

# CS4530 Final Project: Code2Date

Group 708: Aarushi Surve, Anh Nguyen, Bach Pham, Nghia Duong

## Project Overview

Code2Date is inspired by the legacy Stack Overflow, a rather familiar knowledge exchange forum in the industry. Code2Date aims to make the forum to become a more secure, interactive, dustomisable and fun environment for all users.

## Our Features

In our implementation for Code2Date, we added 3 new features. Firstly, we increased the security measures for the website by adding 2FA authentication, GitHub SSO, Remember this device, revoke + display session functionality, suspicious activity detection, time limited API tokens.

Secondly, we gamified the website so that users can now earn and lose reputations points, they earn badges accordingly to the amount of points they have, which will be shown to all users.

Inappropriate activities such as bad words will be limited which will affect their reputation score when abuse. Users can now see whether or not their question already exists.

Thirdly, we introduced a new matching feature that helps users find compatible partners for collaborative coding. Users can create a personalized match profile that includes their skill level, preferred programming languages, learning goals, and working style. The system then recommends potential partners and provides a compatibility score based on shared skills and preferences, making it easier to find someone who matches how they like to work.

## Design

**Frontend:** Service handle all external communication (API, WebSocket). **Hooks** manage states and effects, consume services, and components.

**Components** are for presentation

**Backend:** Layered Architecture.

**Controllers** handle HTTP requests, delegate logic to service modules

**Services** handle all business logic data operations

**Database:** Cloud database service provided by MongoDB

## Tech Stack

### Deployment:

- Render, MongoDB Atlas

### Database:

- Mongo DB

### Testing:

- Jest, Cypress, CI/CD

## What's next

- Filtering out inappropriate language on the matching function
- New database extension for profile picture storage
- Implement a more rigorous / machine learning adjacent matching algorithm
- AI cleanse on content and profile pictures that are deemed to be inappropriate and give warnings and ban the user

