Presentation

NEXT.Js

The React Framework for the Web

Used by some of the world's largest companies, Next.js enables you to create high-quality web applications with the power of React components.

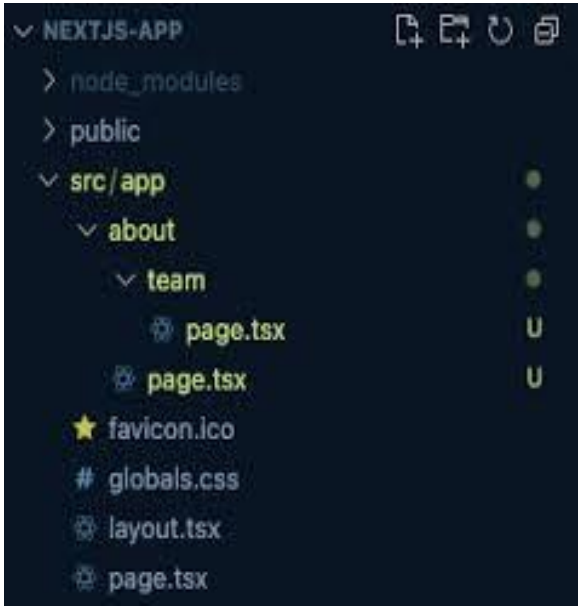
Made By: SAMREEN

TABLE OF CONTENT

1. What is page.tsx file?

Definition:

The page.tsx file is a special file in Next.js that defines a route. Each file in the app directory corresponds to a route in the application.



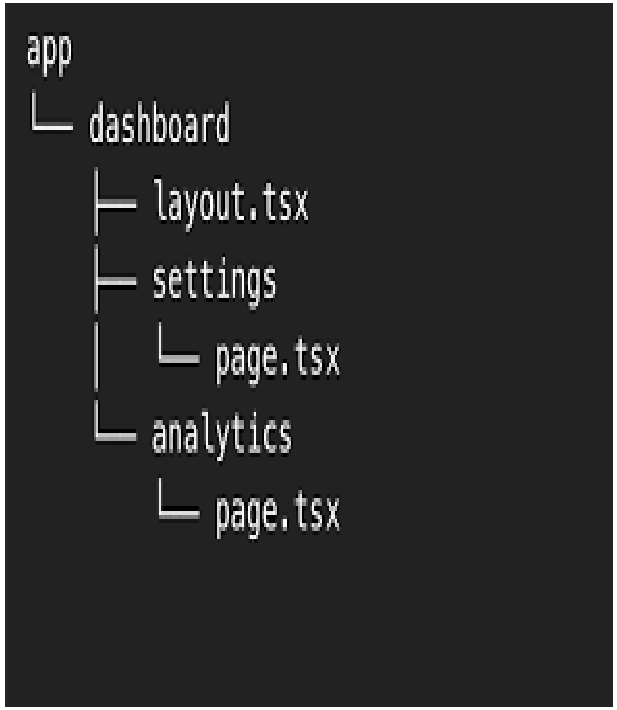
Purpose:

It contains the React component that renders the content for that specific route.

2. What is layout.tsx file?

Definition:
The layout.tsx file is used to define the layout structure for your pages.

Purpose:
It allows you to apply consistent UI elements (like headers, footers, sidebars) across multiple pages without duplicating code.



3- What is the Link tag?

Definition:

The <link> tag **defines the relationship between the current document and an external resource**.
The <link> tag is most often used to link to external style sheets or to add a favicon to your website. The <link> element is an empty element, it contains attributes only.

Example:

import Link from 'next/link';

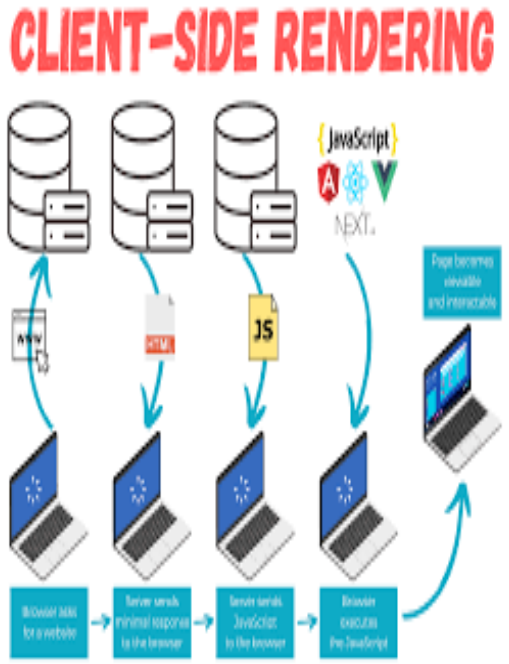
<Link href="/about">About Us</Link>

What are the Purpose of the Link tag?

