

# NextGen **Movie** Recommendation

Discovering your next action

# The amazing team



Mario



Luigi



Link



Peach

# The problem

Finding something to watch



# Value proposition

Bla bla bla

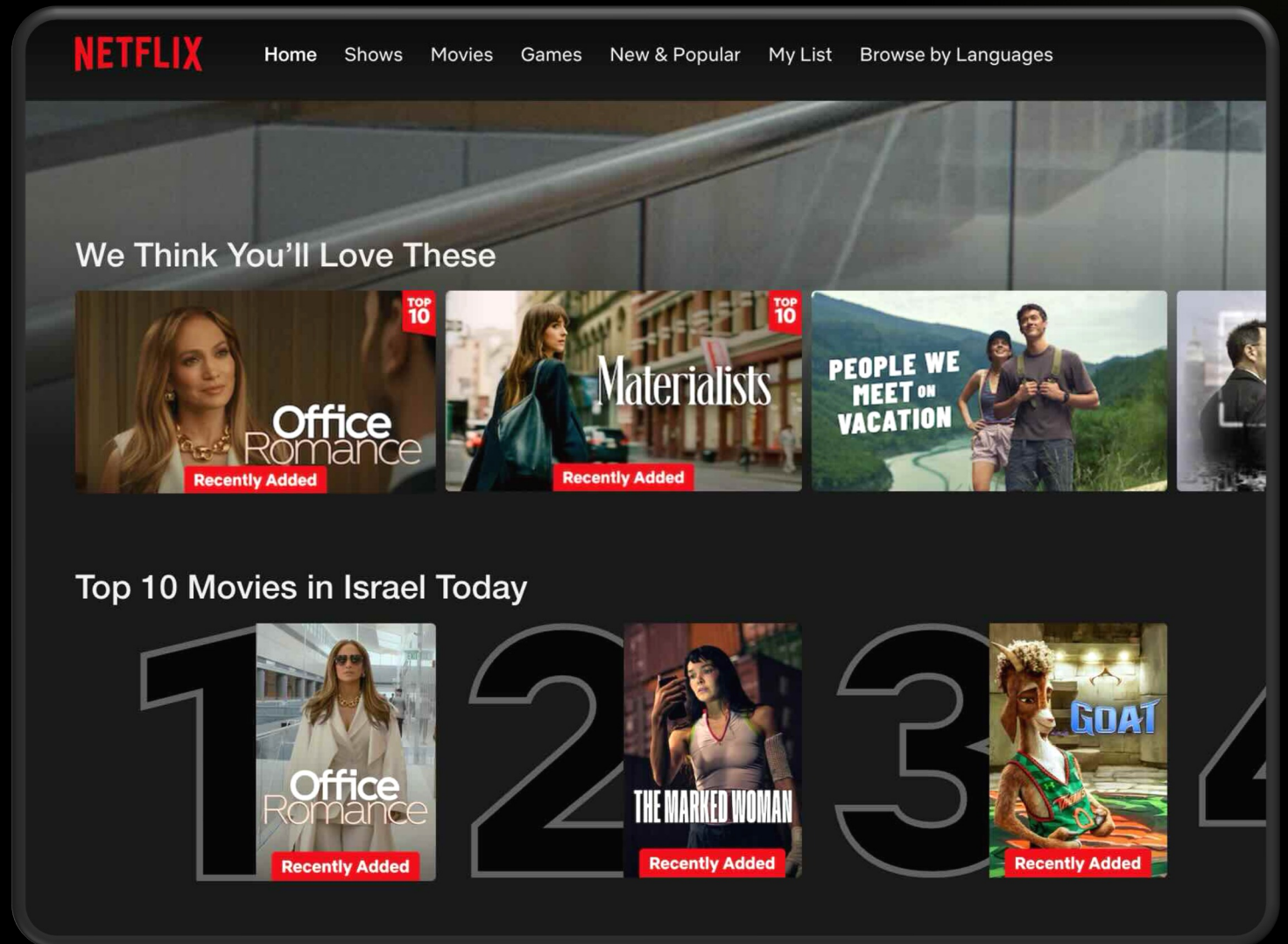


Live DEMO



