

Agora

En digital community-plattform för
dialog om jämställdhet i akademien



F. Betul Demir Korkmaz

FED24

2026-03-06

Abstract

Detta examensarbete undersöker hur en digital plattform kan utformas för att stödja dialog och erfarenhetsutbyte kring jämställdhet i akademien. Arbetet resulterade i en konceptuell community-plattform, Agora, som möjliggör anonym publicering och respektfull diskussion. Genom användarcentrerad design och modern webbutvecklingsteknik utvecklades en prototyp som visar hur digitala verktyg kan sänka tröskeln för deltagande i känsliga samtal. Resultatet visar att kombinationen av anonymitet, strukturerad identitetshantering och intuitiv design kan bidra till ökad inkludering och trygghet i digitala diskussionsmiljöer.

Tack

Jag vill rikta ett stort tack till min handledare, Dr Jasmina Maric, för hennes stöd och vägledning under arbetets gång. Ditt engagemang och dina reflektioner har varit värdefulla genom hela processen, särskilt när projektet ställde krav på svåra avvägningar. Tack för ditt stöd och för de samtal som har bidragit till både projektets och min egen utveckling.

Innehåll

Abstract	1
Tack	2
1. Inledning	4
2. Metod	5
2.1 Designprocess	6
2.2 Teknisk metod och verktyg	6
2.2.1. Motivering av teknikval	7
2.3 Arbetsprocess	8
2.4 Val och utvärdering av teknik	8
3. Genomförande	9
3.1 Avgränsning	9
3.2 Översikt av systemet	9
3.3 Datamodellering och databas	10
3.4 Next.js och applikationsstruktur	10
3.5 Frontend-implementation	11
3.6 Autentisering och säkerhet	11
3.7 Framtida utveckling	12
3.8 Resource Hub	12
3.9 AI-integration med Gemini	12
4. Resultat med analys	13
4.1 Slutresultat	13
4.2 Analys av genomförandet	14
5. Referenser	16
6. Appendix	17

1. Inledning

Jämställdhet inom akademien är en aktuell och viktig fråga, både internationellt och i Sverige. Trots att många initiativ och handlingsplaner finns, kan det vara svårt för individer att veta hur de konkret kan bidra till förändring i sin egen vardag. Vid Chalmers tekniska högskola pågår initiativet GENIE (Gender Initiative for Excellence), vilket har fungerat som en inspirationskälla för detta arbete.

Tidigare forskning visar att hur digitala system uttrycker ton och personlighet har stor påverkan på användarens emotionella upplevelse, särskilt i känsliga sammanhang. I en masteruppsats inom interaktionsdesign undersökte Månström och Nilsson (2025) hur två olika chatbot-personligheter, en handlingsorienterad och en medkännande, påverkade användares upplevelse i kontexten av könsojämlikhet [2]. Resultaten visade att en empatisk och emotionellt validerande ton upplevdes som mer stödjande och meningsfull än en enbart lösningsfokuserad kommunikation. Studien lyfter även att användare ibland vänder sig till AI-baserade system för emotionellt stöd när institutionella strukturer upplevs som otillräckliga. Detta understryker vikten av att designa digitala lösningar med fokus på emotionell trygghet och respektfull dialog, samtidigt som sådana verktyg bör ses som komplement snarare än ersättning för organisatoriskt ansvar.

Tidigare arbete vid Chalmers har dessutom visat att digitala resurser för jämställdhet riskerar att bli underutnyttjade när de är texttunga och saknar interaktiva element. I masteruppsatsen *Genie Action Toolbox* av Chau och Lou (2024) betonas behovet av mer engagerande och användarvänliga lösningar för att öka individens benägenhet att faktiskt agera, exempelvis genom tydlig informationsarkitektur, interaktiva funktioner samt fokus på förtroende och inkludering [1]. Detta visar att det finns ett behov av digitala verktyg som inte enbart informerar, utan också aktiverar och engagerar användare.

