



Notes

May 20, 2025

J & K

Invited Kiyasha Singh Justin Germishuys

Attachments J & K

Meeting records Transcript

Summary

Kiyasha Singh reported issues with the magic link login, leading Justin Germishuys to suggest switching to OTP for better production stability. The discussion covered database setup and testing strategies, review of Superbase including bug reports and row-level security, and the structure of tables for challenges, responses, activity logs, users, and cohorts, emphasizing efficient management and cohort-specific content display. Justin Germishuys provided guidance on front-end interaction with Superbase using React and JWT tokens, bulk user uploads, and the importance of adhering to the project timeline, while Kiyasha Singh shared feedback on existing challenges and anticipated future needs for video management and dynamic content based on language and audience.

Details

- **Magic Link Issues and OTP Suggestion** Kiyasha Singh reported encountering errors while implementing the magic link for login, after previously struggling with the forgotten password feature. Justin Germishuys suggested switching to OTP (One-Time Password) instead, noting that magic links can become problematic when moving from development to production ([00:00:00](#)). Justin Germishuys explained that OTPs still avoid passwords and use email but don't rely on a direct callback link, and the code is similar to magic links. Justin Germishuys believed this switch would save time and headaches, as their own experience with OTP was quick and problem-free ([00:02:29](#)).

- **Database Setup and Testing** Kiyasha Singh has the database set up but needs to sign in to verify its functionality, which is hindered by the magic link issue ([00:02:29](#)). Justin Germishuys advised Kiyasha Singh to temporarily disable sign-in via Cursor to test the database directly, emphasizing that sign-in can be added later but acknowledging the need for initial test users linked to user IDs. Justin Germishuys and Kiyasha Singh discussed the setup of a separate database version for testing with the new sign-in method ([00:03:35](#)).
- **Bug Reports and Superbase Review** Justin Germishuys wanted Kiyasha Singh to show them the Superbase setup and review bug reports, but preferred to look at the database first, assuming the bugs might be related to the current implementation ([00:04:43](#)). Kiyasha Singh indicated the bugs were simple content-related issues, not UX problems. Justin Germishuys also inquired about row-level security in Superbase ([00:05:48](#)).
- **Database Challenges and Activity Logs** Kiyasha Singh mentioned that the challenges table in the new database is currently empty, following Justin Germishuys' earlier advice to keep it hardcoded in the version everyone else is using. Justin Germishuys clarified that a table for magic link tokens or OTPs is unnecessary as the tokens are automatically generated and expire ([00:07:38](#)). Regarding the activity logs table, Kiyasha Singh explained it would track admin actions and completion times ([00:08:52](#)). Justin Germishuys reviewed the columns, particularly the "details" field set as JSONB, suggesting it might need adjustment based on the content. Justin Germishuys recommended using Cursor to get the most logical database structure and SQL as development progresses ([00:09:53](#)).
- **Challenge Responses and Completion Tracking** Justin Germishuys and Kiyasha Singh discussed the "challenge responses" table, clarifying it's for written reflections, as all challenges include reflection questions. Justin Germishuys questioned if there's a mechanism to track when a user simply completes a challenge (e.g., via a "completed" button), separate from submitting a reflection ([00:11:03](#)). Kiyasha Singh confirmed a "completed button" exists, and Justin Germishuys noted the need to store this completion data separately from reflections, as users might complete a challenge without reflecting ([00:12:28](#)).
- **User and Cohort Management in the Database** Justin Germishuys and Kiyasha Singh discussed the "organization members" table, with Justin Germishuys suggesting a more streamlined approach using a "user profiles" table linked to Superbase's default user table ([00:13:45](#)). Justin Germishuys proposed including

fields for email, username, organization, and cohort in the "user profiles" table, allowing for filtering by cohort ([00:15:03](#)). Justin Germishuys suggested eliminating the "organization members" and "organizations" tables ([00:16:30](#)). Justin Germishuys also identified the need for a "cohort" table and a "cohort ID" within the "challenges" table to manage organization and cohort-specific challenges ([00:19:27](#)).

- **Dynamic Challenge Display and Database Structure** Justin Germishuys explained the logic for displaying cohort-specific challenges: upon login, the system identifies the user's cohort, then retrieves the challenges selected for that cohort based on their IDs ([00:25:01](#)). Justin Germishuys considered using a JSON object within the cohort table to store the list of challenge IDs. Justin Germishuys also highlighted the need to store image URLs for challenges within the challenges table, referencing Superbase storage for the image files ([00:47:14](#)).
- **Front-end and Back-end Interaction with Superbase** Justin Germishuys explained that the current front-end is built with React and TypeScript without a separate backend server ([00:26:34](#)). Justin Germishuys sought advice from an AI on the most secure way for the front-end to interact with Superbase in this setup, emphasizing the need for production-grade security. Justin Germishuys recommended enabling row-level security in Superbase and using the public and anon API keys in the browser, while never exposing the service role key ([00:28:08](#)) ([00:32:34](#)). Justin Germishuys advised Kiyasha Singh to create a simple React form that writes to a Superbase table to gain practical experience with this interaction ([00:35:19](#)).
- **JWT Tokens and Superbase Client Setup** Justin Germishuys explained the concept of JWT (JSON Web Tokens) and their role in secure communication with Superbase, noting their temporary nature and automatic regeneration ([00:35:19](#)). Justin Germishuys mentioned that the superbase.js SDK stores access and refresh tokens in local storage by default, with a more secure option of using HTTP-only cookies ([00:33:44](#)). Justin Germishuys guided Kiyasha Singh on where to find the public and anon API keys in the Superbase project settings ([00:32:34](#)).
- **Cohort and Challenge Selection Mechanism** Justin Germishuys proposed creating a screen or form to add cohorts and a separate cohort table in Superbase. For each cohort, there would be a way to select specific challenge IDs from the total pool, potentially with a limit ([00:45:38](#)). This selection would then be stored in the cohort table, likely as a JSON object of challenge IDs ([00:47:14](#)).

