Project Advisor

Khalil Khan

FINAL PRESENTATION

# Multi-Organization Pass System for Children's Extracurricular Activities


Abrorbek Shanazarov

Marzhan Saparbekova

Aniyar Durmagambetova

Samat Sauranbek

Ilyas Kudaibergenov

## PROBLEM

Parents often face the challenge of selecting specific extracurriculars that fit their children's needs and preferences, and especially managing those activities.

## SOLUTION

A centralised platform that would enable parents to find and explore various leisure activities by registering and paying only once.

# THE PROJECT

is a centralized mobile and web system that is designed to simplify finding and managing different useful extracurricular activities for children under a single subscription.

## FEATURES

01 Centralized access to all activities

02 Personalized recommendations

03 Schedule optimization

04 Dynamic filtering

# COMPLETED FUNCTIONALITY

01 Registration and authentication

02 User profiles

03 Organizations registration and activity data upload

04 Schedule upload and management

05 Admin panel

# COMPLETED FUNCTIONALITY

06 Browsing organizations & classes

07 Signing up for classes

08 Scanning QR for class attendance

09 Rating attended classes via feedback forms

10 Class recommender system

# MOBILE APPLICATION

**Architecture & UI**

- SwiftUI (Declarative UI)
- MVVM Architecture
- @State, @Binding, @ObservableObject

**Backend & Data**

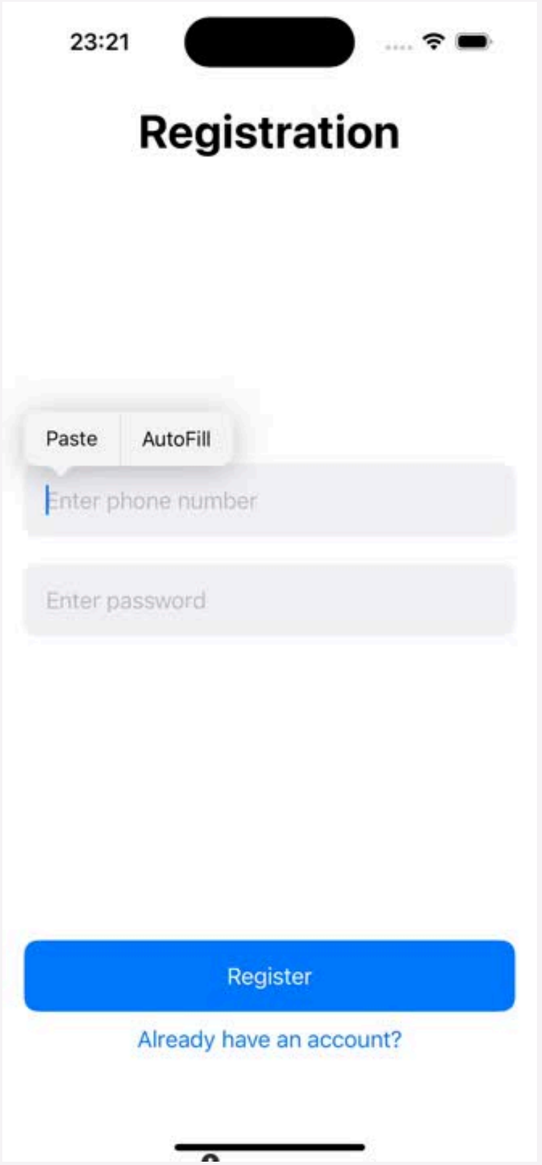
- Apollo GraphQL Client
- JWT Auth with Keychain
- SwiftData for Offline Support

**Core Features**

- Lottie Animations
- AVFoundation QR Scanner
- Dynamic Survey Forms
- Local Recommendation Engine

**Performance**

- Async Image Loading with Caching
- Efficient Memory Handling
- Adaptive Layouts for All Devices



# RATINGS-BASED RECOMMENDER SYSTEM

**Core Stack**

- PyTorch – Deep learning model training & inference
- FastAPI – Lightweight API for serving recommendations

