

JUTJUBIC

YouTube-inspired video sharing platform

Angular · Spring Boot · PostgreSQL · RabbitMQ · Docker · Nginx

Projektna dokumentacija

Pavle Kapuran

pavlekapuran.pages.dev

// sadrzaj

- 01 Opis projekta
- 02 Funkcionalnosti
- 03 Tech Stack
- 04 Arhitektura sistema
- 05 Model baze podataka
- 06 Video procesiranje
- 07 Watch Party
- 08 Autentifikacija
- 09 Klasterovani deployment
- 10 UI/UX dizajn

// opis projekta

Jutjubic je full-stack web aplikacija inspirisana YouTube-om koja omogućava korisnicima upload, pregled, pretragu i interakciju sa video sadrzajem. Platforma podrzava kompletni video lifecycle — od uploada i transkodiranja do streaminga i komentarisanja.

Projekat je razvijen kao dio kursa Inzenjerstvo softverskih arhitektura (ISA) sa fokusom na mikroservisnu arhitekturu, message queue komunikaciju, i klasterovani deployment. Sistem demonstrira produkcione obrasce kao sto su load balancing, caching, i asinhrono procesiranje.

Kljucne karakteristike

- ▶ Upload i streaming videa sa automatskim transkodiranjem
- ▶ Sistem komentara sa kesiranim odgovorima za performanse
- ▶ Watch Party — sinhronizovano gledanje videa u realnom vremenu
- ▶ Pretraga videa po naslovu, opisu i tagovima
- ▶ Like/dislike sistem sa real-time brojacima
- ▶ Klasterovani deployment sa Nginx load balancerom

// funkcionalnosti

Video upload i procesiranje

Korisnici mogu uploadovati video fajlove u razlicitim formatima. Backend prima fajl, validira format i velicinu, i salje poruku na RabbitMQ queue za asinhrono transkodiranje. Video se transkodira u standardne rezolucije (480p, 720p) i cuva na serveru. Korisnik dobija notifikaciju kada je video spreman za gledanje.

Sistem komentara

Svaki video ima thread komentara sa podrskom za nested odgovore. Komentari su kesirani na backend-u radi performansi — cache se invalidira pri novom komentaru. Implementiran je i paginacioni sistem za videe sa velikim brojem komentara.

Pretraga i filtriranje

Full-text pretraga po naslovu i opisu videa. Filtriranje po kategorijama, datumu uploada, i popularnosti. Rezultati se sortiraju po relevantnosti sa paginacijom.

Korisnicke interakcije

Like/dislike sistem, pretplata na kanale, history pregleda, i personalizovane preporuke na osnovu istorije gledanja. Svaki korisnik ima profil sa listom uploadovanih videa i statistikama.

// tech stack

Frontend

Tehnologija	Verzija	Namjena
Angular	17.x	Frontend framework — SPA sa routing-om
TypeScript	5.x	Type-safe development
Angular Material	17.x	UI komponente i tema
RxJS	7.x	Reaktivno programiranje i HTTP interceptori

Backend

Tehnologija	Verzija	Namjena
Spring Boot	3.x	Backend framework — REST API
PostgreSQL	15.x	Relaciona baza podataka
RabbitMQ	3.x	Message queue za asinhrono procesiranje
Redis	—	Caching sloj za komentare i sesije
FFmpeg	—	Video transkodiranje i thumbnail generisanje

Infrastruktura

Tehnologija	Namjena
Docker	Kontejnerizacija svih servisa
Docker Compose	Orkestracija multi-container setup-a
Nginx	Reverse proxy i load balancer
Vercel	Frontend hosting (produkcija)

// arhitektura sistema

Sistem koristi mikroservisnu arhitekturu sa jasnom separacijom odgovornosti. Frontend komunicira sa backend-om preko REST API-a, dok backend servisi medjusobno komuniciraju putem RabbitMQ message queue-a.

Komunikacioni obrasci

- Sinhrona: REST API pozivi izmedju frontenda i backenda
- Asinhrona: RabbitMQ za video procesiranje i notifikacije
- Caching: Redis za kesirane odgovore i sesije

// model baze podataka

PostgreSQL baza sa normalizovanom semom optimizovanom za citanje. Kljucni entiteti su medjusobno povezani foreign key relacijama sa kaskadnim brisanjem gdje je primjenjivo.