Utifrån dessa utgångspunkter syftar detta examensarbete till att utveckla en digital community-plattform, Agora, som utforskar hur digitala verktyg kan stödja dialog och erfarenhetsutbyte kring jämställdhet i akademien. Plattformen är ett konceptuellt förslag och är inte kopplad till någon specifik organisation.

Namnet Agora är inspirerat av den antika grekiska mötesplatsen där medborgare samlades för samtal och diskussion, vilket speglar plattformens ambition att fungera som en trygg digital mötesplats för dialog och reflektion.

Målet är att sänka tröskeln för att dela erfarenheter genom att erbjuda möjlighet till anonym publicering, samtidigt som användare kan välja att använda smeknamn och avatar. Plattformen är inspirerad av forumstrukturer liknande Reddit, men med fokus på respektfull dialog och konstruktiva samtal. Genom projektet har målet även varit att fördjupa kunskaper inom fullstack-utveckling med Next.js och Supabase, samt att praktiskt tillämpa principer inom interaktionsdesign och användarcentrerad utveckling.

Projektets huvudsakliga frågeställning är:

Hur kan en digital plattform utformas för att på ett enkelt och inkluderande sätt stödja dialog och erfarenhetsutbyte kring jämställdhet i akademien?

Arbetet genomförs under handledning av Dr Jasmina Maric, universitetslektor vid Interaction Design and Software Engineering, Institutionen för Data- och informationsteknik (CSE), Chalmers tekniska högskola i Göteborg.

2. Metod

Projektet genomfördes som ett design- och utvecklingsprojekt med en tydlig fasindelad struktur. Arbetet inleddes med en omfattande designfas där applikationens övergripande koncept, funktionalitet, användarflöden och gränssnitt definierades. Wireframes och designförslag togs fram och presenterades för handledaren.

Efter att ha mottagit återkoppling reviderades designen för att säkerställa att struktur, användarupplevelse och funktionalitet var väl genomtänkta innan implementationen påbörjades. Denna process bidrog till att minska osäkerhet och skapa en tydlig riktning inför utvecklingsfasen.

När designen hade fastställts påbörjades implementeringen av applikationen. Utvecklingen genomfördes med fokus på att skapa en stabil, skalbar och underhållbar arkitektur. Kodstrukturen organiserades med tydlig komponentuppdelning och separering av logik och presentation.

Efter att implementationen slutförts genomfördes en övergripande utvärdering av resultatet i relation till projektets mål och ursprungliga designbeslut.

Versionshantering skedde med Git och GitHub för att säkerställa spårbarhet och en strukturerad utvecklingsprocess.

2.1 Designprocess

Designarbetet inleddes med en fördjupande litteraturgenomgång, där tidigare forskning inom området studerades, inklusive arbeten av handledaren. Syftet var att skapa en teoretisk förståelse för digitala verktyg i kontexten jämställdhet och emotionellt stöd, samt att identifiera relevanta designprinciper.

Parallellt genomfördes benchmarking av befintliga community-plattformar och forumstrukturer. Fokus låg på att analysera återkommande strukturer, interaktionsmönster och funktioner såsom anonym publicering, trådstruktur och filtrering.

Designförslag utvecklades därefter i Figma genom wireframes och interaktiva prototyper för att visualisera användarflöden och gränssnitt. Processen var iterativ i designfasen, där idéer diskuterades löpande med handledaren, som har expertis inom interaktionsdesign. Förslagen presenterades, utvärderades och reviderades i flera omgångar innan implementationen påbörjades.

Designen utvecklades med fokus på:

- Låg tröskel för deltagande
- Möjlighet till anonym publicering
- Enkel och tydlig navigering
- Respektfull och inkluderande visuell ton

De slutgiltiga prototyperna fungerade som direkt underlag för implementationen i Next.js.

