Save the changes made to the page.tsx file.

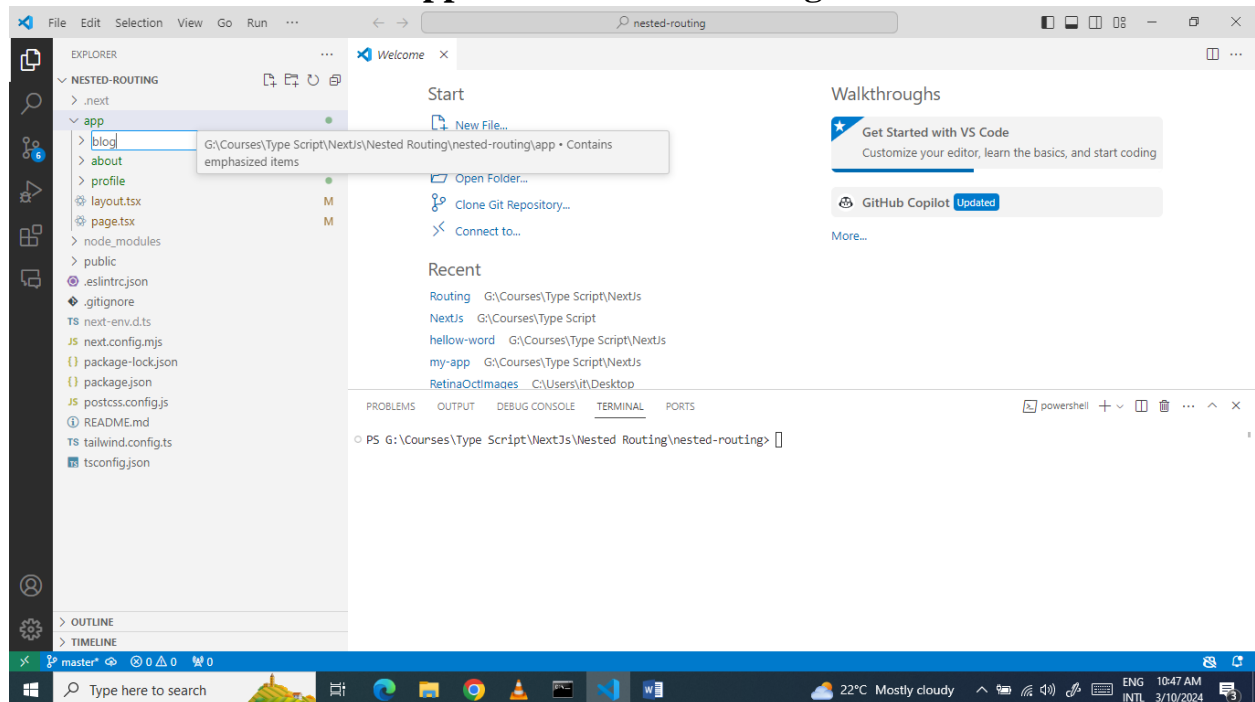
Now you've successfully recreated your app folder and created a new page.tsx file.



