

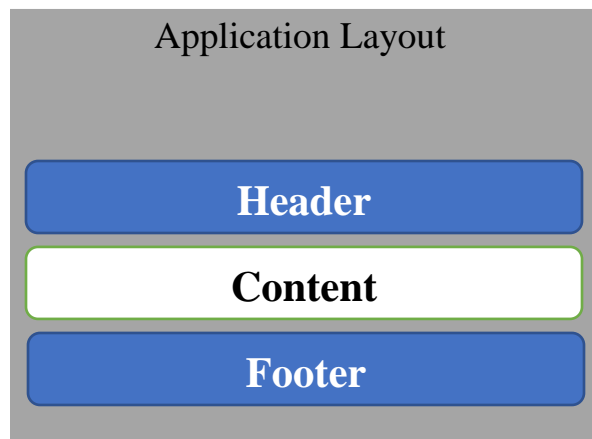
# Nested layouts in Next.js

## What is page

In Next.js, a "page" refers to the specific content or UI (user interface) that is associated with a particular route in your application. Each page corresponds to a specific URL path, and when a user navigates to that URL, they see the content defined by the corresponding page component

## What's a Layout?

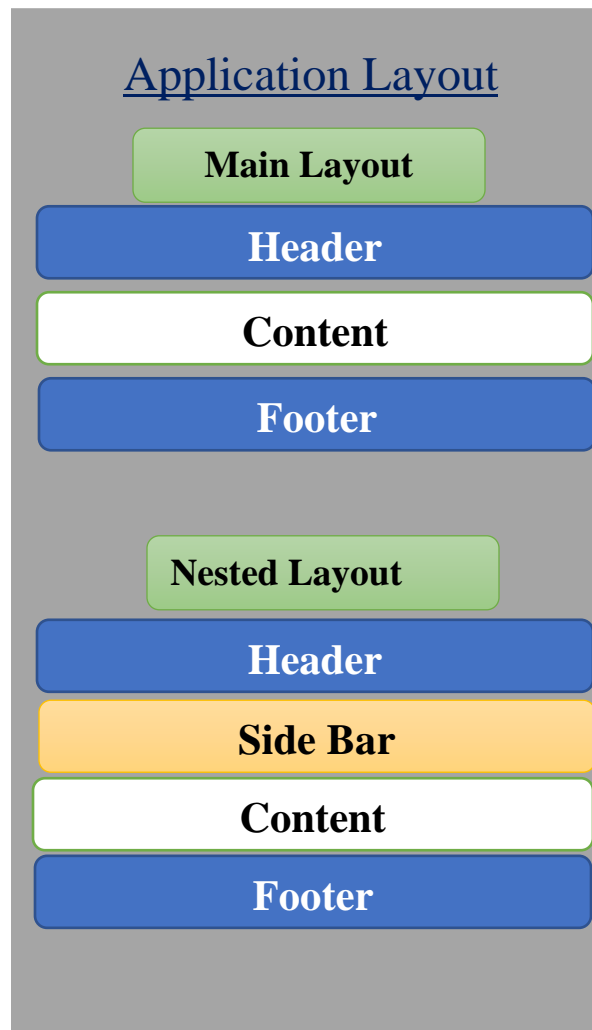
A layout is like a template for the pages on your website or app. It includes common elements that you want to appear on multiple pages, such as a header, footer, or sidebar.



## Nested layouts:

Nested layouts in Next.js refer to using multiple layout components within your application hierarchy. This means having a layout component that wraps around another layout component, creating a nested structure.

A simple block diagram illustrating the concept of nested layout components within an application hierarchy:



At the top level, you have the "Application Layout", which represents the overall structure of your application.

Within the "Application Layout", you have the "Main Layout", which serves as the main container for most of your pages.

The "Main Layout" consists of common elements like a header, content area, and footer.

Additionally, you may have a "Nested Layout" embedded within the "Main Layout". This nested layout can have its own structure, including its own header, sidebar, content area, and footer.

Each layout component encapsulates a specific set of UI elements and can be reused across different pages or sections of your application.

This diagram illustrates how nested layout components can be organized within your application hierarchy to provide structure, consistency, and flexibility in managing the UI of your Next.js application

## Purpose of Nested Layouts:

The purpose of nested layouts is to provide flexibility and organization in structuring the UI of your application. It allows you to compose complex layouts from simpler ones, creating a hierarchy of reusable components.