2.2 Teknisk metod och verktyg

Utvecklingen av applikationen har baserats på moderna webbutvecklingstekniker med fokus på struktur, skalbarhet och underhållbarhet.

Följande tekniker och verktyg har använts:

- **Next.js** som ramverk för frontend och routing
- **TypeScript** för typkontroll och ökad kodkvalitet

- **Tailwind CSS** för responsiv och konsekvent design
- **Supabase** (PostgreSQL) för databas och datalagring
- **Git och GitHub** för versionshantering
- **Figma** för design och prototypframtagning

2.2.1. Motivering av teknikval

TypeScript valdes för att minska risken för typrelaterade fel och skapa en mer robust och lättunderhållen kodbas. Genom statisk typkontroll kan många vanliga fel upptäckas redan under utvecklingsfasen, vilket förbättrar kodkvalitet och långsiktig underhållbarhet i större applikationer [3].

Supabase valdes som backend-lösning eftersom det är en open-source backend-plattform byggd ovanpå PostgreSQL och erbjuder integrerade funktioner för databas, autentisering och realtidskommunikation [4]. Detta möjliggör snabb utveckling utan att behöva konfigurera och underhålla en egen servermiljö, samtidigt som den relationella databasen ger tydlig struktur och hantering av relationer mellan tabeller.

Next.js valdes på grund av dess moderna arkitektur och stöd för både server- och klientkomponenter. Ramverket erbjuder filbaserad routing, server-side rendering samt automatisk koduppdelning, vilket bidrar till god prestanda och skalbarhet [5]. Detta gör det särskilt lämpligt för strukturerade och interaktiva webbapplikationer.

Tailwind CSS användes för att säkerställa en konsekvent och responsiv design. Som ett utility-first CSS-ramverk möjliggör det snabb utveckling av gränssnitt utan omfattande egen CSS-struktur, vilket minskar risken för stilkonflikter och förbättrar kodens läsbarhet [6].

Git och GitHub användes för versionshantering för att säkerställa spårbarhet och strukturerad kodhantering. Versionskontroll är central i moderna utvecklingsprojekt eftersom det möjliggör samarbete, historikspårning och återställning av tidigare versioner vid behov [7].

Figma användes för design och prototypframtagning för att visualisera användarflöden och gränssnitt innan implementation. Genom att testa och iterera designbeslut i ett tidigt skede kunde användbarhetsrelaterade och strukturella frågor hanteras innan utvecklingsfasen påbörjades [8].

Integration av en extern språkmodell valdes för att undersöka hur AI kan fungera som ett kompletterande stöd i känsliga dialogmiljöer. Gemini API möjliggör generering av kontextanpassade textbaserade svar genom ett tillgängligt utvecklargränssnitt [9].

2.3 Arbetsprocess

Projektet har genomförts enligt en förenklad agil arbetsmetod. Arbetet har delats upp i mindre funktionella delar (features), vilka har planerats och implementerats stegvis. En Kanban-baserad struktur har använts för att organisera uppgifter i kategorier såsom:

- To Do
- In Progress
- Done

Denna metod har möjliggjort kontinuerlig avstämning mot projektets omfattning och tidsram. Särskilt fokus har lagts på att avgränsa projektet till en Minimum Viable Product (MVP) för att säkerställa att en fungerande och stabil version av applikationen kan färdigställas inom given tid.

2.4 Val och utvärdering av teknik

I ett tidigt skede övervägdes olika backend-alternativ, såsom Firebase och Supabase. Supabase valdes på grund av:

- Relationell databas (PostgreSQL)
- Tydlig hantering av relationer mellan tabeller
- Enkel integration med Next.js
- Inbyggd autentisering

Valet av teknik har baserats på balans mellan funktionalitet, utvecklingstid och framtida skalbarhet.