- **Bulk User Upload and Additional Data Fields** Justin Germishuys raised the need for a bulk user upload feature, likely using CSV files, to efficiently add many users. Justin Germishuys also identified the requirement to include department or business unit information for users, in addition to organization and cohort ([00:51:15](#)).
- **Project Timeline and Collaboration** Justin Germishuys emphasized the importance of adhering to the project timeline due to commitments to clients. Justin Germishuys requested daily updates on progress and any challenges encountered ([00:52:27](#)). Justin Germishuys acknowledged that the existing platform provides a clear understanding of what needs to be built, and the rebuild is an opportunity to implement everything correctly with the new database structure in mind ([00:53:36](#)). Justin Germishuys suggested that if Kiyasha Singh gets significantly stuck, they can share their front-end files for assistance ([00:49:37](#)).
- **Feedback on Existing Challenges** Justin Germishuys inquired about feedback on the current challenges. Kiyasha Singh reported positive feedback, with the exception of one grammar error pointed out by David. Feedback from DAB students, some with limited English proficiency, suggested they understood the challenges, although it was unclear if they fully engaged with them ([00:56:05](#)).
- **Future Considerations: Videos and Dynamic Content** Justin Germishuys noted the need for a "videos" table in the database to manage video links, especially for cohort-specific videos ([00:57:18](#)). Justin Germishuys suggested including columns for video URL, associated habit, language, and avatar type in the videos table, allowing for dynamic fetching of video links based on cohort or other criteria. Kiyasha Singh anticipated the need for multiple tables to handle cohort-specific variations in displayed text as well ([00:58:41](#)).
- **Language Style and Level** Kiyasha Singh raised the point that language should be adapted based on the audience, suggesting more professional jargon for corporate clients and simpler language for students. Justin Germishuys agreed, proposing the inclusion of columns in their system to track language style (corporate/informal) and level, although they suggested keeping the default settings accessible for now and extending the table later if needed ([01:00:10](#)).
- **Database Architecture and Requirements** Justin Germishuys stated that the transcript likely captures most of the database architecture and requirements, including row-level security and the use of JWT tokens. They suggested Kiyasha

Singh could use the transcript to distill information and potentially create a diagram of the DB schema ([01:00:10](#)).

- **Meeting Conclusion** Justin Germishuys thanked Kiyasha Singh for their time and expressed that they hoped their collaboration would become the norm rather than a nostalgic occurrence. Kiyasha Singh echoed the sentiment, and the meeting concluded ([01:00:10](#)).

Suggested next steps

- Kiyasha Singh will (1) consider temporarily switching off sign-in to test the database, (2) start populating challenges and images in the new database, and (3) consult Cursor on the most logical database structure for writing to Superbase and generating SQL.
- Kiyasha Singh will add a mechanism to track challenge completion separate from reflections.
- Kiyasha Singh will explore creating a 'user profiles' table in Superbase that pulls user ID and email and allows adding username, organization, and cohort.
- Kiyasha Singh will create a 'cohort' table and associate a cohort ID with each challenge.
- Kiyasha Singh will investigate how to pull cohort-specific challenges based on user login.
- Kiyasha Singh will create a simple React form to write user profile information to Superbase.
- Kiyasha Singh will explore setting up a cohort management interface and consider a bulk user upload feature.
- Kiyasha Singh will create a 'videos' table with video URLs and associated habits, considering cohort-level control and language variations.
- Kiyasha Singh will provide daily progress updates.
- Kiyasha Singh will distill the transcript to propose a DB schema, potentially in diagram form.

You should review Gemini's notes to make sure they're accurate. [Get tips and learn how Gemini takes notes](#)

Please provide feedback about using Gemini to take notes in a [short survey](#).



Transcript

May 20, 2025

J & K - Transcript

00:00:00

Justin Germishuys: Hi, Kesha.

Kiyasha Singh: Thank you. Hello,

Justin Germishuys: How are you today?

Kiyasha Singh: I am good.

Justin Germishuys: Well, thanks. Um, I have a little bit more space now, so I'm feeling a lot better.

Kiyasha Singh: That's

Justin Germishuys: Um, yeah. How are things on your front?

Kiyasha Singh: uh agitating

Justin Germishuys: In what way?

Kiyasha Singh: um I'm implementing the magic link and previously I was having an issue with the forget forgot password.

Justin Germishuys: Yeah.

Kiyasha Singh: So now that I'm trying to implement this instead of the whole login and sign up I'm just running into errors. So yeah.

Justin Germishuys: Yeah. Can I suggest an alternative because I try to implement magic link for the first time like yesterday and when you go from development to production magic link becomes a pain in the butt.

Kiyasha Singh: Oh,

Justin Germishuys: So what I actually suggest is rather do OTP

Kiyasha Singh: okay.

Justin Germishuys: um because then you still don't have to deal with the password. an email is still sent, but you don't have to rely on the link to be a proper call back to your site.

00:02:29

Justin Germishuys: So that's um what I discovered. And basically the code for magic link is or for OTP is basically the same as the magic link code. You just have it send an

OT OTP rather than a

Kiyasha Singh: magic.

Justin Germishuys: So I mean give it a bash. I think that that would be will save you some headaches honestly. Um I'm glad I didn't spend too much time trying to get Magic Link to work and instead just opted to do OTP cuz that solved all my problems in like 10 minutes.

Kiyasha Singh: Oh, okay.

Justin Germishuys: Yeah, cool. Um, and how are things going with the DB?

