Contents

[Question 1 – How the root page is being called or configured, what is the high-level flow of the execution 1](#_Toc194409703)

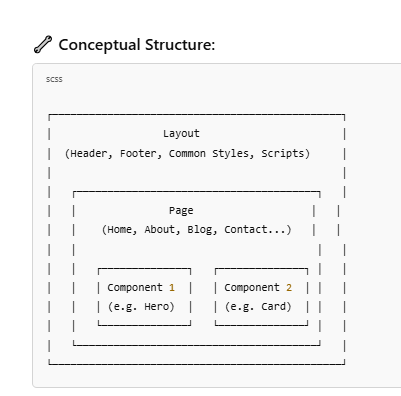
[Question 2 – CSS to Tailwind 1](#_Toc194409704)

[What is REM 2](#_Toc194409705)

[🔁 Why Use rem Instead of px? 3](#_Toc194409706)

## Question 1 – How the root page is being called or configured, what is the high-level flow of the execution

Very minimum you need to understand the concept of layout, page and component



**🔍 Explanation:**

**✅ Layout (/components/Layout.js)**

* Shared UI elements like:
  + **Header**
  + **Footer**
  + **Global CSS or Scripts**
* Wraps around every page

**✅ Page (/pages/index.js, /pages/about.js)**

* Represents each **URL route**
* Uses the Layout and combines different **components**

**✅ Component (/components/Button.js, /components/Card.js)**

* Reusable UI elements like:
  + Button
  + Card
  + ImageSlider
  + TestimonialBlock

**💡 Sample Code (Minimal Demo)**

jsx

CopyEdit

// components/Layout.js

export default function Layout({ children }) {

return (

<>

<header>Header</header>

<main>{children}</main>

<footer>Footer</footer>

</>

);

}

jsx

CopyEdit

// components/Hero.js

export default function Hero() {

return <div className="bg-blue-200 p-6 text-xl">Welcome to the site!</div>;

}

jsx

CopyEdit

// pages/index.js

import Layout from '../components/Layout';

import Hero from '../components/Hero';

export default function Home() {

return (

<Layout>

<Hero />

<p>Some homepage content goes here...</p>

</Layout>

);

}

After Next 13+ version the structure has changed, at minimum we can start with

/app

/components

Header.tsx

Footer.tsx

MainContent.tsx

/about

page.tsx

/contact

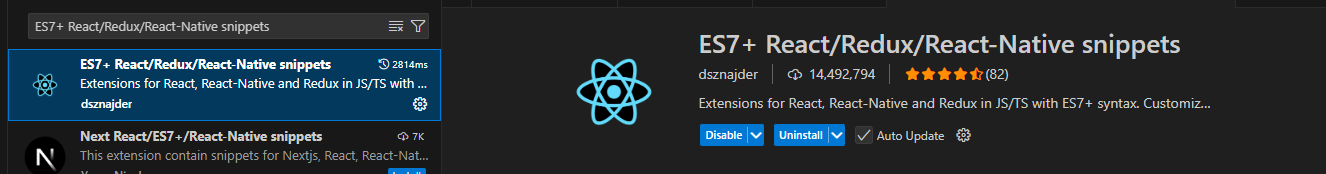
page.tsx

layout.tsx

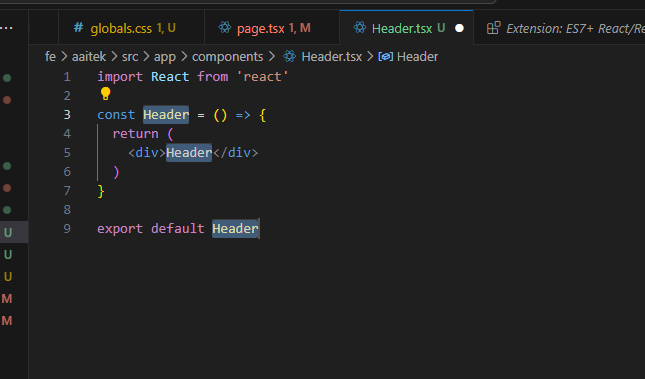
page.tsx <-- this is your Home page

Install this module for the short cut to create the scaffolding of the components