3. Genomförande

I detta avsnitt beskrivs den praktiska implementeringen av plattformen. Den färdiga prototypen omfattar kärnfunktioner såsom inläggsflöde, anonym publicering, kommentarsfunktion, interaktioner (gilla och "helpful"), enkel profilhantering samt integration av Resource Hub och AI-stöd.

De tekniska och strukturella lösningarna avseende systemöversikt, datamodellering, applikationsarkitektur, frontend-implementation samt autentisering och säkerhet diskuteras mer utförligt i följande delavsnitt.

3.1 Avgränsning

Eftersom en digital community-plattform kan bli mycket omfattande, har projektet avgränsats till en tydlig MVP (Minimum Viable Product). Det innebär att fokus ligger på kärnfunktionerna:

- Feed med inlägg
- Skapa inlägg (med anonymt läge)
- Kommentarsfunktion
- Hashtags och filtrering (Den dynamiska hashtag-visualiseringen visas i bild 3.)
- Enkel profil med smeknamn och avatar

Avancerade funktioner såsom notifieringar, bokmärkning eller sparade inlägg och mer omfattande moderationsverktyg planeras som en möjlig framtida version (Version 2) men ingår inte i denna första implementering.

3.2 Översikt av systemet

Projektet resulterade i en webbaserad community-plattform där användare kan skapa inlägg, kommentera och delta i diskussioner kring jämställdhet i akademien. Plattformen är utformad för att möjliggöra anonym publicering, samtidigt som användare kan välja att skapa en profil med smeknamn och avatar. Systemet är mobilanpassat (mobile-first) men fungerar även på större skärmar.

Plattformen är tillgänglig online för demonstration på:

<https://agora-connect.vercel.app/>

3.3 Datamodellering och databas

Databasen är implementerad i Supabase (PostgreSQL). Följande centrala tabeller har skapats:

- **profiles** – lagrar användarens smeknamn och avatar
- **posts** – lagrar inlägg
- **comments** – lagrar kommentarer kopplade till inlägg
- **post_helpfuls** – lagrar användares interaktioner (t.ex. “helpful”)
- **post_likes** – lagrar användares interaktioner (t.ex. “like”)

Relationer mellan tabeller har implementerats genom främmande nycklar, exempelvis:

- `comments.post_id` → `posts.id`
- `comments.author_id` → `profiles.id`

3.4 Next.js och applikationsstruktur

Projektet utvecklades med en komponentbaserad arkitektur där varje funktionell del abstraherades till återanvändbara React-komponenter. Detta möjliggjorde effektiv kodstruktur, minskad duplicering samt enklare vidareutveckling. Denna strategi bidrog till både utvecklingshastighet och långsiktig underhållbarhet.

Applikationen är byggd med Next.js App Router och strukturerad enligt en tydlig mappindelning:

- Sidor (routes) – `app/`
- Komponenter – `components/`, organiserat i undermappar:
 - `ui/` för återanvändbara UI-element (exempelvis Button, Input, Modal och Avatar)
 - `feed/` för inläggsrelaterade komponenter
 - `layout/` för strukturkomponenter
 - `icons/` för ikonkomponenter
- Databaslogik och TypeScript-typer – `lib/`

Genom att strukturera applikationen kring återanvändbara komponenter, både på UI-nivå (exempelvis Card, Button och Modal) och på funktionsnivå (såsom PostCard, PostThread och CreatePost), skapades en modulär arkitektur som underlättar vidareutveckling och underhåll.

3.5 Frontend-implementation

Frontend är utvecklad med React-komponenter och Tailwind CSS. Återanvändbara UI-komponenter har skapats för att säkerställa konsekvens i design och funktionalitet. Exempel på centrala komponenter är: PostCard CommentItem Sidebar WordCloud ProfileCard ThemeToggle Plattformen stöder ljust och mörkt tema via en toggle-knapp i navigeringsfältet. Temabytet implementeras genom att CSS-variabler skrivs över direkt på dokumentelementet med inline-stilar, vilket garanterar korrekt beteende oavsett specificitetsordning i CSS. Det valda temat sparas i localStorage och laddas automatiskt vid nästa besök. Genom att abstrahera funktionalitet i separata komponenter har koden blivit mer strukturerad och lättare att vidareutveckla.