Glavni entiteti

Entitet	Opis	Kljucne relacije
User	Registrovani korisnik	Ima mnogo videa, komentara, subskripcija
Video	Uploadovani video	Pripada korisniku, ima komentare i reakcije
Comment	Komentar na video	Pripada videu i korisniku, self-ref za replies
Reaction	Like/dislike	Pripada videu i korisniku (unique constraint)
Subscription	Pretplata na kanal	Many-to-many izmedju korisnika
Category	Kategorija videa	One-to-many sa videima

Indexing strategija

Indeksi su kreirani na frequently queried kolonama: video.title (GIN index za full-text search), video.created_at (B-tree za sortiranje), comment.video_id (za brzo dohvacanje komentara), i user.username (unique index za login).

// video procesiranje

Video upload pipeline koristi asinhroni obrazac sa RabbitMQ za decoupling upload-a od transkodiranja. Ovo omogućava da korisnik odmah dobije potvrdu uploada dok se video procesira u pozadini.

Pipeline koraci

1. Upload → Spring prima multipart file, validira format/velicinu
2. Storage → Originalni fajl se cuva na disk
3. Queue → Poruka sa video ID-jem ide na RabbitMQ
4. Transcode → Worker cita poruku, pokrece FFmpeg
5. Thumbnail → FFmpeg ekstrahuje frame za thumbnail
6. Update → Video status se azurira u bazi na "ready"
7. Notify → Korisnik dobija notifikaciju da je video spreman

FFmpeg konfiguracija

Video se transkodira u H.264 codec sa AAC audio-m. Podržane rezolucije su 480p i 720p. Bitrate se automatski prilagodjava rezoluciji. Thumbnail se generise iz prvog non-black frame-a videa.

// watch party

Watch Party omogućava grupi korisnika da sinhronizovano gledaju isti video u realnom vremenu. Host kontrolise playback (play, pause, seek), a svi ucesnici automatski prate stanje.

Implementacija

- ▶ Host kreira party i dijeli link sa ucesnicima
- ▶ WebSocket konekcija za real-time sinhronizaciju
- ▶ Host akcije (play/pause/seek) se propagiraju svim ucesnicima
- ▶ Latency kompenzacija za sinhronizovani playback
- ▶ Chat u okviru party-ja za komunikaciju tokom gledanja

// autentifikacija

Sistem koristi JWT (JSON Web Token) autentifikaciju sa Spring Security. Tokeni se izdaju pri uspješnom loginu i sadrže korisnicke role (USER, ADMIN).

Aspekt	Implementacija
Token tip	JWT sa HS256 algoritmom
Trajanje	Access token: 1h, Refresh token: 7d
Storage	HttpOnly cookie (secure)
Roles	USER — osnovni pristup, ADMIN — moderacija
Password	BCrypt hashing sa salt-om
Validacija	Spring Security filter chain

// klasterovani deployment

Produkcioni deployment koristi Docker Compose sa multiple instancama Spring Boot aplikacije iza Nginx load balancera. Ovo demonstrira horizontalno skaliranje i high availability obrasce.

Docker Compose konfiguracija

services:

- nginx: → Reverse proxy, port 80
- app-1: → Spring Boot instanca #1
- app-2: → Spring Boot instanca #2
- app-3: → Spring Boot instanca #3
- postgres: → PostgreSQL baza
- rabbitmq: → Message queue
- redis: → Cache layer

Nginx load balancing

Nginx koristi round-robin algoritam za distribuciju zahtjeva izmedju Spring Boot instanci. Health check endpoint osigurava da se saobračaj ne salje na neaktivne instance. Sticky sessions su konfigurisani za WebSocket konekcije (Watch Party).

// ui/ux dizajn

Frontend koristi Angular Material za konzistentan UI sa custom temom inspirisanom YouTube-ovim dizajnom. Responsive layout se prilagodjava svim velicinama ekrana — od mobilnog do desktop prikaza.

Ključni ekrani

Ekran	Opis
Home feed	Grid videa sa thumbnail-ovima, naslovom, i statistikama
Video player	Custom player sa kontrolama, komentarima i related videima
Upload	Drag-and-drop upload sa progress barom i metadata formom
Profile	Korisnikovi videi, statistike, i podesavanja
Search	Full-text pretraga sa filtriranjem i sortiranjem
Watch Party	Sinhronizovani player sa chat panelom

Responsive breakpoints

- ▶ Mobile (< 768px): Jednokolonski layout, hamburger meni
- ▶ Tablet (768px - 1024px): Dvokolonski grid, sidebar collapse
- ▶ Desktop (> 1024px): Trokolonski grid, fiksni sidebar