Output will be

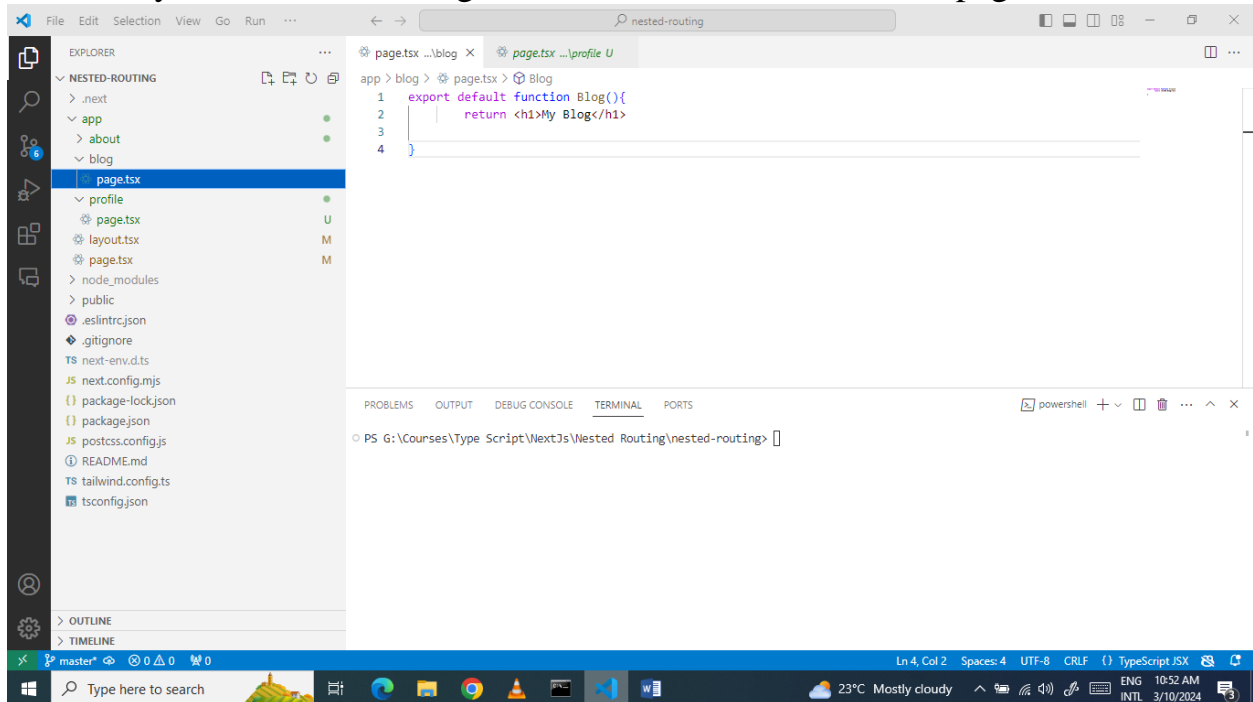


Create the new folder in app folder with name bolg



Create a New page.tsx File in the blog folder:

Now that you're inside the blog folder, create a new file named page.tsx.



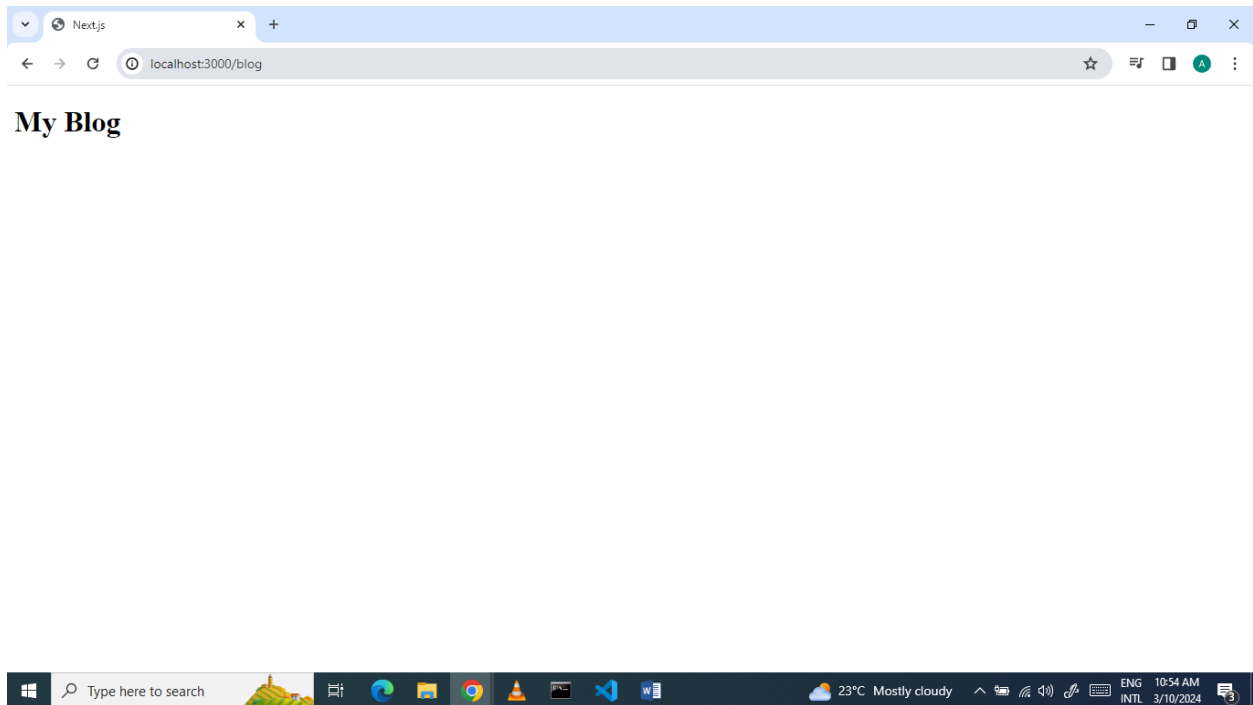
Edit the page.tsx File:

Open the page.tsx file located inside the blog folder in your text editor.
Add the necessary code for your blog page.

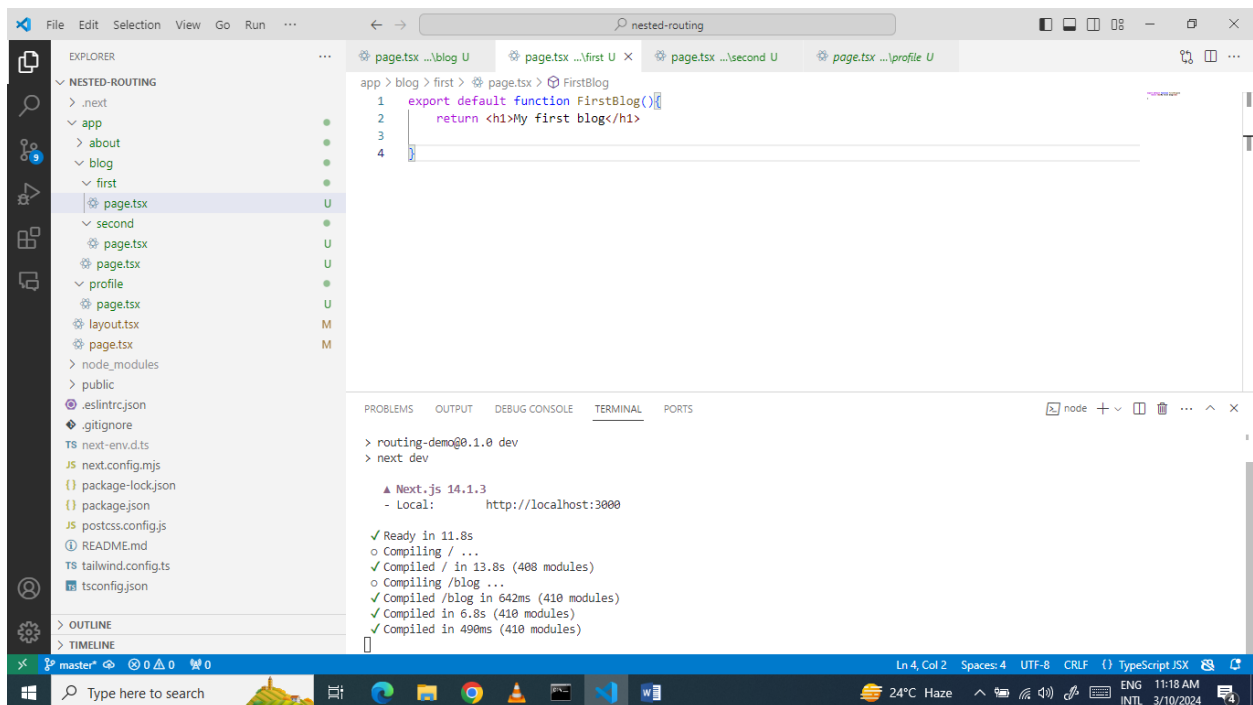
Save Changes:

Save the changes made to the page.tsx file.