3.6 Autentisering och säkerhet

I denna version används en förenklad identitetsmodell istället för traditionell autentisering. Användaren skapar ett smeknamn och väljer en emoji-avatar, varefter det databasgenererade id-värdet tillsammans med profildata sparas i webbläsarens localStorage (Profilskapandet visas i bild 2). Detta möjliggör låg tröskel för deltagande utan krav på e-post eller lösenord.

Row Level Security (RLS) är aktiverat i Supabase för samtliga tabeller. Policies är dock i nuläget öppna eftersom Supabase Auth inte är implementerat. I avsaknad av autentiserade användarsessioner kan systemet inte särskilja åtkomst baserat på användaridentitet. Strukturen för åtkomstkontroll är därmed tekniskt etablerad, men striktare regler och fullständig autentisering behöver införas i en framtida version för att möjliggöra faktisk begränsning av läs- och skrivåtkomst per användare.

Användare kan:

- Skapa inlägg
- Kommentera
- Markera inlägg som hjälpsamma
- Gilla inlägg genom att klicka på en hjärtikon

Anonym publicering påverkar endast presentationen i gränssnittet och inte den faktiska datalagringen, vilket möjliggör grundläggande moderering och framtida spårbarhet inom systemet.

3.7 Framtida utveckling

För att säkerställa genomförbarhet inom projektets tidsram har fokus lagts på en fungerande MVP.

Funktioner som identifierats för en eventuell vidareutveckling inkluderar:

- Avancerad moderering
- Notifikationssystem
- AI-baserad rådgivning
- Statistik och analysverktyg

Den modulära arkitekturen gör det möjligt att vidareutveckla systemet efter projektets avslut. En begränsning i projektet är avsaknaden av användartester i slutskedet. På grund av tidsramen prioriterades implementation och rapportskrivning. Detta innebär att användarupplevelsen inte har validerats empiriskt, vilket bör beaktas vid framtida vidareutveckling.

3.8 Resource Hub

Agora innehåller även en sektion kallad "Resource Hub" (Resource Hub-sektionen illustreras i Appendix, bild 1.), där användare kan ta del av artiklar och externa resurser kopplade till jämställdhet i akademien.

Syftet med denna funktion är att kombinera dialog med kunskapsstöd. Medan forumet möjliggör erfarenhetsutbyte, erbjuder Resource Hub strukturerad information som kan stärka användares handlingsförmåga. Funktionen är avsedd som ett komplement till diskussionsdelen.

3.9 AI-integration med Gemini

För att komplettera den användardrivna dialogen har en AI-funktion implementerats genom integration med Googles Gemini API (AI-integration visas i bild 4). Funktionen möjliggör att användare kan ställa frågor relaterade till jämställdhet i akademien och erhålla generella, kontextuellt anpassade svar genererade av en språkmodell.

AI-komponenten fungerar som ett orienterande stöd och är avsedd att sänka tröskeln för deltagande, särskilt i situationer där användare upplever osäkerhet inför att formulera eller publicera ett inlägg i forumet.

Tekniskt sett skickas användarens fråga till Gemini API via en serverfunktion i Next.js, varefter svaret returneras och presenteras i användargränssnittet. Ingen permanent lagring av AI-konversationer sker i denna version av systemet.

Det är viktigt att betona att AI-svaren är generella och inte ersätter professionell rådgivning, juridisk expertis eller institutionella processer.

4. Resultat med analys

Resultatet från Agora kan relateras till tidigare arbete kring digitala jämställdhetsverktyg. *Genie Action Toolbox* beskriver att textbaserade resurser ofta leder till låg användning och att interaktiva och engagerande element behövs för att stärka användares motivation och handlingskraft. [1] I Agora har detta omsatts genom ett forumliknande upplägg och funktioner som underlättar deltagande även i känsliga diskussioner.

