Clarifying Questions Regarding Figma and Assignment Scope

Thanks again for the opportunity!

After reviewing the Figma design and task description, I’ve compiled a short list of clarification questions to ensure we’re fully aligned before development begins.

I have to admit that mostly those question deals with edge cases and extend the demo assignment towards a few weeks of development if they are addressed.

From the other hand I have to raise some flags as experience demonstration, which took like a hour of work on the questions:

**Form Behavior & Input Handling**

1. The Figma currently shows only red borders for validation errors. Should we also display descriptive error messages above/under the red border like:
   1. “Display Name is required”
   2. “Invalid URL”
   3. “404 – Not Found”
2. Should we add a placeholder for API errors? like:
   1. “Database unavailable”
   2. “Server Side link validation failed with 403 error (authentication required)”
3. The “Save” button appears to support both “Add” and “Edit”. Should I assume the same component handles both use cases?
4. Should validation occur on the frontend, backend or both?
5. Should we check for duplicate links e.g., prevent submitting the same URL more than once?
6. Same about Display Name.
7. How should we handle situation when link fails validation e.g., unreachable or private?
   1. Should we validate the URL structure prior HTTP request, to notify the user about malformed link, and prevent going further into validating the HTTP response?
   2. If a link has a valid format, but returns a problematic HTTP response (403, 404, timeout), should we still allow saving?
   3. What will be the timeout setting? Need to make sure it will not exceed the API timeout setting in case of synchronous backend validation.
   4. Should we introduce “force save” in case link has a valid format, HTTP response validation was failed, but user insists on saving regardless of the warnings system generated?

**URL & Video Handling**

1. Is it acceptable to validate YouTube link availability using this public endpoint?
   1. <https://www.youtube.com/oembed?url=https://www.youtube.com/watch?v=VIDEO_ID&format=json>
2. If a media item becomes private or deleted after validations passed, how should we address that when video playback fails or client hit a 403/404 link?
   1. Periodic revalidation job on backend?
   2. Notifications of users about content that became unavailable?

**Media Display & Rendering**

1. When the Library is empty, what should we show? (e.g., placeholder image, message, or blank state)
2. Regular & search results, what should be the ordering? By age/popularity/other?
3. Should we implement pagination/lazy loading/infinite scroll for long item lists?
4. The “Case Studies” label under each item — what is the origin of that string?
5. The design uses a custom preview for Loom — should Loom links be handled differently, or treated as regular external links?
6. The “Loom” Under the most left avatar:
   1. Is this part of an advanced assignment version (e.g., Loom-specific behavior)?
   2. Is it a domain part and I should parse the domain name to generate that label (e.g. [www.loom.com](http://www.loom.com) -> loom )?

**Avatars & Ownership**

1. Where should the avatar image come from?
   1. Uploaded during user creation and added into the MongoDB collection of users as a user property?
   2. Pulled from the owner user’s SSO account?
   3. Extracted from the linked domain (e.g., favicon)?
2. If avatars are derived from external sources:
   1. There is a security concern - embedded attack surface, how to address?
   2. Should the avatars be cached and persisted, or always fetched live?
   3. What is the caching policy?
   4. Any cleanup requirements for avatars?

**System Feedback & Error Reporting**

1. Should the frontend / backend errors be reported to a service like Sentry or Bugsnag?
2. Should API errors return user-visible feedback (e.g., “Database error”, “Access violation”, “Item already exists”)?
3. How should we differentiate between infrastructure errors (e.g., database unreachable) and data/logic errors (e.g., duplicate URL)?