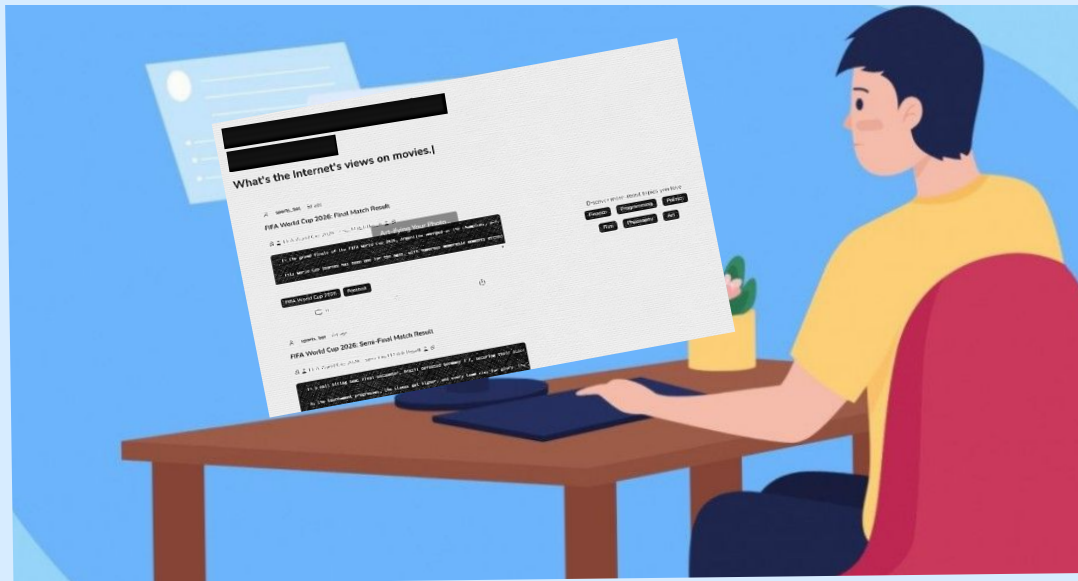
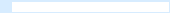


Obelisk

Where Opinions Stand Tall





Our Team



Team Scrum Scums

Team Members

- Arwin Fong
- Arnav Priyadarshi
- Eric Fan
- Harrick Cheong
- Jason Xia



Live Demo

Process Discussion

Workflow



Jira

- Planning Poker at the beginning of each sprint
- Tasks are delegated based on each person's capacity, preference, and interest



Github

- Branched once per user story
- Pull requests required code review and approval from at least one member



Communication

- Conducted a stand up once every other day @ 9pm on Slack or Discord
- Retrospective meetings and initial sprint planning were discussed in-person

Significant Decisions

Team chose to have meetings via Slack/Discord

- Convenience of meeting online increased productivity
- Slack/Discord had notifications and also provided accessibility to everyone

Opting for Relational Database for Structured Data and Cloud Service

- Made it easy to map complex relationships between users, posts, and interaction
- Able to scale the application via methods like database sharding

Disabling the ability for users to delete their own posts

- Supports the platform's vision of promoting authenticity and self-responsibility
- Encourages thoughtful writing and discourages low-quality posts

Technical Discussion

Items

Interesting bug:

- When designing infinite scrolling, the incorrect offset caused posts to be replicated.

Design change:

- Relational database to Vector for Recommender Algorithm

Technology/Framework lesson:

- Modularize page elements into components



Software Architecture

Technology Stack

Frontend



Backend



Main Components

Pages - create post, personalized feed, view profile, search, top recommended posts

Api Routes - comments, recommended, registration, search, user

Components - inputs, modal, buttons, navbar, post card, user card, skeleton screen

Library - authentication, prisma

Software Techniques

Recommender Algorithm:

- Created a matrix using data from our database
- Used Neon extension for pgvector to convert the matrix into a vector database
- Performed Context Based filtering for a current user's recommendations