4.1 Slutresultat

Resultatet av projektet är en fungerande webbaserad prototyp av en anonym community-plattform för dialog kring jämställdhet i akademien. Plattformen möjliggör för användare att:

- Skapa och publicera inlägg anonymt
- Kommentera och interagera med andra användare
- Använda och filtrera innehåll via hashtags
- Navigera i en feed med sorterade diskussioner
- Se en dynamisk hashtag-visualisering (word cloud) baserad på aktivitet

Prototypen är utvecklad med Next.js och TypeScript samt integrerad med Supabase för datalagring. Designen har tagits fram i Figma och implementerats med fokus på tydlig struktur, återanvändbara komponenter och responsivitet.

Projektets ursprungliga ambition var att inkludera ytterligare funktionalitet såsom rapporteringssystem, sökfunktion, redigering och borttagning av inlägg samt förbättrade navigationskomponenter (exempelvis breadcrumb och empty state-hantering). På grund av tidsramen och behovet av att prioritera kärnfunktioner implementerades dessa inte i den första versionen utan planeras för en framtida version (Version 2).

Bidraget i detta arbete ligger inte i att skapa ännu ett forum, utan i att undersöka hur en strukturerad och tekniskt genomförbar plattform kan designas för att stödja dialog kring jämställdhet inom högre utbildning.

4.2 Analys av genomförandet

Resultatet visar hur en sådan plattform kan konstrueras och implementeras tekniskt. Effekten på inkludering och upplevd trygghet har dock inte empiriskt utvärderats inom ramen för detta arbete.

En central del av genomförandet innebar etiska överväganden kring användarprofiler, anonymitet och moderering. I dialog med handledaren diskuterades balansen mellan att skapa en så fri och anonym miljö som möjligt, samtidigt som ansvar och säkerhet behöver kunna upprätthållas i ett community-system. Beslutet att initialt prioritera anonymitet och låg tröskel för deltagande var ett medvetet designval, motiverat av plattformens syfte att möjliggöra delning av potentiellt känsliga erfarenheter.

Detta innebar dock att funktioner såsom rapporteringssystem och mer avancerade moderationsverktyg inte implementerades i denna version. Dessa identifierades som viktiga framtida utvecklingsområden, men krävde ytterligare etiska och tekniska överväganden som inte rymdes inom projektets tidsram.

Att projektet resulterade i en fungerande prototyp beror till stor del på en tydlig komponentbaserad arkitektur och en iterativ arbetsprocess. Genom att arbeta strukturerat med återanvändbara komponenter (exempelvis PostCard, CommentItem och WordCloud) skapades en stabil och skalbar kodbas.

Samtidigt uppstod tekniska utmaningar, särskilt kring autentisering, datamodellering och hantering av anonymitet i relation till användarkonton. Dessa moment krävde mer tid än förväntat och bidrog till att vissa funktioner fick prioriteras ned.

En avgörande faktor för att projektet kunde slutföras inom tidsramen var den tidiga avgränsningen till en MVP samt en strikt prioritering av kärnfunktioner. Genom att implementera och testa funktionalitet stegvis kunde tekniska problem identifieras tidigt, vilket minskade risken för omfattande omstrukturering i senare skeden.

En viktig lärdom är att community-baserade system innebär komplexa sociala, etiska och tekniska överväganden. Att möjliggöra anonymitet samtidigt som ansvar och trygghet upprätthålls är en balansgång som kräver genomtänkt design och tydliga framtida modereringsstrategier.