**Model Architecture**

- Hybrid of Matrix Factorization & MLP
- Collaborative and Content-based filtering
- Embeddings for user/item IDs, optionally user/item metadata.
- Linear (cosine similarity) + Non-linear (hidden layers)

**Usage Flow**

- Input: user_id, k → Output: Top-k item IDs
- Caching: faster for frequent requests
- Cold-start fallback for new users (popular items)

**Data & Training**

- Ratings data: (user_id, item_id, rating, user metadata, item metadata)
- Continuous updates with new user feedback
- Daily retraining with dual model instances (train + serve)

**Evaluation**

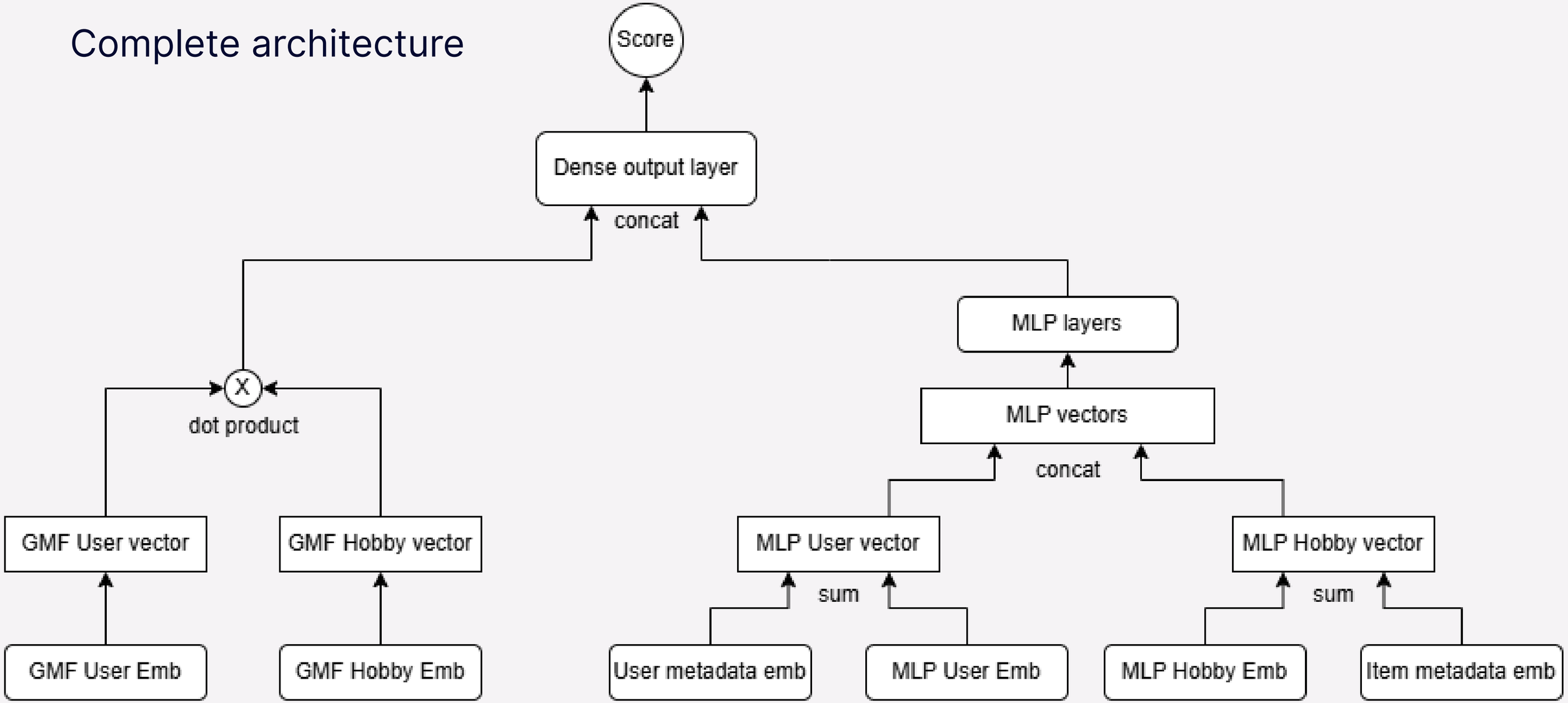
- MovieLens 100K Benchmark
- 1000 users and 1700 movies
- Achieved 0.90 MSE on test set

