*GROUP* ***DISCUSSIONS***

|  |  |
| --- | --- |
| *GROUP MEMBERS* | * Ory Chanraksa * Sao Visal * Hong Layeang * Kong Visal |

 **Discuss** in group (3 to 4)on bellow topics.

A person pointing at a board

AI-generated content may be incorrect. **Present** your outcomes to the whole group.

**DISCUSSION 1: LAST PRACTICE DISCUSSION**

* First, share the points you **could not perform** well during the last practice: does someone know how to solve them?
* **Ory Chanraksa:**  
  I think my code worked fine overall, but I could have made it a bit cleaner. I didn’t really focus on optimizing, just got it to work. Also, I’m not sure if I used the key attributes properly.
* **Sao Visal:**  
  My component does what it’s supposed to, but I feel like I could have made it more reusable. I just looped through them normally, but maybe there’s a better way.
* **Hong Layeang:**  
  I got everything working, but I struggled a bit with changing the score color when it was below 50. I managed to do it, but I feel like my approach was a bit messy.
* **Kong Visal:**  
  I had the same issue at first. I ended up just adding a condition inside the styling, and it worked fine, but I’m wondering if there’s a better way to do it.
* Then, **compare your codes** among the team: which solution is the better? Cleaner?

After evaluating everyone’s code, we would say that Layeang’s code is the cleanest because it’s well-structured and easy to read. He also added some extra functionality, like making the table more responsive, and improving readability with comments. It made the component more reusable without overcomplicating it.

**DISCUSSION 2: IMAGINE AN APP USING COMPONENTS!**

Do you remember last practice? You had to design a component **Place** that takes as an argument **a place data structure.**

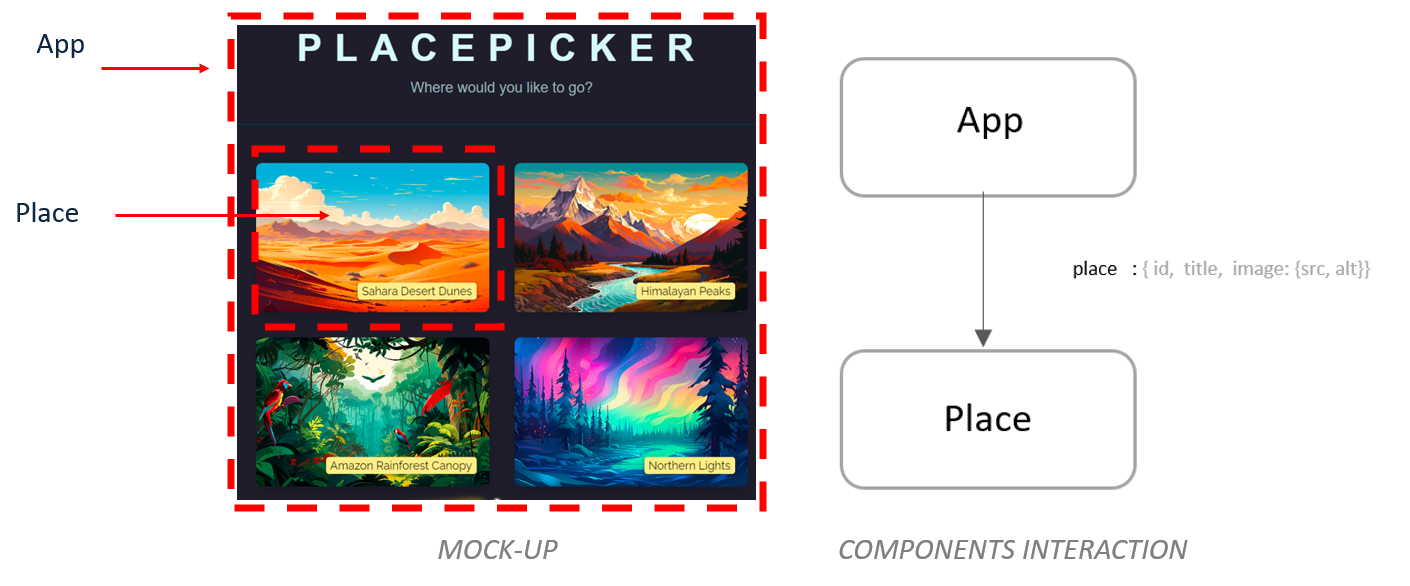
****

Image another app!

* Thing about an application and identify how you can use components to re-usage similar widgets
* Your component must take at least 1 parameter
* Your component must be used at least 2 times
* A book-tracking app, like a reading list.
* A **BookCard** component would take in props like **title, author, and status** (e.g., "Read" or "Not Read").
* Instead of just displaying the book info, we add a **button to update the status**, so users can mark books as "Read" when they finish them.
* We could also add a **"Recommended by"** field to track who suggested the book.
* This follows the same idea as the last exercise, where we passed data as parameters to a component and displayed it dynamically.
* The **BookCard** component would be used multiple times, just like in the previous practice.

**Example usage:**

<BookCard title="The Great Gatsby" author="F. Scott Fitzgerald" status="Not Read" />

<BookCard title="1984" author="George Orwell" status="Read" />

To present

* Your app mockup
* Your component(s) you have identified
* Your component(s) parameters