

# ESOL Communication App

Sam Treadwell

2/20/2025

## Introduction

This app's UI is inspired by a modern social media platform tutorial, leveraging cutting-edge technologies for a seamless experience. It is built with **Next.js 15** and incorporates:

- **React Query** for efficient data fetching
- **Infinite loading** for smooth scrolling
- **Optimistic updates** (e.g., instant follow/like button toggles)
- **Cache mutation** for real-time updates
- **Server components & actions** for features like:
  - "For You" vs. "Following" feeds
  - Profile-specific posts

The backend is powered by **PostgreSQL** with **Prisma ORM** for easy database queries, and **Lucia Auth** for secure authentication.

For styling and UI components, the app uses:

- **Tailwind CSS** for a sleek, responsive design
- **shadcn/ui** for fully customizable UI elements
- **React Hook Form** for intuitive form handling

Additionally, it supports **file uploads via UploadThing**, with:

- Custom URL masking on the backend
- Image resizing is restricted to app-generated files
- Integration with **Amazon S3** for cloud storage
- Vercel's limit of **5,000 image resizes per month**

Finally, the app is deployed on **Vercel** and integrates the **Streams Chat SDK** for real-time messaging.

---

## Frontend (Client-side)

### 1. Next.js 15

- React framework that enables **server-side rendering (SSR)**, **static site generation (SSG)**, and **API routes**.
- Next.js 15 likely includes improved **Server Components and Actions**, making data fetching more efficient.

### 2. React Query

- A powerful data-fetching and state-management library.
- **Handles caching, background syncing, and automatic refetching** to keep UI data fresh.

### 3. Infinite Loading

- A UX pattern that **loads more content dynamically** as the user scrolls (e.g., social media feeds).
- Implemented with **React Query** and `useInfiniteQuery()`.

### 4. Optimistic Updates

- A technique where the UI **immediately updates** before the server confirms the change.
- Improves UX by making interactions feel faster and more responsive.
- Often paired with **mutating the cache** in **React Query**.

### 5. Mutating the Cache

- Instead of waiting for the server response, we **update the local cache immediately** to reflect changes in the UI.
  - Ensures better performance and responsiveness.
-

## Backend (Server-side)

### 6. Server Components

- Next.js feature that **renders React components on the server** and sends minimal JavaScript to the client.
- Improves performance and reduces client-side load.

### 7. Server Actions

- A newer **Next.js feature that simplifies server-side logic** inside client components.
- Handles form submissions, data mutations, and API interactions without needing API routes.

### 8. PostgreSQL

- A **relational database** is used for storing structured data (users, posts, comments, etc.).
- Known for scalability, transactions, and strong data consistency.

### 9. Prisma ORM

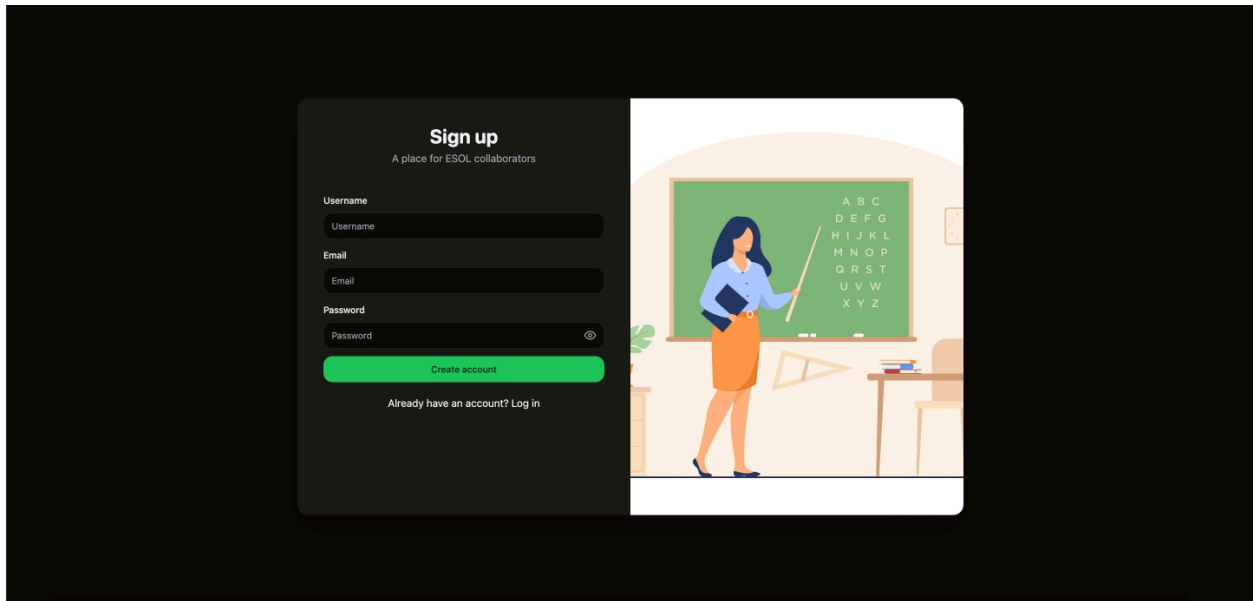
- An **Object-Relational Mapper (ORM)** for working with PostgreSQL in **TypeScript**.
- Provides an easy-to-use API for querying and managing database records.
- Generates **TypeScript types** for safe database interactions.