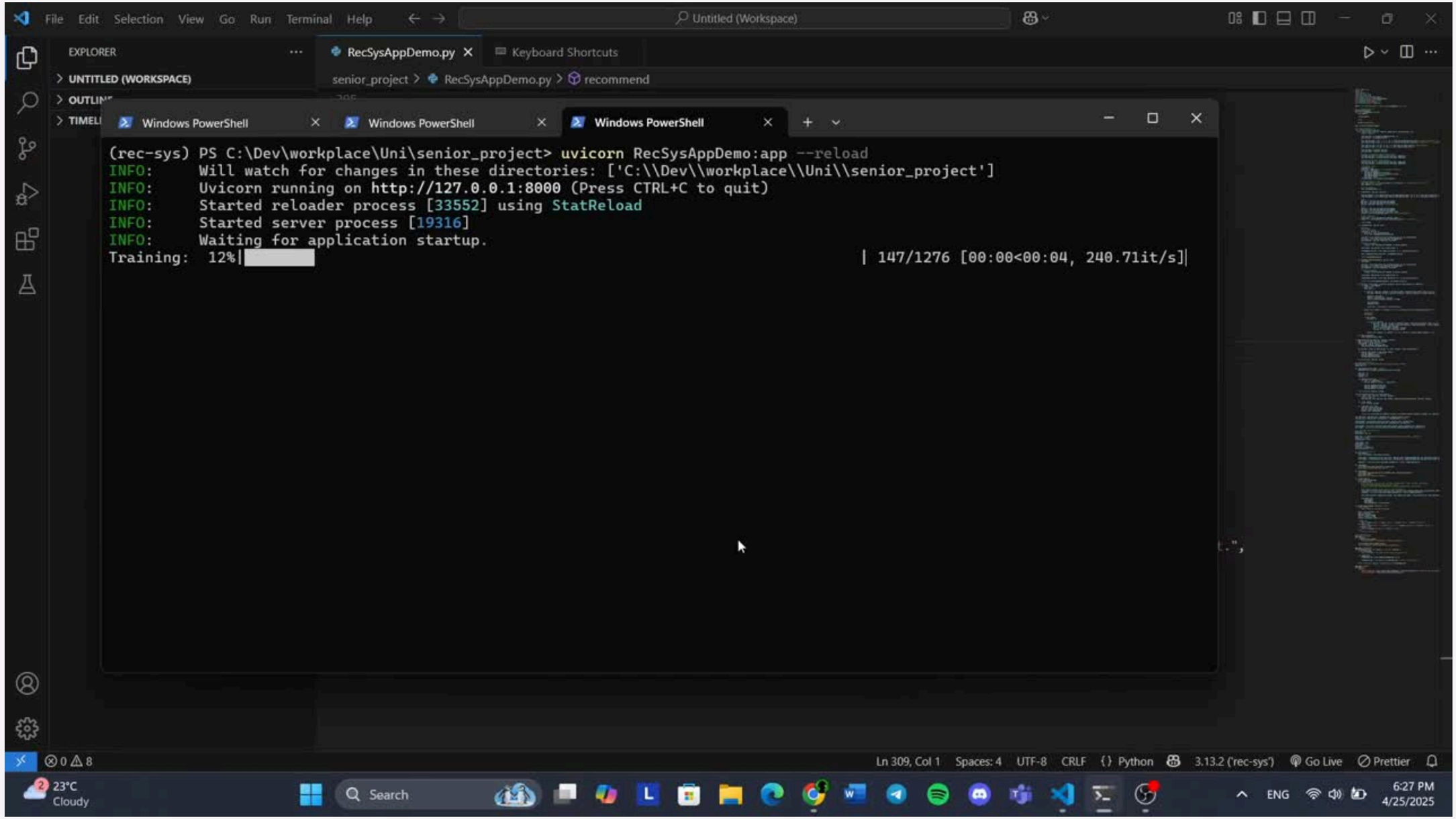
**API Endpoints**

- /recommend – Get recommendations
- /retrain – Trigger model update
- /status – Check last update time