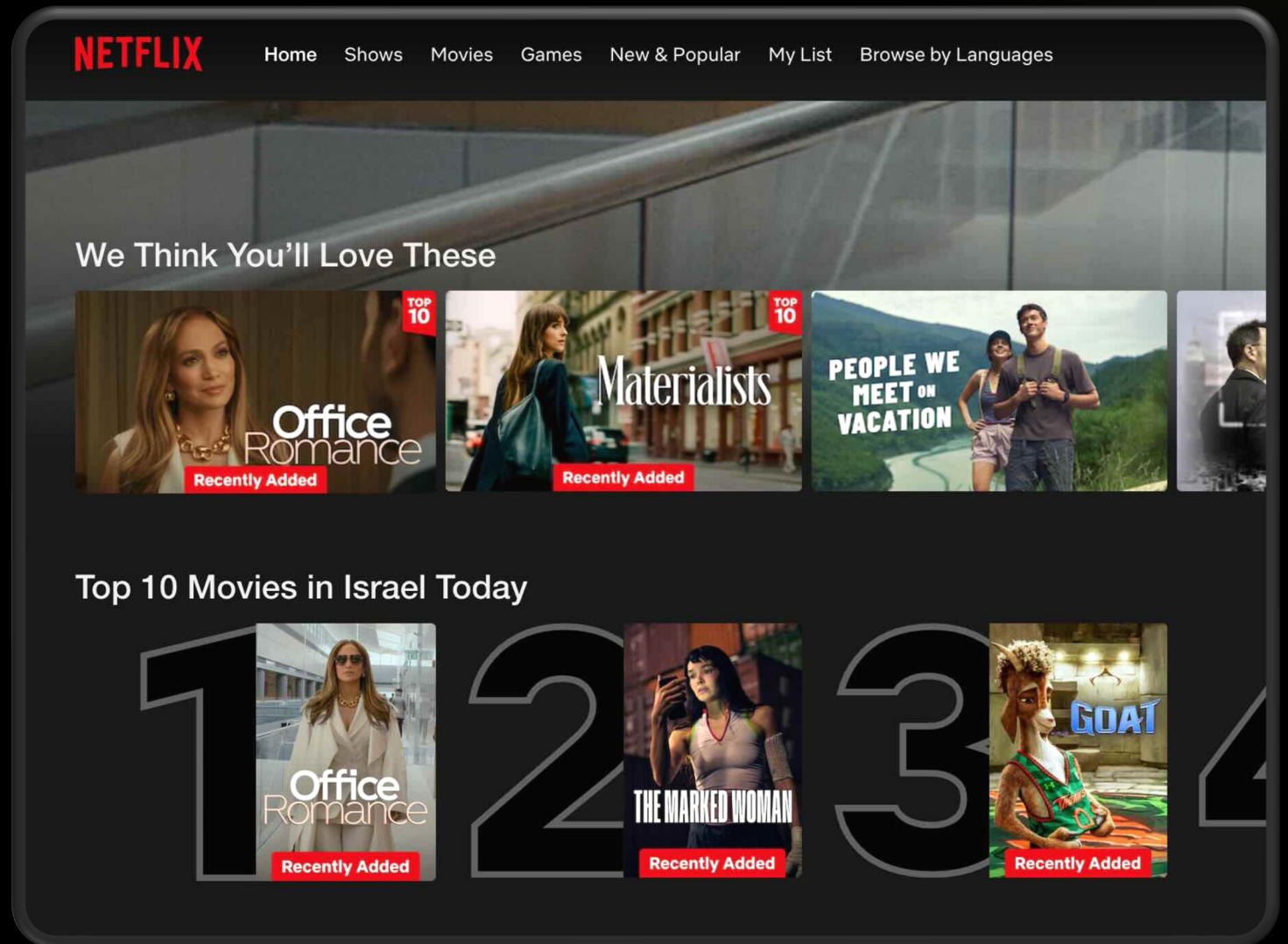
# Screenshot – Feature 1

## Describing feature



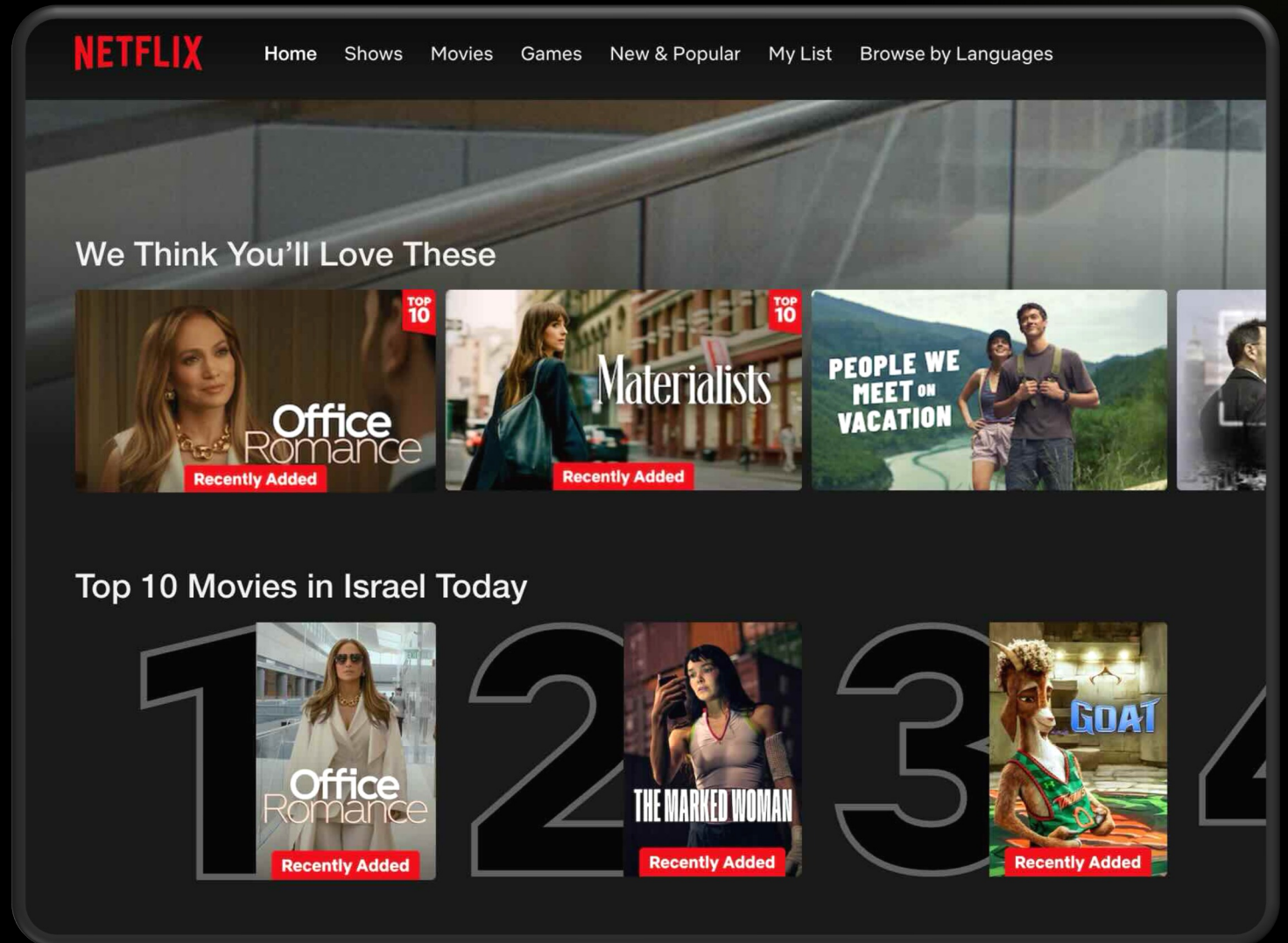
# Screenshot – Feature 