Client-Side Navigation:
The <Link> component allows for seamless navigation without a full page reload. This enhances user experience by making transitions faster.

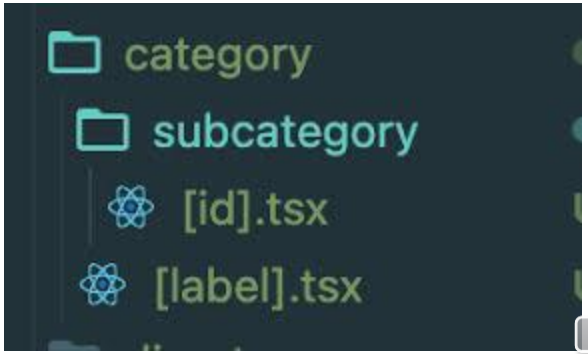
Prefetching:
Next.js automatically prefetches linked pages when they come into view (if the user is on a page long enough). This means that the linked page is already loaded in the background, making navigation instantaneous.

Accessibility: The <Link> component handles accessibility concerns, such as ensuring that the linked elements are keyboard navigable.



4- How to Create Nested Pages in Next.js?

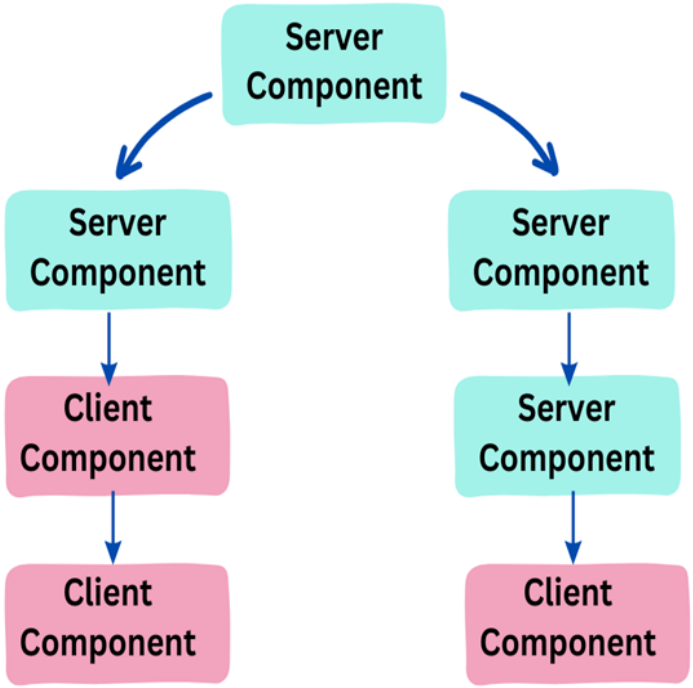
Next.js uses file-system routing where folders are used to create *nested* routes. Each folder represents a route segment that maps to a URL segment. page.tsx is a special Next.js file that exports a React component, and it's required for the route to be accessible. In your application, you already have a page file: /app/page.tsx - this is the home page associated with the route / . /app/dashboard/page.tsx is associated with the /dashboard path.



5- What are the Components

Components allow you to build self-contained, reusable snippets of code. you can take these individual bricks and combine them together to form larger structures.