ES7+ React/Redux/React-Native snippets



It will create this structure



Short cut

After installation, you can type shortcuts like:

| **Shortcut** | **Expands to** |
| --- | --- |
| rafce | React Arrow Function Component Export |
| rsc | React Stateless Component |
| rfc | React Function Component |
| rscp | React Component with PropTypes |

**Example:**

Type rafce and hit **Tab** → it becomes:

tsx

CopyEdit

import React from 'react';

const MyComponent = () => {

return (

<div>MyComponent</div>

);

};

export default MyComponent;

## Question 2 – CSS to Tailwind

Regular CSS

<h1 class="title">Welcome</h1>

<style>

.title {

font-size: 32px;

font-weight: bold;

margin-top: 20px;

color: #1e293b;

}

</style>

**🔹 Tailwind version:**

tsx

CopyEdit

<h1 className="text-3xl font-bold mt-5 text-slate-800">Welcome</h1>

Same thing. But faster, cleaner, and no separate CSS.

**🧩 BREAKING DOWN COMMON CLASSES**

| **Tailwind Class** | **Meaning** |
| --- | --- |
| p-4 | padding: 1rem |
| mt-6 | margin-top: 1.5rem |
| text-xl | font-size: 1.25rem |
| font-bold | font-weight: bold |
| bg-gray-100 | background-color: light gray |
| text-center | text-align: center |
| rounded-lg | border-radius: large |
| shadow | apply a box shadow |
| **🛠 RECOMMENDED TOOL: Tailwind Cheat Sheet**  This is a visual tool that shows:   * All utility classes * Live examples * Copy-paste ready code   <https://nerdcave.com/tailwind-cheat-sheet>  https://play.tailwindcss.com/ |  |

### What is REM

**💡 What is rem in CSS?**

**rem** stands for **“root em”**, and it’s a **relative unit** of measurement.

* 1rem = **the font size of the root element** (usually <html>).
* By default, most browsers set this to 16px.

So:

* 1rem = 16px
* 0.5rem = 8px
* 2rem = 32px
* and so on…

### 🔁 Why Use rem Instead of px?

| **px** | **rem** |
| --- | --- |
| Fixed size | Scales with root font-size |
| Less responsive | Better for accessibility |
| Doesn’t adapt easily | Adapts if user changes font size |

**📘 Example:**

css

CopyEdit

/\* Traditional CSS \*/

.title {

font-size: 32px;

padding: 16px;

}

/\* With rem \*/

.title {

font-size: 2rem; /\* 2 × 16px = 32px \*/

padding: 1rem; /\* 1 × 16px = 16px \*/

}

Now if you **change the base size** (e.g. html { font-size: 18px; }), everything adjusts accordingly.

**🛠 In Tailwind:**

Tailwind uses rem behind the scenes.

| **Tailwind class** | **Equivalent** |
| --- | --- |
| p-4 | padding: 1rem (16px) |
| text-xl | font-size: 1.25rem (20px) |
| mt-2 | margin-top: 0.5rem (8px) |

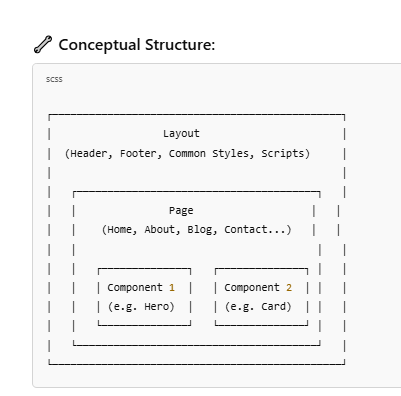
You never need to type rem — just use Tailwind's scale.