## Use of Nested Layouts in Simple Words:

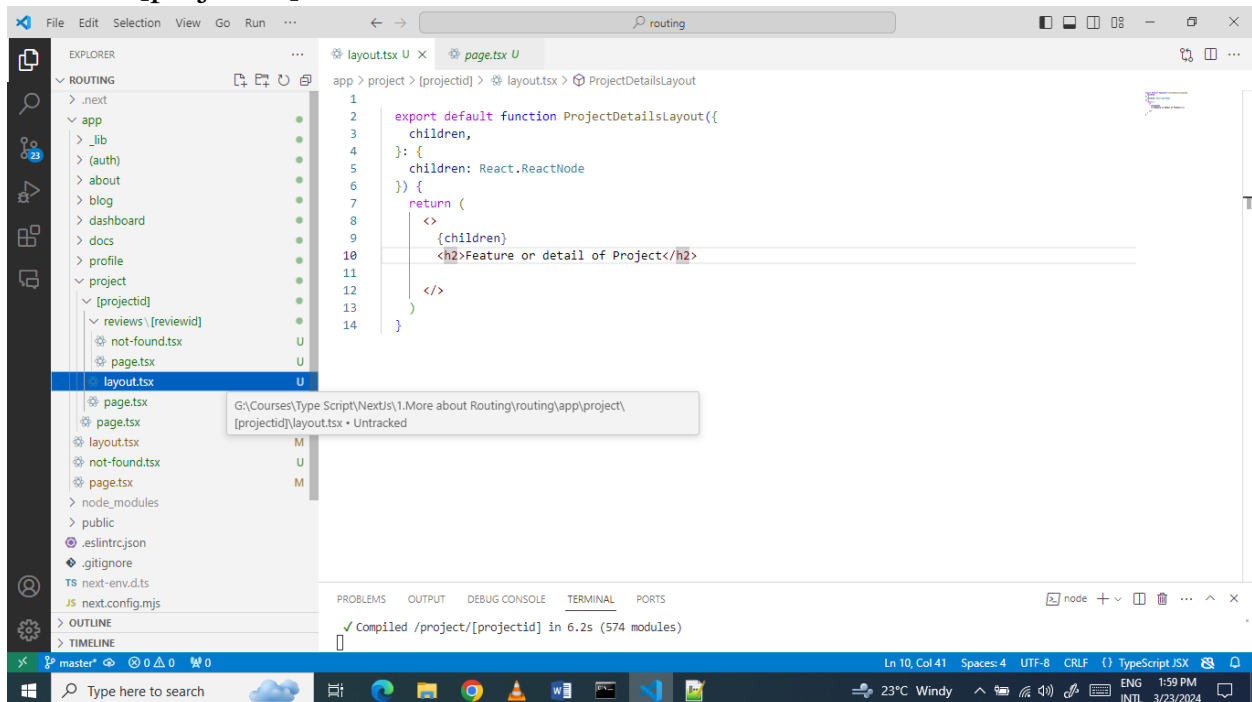
Imagine you're building a house. You start with a blueprint (a layout) for the entire house. But within each room, you might have different designs or features. Nested layouts are like having blueprints for each room within the overall blueprint of the house.

For example:

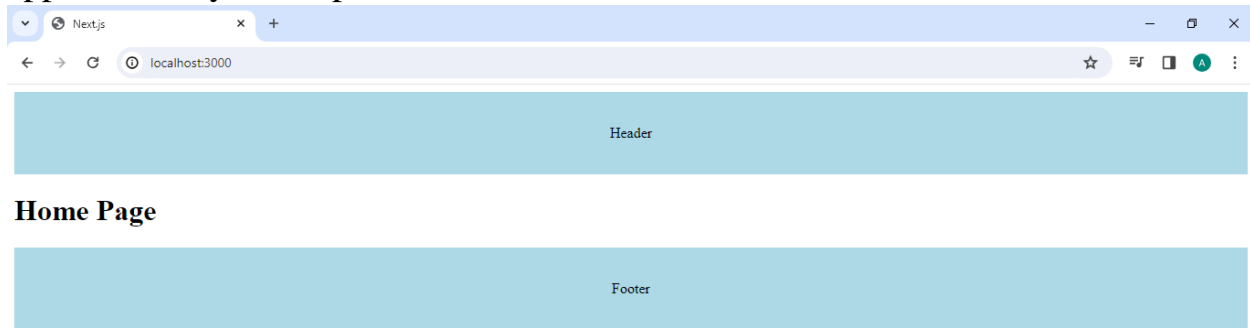
Your main layout (the entire house blueprint) might contain a header, a footer, and a main content area.

Within the main content area, you might have different sections (like rooms in a house) with their own layouts. These nested layouts might contain specific elements or styling unique to that section.

Create a simple example of a nested layout with a React component. Below is the structure for the layout.tsx file and a simple React component (layout.tsx) within a [projectid] folder.



Now, if we navigate the browser to <http://localhost:3000>, it will display the main application layout output.



Also, if we navigate the browser to <http://localhost:3000/Project>, it will display the main application layout output.



Now, if we navigate to <http://localhost:3000/project/1>, it will display the nested layout output.