### 10. Lucia Auth

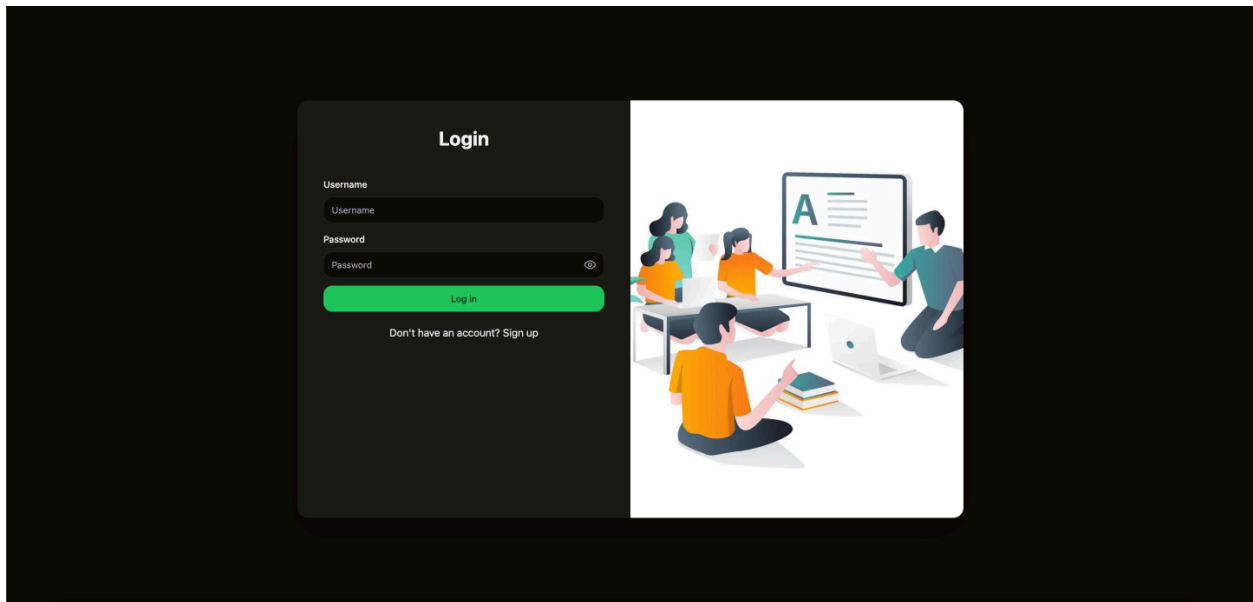
- A lightweight **authentication library** for handling user sign-in/sign-up.
  - Works with **PostgreSQL** and integrates well with Next.js.
  - Provides session management and security features.
-

## Login/Authentication

### Sign up



### Login



Login and authentication are achieved by leveraging [Lucia Auth](#). This approach is beneficial because it works well with PostgreSQL and integrates easily into Next.js projects. It also allows for enhanced control over custom styling. Passwords are hashed in the database for security using [argon2](#).

Lucia **does not use JWT tokens** but instead relies on **sessions stored in the database** and a **secure cookie** in the user's browser.

## Lucia Auth

- A lightweight **authentication library** for handling user sign-in/sign-up.
- Works with **PostgreSQL** and integrates well with Next.js.
- Provides session management and security features.

### User Signup/Login Process

- A new **user** is stored in the **users** table in **PostgreSQL**.
- A **session** is created in the **sessions** table and linked to that user.
- A **secure, httpOnly cookie** (containing the session ID) is sent to the user's browser.

### How Authentication Works

- When the user makes a request (e.g., visiting a protected page), the browser sends the **cookie**.
- The server checks the **sessions** table in **PostgreSQL** to see if the session is valid.
- If the session exists and matches the cookie, the user is **authenticated**.