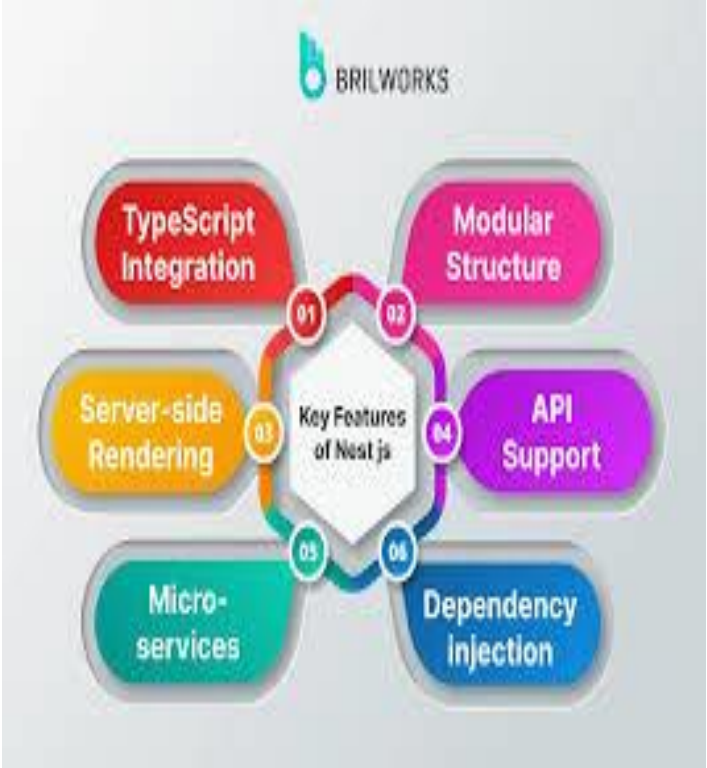
Next.js components are essentially React components. They can be functional or class-based and are used to define how a part of the UI should appear and behave.



Why Use Components in Next.js?

Reusability:
Components can be reused across different pages and parts of the application, making your code DRY (Don't Repeat Yourself).

Organization:
Breaking your UI into components helps keep your code organized and manageable, allowing for easier navigation and maintenance.



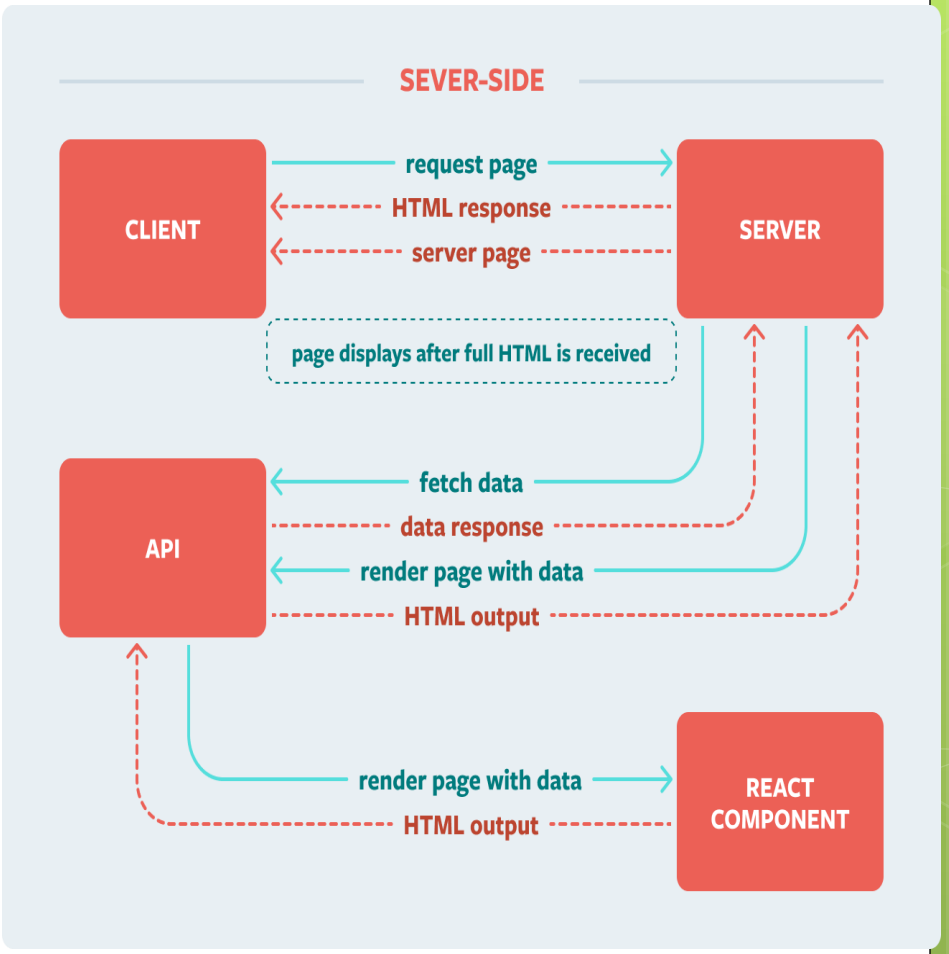
Use Components in Next.js?

