

1. Тема: LandCast е централизирана онлайн стрийминг платформа за качване, гледане и провеждане на уебинари в реално време.

- Приложението е публично достъпно на адрес:

<https://landcast-frontend.vercel.app>

2. Автор:

Трите имена: Никола Диянов Илиев, ЕГН: 0345178720, Адрес: Дедеагач 11 вход 5 етаж 8 ап.119, Телефон: 0887276031, Имейл: nikimen2003@gmail.com, Училище: ППМГ “Нанчо Попович”

3. Ръководител:

Даринка Георгиева Вълкова, 0882289147, valkova_darina@abv.bg, старши учител.

4. Резюме:

4.1. Цели:

- Идеята за създаването на платформата е породена от факта, че в България липсва специализирана платформа за уебинари.

В момента уебинари се обявяват и рекламират **в социалните мрежи**, после се качват **в други платформи** като YouTube, което не е оптимално за даден потребител, тъй като той **не може** винаги да следи кога, къде и какви уебинари има и кога ще се проведат.

LandCast дава възможност да се качват, гледат и провеждат уебинари в реално време, в една централизирана платформа, която предлага всичко необходимо на едно място.

4.2. Основни етапи в реализирането на проекта

- обмисляне на идея

- проучване дали идеята решава **актуален** проблем

- обмисляне на основните функции, които да постигнат необходимата крайна цел (**MVP** / Minimum viable product)

- избиране на подходящи програмни езици и технологии

- постепенно се изграждат “подсистеми” (**FDD** - Feature Driven Development), които накрая образуват крайния продукт.

- **валидация и верификация** на софтуера

- потребителско тестване (дава се линк към крайния продукт на потребители, които **оценяват и критикуват** платформата за оптимизации)

4.3. Ниво на сложност на проекта:

- Сложността на проекта основно е в изграждането на **VoD** (Video on Demand) услуга и **стриймिंग на живо** към много потребители.

4.4. Логическо и функционално описание на решението:

Архитектура на проекта:

- Цялото приложение за момента се състои от четири сървъра - клиентско уеб приложение, бекенд GraphQL API, сървър за транскодиране/обработка на видеа и специализиран SFU стрийминг сървър, който играе важна роля в провеждането на уебинарите на живо.

4.5. Реализация

Фронтенд:

За фронтенда се използва React с фреймуърк NextJS, тъй като позволява SSR и предлага много вградени оптимизации. За статичните файлове използвам Cloudinary CDN с цел по-бързото им достигане до потребителите. За заявки към GraphQL АПИ-то се използва React-Query, който предоставя редица оптимизации като client-side кеширане, намалява броя на изпратените заявки към сървъра (deduping) и много други. Изграждането на визуалните компоненти преминава през изграждане на прототип във Figma и после се конвертира директно в приложението.

Бекенд АПИ:

За изграждане на бекенд АПИ-то съм избрал езика Go, тъй като е приятен, лесен и ефикасен език. Използвам Ent за ORM, тъй като улеснява работата с GraphQL и решава известни проблеми като N+1. Сървърът седи зад GraphCDN, който е специализиран CDN за GraphQL.

Стрийминг SFU сървър:

Възможно е един или няколко водещи да организират уебинар и да влизат десетки хора да ги гледат.

В момента използвам WebRTC протокола за връзка в реално време, който е Peer To Peer. Тъй като не е подходящ сам по себе си за много потребители, използвам NodeJS стрийминг сървър (Mediasoup), който служи като SFU. Абстрактно погледнато, изпращаме нашия стрийм (аудио с OPUS кодек и видео с H264 кодек) към сървъра, а сървърът ги препраща към хората в съответната виртуална стая, в която се води

уебинара. Комуникацията между клиента и сървъра се извършва посредством уеб сокети, които позволяват двупосочна връзка.

VoD:

За създаването на VoD услуга е нужно видео съдържанието, което се качва да може да се стриймва онлайн. За целта съм избрал протокола HLS (HTTP Live Streaming), който позволява на потребителя да тегли видеото на отделни части/сегменти с различно качество на клипа, което зависи от интернет връзката. Причината за това е, че един уебинар варира между 30 минути до 2 часа и не е удобно за потребителя да чака да се зареди целия клип преди да може да го гледа.

За цялостната функционалност на приложението е нужен облачен транскодер, но тъй като тези услуги са много скъпи, засега използвам собствен сървър за обработка на видеата. Генерираните файлове се качват в облачно хранилище като DigitalOcean Spaces, които предоставят безплатен CDN с цел по-бързо да достигат до потребителя.

4.6. Описание на приложението:

- Преди да използваме функциите на системата, трябва да влезем в акаунта си. След това ще ни пренасочи към всичките качени уебинари, които са разделени по категории. Ако влизаме за пръв път, ще ни каже, че нямаме избрани интереси и че можем да си ги изберем, за да получаваме персонализирано съдържание. Засега всеки може да създава виртуални стаи и всеки да влиза в тях, но на по-късен етап само определени хора ще могат да създават уебинари на живо и само ако потребителят се е записал и си е заплатил, ще има право да влезе. Качването на видеа за момента е в ранна фаза, тъй като се разработва специализиран сървър, който да служи като транскодер. Също така, само определени хора, преминали през проверка, ще могат да организират и качват уебинари (content creators).

- Приложението е лесно за инсталация, тъй като сървърите са контейнеризирани с Docker и са лесни за инсталиране и експлоатация. Целият код се намира в Github, който се използва за контрол на версиите. Добавени са непрекъснатата интеграция (CI) и непрекъснатата доставка (CD) за автоматично тестване и деплойване на кода. (Build, Lint, Test, Deploy). Фронтендът се хоства във Vercel, а другите сървъри в различни VPS машини в DigitalOcean (Droplets).

4.7. Заключение

- LandCast е иновативна централизирана платформа, която улеснява живота на човека, давайки му възможност да гледа, да качва и да провежда уебинари на живо. Всичко на едно място.