### Handling Logout

- The session entry in the **sessions** table is **deleted**.
- The cookie in the browser is **cleared**.

## Why Lucia Auth?

### 1. Low-Level and Flexible

Lucia Auth is a **low-level authentication library**, meaning it doesn't impose a rigid structure on your application. Instead, it provides **fine-grained control** over authentication logic, allowing you to define how users are created, authenticated, and managed within your database.

### 2. Database-Backed Sessions (No JWT Tokens)

Unlike many authentication systems that rely on **JWTs (JSON Web Tokens)**, Lucia uses **database-backed sessions** for handling authentication.

- When a user logs in, a **session is created** in the database.
- A **corresponding session cookie** is stored in the user's browser.
- On each request, Lucia checks if the **session in the database** matches the **session cookie**, verifying the user's identity.

This approach ensures **better security** and **simplifies session management** compared to JWT-based authentication.

### 3. Sessions Are More Secure and Easier to Use Than JWTs

JWTs are **stateless**, but can be **difficult to implement securely** because:

- Many tutorials suggest using **one long-lived JWT token**, which is **not secure**—if an attacker steals it, they can impersonate the user indefinitely.
- You **can't force a logout** with a JWT alone. Even if you delete a JWT from your database, a user with a valid token can still make requests until the token expires.

In contrast, **sessions are easier to manage and inherently more secure**:

- If a user logs out, **you only need to delete their session from the database** to invalidate their authentication immediately.
- If an attacker steals a session cookie, **you can revoke that session instantly**, ensuring better security.
- **No risk of token misuse** or needing complex token revocation strategies.

### 4. Better Control Over Authentication

With Lucia, authentication is tightly integrated with your database, allowing:

- **More control over user roles, permissions, and expiration policies**
- **Stronger security practices**, such as limiting session duration or enforcing re-authentication
- **Easier integration with existing databases** without the need for external authentication services

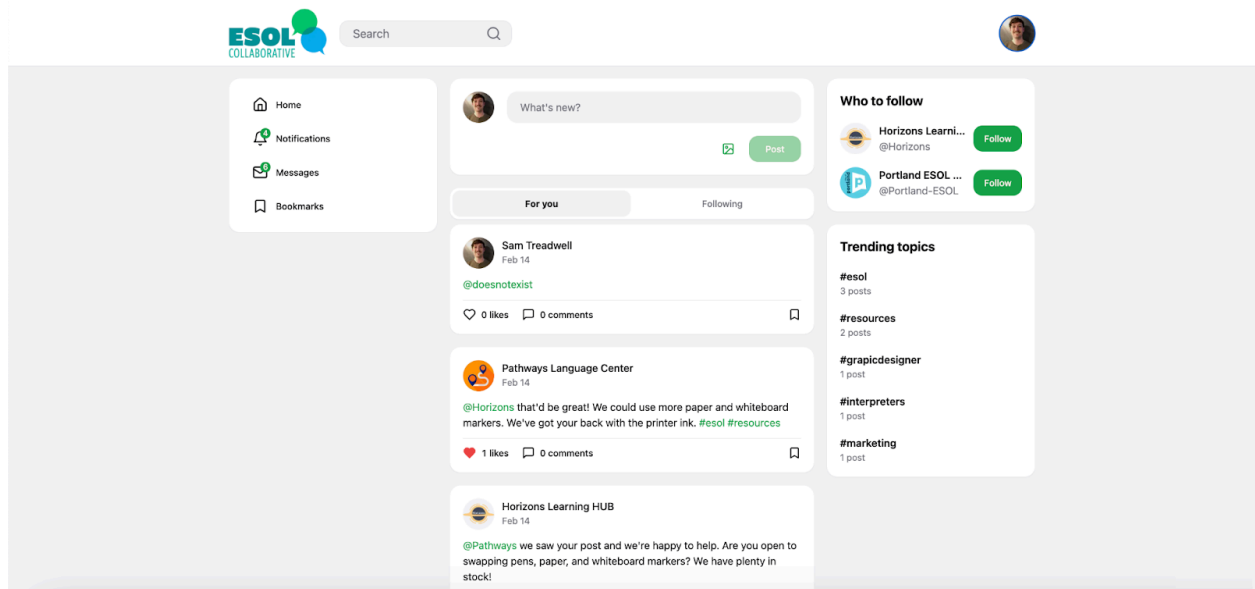
### Conclusion

Lucia Auth provides a **simpler, more secure alternative to JWT authentication** by relying on **database-backed sessions** instead of **long-lived tokens**. It allows **better session control, immediate logout functionality, and stronger security**, making it an excellent choice for modern web applications.