Kiyasha Singh: So, I have it set up, but in order for me to actually check that everything is working properly, I have to sign in. But because of the whole magic link problem, I've been struggling with that.

Justin Germishuys: Ah, all right. I see. So, you are you are you rebuilding it? Are you creating a a cloned version with the magic link now?

Kiyasha Singh: Yes.

Justin Germishuys: Right.

00:03:35

Kiyasha Singh: So, this version is separate from the one that everyone has the link to.

Justin Germishuys: All right. So that's good. Um so basically um I would also suggest you switch off sign in so that you can test the DB. So you can always go into cursor and say look can you just temporarily switch off sign in so that I can test the you know so that I can get straight to the app and then you can just reintroduce it at the end.

Kiyasha Singh: Okay.

Justin Germishuys: Otherwise it gets in the way of development if you have to sign in. So it's not it's also the kind of thing you can add at the end because well I suppose it is and it isn't. So you want to set up sign in because you want to see that it's you know you've got the a lot of your tables link to the user ID.

Kiyasha Singh: Yes.

Justin Germishuys: So it's useful to at least have one or two test users in the DB already. But um and make sure that you know that ID is you know being used.

00:04:43

Justin Germishuys: Okay. Um so the question is I would like you to Allison asked us to

spec did she specifically ask us to look at something or was that No. Was that with me and Yiannis?

Kiyasha Singh: I think that was a few minutes.

Justin Germishuys: All right. Um, so I think I'd just like you to show me. So I don't want to focus on the emailers. I'm just going to do that this afternoon. Um, so not to worry about that. Uh, the site content we can, you know, edit a little bit as we go. So that's there. We might change one or two things in light of the feedback. Oh, she did say she wanted you to go through the bug reports with me.

Kiyasha Singh: Oh, okay. Uh,

Justin Germishuys: Yes.

Kiyasha Singh: okay. The bug reports

Justin Germishuys: But

Kiyasha Singh: it's

Justin Germishuys: I also want to see how it looks in your superbase. So

Kiyasha Singh: nice.

Justin Germishuys: like Yeah.

Kiyasha Singh: Okay.

Justin Germishuys: So both both of those two things we can have a look at quickly.

00:05:48

Kiyasha Singh: Uh, okay.

Justin Germishuys: What is today? Today is the 20th.

Kiyasha Singh: Oh, yes.

Justin Germishuys: Okay. So hopefully we can get all of the the content done before the end of this week and then that'll give everybody 5 days to get everything else ready.

Okay. So what should we do first? The DB or the bug? Let's do the DB first.

Kiyasha Singh: Oh, okay. Sorry, I'm saying the wrong thing then.

Justin Germishuys: Um it's all fine. It's just I don't want to look at the the bug reports first if I know that these bugs are going to disappear when you do something else.

Although

Kiyasha Singh: It's

Justin Germishuys: again,

Kiyasha Singh: very simple. It's not anything difficult at all.

Justin Germishuys: okay, so it's mostly just content stuff and no UX. All right. And even if it is a UX thing, it's not like super All right. Um, so do you have row level security associated with those? I'm not seeing a

Kiyasha Singh: Um, I actually have to implement it for the profile if I'm not mistaken cuz I was running into errors and I'm literally using cursor and asking you to explain things very simply.

00:07:38

Justin Germishuys: Okay. Um, okay. So, let me just have a look at the challenges.

Kiyasha Singh: There's nothing noted yet.

Justin Germishuys: Okay. So, there's nothing in the

Kiyasha Singh: This

Justin Germishuys: database

Kiyasha Singh: is just empty. Yes, because I recall you saying that I shouldn't try to add them there. Just have them hardcoded for now. So,

Justin Germishuys: um in the version that everybody else is seeing. Yes.

Kiyasha Singh: okay.

Justin Germishuys: Um, but obviously you're building a new one. with signin. Um, which if we can get it up and running would be great. Um,

Kiyasha Singh: Okay, I'll start populating the challenges and the images here.

Justin Germishuys: um, okay, let's just before we carry on and I send you off on the wrong direction, uh, let me just think this through.

Kiyasha Singh: Okay.

Justin Germishuys: You don't need a table for magic link tokens and you would never have needed it anyway.

Kiyasha Singh: Okay.

Justin Germishuys: Um largely because the token is the magic link is um automatically generated and expires after an hour.

00:08:52

Justin Germishuys: So there's no point in storing anything. And the same with OTPs. I do think you should switch to OTP for now. It'll be easier and it amounts to the same thing basically. Um, but you won't need a table for that. And then let me just look at activity logs. What do you need? What do you intend to put into activity logs?

Kiyasha Singh: So it's it's the what they did specifically and when they completed it. So for example, if they logged in but they didn't do it. This will mainly just track the admins.

Justin Germishuys: Okay.

Kiyasha Singh: Yes.

Justin Germishuys: So, just scroll back. I just want to see the columns. When you say the when you Okay. So, action type is text. Uh, scroll over detail. What goes under details?

Kiyasha Singh: Uh well, I'm not too sure yet. I

Justin Germishuys: Okay.

Kiyasha Singh: just have it sit there just in case I might add like, oh, they went into this.

00:09:53

Kiyasha Singh: They changed or deleted something.

Justin Germishuys: Okay, I just see that it's set up as a JSON B file format and or a a data format and so you may want to change that

Kiyasha Singh: Okay.

Justin Germishuys: depending on what your details look like. Okay. Um, so I think it's useful to have a general sense of what you what kinds of things you're writing to a database. But it's also useful that once you're in cursor and you're setting things up, you can say look, I want this thing to write to superbbase. Can you give me the lo most logical structure given the current implementation? And then what it can do is you can say can you please give me SQL that I can paste directly into the superbbase uh dashboard. And then it'll generally give you the SQL that sets up your tables perfectly at that point in at at that point. Otherwise, you end up it's good to conceptually say like what different tables you want so that you don't end up trying to make everything in one table.