# RATINGS-BASED RECOMMENDER SYSTEM

Complete architecture

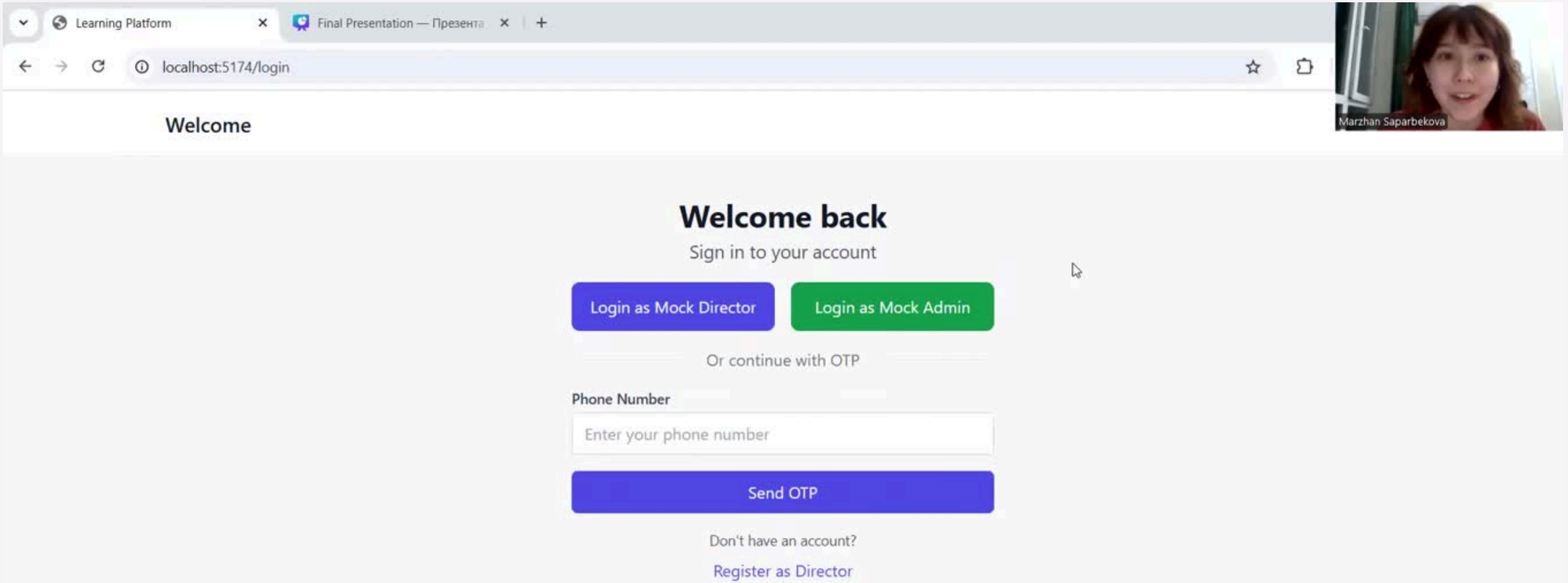





# RECOMMENDER SYSTEM DEMO

# ADMIN PANEL

**DEMONSTRATION**

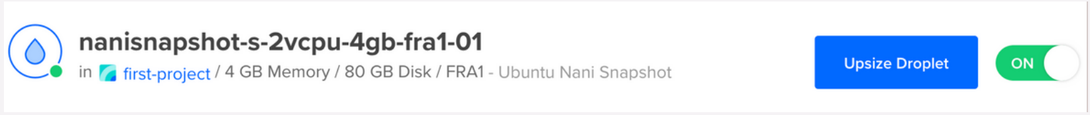


