eBuddy Technical Test – Syahrul Rofi

🔗 Links

- **GitHub Repository**: https://github.com/RofiSyahrul/ebuddy-technical-test
- **LinkedIn Profile**: https://www.linkedin.com/in/syahrul-rofi/
- **Personal Website**: https://rofisyahrul.com/

📦 Part 4: Bonus Firebase Technical Questions

1. Firestore Query Strategy for High Potential Users with Pagination

Firestore does not support weighted multi-factor sorting natively. To enable efficient pagination and ranking, we can precompute a custom field `rankingScore` that combines the priorities:

```
const rankingScore =
  totalAverageWeightRatings * 10000 +
  numberOfRents * 100 +
  recentlyActive / 1000000;
```

This formula:

- Prioritizes `totalAverageWeightRatings` as the most important factor.
- Uses `numberOfRents` and `recentlyActive` for tie-breaking, while maintaining significance.
- Produces a single numeric value suitable for sorting and pagination.

Query:

```
db.collection('USERS').orderBy('rankingScore', 'desc').limit(10);
```

---

2. Updating Online/Offline Status and recentlyActive

While I haven't worked with Firestore or RTDB in production yet, here's the strategy based on Firebase docs:

- Use **Realtime Database's** `.info/connected` to detect if the user is online.
- On disconnect, update `recentlyActive` using `onDisconnect().set(...)`.
- Use a **Cloud Function** to sync `recentlyActive` from RTDB to Firestore if needed for Firestore queries.
- For online status, consider writing a `status` field as `"online"` or `"offline"`.

This provides real-time presence tracking with persistence and Firestore query compatibility.

💬 Part 5: Personality & Technical Questions

1. Non-Server Component on Reddit Settings Page

Non-Server Components are typically interactive client-side elements those rely on JavaScript in the browser to function, such as click handlers, DOM mutations, or modal triggers. Based on this understanding and by inspecting the behavior and network activity in the browser (without accessing the codebase), here are parts of Reddit's settings page that are likely not Server Components:

- The Settings Tabs (e.g., Account, Profile, Privacy). The switching tabs happens instantly without a full page reload.
- The Search Bar. It accepts input and filters content client-side.
- The Toggle Button for collapsing/expanding the left navigation menu.
- The Accordions inside the left navigation. They expand/collapse via DOM manipulation (e.g. toggle className or any DOM attributes).
- The Chat Trigger Button in the top navigation.
- The Profile Menu Button. It opens a dropdown menu.
- Form Fields and Buttons. They open popups or inline modals for editing, which implies dynamic interactivity.

These components rely on client-side JavaScript and would be implemented as Client Components in a Server Component architecture like React Server Components.

2. Most Difficult Technical Problem

One challenge was implementing **complex dynamic filtering with pagination**. The difficulties included nested filters, performance, and maintaining state on refresh.

Solution:

- Created a **config-based filtering system** for scalability.
- Used **URL query params and pathnames** to persist filters.
- Applied **debounce + memoization** to optimize re-renders.
- Encapsulated logic into **custom React hooks** with unit tests.

This improved performance and made the filters more maintainable.

3. Project Approach from Start to Finish

1. Understand the scope and requirements
2. Design architecture and the user flow
3. Break the problem into actionable tasks
4. Build iteratively, test frequently
5. Polish UI/UX and write documentation
6. Deploy and monitor

4. Learning Strategy

- Start with **official documentation** or a trusted source.
- Build a **small demo or feature** using the new concept.

- Repeat and revise, i.e. revisit docs or advanced use cases.
- Avoid passive content; prefer **applied, hands-on learning**.

5. “Consistency” vs “Fast & Efficient”

I prefer **Consistency**. It builds trust, reduces tech debt, and ensures long-term velocity across a team.

6. Do You Own Any Apple Products?

Yes, I use a **MacBook Pro** on daily basis for development.

7. Availability

I’m available to start **immediately**, or within **1 week** if needed.

Thank you for reviewing my answers and project. Looking forward to the next steps!

Syahrul Rofi