00:11:03

Justin Germishuys: But the specific layout of that table needs to occur as you're building the thing. Does that make sense?

Kiyasha Singh: Yes.

Justin Germishuys: Cool. Um, okay. So, I see you've got challenge responses. Just click into that quickly. Sorry, I'm just going to go through each of these things.

Kiyasha Singh: Okay.

Justin Germishuys: Okay. So, each challenge will have its unique ID. So, basically, when you put challenges into the the challenges table, it's each individual challenge is going

to get its unique challenge ID, right?

Kiyasha Singh: Yes.

Justin Germishuys: And then when you say challenge responses, what does that mean? Does that mean that they've said yes, I've done it or that they've written a reflection?

Kiyasha Singh: They've written a reflection because all of the challenges will have reflection questions including the first one.

Justin Germishuys: Okay. But is there I suppose I haven't checked this myself. I've definitely been too busy elsewhere. When somebody's using um the cyborg habits interface, is there like a check box where they can say yes, I have done this challenge.

00:12:28

Kiyasha Singh: No, it automatically checks after they've opened and clicked on the completed button

Justin Germishuys: So the completed button. So basically my question is is there some way that the system signals that somebody actually completed the challenge?

Kiyasha Singh: with the completed button.

Justin Germishuys: Yeah. Okay, that's fine. And then where is that being stored?

Kiyasha Singh: I don't have that. Okay.

Justin Germishuys: Okay,

Kiyasha Singh: The time stamp uh is the only thing that's

Justin Germishuys: it's possible that somebody completed the challenge but they don't share a reflection which means that we need to know if people have said that they completed the challenge even if they haven't um cool I mean we can refine that behavior because sometimes times people can lie about finishing it and that's fine I suppose but I imagine everybody's just going to say yes they completed it and as long as they see the challenge it's already like a little bit of the work done so it's even if they don't actually do it but they kind of do it imaginatively that's still not nothing all right okay so you're going to need a you're going to need somewhere to put that

00:13:45

Kiyasha Singh: Okay.

Justin Germishuys: um and not in challenge responses because as I said you want to keep things that are conceptually different separate. Saying that you completed a

challenge, that is one set of data that we care about. Actually responding and reflecting, sharing a reflection is a different thing. And so it's unlikely that somebody's going to reflect if they didn't complete the challenge. But it is possible that they can complete the challenge and not reflect. So we do want to keep those points separate. Okay. So then basically the challenges will pull through. Then you have organization members. What does organization members do?

Kiyasha Singh: So, this just uh loads the organizations. So, if you have an organization, it'll list them. And then I haven't actually thought this one out properly, but supposed to have the people within the organizations loaded in.

Justin Germishuys: So I think an easier thing is you have you load a user on. So there needs to be like a user upload section

Kiyasha Singh: Yes.

00:15:03

Justin Germishuys: and then you need to give their email address. I assume you give their name, then you'll give their organization, and then you're going to give their cohort because you might have multiple instances of cyborg habits run within the same organization,

Kiyasha Singh: Okay.

Justin Germishuys: but and so you want to separate it up by cohort, but all of that information can be in a um a extended user um database so or table. So currently Superbase has its default user or table that you can't edit and that's cool. What you'll then do is you create another one and you'll call it um user profiles or whatever. You can decide what you want to call that as long as it means and in there you'll have the user ID that pulls through from the original table and then what you'll have is their email which you we'll have which can also pull through from the uh user table. So that's also fine. So we can have the user ID and the user email both automatically pull through.

00:16:30

Justin Germishuys: So you can actually set up your table to have a rule that it pulls information from another table.

Kiyasha Singh: Oh, okay.

Justin Germishuys: So you can also just ask cursor you can say look I want to create a

user profiles table. I need it to pull through the user ID and the user email address from that from the you know uh foundation user table in superbase. But I am going to have a an upload or a profile form for the users where I can add their username and their um organization and their cohort. And we can then split we can then filter or uh you know use logic to determine which users belong in which cohort based on the information in that table. Okay. So I think that that's pretty clear. So I don't think you need organization members and organizations. I don't think those tables really serve a purpose. I see you have profile. So that's probably where we would do it.

Kiyasha Singh: Yeah.

Justin Germishuys: Okay. And then you have the the serve the survey content in one table and then the survey responses.

00:17:57

Justin Germishuys: Now I wonder if you I think it's interesting. I don't know if you're going to store your surveys in this DB because where where where will the users be doing the surveys on the system under their profile? If that's the case, that's awesome. But often surveys are emailed. But

Kiyasha Singh: Okay.

Justin Germishuys: it is also possible that we could say, look, your challenge today is to complete the survey. And then we give them a link that takes them to the platform and then they do the survey there. And the nice thing about that is the survey responses table will also have the survey question. And so essentially the survey and the survey request survey responses are essentially in the same table because you can't really suddenly change the survey. it'll invalidate your survey responses table. Um, so I think again you can so basically I think we can probably get rid of organization members, organizations and the surveys tables and then everything else. What other things do we do from an admin point of view?

00:19:27

Kiyasha Singh: Um, so apart from adding users, uploading content in terms of like new habits or information. I don't think there's much else.

Justin Germishuys: Okay,

Kiyasha Singh: Viewing profiles, deleting.

Justin Germishuys: click back into challenge challenges quickly.

Kiyasha Singh: Okay.

Justin Germishuys: Okay, scroll to the right. Okay, so here's the thing. remember that we have a need to create um core challenges but also um organization and cohort specific challenges. So that makes me wonder earlier I said under profile you have um I think we do need a cohort table.

Kiyasha Singh: Okay.