## Frontend:

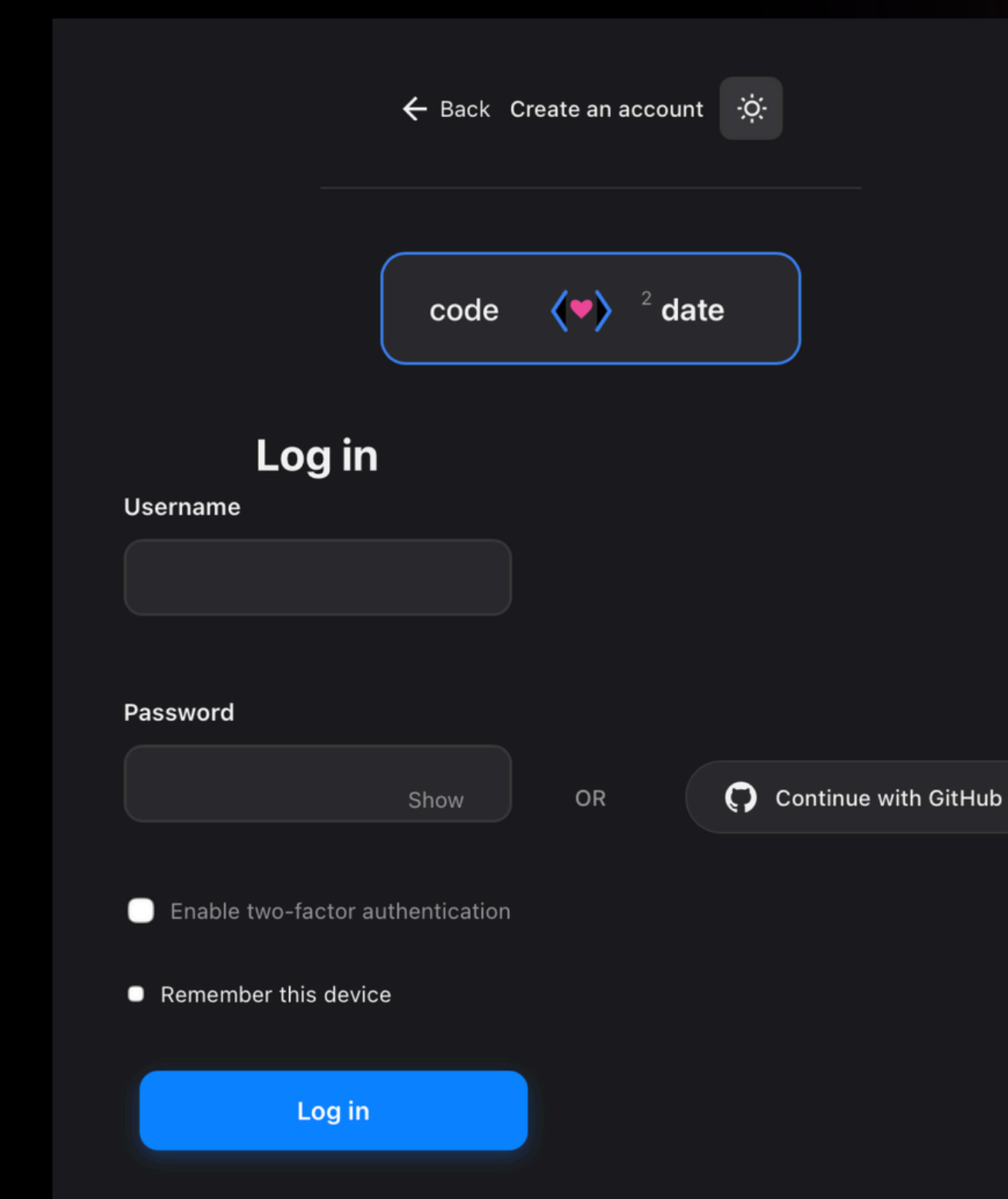
- React, Typescript

## Backend:

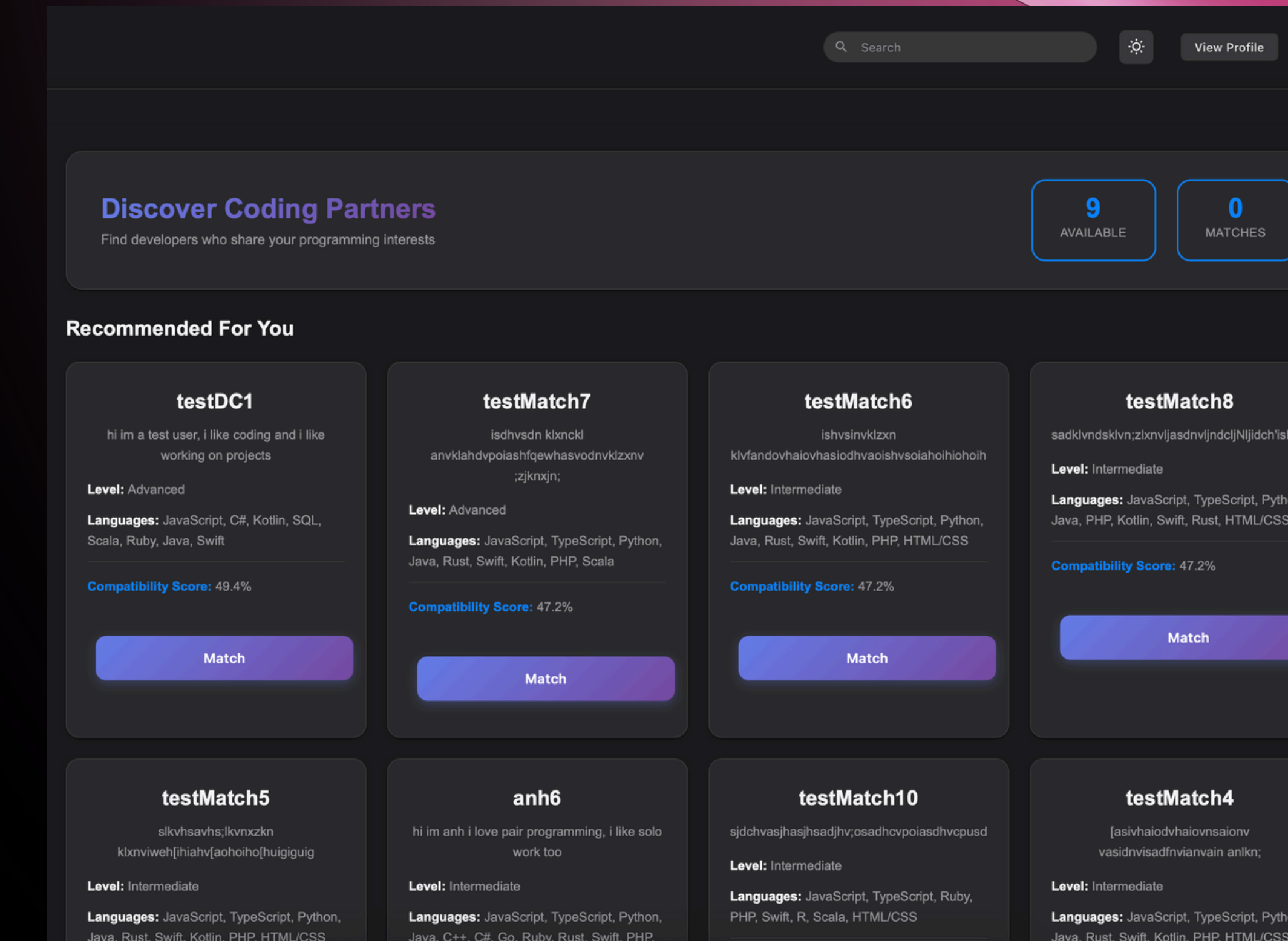
