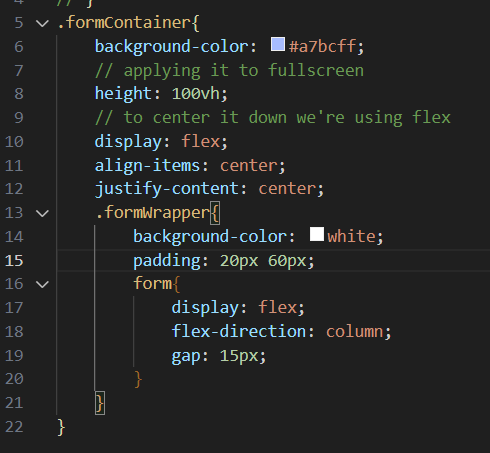
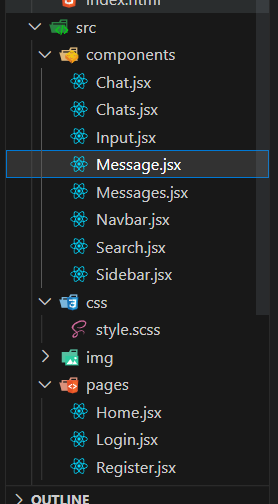
***CHATAPP***

* We started by creating out react app using “npx create-react-app chatapp”
* And then we created pages for **“Login.jsx”, “Register.jsx” and “Home.jsx”**
* we installed **sass** for the css part **“npm i sass”.**
* For it we’ll create **“style.scss”** for it.
* The best feature about **sass** is that we can write css classes as children inside other classes

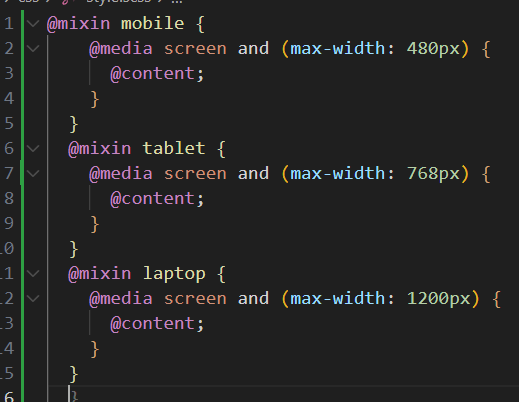


Afte we’ve designed our app at then end we’ve these many components



**FROM 45:55 SEC WE’LL START WORKING ON *MIXIN***

We added the following code **styles.scss**



The above given code defines three mixins for applying styles at different screen sizes.

The first mixin is called "**mobile**" and is defined using the **@mixin directive.**

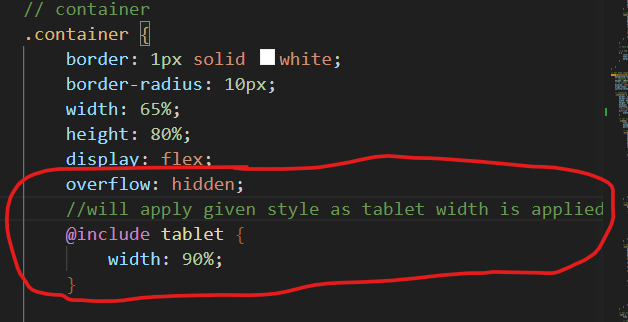
Inside the mixin, it **defines a media query** using the **@media directive**, which targets screens with a maximum width **of 480 pixels**. The **@content directive** is **used to insert the styles defined in the mixin wherever the mixin is called.**

The second mixin is called "**tablet**" and is similar to the first, except it targets screens with a **maximum width of 768 pixels**.