Justin Germishuys: The reason we need a coord, okay, we just need to think this through. Essentially in the database or in the table the challenge needs to get a cohort ID as well because we might create cohort specific challenges and that cohort ID needs to be mapped to some cohort name that we agree on. Right? So it might be uh Ned Bank HR cohort 1, right? That might be our naming convention for then it might be Nedbank HR cohort 2 and then it might be Nedbank IT cohort 5. See there might be different there's organization department cohort depending on how we want to run it and then that gets its own unique identifier.

00:21:42

Justin Germishuys: So there's no reason that we can't use the name as a unique identifier but there's no guarantee that we're not accidentally going to duplicate it. So we need to make sure that we almost encode like a unique identifier for Nedbank HR cohort 5. So you can almost, you know, get from one to the other and then when you're um adding challenges for a specific cohort, you can say, "Okay, we want to add these seven challenges for the cohort." However, we need a way to identify which challenges we're selecting for the cohort and um which ones do not display. So, we need a way to almost select which challenges we're going to um display and which ones we're going to hide from the total pool. So, as we kind of go into this, you can kind of see that there's a a kind of an increasing complexity here. But now the problem is earlier I said to you that hard coding things is fine if we only have one client and one cohort because we need to buy time. But now we're seeing that um we're selling cyborg habits to lots of different people which means we're going to need this.

00:23:28

Justin Germishuys: In fact, it's already shifted from Pragma to I think Vitz for a roll out

in June, which means that we can't hardcode one for Pragma and for Vitz. We can't keep cloning the platform. So, I do think that um if we can, we need to try and resolve this in this week. Um now the thing is that if needs be I can do this I think quite quickly or at least get like the uh the code that kind of drives the behavior but have you you haven't yet figured about how to have you made anything that interfaces with Superbase yet?

Kiyasha Singh: No, not it.

Justin Germishuys: All right. Okay. So, that's something that we need to just look into. What are you using React?

Kiyasha Singh: Yes.

Justin Germishuys: All right. Okay. Okay. Let's just kind of close that.

Kiyasha Singh: Should I stop sharing?

Justin Germishuys: Um, no, it's all right. Well, you can.

Kiyasha Singh: Okay.

Justin Germishuys: Uh, a lot of what I've said about the databases holds.

00:25:01

Justin Germishuys: I think the only thing that we need to figure out is how are we going to get the challenges to pull through and the right one for the right cohort. So essentially what happens is conceptually a user logs in and if they've already been loaded on the system then we already know this is their user ID. From their user ID we can see this is the cohort that they belong to. Then once we know what the cohort is we know what challenges were selected for that cohort and then it has to pull those in. So that's basically the logic that we need to make provision for. Then obviously once it knows which challenges based on the unique challenge ids that are associated with the cohort, then it will go and look for each of those challenge IDs in the lookup, pull that content and then inject it into a card that you have on your screen. I just want to see is this just I'm just actually trying to see if um All right, let's just kind of go here cuz I think you and I need to understand this together.

00:26:34

Justin Germishuys: Basically, I've been working with Superbase in two ways and neither of them is right for what you're doing. Otherwise, we're going to end up creating a lot of complexity. In time, we might want to add like a server, but right now you've just been

building the website straight React. Nothing else.

Kiyasha Singh: React Jav a bit of JavaScript just for like the

Justin Germishuys: No, that's fine. React

Kiyasha Singh: Yeah.

Justin Germishuys: is JavaScript, so it's doesn't really change anything. It's just a framework built on top of JavaScript.

Kiyasha Singh: Okay.

Justin Germishuys: Um, so or TypeScript, which is also a variation of JavaScript, so it doesn't really matter. Also worth knowing, are you using React TypeScript or React JavaScript?

Kiyasha Singh: Yeah.

Justin Germishuys: you.

Kiyasha Singh: Take

Justin Germishuys: Okay, cool. So, that's good. So, let me share my screen and then I'll show you what I'm considering here. Okay, let's just see. I'm building a learning platform using React and TypeScript and I assume Vite. Okay. Essentially I want to link this to a superbase um database with a series of tables.

00:28:08

Justin Germishuys: The learning platform is such that we don't have any crazy logic um that needs processing from a backend server. Essentially what we want is a user to be able to log in. Then based on their unique user ID, it finds the content that is relevant for the cohort ID associated with them. So there also the content is stored in superbase as well. What is the most secure way given the setup to interact between our front end and superbase? I wonder if I should put on 03. What do you think? Advanced reasoning or let's just go for it. We can do both. Do you think this kind of thing would be helpful?

Kiyasha Singh: Kiss.

Justin Germishuys: Right. Okay. So that's why I was looking for rowle security because that's kind of the standard. So when you're asking cursor to generate your the SQL that you paste into superbase, ask it to just make sure that it has rowle security enabled.

Kiyasha Singh: Okay.

Justin Germishuys: Um, I don't know if you're familiar. Have how how much have you done with Superbase to date?

00:29:57

Kiyasha Singh: Uh, I've created I've dropped just the like inserting, removing manually doing it step by step because I didn't realize at the time that I could just get SQL and just plug it in and then I realized I could get SQL and then I've been doing that.

Justin Germishuys: All right. Um, so I think what I wanted to ask you is let's just This is the one I've been working with lately. All right. So when you say what you've said, have you been programmatically reading and writing to and from Superbase?

Kiyasha Singh: No, I've been using

Justin Germishuys: Okay. So I guess what I'm asking you is have you written any scripts or used any programming language that automatically connects to Superbase in any way

Kiyasha Singh: uh

Justin Germishuys: that looks for things and fetches it that

Kiyasha Singh: Yes.

Justin Germishuys: writes to it?

Kiyasha Singh: Yes. In cursor, but I don't pay close enough attention, so to say.

Justin Germishuys: Okay. So the answer then is unfortunately no. you haven't built anything that uh writes to cursor.