# BACKEND

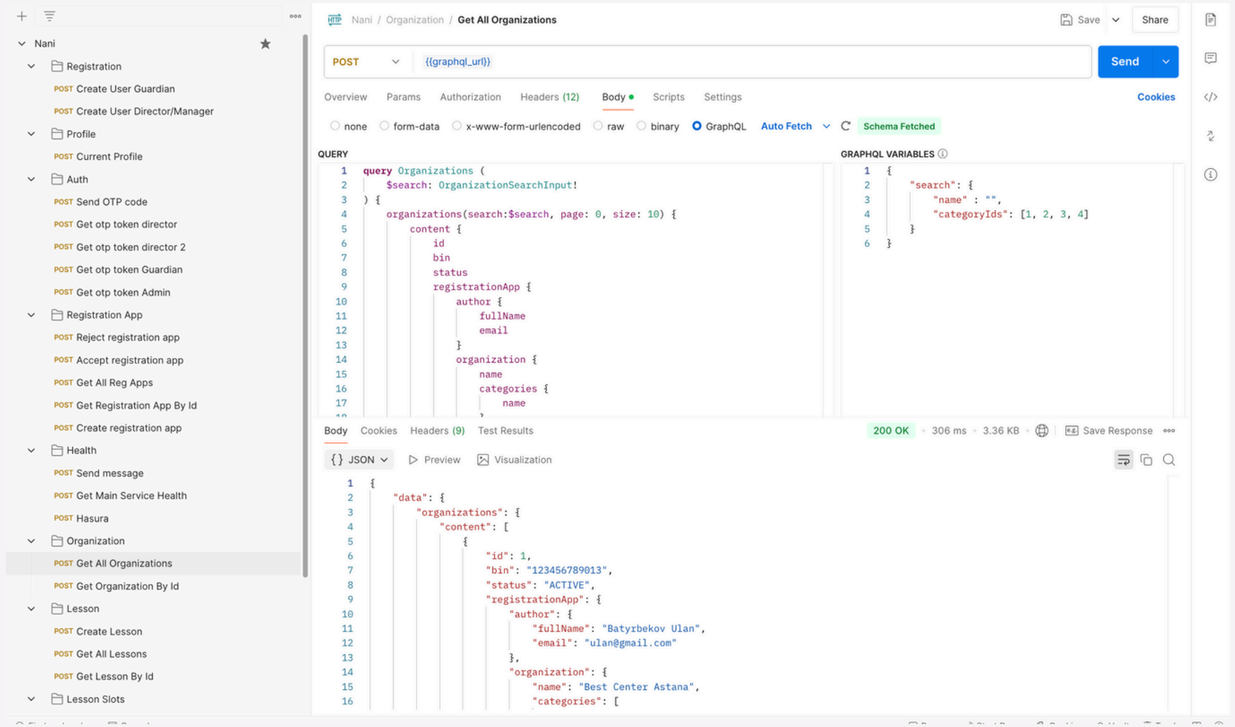
Technologies Used:

- Java 17
- Spring Boot 3
- GraphQL
- PostgreSQL DB
- Keycloak OAuth
- Temporal IO