Now you have successfully created a new page named blog in your Next.js app.
You can access this page by navigating to /blog in your browser once the development server is running (npm run dev).

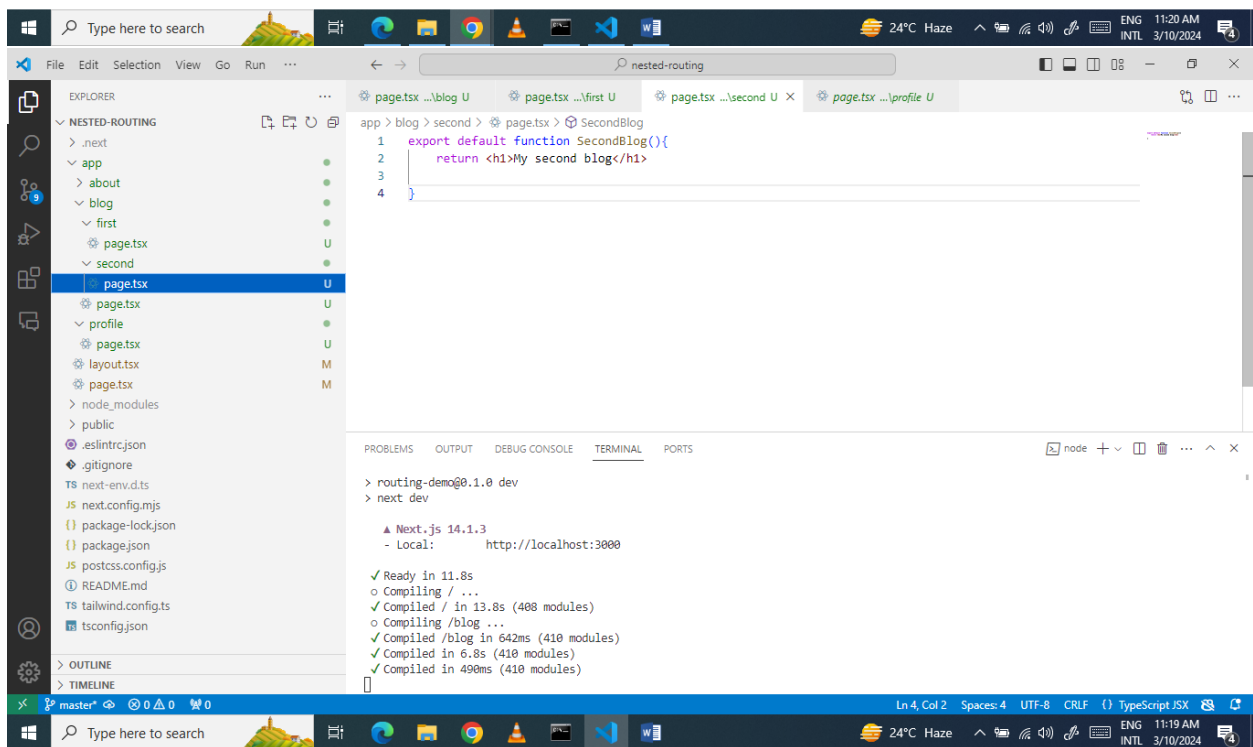


Now create two sub folder inside blog folder name first and second, then create page.tsx file in both folder edit and save it , Now go to the browser and open it one by one by address <http://localhost:3000/blog/first> and <http://localhost:3000/blog/second>





My first blog





My second blog