00:31:19

Justin Germishuys: Now, I just wanted to be clear on this because there are a couple of things you need to be aware of and I'm going to take you through that now so that you know it's always good practice to build a toy thing that does something. So, if you go into cursor and okay, so firstly you've created a project. Have you created any projects in Superbase? I presume you have because you just showed it to me. You can create another pro project if you want, just a test project because then what you're going to need to do is just set up like a I suppose if you have login at all in your current app and the login is connected to Superbase, then you're doing it. It's just you haven't had to do it consciously. All right, where to start? Let's just first see what this says and then I'll I'll actually walk you through it.

Kiyasha Singh: Okay.

Justin Germishuys: So it says below is the approach most teams use when they want 100% front-end only stack with superbase but still need production grade security.

00:32:34

Justin Germishuys: So use the public and non-API key in the browser, nothing else. Okay.

Kiyasha Singh: This

Justin Germishuys: So, where do you find the public and non-API key? It's over here. You go to project settings and you go over here to data API and then you're going to see there a nonpublic that's your a non API key. You will also need to use this URL so it knows exactly which you know superbase URL. So just to reiterate, you go to project settings, you go to data API, and the the two keys you mostly need is are this one and this one. The service ro secret is something you need if you want to have like an admin ability to overwrite a whole bunch of things. It's not usually something that you're you're going to use in something like this, I don't think. All right. So cool. And then Whoopsie. Then the second thing because you're using that it cannot bypass rowle security which means you need those keys in order to manipulate anything in the database.

00:33:44

Justin Germishuys: Now so leaking it is no worse than leaking your project URL. Never ship the service ro key or any key that can bypass RLS. So that's what I said the service ro key can bypass the security. So essentially what's happening is there's a kind of a an encryption that's taking place between your key. But I suppose you don't need to understand how that works. So where do you put it? Uh you'll probably just create a superbase client typescript file. You put it in there. Okay. So then let's superbase or issue the JWT and store it safely. So the superbases SDK keeps the access/refresh tokens in local storage by default. The more secure option is swap storage to at HTTP only cookies. So JavaScript can't read the token superbase providers helpers. Okay. So turn on rowle security. As I said when you ask it to generate the SQL um you can create the right policy for it to block all and then okay so let's just quickly see all right so I think that this is generally so the JWT token that can possibly introduce some complexity.

00:35:19

Justin Germishuys: Let me show you what that is. So, essentially it generates a JWT secret. The JWCT secret is a it's like a temporary token that's generated and then what it

does is it uses this token to communicate with the database, but the token expires.

Kiyasha Singh: All

Justin Germishuys: So what happens is periodically without the user knowing there's a cycle behind between JWT keys. So even if somebody can get a JWT token but they can't log in and that token expires they can't use it to alter or you know insert things or extract things from your database. Okay. So, the three things you're probably going to need is

Kiyasha Singh: right.

Justin Germishuys: the JWT token, the anon uh API key, and your API. So what I would like you to do just so that you know what's happening at least is create in cursor a very simple um react form and then create a very simple superbase table. And so essentially what you're going to want it to do is you can actually use it's a little toy problem to see if you can put user profile information in. Why is some child screaming in the background?

00:36:57

Justin Germishuys: It's dance class. I don't know what's okay. In any case, let's get back to this. Um, yes. And this shouldn't take you more than 5 or 10 minutes to do, just so that you have a feel for it. You create a basic uh React form. So, not multi- pages, no like hectic styling. That's not the point. Basically, what needs to happen is you can set up an OTP login just so that you've been through the steps to do that. have that OTP login generate your JWT token because as the user logs in um I think that's assigned.

Kiyasha Singh: Um, can I ask? So that means I'll need the token, the JWT, not the secret, but a token.

Justin Germishuys: You're going to need both. So what's going to happen is um let's if I open I don't want to create this because um if I were making it Okay, I would do maybe I'll just because we have a little bit of time. We do need to go through the you said the bugs are all easy to fix.

00:38:13

Justin Germishuys: So, I'm not going to worry too much about that. And then after this call, I need to just write those emails um or at least enough of them for it to be useful. All right. So, let's just actually So, I'll create a new folder. I'll call it um sort of React um superbase test. No, that's there's no reason to actually do it like this because the

underscores are not necessary. All right. So now the thing is um I don't want to I've already like run out of like free projects. So

Kiyasha Singh: Wait, how many free projects can we have?

Justin Germishuys: three. So we're going to have to get uh like a superb basease for team like a team superbase where we can all go in and make our projects. So we're going to have to migrate everything over to that anyway. And superbase isn't expensive. It's just uh why pay before you need to pay? We haven't hit that limit yet. But okay.

00:39:30

Justin Germishuys: Um, so let me just kind of I think I can probably take max off because that's not absolutely essential. Um, okay. Let's just I don't feel like typing. So let's just record. I would like to create a react and type script uh front end using vite. And what I essentially want is a form in which I can put in uh user information that can then write to a superbase um database. So right now don't worry about superbase just give me the vite setup and I would like a form in which I can put in the users's email address their username their the cohort that they belong to and their organization please okay I'm just using this for speech to text because cursor doesn't have it

Kiyasha Singh: Yeah.

Justin Germishuys: Um, okay. Okay. And then that will probably just make it. Um, and then in the meantime, uh, let me just kind of if I go into stride shift, um, I need to get into the stride shift super basease. Okay, let's Ah, sorry.

00:41:26

Justin Germishuys: I should just do that. Okay. Yes, proceed. Stop asking me so many questions. Okay.

Kiyasha Singh: Okay.