Separation of Concerns:

Components can encapsulate specific functionality, separating the logic of a UI element from other parts of the application. This makes it easier to understand and test each component in isolation.

Dynamic Rendering:

Next.js components can leverage features like Server-Side Rendering (SSR) and Static Site Generation (SSG), allowing you to create dynamic, fast-loading applications that are optimized for performance and SEO.



global.css U ×

```
:root {
  --color-primary: #da3654;
  --color-text: #000;
  --color-text-2: #3c4856c2;
}

@font-face {
  font-family: "Recoleta";
  src: url("https://bejamas.i
}

body {
  margin: 0;
  font: 100%/1.55 "Poppins",s
  scroll-behavior: smooth;
}

:is(h1,h2,h3,h4) {
  font-family: "Recoleta", "P
}
```

6- How can we apply CSS in Next.js?

In Next.js, you can apply CSS in several ways, allowing for flexibility depending on your project needs. Here are some common methods:

❑ Global CSS

To include global styles, you can create a CSS file and import it in your _app.js file:

- **Create a CSS file in the styles directory (e.g., styles/global.css).**
- **Import the CSS file in your pages/_app.js:**

❑ CSS Modules

CSS Modules allow you to write scoped styles. This means styles defined in a module are specific to the component, preventing naming conflicts.

Create a CSS Module file with the .module.css extension (e.g., styles/Home.module.css).

Import the CSS Module in your component:

Shown in side image:

❑ Styled JSX

Next.js supports styled-jsx out of the box, which allows you to write scoped CSS directly within your components.

javascript

```
function Home() {
  return (
    <div>
      <p>Hello, World!</p>
      <style jsx>{`
        p {
          color: blue;
        }
      `}</style>
    </div>
  );
}

export default Home;
```

index.js ×

```
1   var React = require('react');
2   var ReactDOM = require('react-dom');
3   require('./index.css');
4
5   // state
6   // lifecycle event
7   // UI
8
9   class App extends React.Component {
10      render() {
11          return ( <
12              div >
13              Hello World!
14              <
15              /div>
16          )
17      }
18  }
```

❑ Styled Components

You can also use popular CSS-in-JS libraries like Styled Components. First, install the library:

```
npm installed styled-components
```

Then, create styled components:

```
import styled from 'styled-components';

const Container = styled.div`  color:red;`;

function Home() {

return <Container>Hello, World!</Container>;}

export default Home;
```

❑ Tailwind CSS

Next.js can easily integrate with Tailwind CSS for utility-first styling. To set it up, follow these steps:

Install Tailwind CSS:

```
npm install tailwindcss
postcss autoprefixer
npx tailwindcss init –p
```

In your globals.css, add the Tailwind directives.

```
/* styles/globals.css */
@tailwind base;
@tailwind components;
@tailwind utilities;
```

```
<div class="p-6 max-w-sm mx-auto bg-white rounded-xl shadow-lg flex items-center sp
  <div class="shrink-0">
    <img class="h-12 w-12" src="/img/logo.svg" alt="ChitChat Logo">
  </div>
  <div>
    <div class="text-xl font-medium text-black">ChitChat</div>
    <p class="text-slate-500">You have a new message!</p>
  </div>
</div>
```

What is Tailwind CSS?

Tailwind CSS

is a utility-first CSS framework that allows developers to build custom designs quickly and efficiently. Unlike traditional CSS frameworks that provide pre-designed components, Tailwind focuses on offering low-level utility classes that can be combined to create unique designs directly in your markup.

Differentiate b/w the Tailwind CSS & Standard CSS.

S. #	TAILWIND CSS	STANDARD CSS
01	Utility-First Approach Tailwind CSS uses a utility-first approach, providing small, reusable utility classes for styling directly in your HTML.	Utility-First Approach This contrasts with standard CSS, where styles are usually defined in separate stylesheets.
02	Customization Tailwind is highly customizable through a configuration file, allowing developers to define theme colors, spacing, and breakpoints.	Customization Standard CSS requires manual adjustments for each project.
03	Readability Tailwind can improve readability by making the styles explicit within the markup.	Readability Standard CSS often requires looking up styles in separate files, which can make it harder to understand the overall design at a glance.
04	Development Speed Using Tailwind can speed up development by reducing the need for custom class names and allowing for rapid prototyping.	Development Speed Standard CSS may require more setup and boilerplate code.

Submitted to:

Thank you and your countless efforts Sir Hamza Alvi.