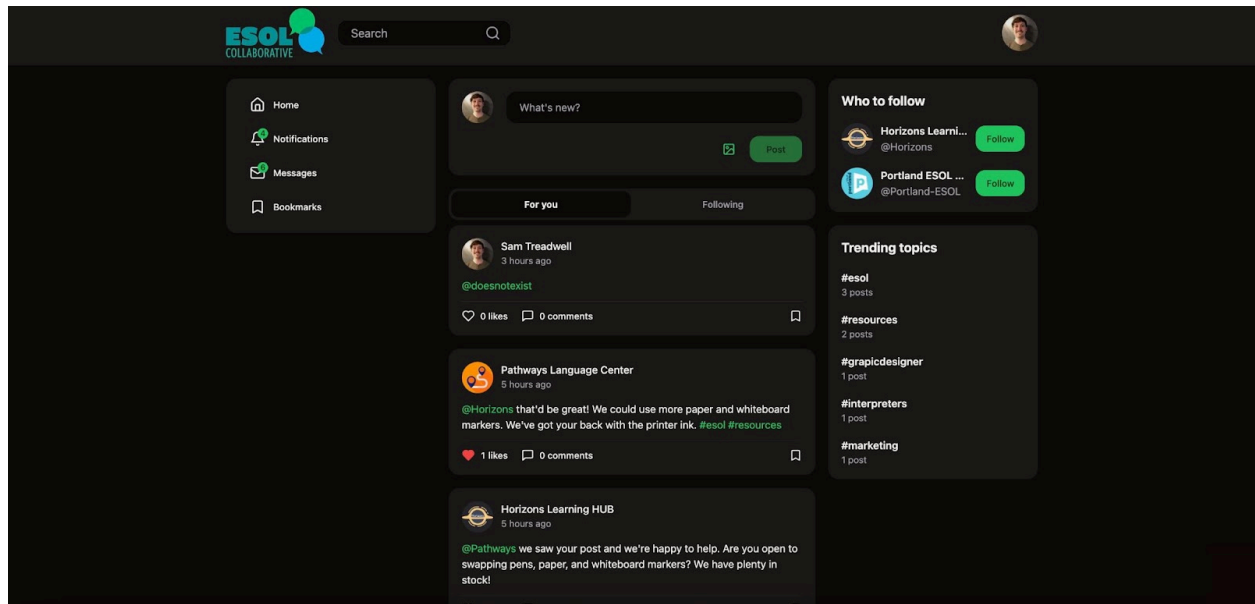
---

# User Interface

## Light Mode



## Dark Mode



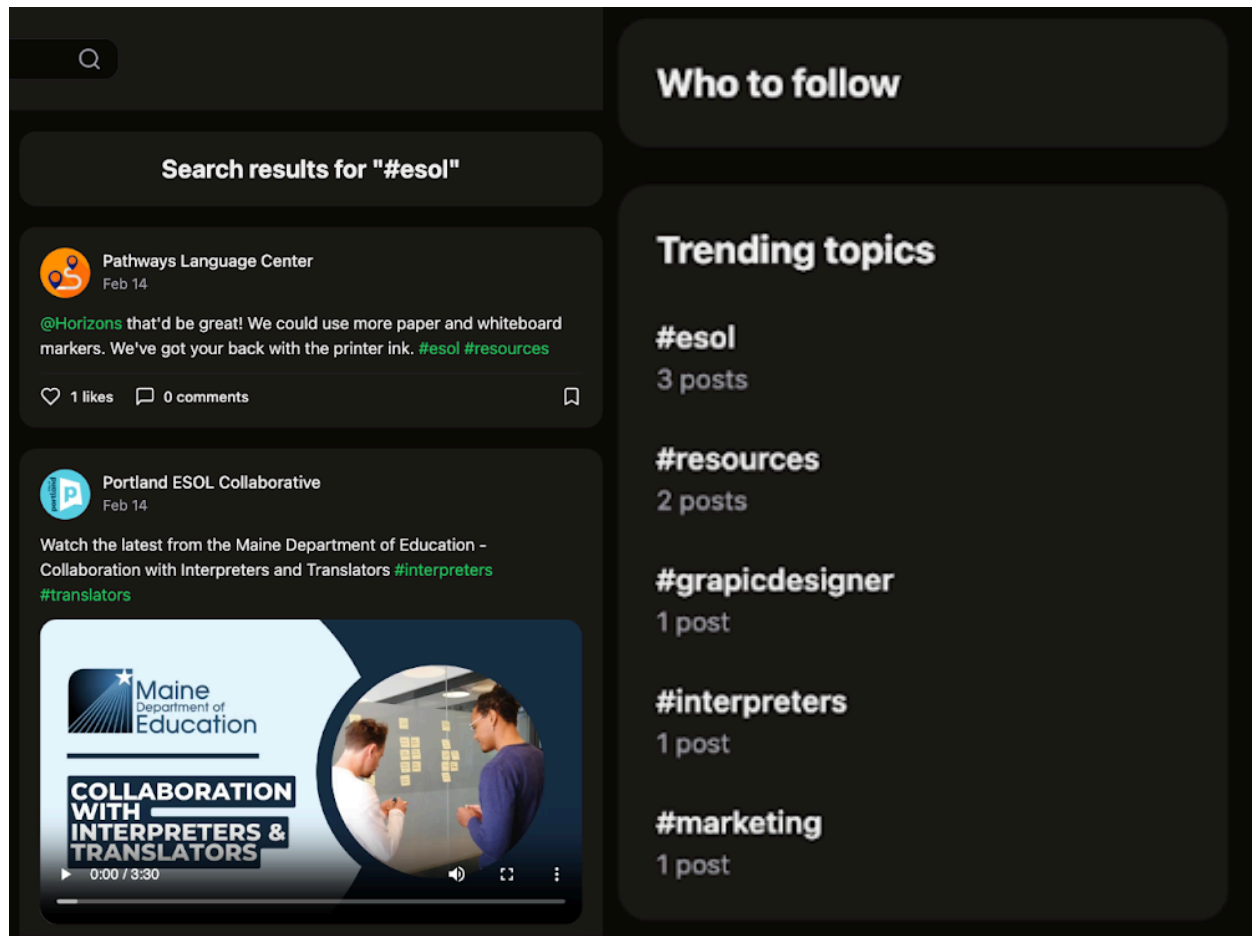
## Profile Page

The screenshot shows the profile page for 'Horizons Learning HUB' (@Horizons) on the ESOL Collaborative platform. The header includes the ESOL Collaborative logo, a search bar, and a user profile icon. The left sidebar contains navigation links: Home, Notifications, Messages, and Bookmarks. The main profile section displays the 'Horizons' logo, the name 'Horizons Learning HUB', the handle '@Horizons', and a bio: 'Welcome to Horizons! We're a community-based initiative offering a variety of youth programs, including ESOL classes to prepare the next generation with the language skills they need to enter the local workforce'. It also shows 'Member since Feb 14, 2025' and 'Posts: 5 Followers: 2'. To the right, a 'Who to follow' section lists trending topics: #esol (3 posts), #resources (2 posts), #graphicdesigner (1 post), #interpreters (1 post), and #marketing (1 post). Below the profile, a section titled 'Horizons Learning HUB's posts' shows a post from Feb 14 by @Pathways, which says: '@Pathways we saw your post and we're happy to help. Are you open to'.

## Notifications

The screenshot shows the 'Notifications' page on the ESOL Collaborative platform. The header is identical to the profile page. The left sidebar remains the same. The main content area is titled 'Notifications' and lists three notifications. The first two are from 'Sam Treadwell' (Sam Treadwell liked your post), with the first notification mentioning a new class schedule for Professional Writing. The third notification is from 'Pathways Language Center' (commented on your post), asking for a good graphic designer. The right sidebar, containing 'Who to follow' and 'Trending topics', is also visible and identical to the profile page.

## Trending Topics



The trending topics feed in the right sidebar is updated hourly, ensuring users stay informed about the latest trends. Clicking on a hashtag redirects users to a dedicated page that filters posts containing the selected keyword (e.g., clicking on **#esol** brings up all posts featuring 'esol').

