Assignment 3: Finishing Your Frontend

Javier Sin Pelayo

URL (main branch): https://github.com/javisin22/csc372_projects
URL (client_site_v2): https://github.com/javisin22/csc372_projects/tree/client_site_v2
URL (live site): here

Part I: Researching & Selecting a Frontend Framework

1. **Which frontend framework did you choose? Why?**

   React is the chosen framework. I chose it because of the component based architecture. It enables you to break the UI into reusable, modular components (Calendar, Chat, Booking Form, Dashboard, etc.). Also, I have some previous experience with React, reducing the learning curve and speeding up development.
   It has an extensive ecosystem, with a lot of useful libraries for routing, state management, and UI component libraries to streamline design implementation.
   Additionally, there's a large community that offers good documentation and many online resources.

2. **How does this framework align with your client's website needs? Consider:**

   - **Content Structure** (Does it support the way your site is organized?

     With its modular organization, React's component-based structure allows me to design and implement distinct sections (public pages, student portal, tutor portal) that mirror my sitemap.
     Also the dynamic routing efficiently renders and updates parts of the site such as calendars, booking lists, and dashboards without reloading the entire page.

   - **Layout & Design** (Does it align with your wireframes and sitemap?)

     React components can be styled to match your approved wireframes and UI designs using CSS, or CSS-in-JS libraries (styled-components).
     It's also straightforward to implement responsive layouts, ensuring the site works seamlessly across devices (smartphones, tablets, PCs) as required by your target audience.
     Common elements like headers, navigation menus, and buttons can be reused across pages, ensuring design consistency and easier maintenance.

   - **Interactivity** (Does it provide features that make interactions easier to implement?)

     React's declarative approach simplifies the implementation of interactive features such as real-time chat, booking appointments, and calendar updates. There's

also libraries that support state management, easing the implementation of complex state interactions such as: tracking booking status, payment history, and notifications.
With the use of WebSockets, features like notifications and chat can be efficiently integrated.

3. **What challenges do you anticipate when using this framework? How do you plan to address them?**

   I have never used libraries like Redux to organize and manage state systematically, but if the app isn't really big it could be handled with other React features like the contexts.
   Another challenge I want to face is performance optimization. I plan to use the React DevTools to identify bottlenecks, avoid unnecessary re-renders, and apply general performance optimization techniques.
   Also, I would like to follow best practices and carefully review documentation, as I've created previous React apps but I want to make them more and more in a cleaner and more understandable way.

Part II: Using A Frontend Framework

(Implementation)

Part III: Hosting Your Site

As I'm using node, I had to host it in *Render*. Here's the link to access the live site:

https://classconnect-18h8.onrender.com

As I'm using the free tier version of the *Render* app, the website got into "Sleep" mode and it's supposed to be up and running when new requests are made. I've tried it and it works when the service is just deployed, but after 15 minutes of inactivity it puts into this "Sleep" mode and doesn't provide the website, but it keeps trying to load the page instead.

Part IV: Client Feedback & Approval

A) **Demo Presentation & Client Feedback**
   The demo was done in a meeting with screen-mirroring. The client was more than satisfied with the result.

   Is the site easy to navigate? → Yes, it's a smooth app that allows the user to navigate without any complication thanks to all the different links and buttons provided.

   Does it align with their brand identity → Yes, it's exactly what I was looking for!

   Does the site display the correct content dynamically? → At this moment there's not

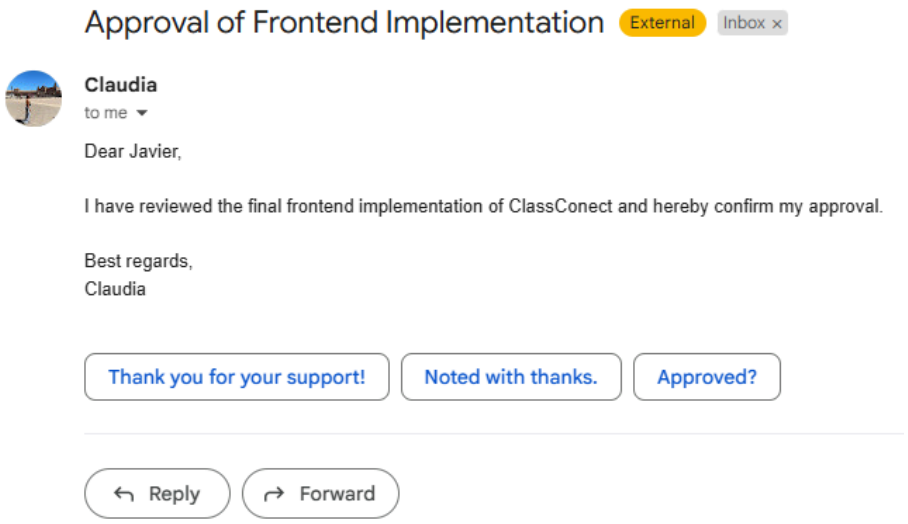
much dynamic content, but I like what I see.

Do interactive elements work as expected on different screen sizes? → Yes, those responsive designs will do the job.

Are any elements missing or requiring further refinement? → There's several features that have not been implemented yet (that's because I wanted to show the client before implementing a lot. At the demo, some placeholders were shown instead).

Summary:
The meeting was held on Tuesday 2/11 in a Google Meet video call as the client is in Spain. I basically showed her a demo of the app and discussed the next steps. I told her how the app will be improving week by week with more and more additions. Now, the next step is to implement a valid calendar (the live page now has a first provisional version of it because I've advanced since the meeting but it wasn't when I showed her the site). She didn't request any change, just told me to keep implementing it as I'm doing it because she thinks I'm quite aligned with her thoughts.

B) Final Approval



C) Reflection on Client Feedback

I just added more buttons to ease the client's experience. For example, including both the login and signup buttons on the header and below the "Features" section.
Also, in the dashboard, I added some quick action buttons in the default page.

There's no before and after as I just implemented the wireframe ideas (where she told me about the button additions).