5. Referenser

1. Chau, A., & Lou, Y. (2024). *Genie Action Toolbox: Designing an interactive digital toolbox for academia to increase user actions toward gender equality*. Master's thesis, Interaction Design and Technologies, Chalmers University of Technology / University of Gothenburg. Hämtad <https://odr.chalmers.se/items/761c43e8-8d50-4424-b360-f42739869cf0>
2. Månström, O. & Nilsson, T. (2025). *Understood and supported by AI*. Master's thesis, Interaction Design and Technology, Chalmers University of Technology.
3. Microsoft. *TypeScript Documentation*. Hämtad 2026-02-21 från <https://www.typescriptlang.org/docs/>
4. Supabase. *Supabase Documentation – Introduction*. Hämtad 2026-02-21 från <https://supabase.com/docs>
5. Vercel. *Next.js Documentation*. Hämtad 2026-02-21 från <https://nextjs.org/docs>
6. Tailwind Labs. *Tailwind CSS Documentation*. Hämtad 2026-02-21 från <https://tailwindcss.com/docs>
7. Git. *Git Documentation*. Hämtad 2026-02-22 från <https://git-scm.com/docs>
8. Figma. *Figma Design Overview*. Hämtad 2026-02-22 från <https://help.figma.com>
9. Google. *Gemini API Documentation*. Hämtad 2026-02-22 från <https://ai.google.dev>

6. Appendix

Bild 1. Resource Hub-sektionen i mörkt tema.

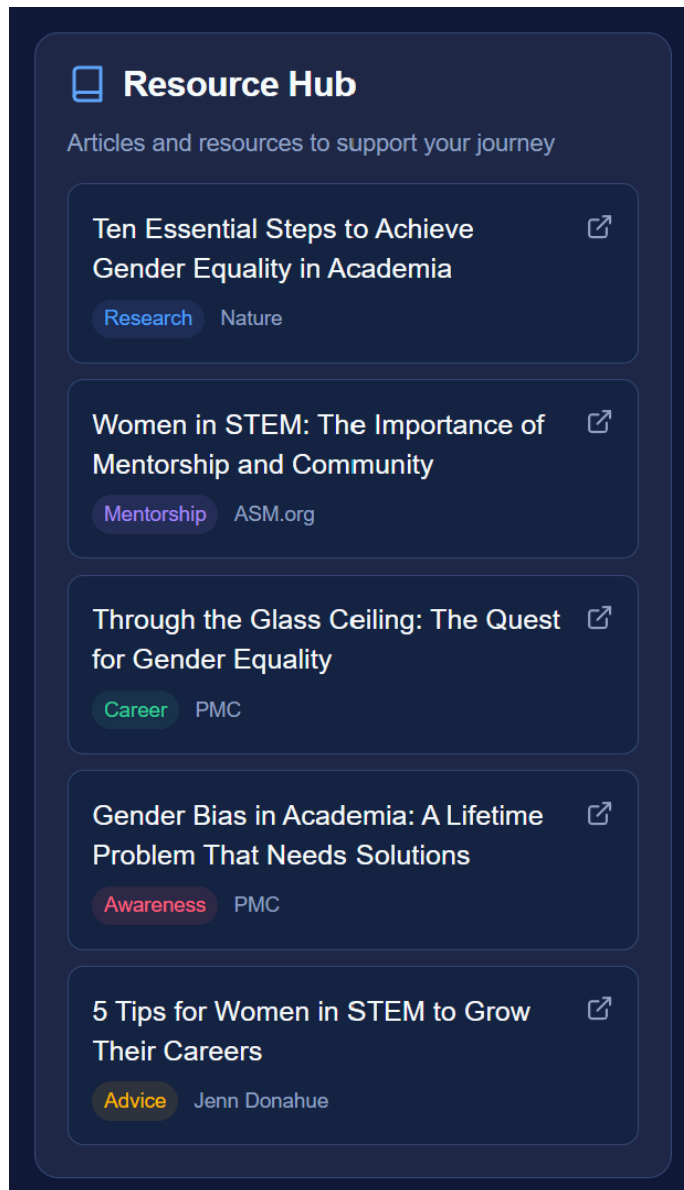


Bild 2. Profilskapande med smeknamn och emoji-avatar.