2

## Describing feature



# Screenshot – Feature 3

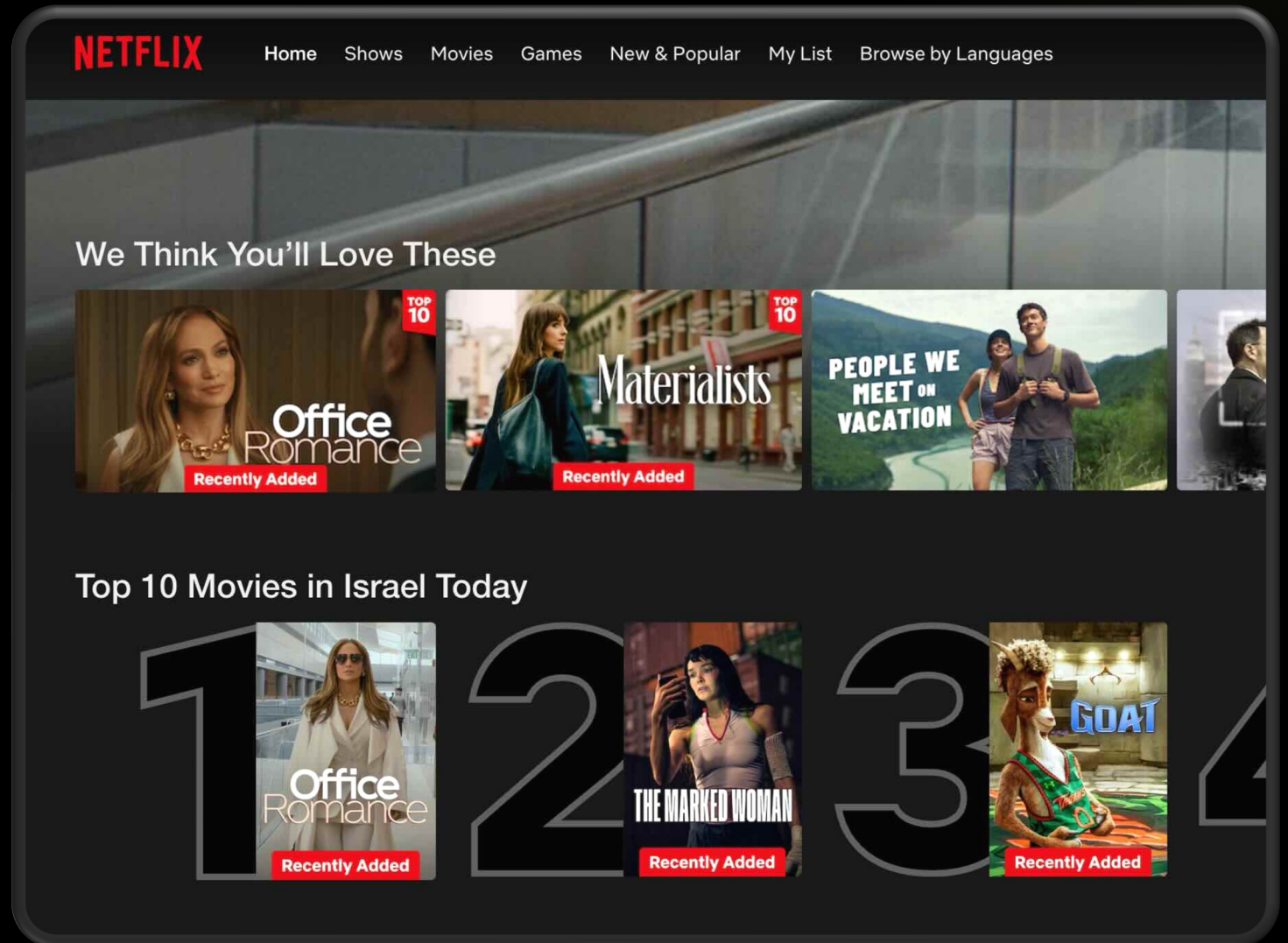
## Describing feature





# Screenshot – Feature 4

## Describing feature



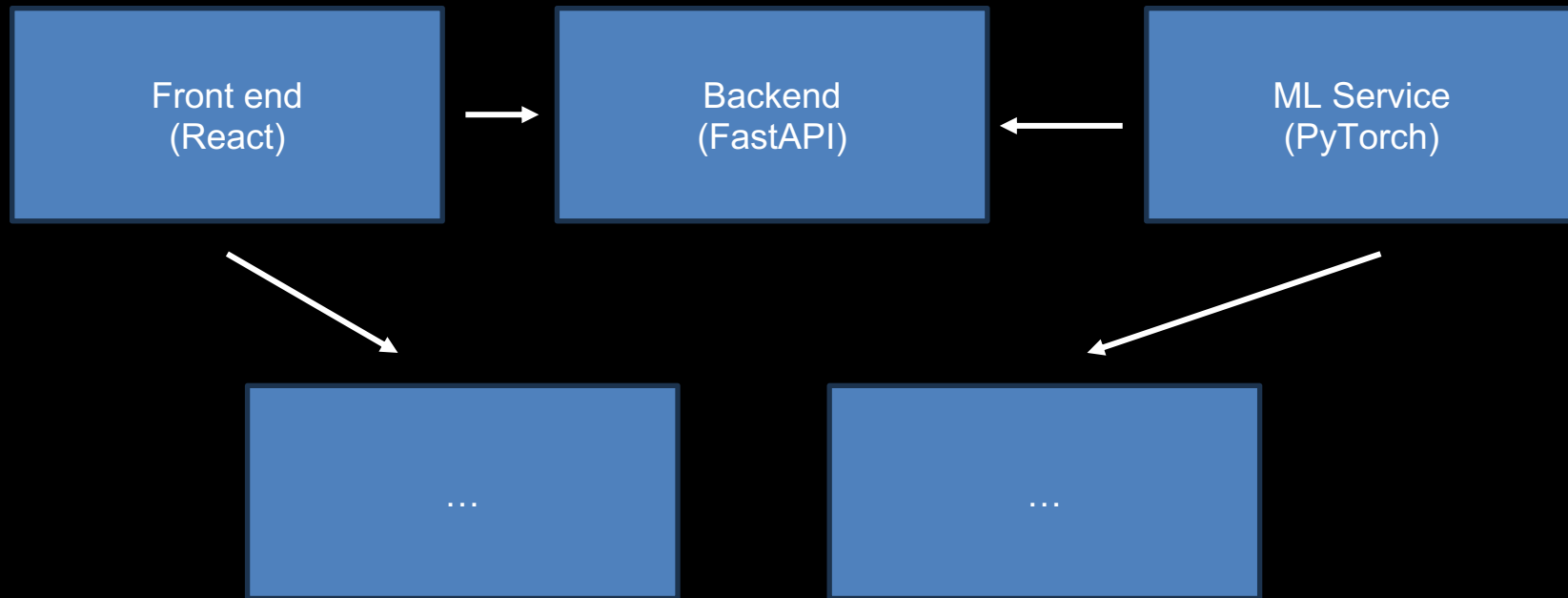
# The Data

10M ratings from TMDB  
(1M users, 500K movies)