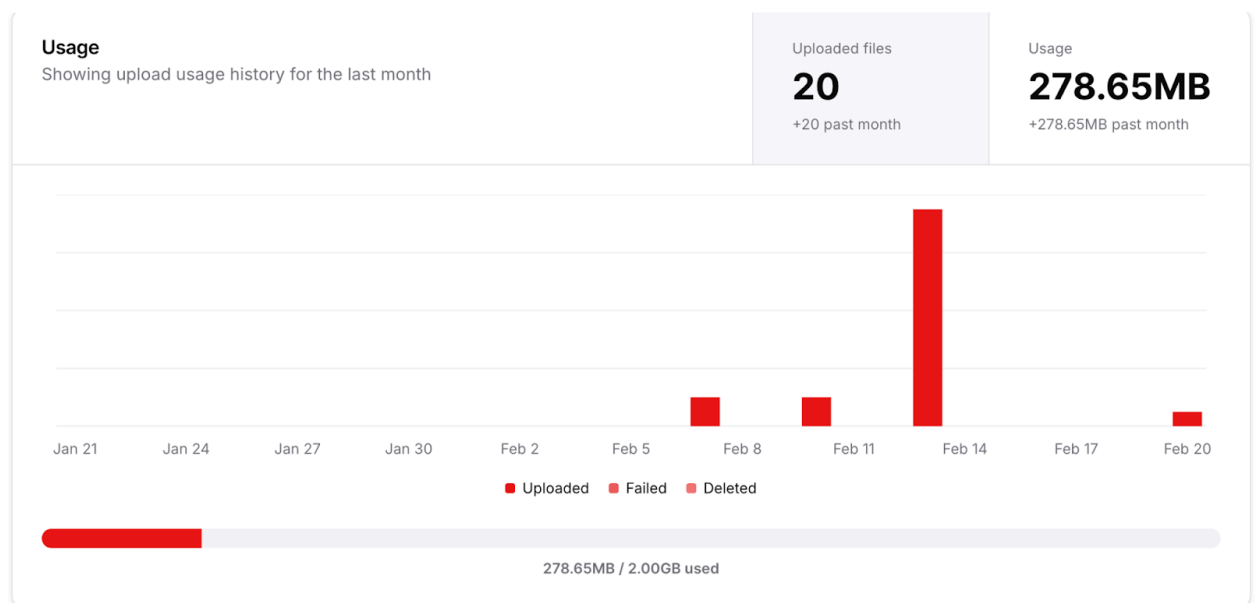
---

## UploadThing

<https://uploadthing.com/>

The current file limit is set to allow a maximum file size of **4MB for images**, and up to **64MB for videos**.

File uploads for posts are supported via media attachments that can be achieved by clicking the upload media button or drag-and-dropping media onto a post. A daily cron job automatically clears any unused or orphaned media files from UploadThing.



Recent Uploads	Largest Files
attachment_5c3475a4-e8c4-4fa1-9897-fefde... Feb 21, 2025, 7:...	attachment_4d1cfa5b-a764-4e15-b2a9-5c4e439bffe... 60.86...
attachment_e7fd0ee9-697a-4b55-bf09-f500d... Feb 14, 2025, 10:...	attachment_36679db8-fe92-49be-abe6-83efe83acdbe.... 57.47...
attachment_4d1cfa5b-a764-4e15-b2a9-5c4e4... Feb 14, 2025, 9:...	attachment_4b4badfa-f2d5-4de1-b371-7e4e90f238b9.... 57.47...
attachment_0f647a95-096a-4234-96a7-0848... Feb 14, 2025, 9:...	attachment_464dd99d-67ee-44e8-abe2-f4ed9e311113.... 48.03...

## **Streams Chat SDK**

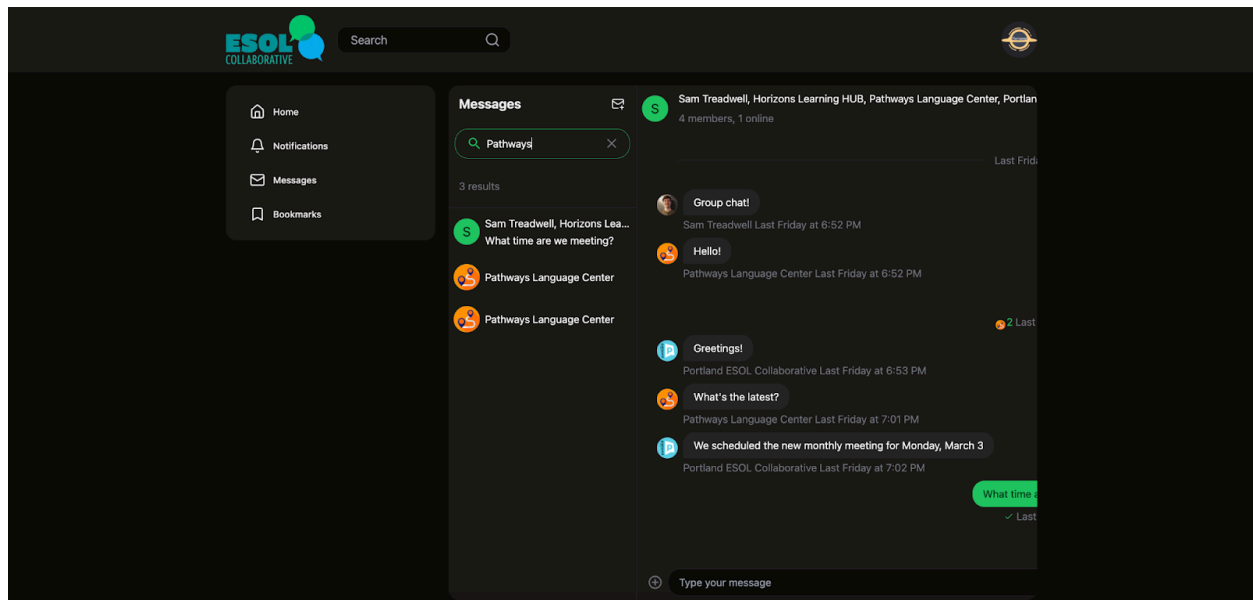
### [Chat SDK - Real-time In-App Messaging SDK](#)