The third mixin is called "**laptop**" and targets screens with a **maximum width of 1200 pixels.**

**Example case:**

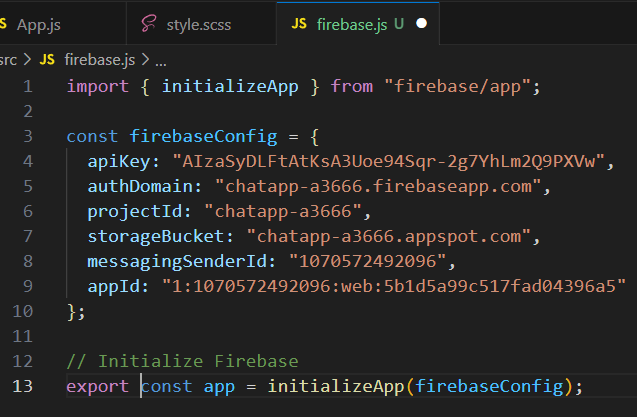
* ****
* **In this example, the .my-class selector sets a default background color of white.**
* **But when the screen size is less than or equal to 480 pixels, the background color will be set to red; when the screen size is between 481 and 768 pixels, the background color will be set to green; and when the screen size is between 769 and 1200 pixels, the background color will be set to blue.**

****

**Tablet ki size pr aate hi width 90% ho jayegi.**

[**48:58**](https://www.youtube.com/watch?v=k4mjF4sPITE&t=2938s) **React Firebase v9 Authentication Tutorial (Login - Register)**

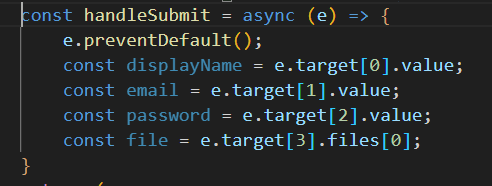
We started by creating a new project and then adding our webapp to the project. Then from the credentials we get adding webapp to the project we add them in **firebase.js**  in the src folder.



Then we move to register.jsx page to work on signup process

\*e.preventDefault(), prevents the default action of the form submission from occurring, which usually involves reloading the page or navigating to a new URL.

\*The displayName, email, and password fields are extracted using the e.target property, which returns the element that triggered the event, in this case the form. The values of these fields are accessed using the value property, which returns the current value of the field.



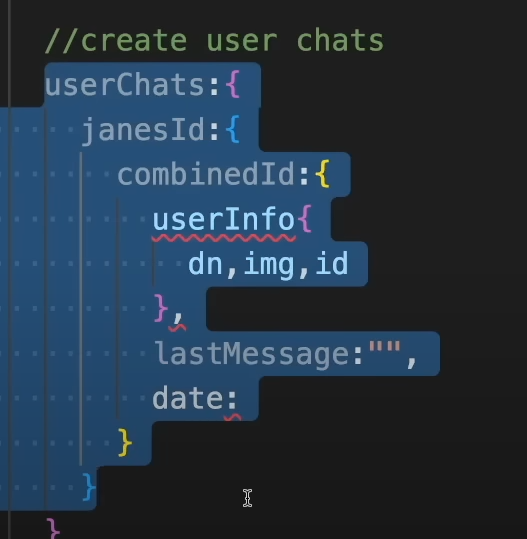
***Onwards*** [***01:23:33***](https://www.youtube.com/watch?v=k4mjF4sPITE&t=5013s) ***Firebase Chat App Database Structure***

We started to created collection in firestore database to store chat.

**So what we aim is to join User1 uid and User2 uid and join them write together to create a entry in collection so that it stores the chat between them.**

Uptill new used to creat **userChat(collection)** when we **register a user** but inside we’re gonna store the other persons info

For example: I(Aishwar) is talking to jane then Aishwar’s collection in userChat gonna store info about **jane’s profile picture, name, uid, there combined uid and the time of the chat**, **last message** as well



**Onwards** [**01:34:22**](https://www.youtube.com/watch?v=k4mjF4sPITE&t=5662s) **Fetch Chat Conversations we’re gonna fetch real time data from firebase.**

For it we’re gonna use realtime datbase

We’ll use snapshot function for it