Semantic Data from IMDB



# Architecture



# System Architecture Overview

## Full-Stack Integration

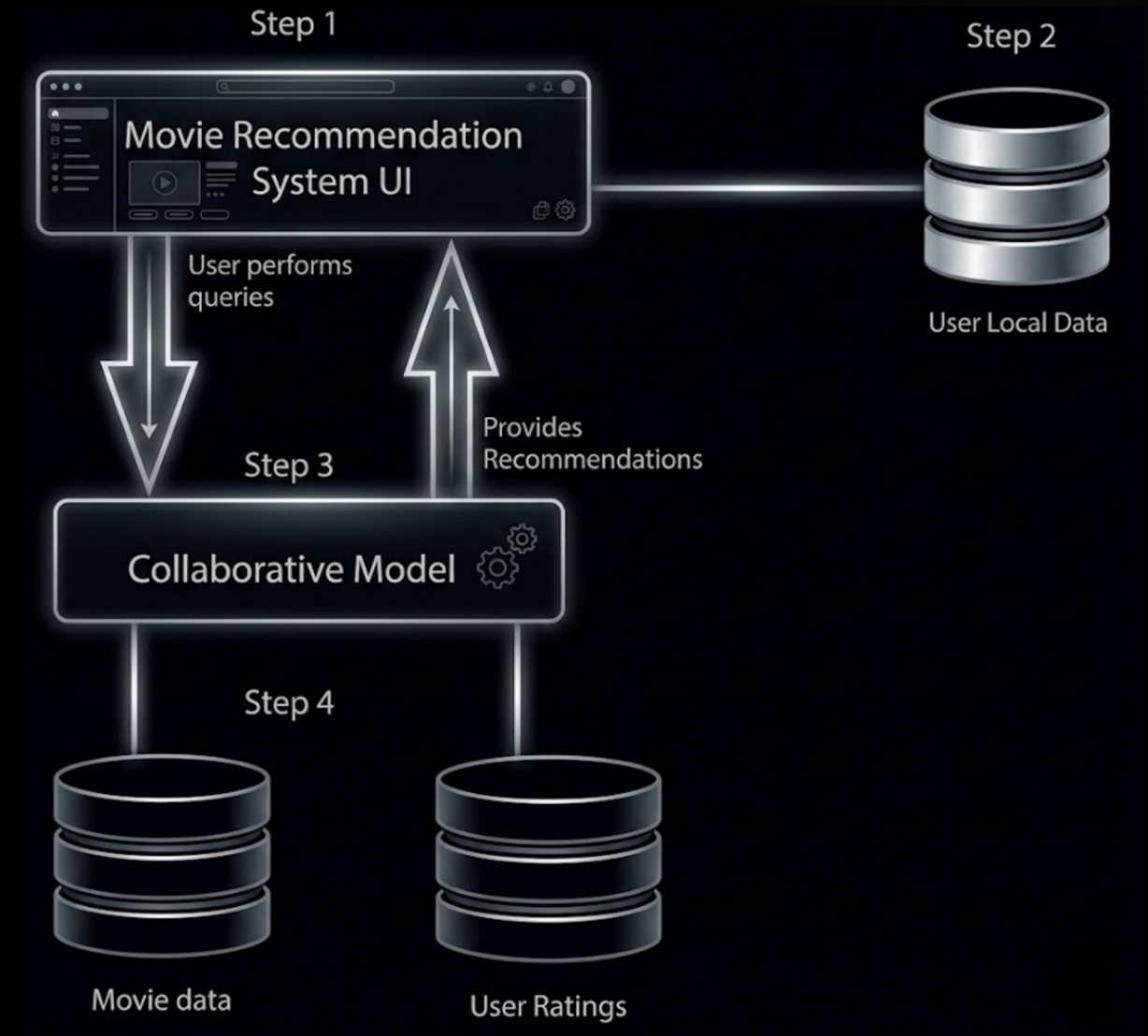
The system utilizes a **Client-Server** architecture designed for scalability:

**Frontend:** React/Streamlit for responsive UI and poster rendering.

**Backend:** Flask API serving as the orchestration layer.

**Recommendation Engine:** scikit -learn and PyTorch models computing similarity scores.

**Database:** PostgreSQL for user metadata and Redis for caching recommendations.





# Algorithmic Strategies

## Content-Based Filtering

- which algorithm was selected for each recommendation approach, and why

## Collaborative Filtering

which algorithm was selected for each recommendation approach, and why

# Future work



Implement feature 1

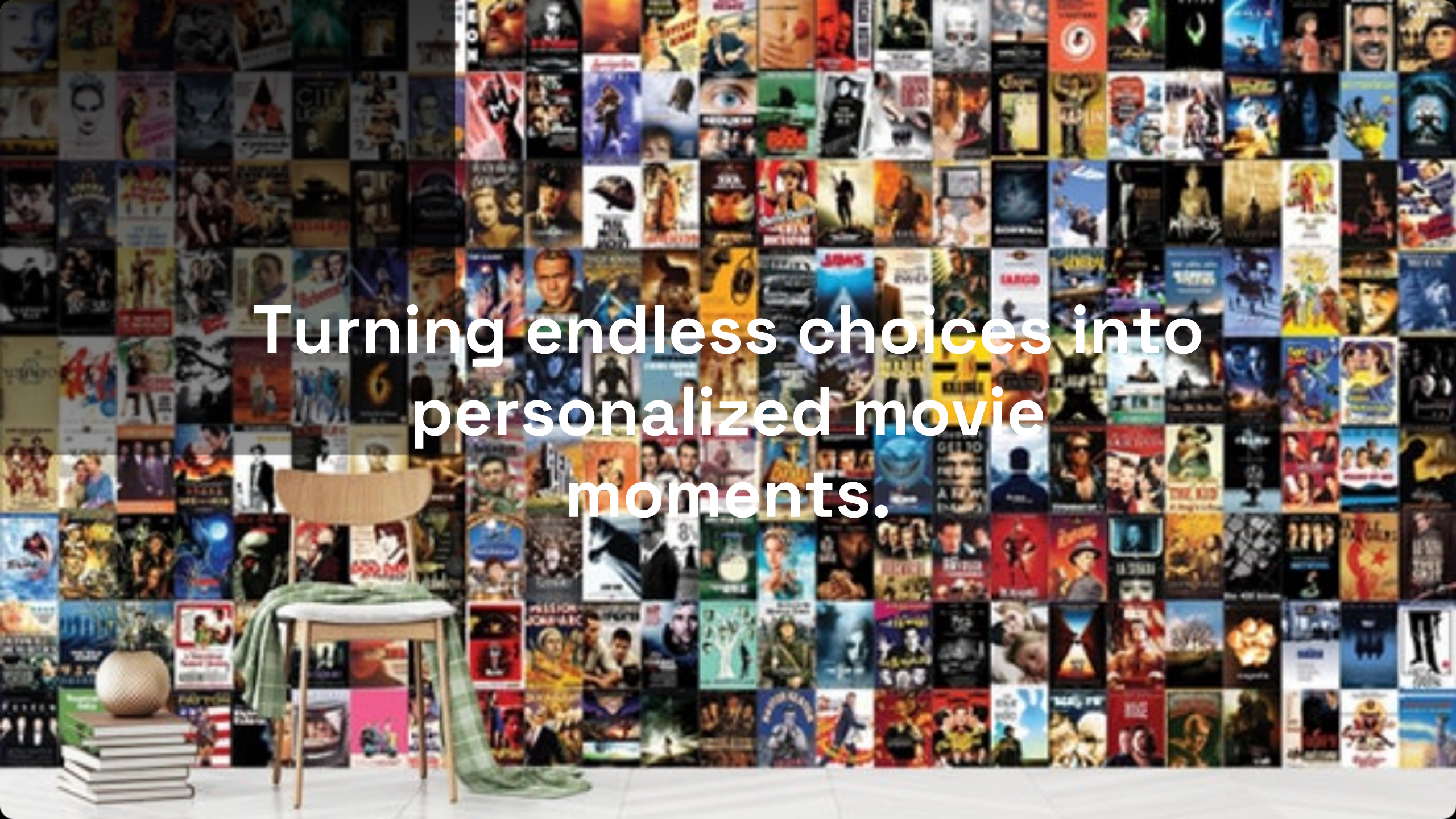
Implement feature 2

Add DB support

Fix data entity matching

...



The background of the image is a vast, dense grid of movie posters, creating a colorful and chaotic wall of choices. In the lower-left foreground, a simple wooden chair with a light-colored seat is positioned. A green cloth is draped over the back of the chair and hangs down to the floor. To the left of the chair, there is a stack of several books, with a round, textured object (possibly a lamp or a decorative sphere) resting on top of them. The floor is a light, neutral color.

Turning endless choices into  
personalized movie  
moments.