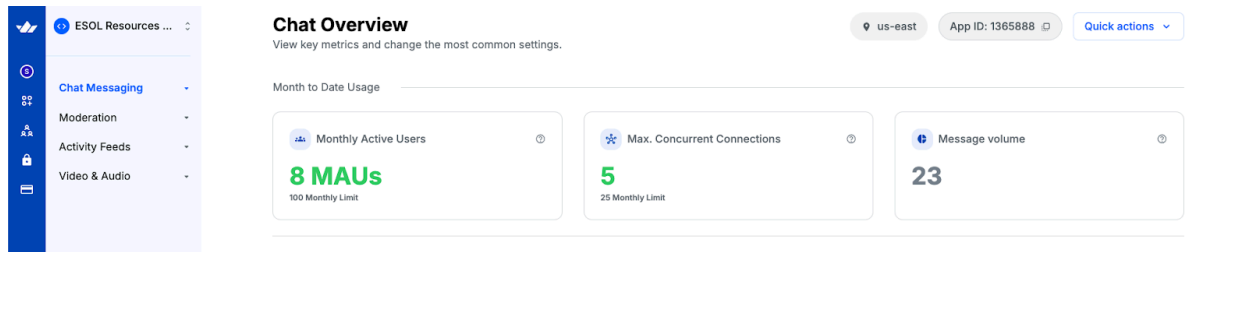
The **Stream Maker Account** offers free access to Stream's Chat API for eligible small teams and early-stage projects. This plan includes the **Chat Startup Plan**, modified to support up to **2,000 Monthly Active Users (MAUs)** and **100 peak concurrent connections** per month. These hard limits ensure you won't encounter unexpected overage charges.

#### **Eligibility Requirements:**

- Team size: **5 or fewer members**
- Funding: Less than **\$100,000**
- Revenue: Less than **\$10,000 per month**

Stream Chat API enforces a default character limit of **5,000 characters per message**.





## Vercel

Vercel is a powerful hosting platform optimized for **frontend applications**, particularly those built with **Next.js**. It provides **fast deployments, automatic scaling, and seamless integration** with various backend technologies, including **PostgreSQL** with **Prisma ORM**. Here's how Vercel enhances both **hosting** and **database management**.