- Typescript

## Libraries:

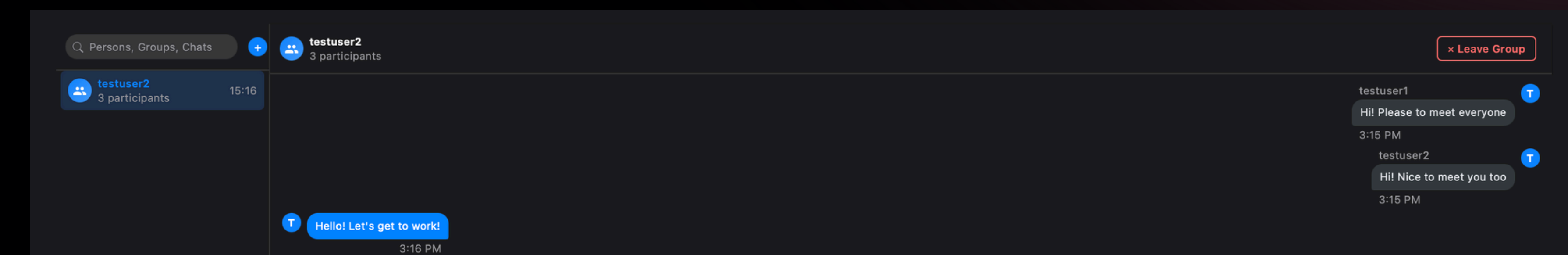
- Obscurity
- Multer
- Jsonwebtoken
- sendgrid/mail