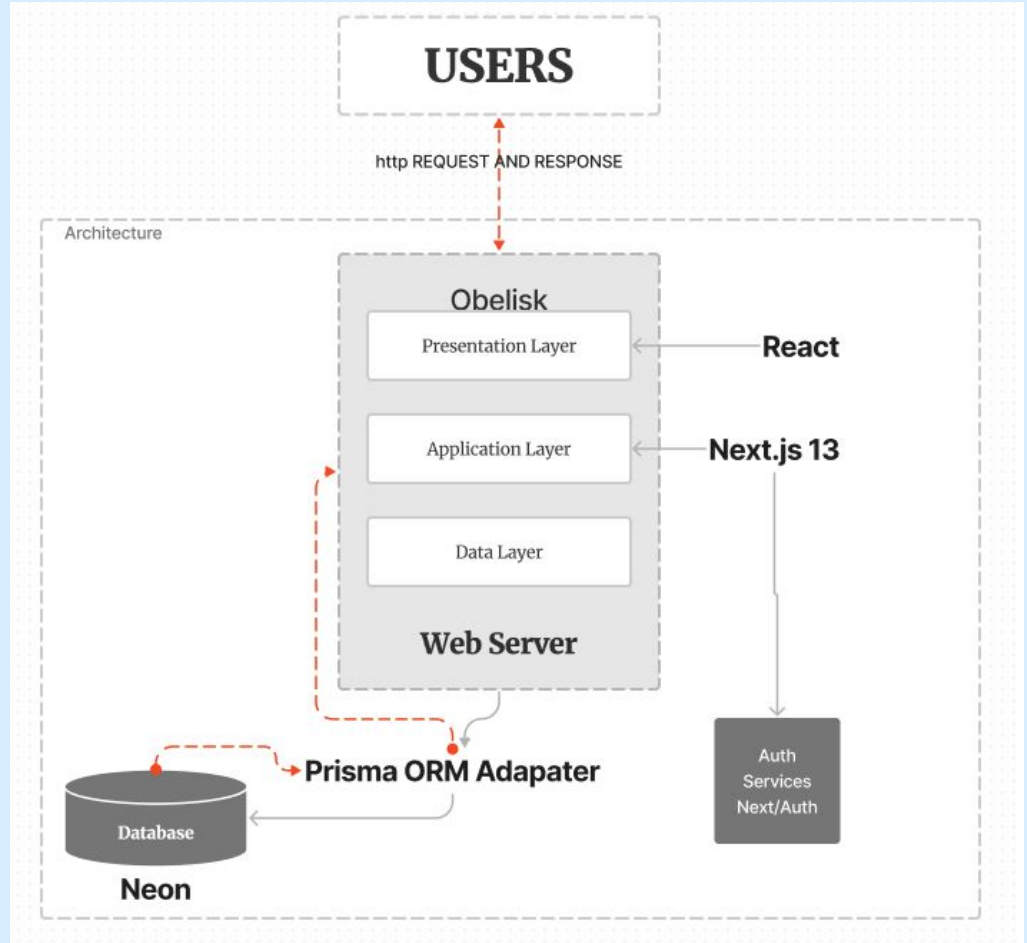
Infinite Scrolling

- Used Prisma cursor-based pagination to make this feature scalable
- Created infinitely scrollable pages using Infinite Scrolling React Component

Search Engine - FullTextSearch

- Provides flexible and accurate search results
- Leverages rank for priority searching
- Streamlines well with PostgreSQL

Architecture





Contributions

Individual Contribution

Arwin Fong - Comments, Markdown Toolbar, Recommender Algorithm + Top Recommended Post

Arnav Priyadarshi - Scrum Management, Posts, Comments, Infinite Scrolling, Profiles, Database, Password Auth, Site UI, Tags, Follows, Likes, Markdown Renderer, Share Posts

Eric Fan - Like on Posts, Profile Customization, Post Category Adding and Creation

Harrick Cheong - Login, Sessions, Edit Profile, Cloud Storage, Frontend Refinements

Jason Xia - Comments, Likes, Full Text Search

Questions?