**✅ Summary**

* rem = **root em** = **relative to root <html> font-size**
* **1rem = 16px** (by default)
* It's great for **responsive design** and **accessibility**
* Tailwind uses rem behind the scenes so you get the benefits automatically

## Question 3 – how to setup the responsive design

Very minimum you need to understand the concept of layout, page and component



**🔍 Explanation:**

**✅ Layout (/components/Layout.js)**

* Shared UI elements like:
  + **Header**
  + **Footer**
  + **Global CSS or Scripts**
* Wraps around every page

**✅ Page (/pages/index.js, /pages/about.js)**

* Represents each **URL route**
* Uses the Layout and combines different **components**

**✅ Component (/components/Button.js, /components/Card.js)**

* Reusable UI elements like:
  + Button
  + Card
  + ImageSlider
  + TestimonialBlock

**💡 Sample Code (Minimal Demo)**

jsx

CopyEdit

// components/Layout.js

export default function Layout({ children }) {

return (

<>

<header>Header</header>

<main>{children}</main>

<footer>Footer</footer>

</>

);

}

jsx

CopyEdit

// components/Hero.js

export default function Hero() {

return <div className="bg-blue-200 p-6 text-xl">Welcome to the site!</div>;

}

jsx

CopyEdit

// pages/index.js

import Layout from '../components/Layout';

import Hero from '../components/Hero';

export default function Home() {

return (

<Layout>

<Hero />

<p>Some homepage content goes here...</p>

</Layout>

);

}

After Next 13+ version the structure has changed, at minimum we can start with

/app

/components

Header.tsx

Footer.tsx

MainContent.tsx

/about

page.tsx

/contact

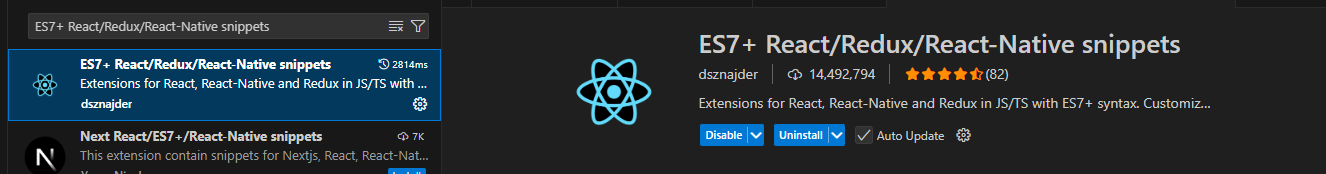
page.tsx

layout.tsx

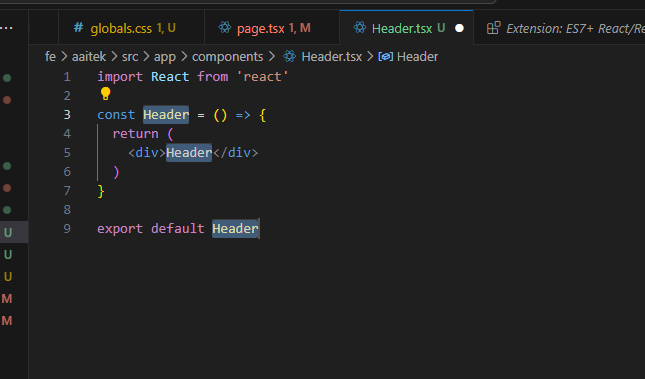
page.tsx <-- this is your Home page

Install this module for the short cut to create the scaffolding of the components

ES7+ React/Redux/React-Native snippets



It will create this structure



Short cut

After installation, you can type shortcuts like:

| **Shortcut** | **Expands to** |
| --- | --- |
| rafce | React Arrow Function Component Export |
| rsc | React Stateless Component |
| rfc | React Function Component |
| rscp | React Component with PropTypes |

**Example:**

Type rafce and hit **Tab** → it becomes:

tsx

CopyEdit

import React from 'react';

const MyComponent = () => {

return (

<div>MyComponent</div>

);

};

export default MyComponent;