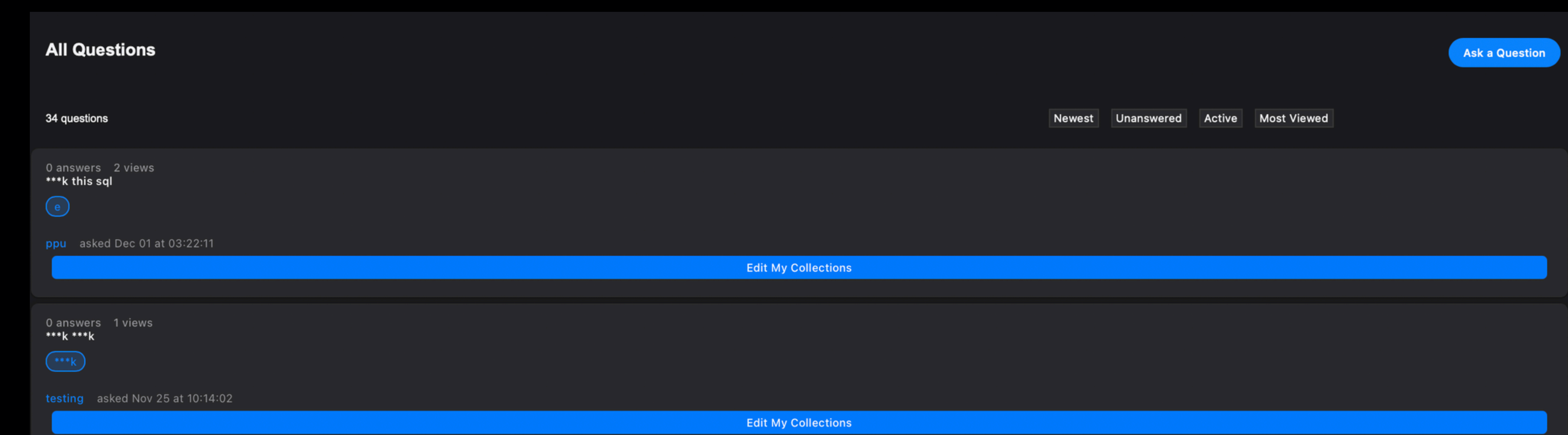
Secure & convenient login options



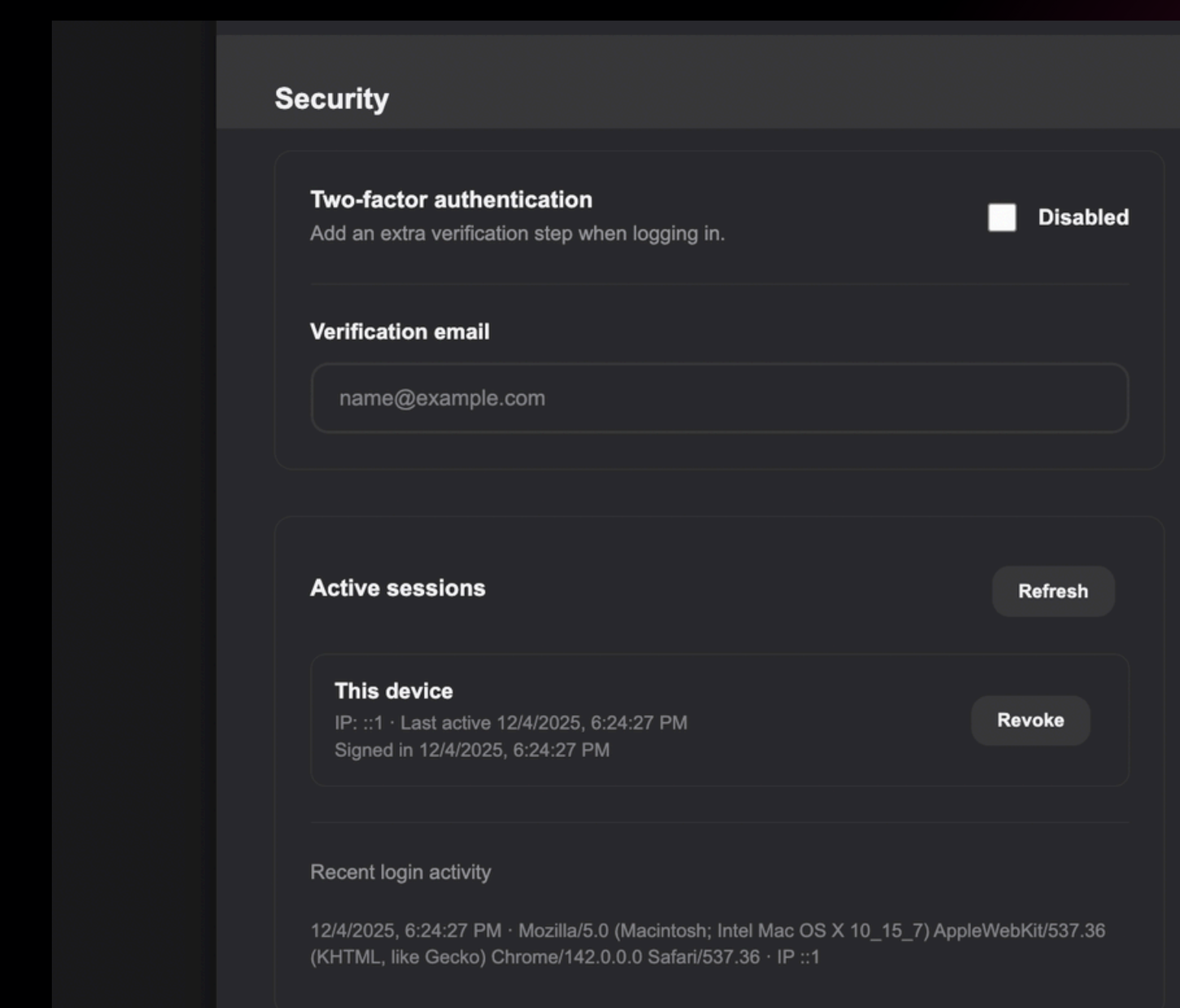
Possible matches for an user



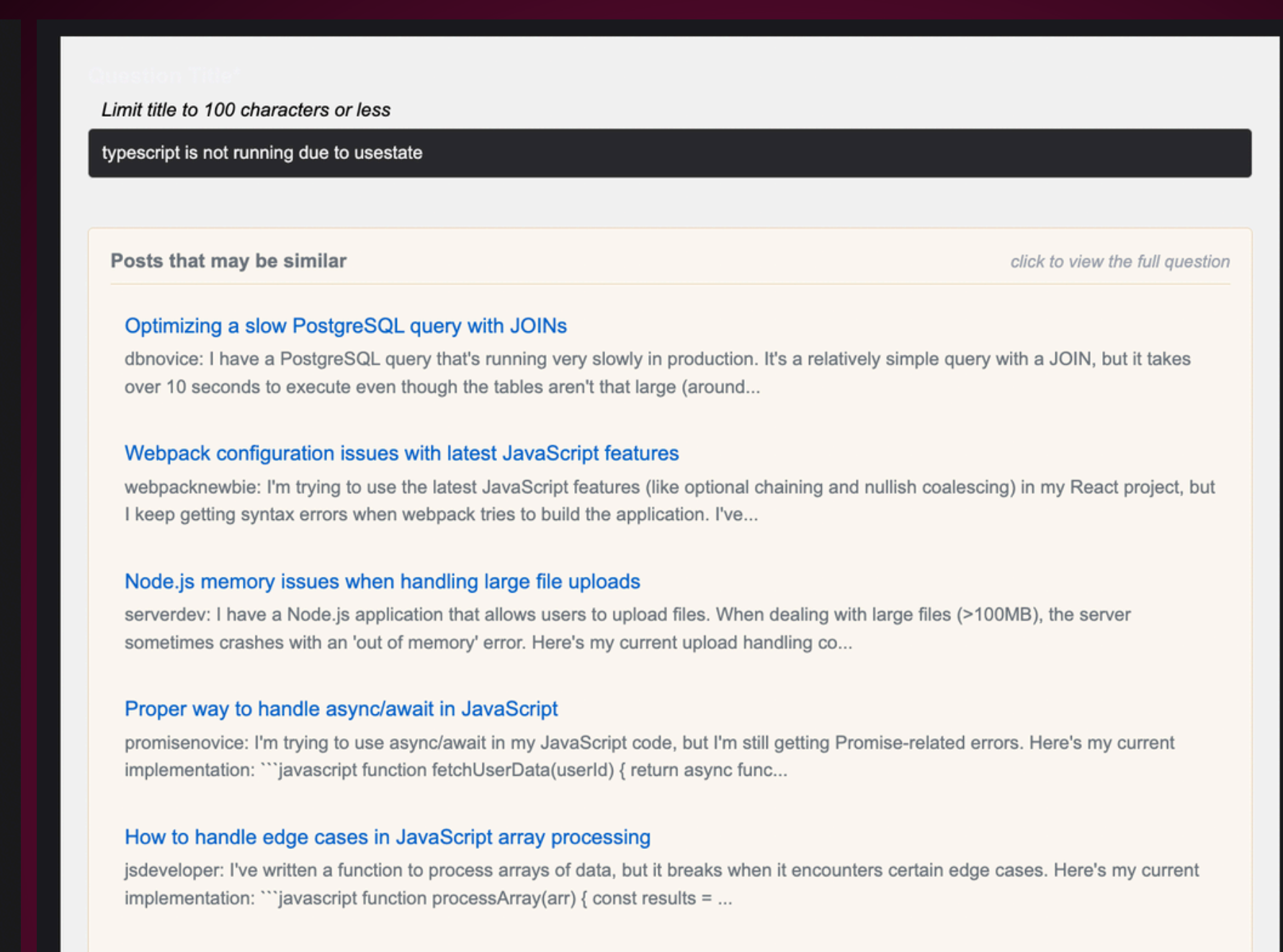
Group / Individual direct messaging



Inappropriate Languages Censored



Two Factor Authentication and Active Sessions



Similar questions suggestions