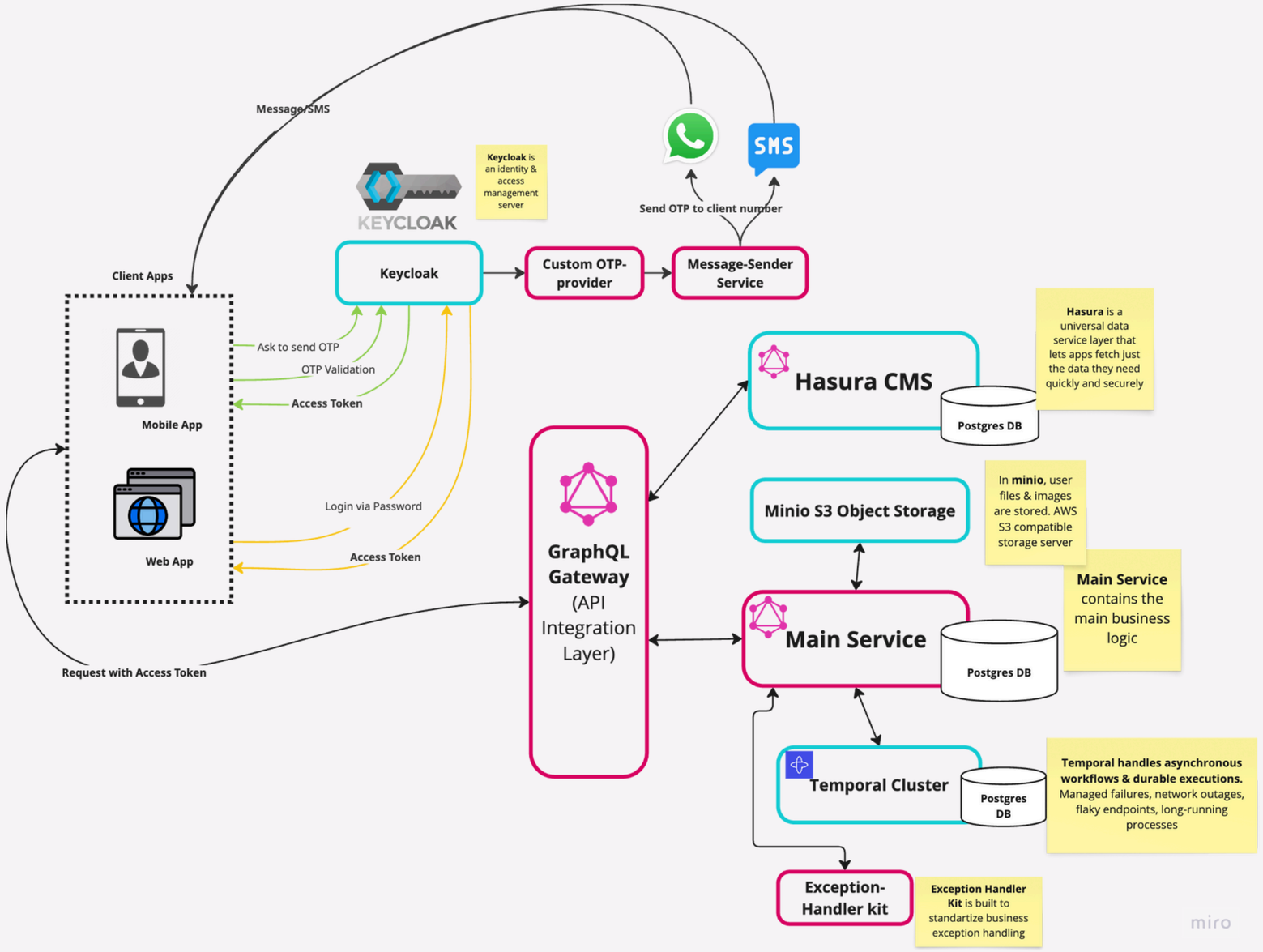
Hosting: DigitalOcean




All endpoints accessible via Postman



# BACKEND: SOLVING IMAGINARY SCALING PROBLEMS




To address future problems with scaling:

- Micro-Service architecture was implemented
- File-storage was moved to Minio S3 (AWS)
- OAuth was moved to Keycloak with custom OTP provider
- Temporal IO was added to orchestrate asynchronous processes

# FUTURE PLANS

SPRING SEMESTER

01 Expanding existing functionality

02 Schedule optimization model for organization owners

03 Add financial infrastructure to manage payments

04 Improve UI & UX

05 Deploy the project as a startup