Justin Germishuys: Sorry. Okay, sorry. I'm just trying to um Okay, sorry. I just want to log into Superbase. Okay, that's the problem is I signed in with my field account, so I should probably get a stride shift one, too. Um, heat Okay, let's just create a new one. Um, user test. Um, okay. Generate a new project. Okay. Sorry. Okay. Now it needs to create components. Okay, it's creating a user form component. Okay. Okay. So, here I have May as well just close this. Okay. So, this is where it's created. I can probably get um So, I'm not going to suggest you do what I'm about to do, but nonetheless, uh I'm going to do it

anyway. So, I'm going to go to data API.

00:44:15

Justin Germishuys: Okay. So, all of that stuff is ready. Okay. to run. I assume this is more or less how you do it.

Kiyasha Singh: Yeah. The only thing that I actually didn't quite understand was the JWT token because I kept running into timeout errors and things for a while and I didn't understand what was going on.

Justin Germishuys: Yeah. So, I think that's kind of where it has gotten sticky for me. Uh, I think I got stuck on it for quite some time at one point and then everything I've been working with lately, I've already just gotten it right and so I haven't had to worry about it again. So, I don't know if um Right. So basically if it's created this uh or once it's created this um I'll just try to implement the uh the superb base stuff but basically the fact is cursor can probably just guide you through it and just say but what I would like you to do so that you probably don't have to watch me do the whole thing and you can rather just go for it self.

00:45:38

Justin Germishuys: Um, so this this brings me to the other thing that we do need to just settle on is how the challenges pull through. So not how the challenge is I think that's easy. What is I think worth figuring out is you almost need to set up a cohort. So you almost need a screen or a form where you add a cohort. And then that's why I said I think we need it's an our own separate cohort table. So we can set up the cohort. Then for the cohort we can also set up the organization. And then we need a section where we say okay uh here is a list of all of the relevant challenge ids that we've selected or it is even possible that we we've we might have a page where you can almost see all the challenges and then you can select select which challenges you want to include up to a maximum of like 20 or 30 or whatever and then they immediately get added to a list that is stored in a row in your cohort table.

00:47:14

Justin Germishuys: So then whenever the system needs to know which challenges do I need to pull through, it'll say okay let's go and look at the cohort table. Let's go and look at which challenges and they are perhaps the ids. I don't know whether it's best to do comment separation or to create like a python list or to get like a JSON maybe a JSON object actually seems like the best option there. So essentially it'll write to like a JSON B uh type column and then we will have a list of all the challenge ids. Then what happens is um the various components will be populated based on the sequence of those ids and it'll basically go and say okay let's go to look in the challenge um table for the content and then there will certainly be if you're going to use superbase storage that's the other thing superbase storage is here also pretty easy to set up but you're going to need to put your challenge images in here because you're probably going to set it up so that for new challenges, you're going to have different images.

00:48:33

Justin Germishuys: So, what you can do is you can upload your images, but then the URL to that image gets placed

Kiyasha Singh: I don't

Justin Germishuys: in that row with the challenge. So you have your challenge ID, you have your challenge copy, you have the URL to where that image is stored in your storage bucket and then when the card populates it'll say ah okay let's pull this content and we put it on the right let's pull this image and we put it on the left and then that's typically how that works. Um I can show you something that I've made that uses that. Let's see. Although no, I have to spin it up then. Okay. So, does any of this help you move forward in any way?

Kiyasha Singh: It does. So, in terms of how you mentioned, what I was planning on doing, I think is not too far off from what you said, but you've added more steps to it, so to say. So, it's added some stuff that I was still overseeing.

00:49:37

Kiyasha Singh: So I think yeah so far I have steps that I can follow and actually try to do

Justin Germishuys: Cool. Um, look, if you do get totally stuck, what you can always do is

you can give me um you can push to GitHub or you can give me your um files, your front-end files if you want to. And then if you get stuck with the back end, you know, I can always just sit down over the weekend or whatever and get that over the line if you get like woefully stuck.

Kiyasha Singh: Okay.

Justin Germishuys: But the other thing that we need to do is it's one thing, so I think we've dealt with it is to say, okay, we've managed to select which challenges we're going to show to the cohort. Now, this is the other thing that you and I haven't necessarily resolved completely. Let me just stop presenting. Oh, wait. Hold on. Let's just see how this looks. Okay. So, this is kind of what it built. It's basically all that we're going to need to set up like a Now, the thing is that I suppose we also have bulk upload.

00:51:15

Justin Germishuys: So, nobody wants to type in one name and hit submit and type in another name and hit submit. So we do need to think, okay, if we have like 300 people, we don't want to do it one at a time. We want to upload some sort of a CSV and have it automatically read it and populate it. So if we can get a CSV in the right format, we upload the CSV and then that basically loads everybody. Um and then there'll be edge cases where in the CSV we don't get the cohort name or the organization name or there's some uh oh aside from organization we also need to know their department or business unit

Kiyasha Singh: Yeah.

Justin Germishuys: um so that we can get a breakdown like that. Right. Um so what's next? Do you think any of this seriously interferes what you've with what you've already built? Or do you think that what you've already built can still hold, but can now just connect to this or can be adapted for this?

Kiyasha Singh: uh well, if you're talking about the link that everyone has, it's broken.

00:52:27

Kiyasha Singh: Everything will not work. Okay, not everything, but there's a lot of things that might not work. Whereas the one that I'm working on now, if we do this now once, I think it'll be fine.

Justin Germishuys: All right. The thing is that I I want you to kind of I know that we're

getting towards the end of the month and we're making a commitment about when this is going to be done and we need to try and stick to that as far as possible because you know people are making promises to companies and if we make them wait they'll go cold and we'll lose the opportunity. So there is a little bit of end pressure to just get this done. So let's just kind of talk quite frequently during the course of the week.

Kiyasha Singh: Yes.