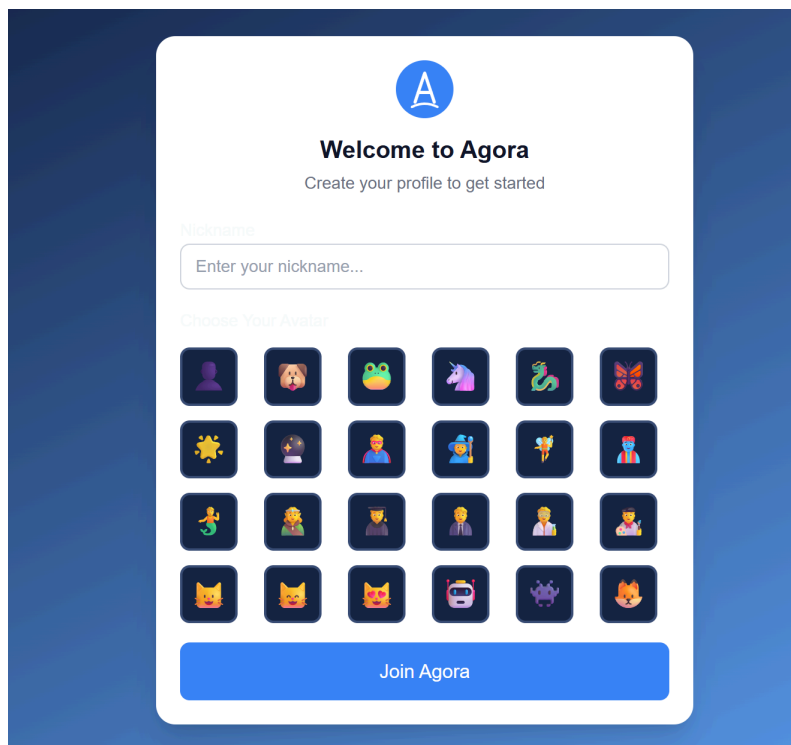


Bild 3. Trending Topics med dynamisk hashtag-visualisering.

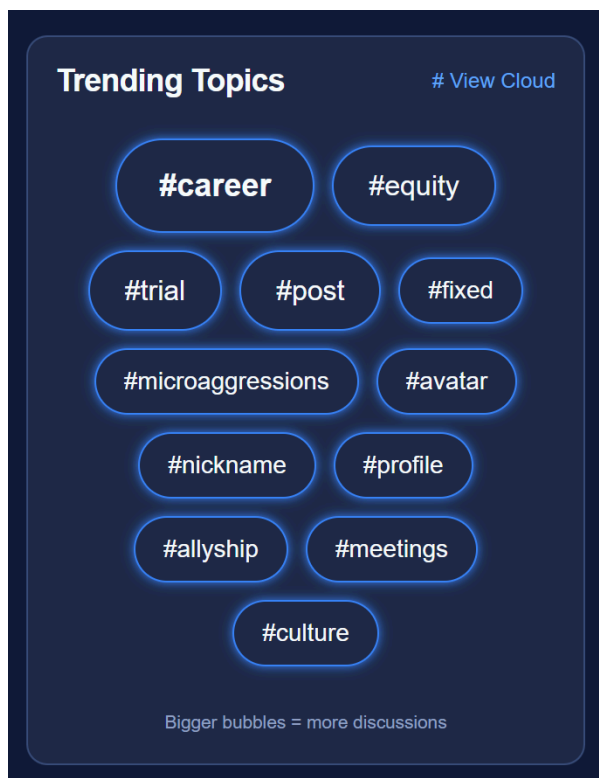


Bild 4. AI-integrationen via Gemini API, implementerad som en frågekomponent i gränssnittet.