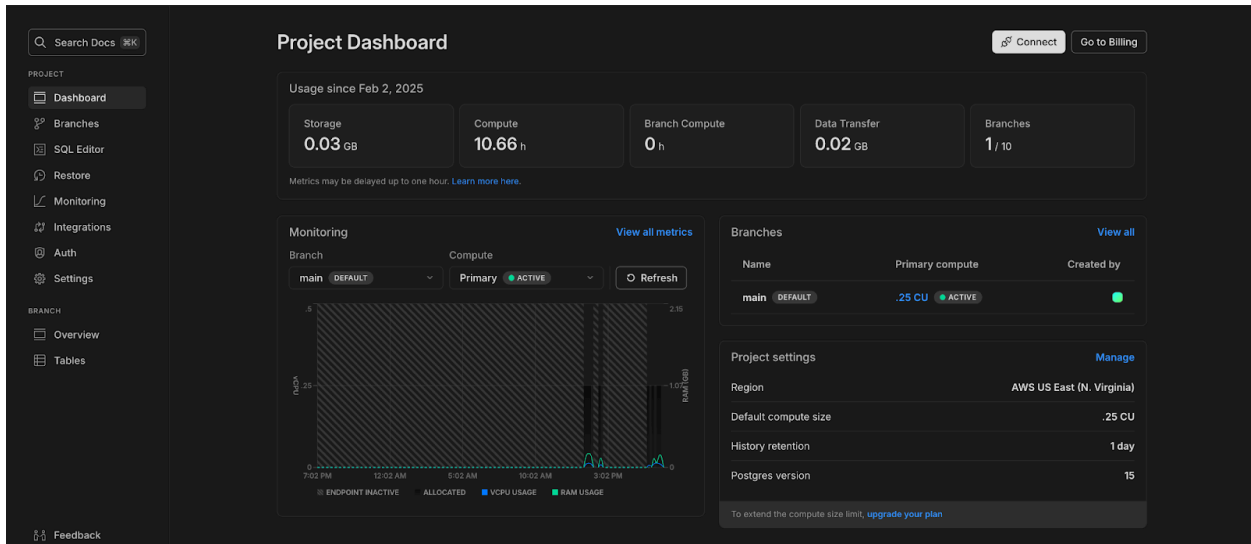
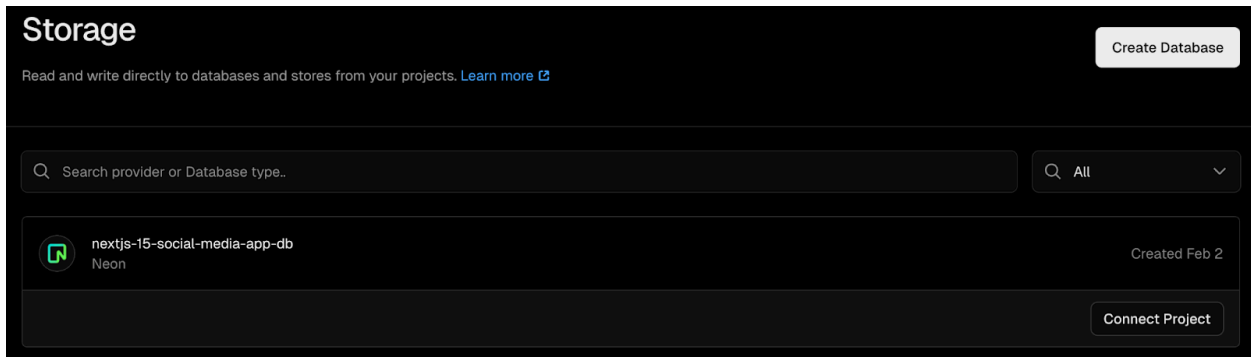
### Vercel Hosting Benefits

- **Optimized for Next.js** with seamless integration, automatic static optimization, and Edge Functions for fast execution.
- **Global CDN & auto-scaling**, ensuring low latency and high performance.
- **Instant deployments & Git integration**, allowing automatic builds and preview deployments for each pull request.
- **Secure by default** with built-in HTTPS, SSL, and DDoS protection.

### PostgreSQL + Prisma on Vercel

- **Managed database connections** via Vercel Postgres, Supabase, or Neon, with Prisma ORM simplifying queries and schema management.
- **Serverless-friendly** with Prisma's connection pooling and Data Proxy to optimize database performance.
- **Fast & efficient** queries using Next.js 15 server actions and React Query for real-time updates.
- **Reliable and scalable**, supporting automatic backups, migrations, and edge functions for reduced database latency.

Vercel's **serverless architecture, automatic scaling, and CDN** make it a top choice for modern web apps, while **PostgreSQL with Prisma** ensures robust, efficient, and scalable database management.



## Works Cited

- [Build A Full-Stack Social Media App With Next.js 15 \(React Query, Lucia Au...](#)
- <https://github.com/codinginflow/nextjs-15-social-media-app>

**Vercel PostgreSQL** - Easy to set up, and the first database is free.

**Prisma ORM** - A library that makes it easy to query the database. Avoids raw SQL; instead, use pre-defined functions.

**npx prisma init**

**prisma.schema** - Creates a schema to define tables in the database

**.env** - Contains sensitive credentials; database passwords

**Lucia Auth** - Lower-level than Next Auth. The developer writes the logic that creates a user in the database. Lucia offers more flexibility and the ability to customize the login forms. With Lucia, Users and Sessions are added to the database schema. Sessions are added every time a user is logged in creating a session in the database and a corresponding cookie in the browser of the user. Only if both parts match, if there is a session in the db, and the user has the corresponding cookie with the correct key in the browser, then the user is authenticated and can login.

**Server components, server actions (actions.ts)** - A Convenient way to write backend code in Next.js. Create functions that create POST endpoints that are callable via POST requests. Create a new user.

**import { z } from "zod";** is how you import the main validation object from the **Zod** library, a TypeScript-first schema validation tool.

**Zod** is a type-safe schema validation library that lets you:

- Define data schemas
- Validate data (e.g., API request/response, form inputs)
- Infer TypeScript types directly from the schemas

**ky** is a tiny and elegant HTTP client based on the modern fetch API — basically a lighter and more convenient alternative to axios.