Justin Germishuys: Um and in the morning every day just give me kind of an update on where you're at, what you're struggling with. I think that you the first version that you created that everybody's playing with gives you a very clear sense of the direction of what to build.

00:53:36

Justin Germishuys: But the rebuild is an opportunity to do everything right and we do need to do it with this kind of database in mind. We need to say okay where do we have a place where we can now granted we can upload challenges via SQL we don't actually need so separately without a you know a page for it we could say look here is my CSV so you basically create another applet until you've created that section in your main app. You can create a little applet that also connects to Superbase using the same credentials, but you say or maybe not even you can decide, but something where you can say here's a SQL thing. Here's the information. Please write the SQL or you know, here's a CSV. Please write the SQL that uploads all of this to Superbase. And if you're not ready to build that page yet, we can still get people in to the system and still have that interface with your main app. So, it's not the end of the world. Cool.

00:54:55

Justin Germishuys: Okay. But I think for now, you just need to experience what it's like setting up a little applet or a form that writes to Superbase. I think that that's like the first thing you really need to experience. Otherwise, it's not going to make as much sense. So, don't kind of proceed with too many other things before seeing that you can get this done. And if you want to continue developing without getting stuck with the OTP or the magic link or whatever, you can just temporarily switch that off. But it is nice to have it because we're going to need it for production. It's nice to know how you get from login save that bit of information and how that bit of that bit of information connects to

all the pages and pulls things through. Okay. I don't think there's anything else. I think you can just keep running and then we need to just probably speak a few more times this week.

Kiyasha Singh: Okay, hold.

Justin Germishuys: Oh. Um I just wanted to ask you one last question.

00:56:05

Justin Germishuys: Is everybody feeling quite positive about the challenges? Nobody's complained about them.

Kiyasha Singh: No, the only thing that would be problematic in terms of the challenges is from David's point of view, there was just one challenge that had incorrect grammar. So, but apart from that, they say they understand. Yeah. And also with some of the DAB students that were doing the testing, they said they understood it. And these students are students that don't really understand English. So, I'm not sure if they're lying or, you know,

Justin Germishuys: So I think okay so on the one hand yes I'm interested to know if they understood it that's super important on the other hand did they try it and then have an experience that was useful for

Kiyasha Singh: I'm not sure with the DAC team, but with the students, they two of them specifically responded. one did all of the explaining challenges and his feedback was that he actually uses it more often than not compared to before. And then the other student didn't really go into much detail.

00:57:18

Kiyasha Singh: So I'm not too sure if they did do the challenges itself.

Justin Germishuys: Okay. So, that's the next thing that we need to sort out. So, if we're going to launch this, we need to figure out who our first major client is and then we need to create the the challenges. But I think as we saw before, we actually have a lot of really usable challenges. I think what we need to do is probably just And we can worry about this over time. So, right now, we just need to get a platform that works and where we can dynamically update the content.

Kiyasha Singh: Yeah.

Justin Germishuys: Um, I don't know about the videos. Yes, we're going to need one for

videos.

Kiyasha Singh: Oh, yes.

Justin Germishuys: We're

Kiyasha Singh: Yes.

Justin Germishuys: going to need a we're going to need a a table for for videos. And then that link, if we're going to keep using YouTube, it actually doesn't matter what service we use. Ultimately, what's important is that we have a table where not only do we have the links to the challenges and for the challenges a link to the URL where the images exist, we are going to need cohort level uh control over the v videos as well.

00:58:41

Justin Germishuys: But we do need to set some logic so that it goes to our default videos. So, let's assume that the videos we have now are the default videos, but then we might have Spanish or a different avatar or whatever the case may be. So, I think that what we're going to need is a video a video um table and we're going to need the video URL, the habit that it connects to, a description of the video or maybe some columns like language, avatar type, and then it will have the name of the avatar so that we can kind of see like what all the variables are for the video. Typically, it'll just be like the avatar and the language for now, but we definitely need that. And then what happens is the component that pulls through the video on the various pages. We'll also have to go and fetch that link from the DB and then pull it through. So, that won't be hardcoded either.

Kiyasha Singh: Okay. Okay. So I'm thinking that we might have a quite a few tables because if we are going to do this based on cohorts also the type of text that displays will have to change.

01:00:10

Kiyasha Singh: So for example if it's for corporate or really big companies more professional jargon will be used whereas if it's students in DAB and some of their English levels are really not great they'll have to be simplified.

Justin Germishuys: Yeah. So I think then you're right. We should but look we we can have columns for it for now. We can anticipate that we need the columns and we can say look um language style corporate informal we can have language level let's not go

too overboard with that for now because I think the default videos are already accessible given what you've said

Kiyasha Singh: Yeah.

Justin Germishuys: so let's take the def let's uh keep the default for now but we can always extend end the table to add those extra fields and then um that way we can also track what we've created and we can see what options we have. So I think that on the whole most of the database architecture is probably captured on this transcript

Kiyasha Singh: Yes.

Justin Germishuys: and also the requirements for role security. um the use of JWT tokens, all of that stuff. So I think that um you could probably also take this transcript at some point and then distill it and then use that to direct at least in some part how you propose at the very least you can turn it into a diagram as well and then you can have a diagram of your DB schema. Awesome. hopefully get them done within the next two hours.

Kiyasha Singh: Okay.

Justin Germishuys: Cool. Okay Kiyasha. Thank you for your time. It's always good. Talking to you and it's good to connect and work together on this. Okay.

Kiyasha Singh: Nostalgic.

Justin Germishuys: Nostalgic,

Kiyasha Singh: Thank you.

Justin Germishuys: I believe it's not gonna be nostalgia anymore. Hopefully it'll become the norm, okay? Bye sure, right.

Transcription ended after 01:02:39

This editable transcript was computer generated and might contain errors. People can also change the text after it was created.