Créer un clone de Netflix en 5h

Tout ça, de A à Z en partant de 0

PlanetHoster®
Host Your World

@LeDevULTIME

Pourquoi je fais ce live ?

Inspirer les développeurs juniors

Inspirer les entrepreneurs tech

Vulgariser

Montrer que c’est possible

“Tous ceux qui font des efforts n’auront pas forcément de résultat

Mais ceux qui ont des résultats ont dû faire des efforts”

Genji Kamogawa - Hajime No Ippo

Mon but ULTIME

Pour me soutenir

- Abonnez-vous à la chaîne YouTube @LeDevULTIME
- Abonnez-vous sur instagram @LeDevULTIME
- Abonnez-vous sur TikTok @LeDevULTIME

Sachez une chose

Je peux créer ce prototype de Netflix en 2h

Alors pourquoi 5h ?

Les étapes d’un projet web de A à Z

Même sans être développeur vous allez apprendre 🤯

Dans les grandes lignes

- Conception
- Modélisation
- Le streaming de vidéo
- Le code (frontend, backend, ...)
- Le paiement
- La mise en ligne

À la fin de ce live

Vous saurez

- Comment fonctionne une application web
- Comment on crée une application web
- Combien ça peut coûter
- Qui sont les différents acteurs pour ce genre de projet
- La différence entre faire un site pour 10 000 et 10 000 000 de personnes

Les différents types de développeurs et de projets.

Il y a plein de haters

Ils disaient

- C’est faux, tu ne peux pas le faire
- Tu vas voler le code de quelqu’un d’autre
- Tu vas juste faire du copier coller

Tant pi pour eux

Vous, vous allez tout voir 😀

Chaque projet est différent

Le but n’est pas de faire un code identique

à quoi ça servirait ?

On veut pas concurrencer Netflix, on veut apprendre

Le but

C’est d’avoir un clone qui fonctionne

Voir TOUTES les étapes pour en arriver là

Netflix de 2008
est différent
Netflix de 2024

16 ans

Netflix en 2024

c’est près de 3 000 ingénieurs sur près de 16 000 employés

Un aperçu globale

Et montrer qu’on peut prototyper rapidement

Je n’ai rien à vous vendre

5 mois
167 vidéos
Tout ça, gratuitement

Qui suis-je ?

Je suis Le Dev ULTIME

En quelques mots

- Ariel, 32 ans
- J’ai appris le code à 15 ans grâce à une passion pour les mangas
- J’ai fait un bac + 3 dans une école du multimédia
- 11 ans que je suis freelance
- J’ai bossé pour l’état français (Le ministère de l’intérieur et le 78)
- J’ai des articles qui sont lu par des gens chez Alibaba (Amazon chinois)
- 5 ans j’ai mes propres logiciels

J’ai bossé sur plus de 100 sites web et app web

Maintenant passons au concret

1. Conception

Les étapes

- Une idée / Identifier un problème
- Définir les fonctionnalités
- Définir le parcours utilisateur (user flow)
- Faire des croquis
- Faire des maquettes graphiques

Les outils

- Excalidraw
- Lucidchart
- FlowMapp
- Figma
- Draw.io
- Notion
- Dribbble.com (pour l’inspiration de design)

Ici plein d’outils de conception : https://uxtools.co/tools/design

Les acteurs

- Celui qui a eu l’idée
- Le chef de projet
- Le product owner / manager
- Le designer UX
- Le designer UI

Parfois, c’est la même personne

2. Modélisation

Les bdd

Une BDD c’est un gros fichier excel pour trier et ranger les données.

- SQL (MySQL, MariaDB, PostgreSQL, SQLite, ...) - tableaux
- NoSQL (Mongo, firebase, ...) - des documents

Comprendre les relations :
https://blog.devart.com/types-of-relationships-in-sql-server-database.html

Dans un même projet on peut

- plusieurs bases de données
- plusieurs serveurs / hébergements
- plusieurs langages de programmation

Tout dépend de la taille du projet !

3. Stack (technologies)

3 types de développeurs

- Développeur over engineering (il se prend la tête)
  - c'est bien si on fait un projet avec :
    - de la complexité
    - une grosse équipe
    - gros budget
    - temps
- Développeur à l'arrache (ils codent mal)
  - pas trop de sécurité
- Développeur qui DOSE
  - il sait compliqué quand il faut
  - il sait faire simple quand il faut
  - il sait mesurer le contexte : budget, temps, type de projet (équipe, besoins...)

2 façons d’apprendre, ceux

- qui comprennent par la pratique
- qui comprennent par la théorie

Pour développer on va avoir besoin

- D’un éditeur de code (VScode, PHPStorm, ...)
- D’un navigateur (Google Chrome, Firefox, Safari...)
- D’outils pour le frontend
  - le HTML et CSS (library, frameworks, preprocessors)
  - le JavaScript (library, frameworks, bundler)
- D’outils pour le backend
  - Langage, library, frameworks, CMS

Les définitions

- Library (bibliothèque) (que du code)
- Framework (cadre de travail) (que du code)
- CMS (content management system) (du code et une interface)

Ma stack

- HTML et CSS (SCSS, Bulma CSS)
- JavaScript : Vue JS (communication avec le backend, animations, UI...)
- CMS headless : Directus (Authentification, DB, API...)

Les types de rendus

- Static (HTML et CSS)
- Dynamique classique monolithe (PHP, Ruby...)
- Dynamique client (SPA single page application, web app)
- Dynamique SSR (mix de classique + SPA, avec NodeJS et un framework frontend)

Choix d’une techno

- votre expérience
- votre équipe
- les choix du clients
- le type de projet

4. Streaming vidéo

Les 5 façons de stream

- servir directement le fichier
  - compression / transcoding à gérer vous même
  - c’est lent car on charge “tout d’un coup”
  - ce n’est pas protégé
- serveur de stream
  - un fichier stocké
  - transcoder / compress
  - plusieurs variations de qualité (360, 720, 1080, 4k)
  - outil de stream RTMP ou HSL (renvoyer des petits fichiers par petits bouts)
  - ça peut vite coûter cher car besoin d’un gros CPU, GPU
  - authentification
  - pour les petit projets : c’est un peu cher car serveur dédié
  - pour les moyens : c’est le plus rentable
  - pour les gros : ça devient très cher

Les 5 façons de stream

- service cloud / de déléguer par API (bas niveau)
  - ils font just la grosse partie (moins cher)
  - AWS transcoding
  - moins cher sur le long terme
- service d’hébergement vidéo via API (haut niveau)
  - Bunny net, cloudflare, api.video, etc
  - plus cher sur le long terme
- plateformes vidéos intégrable
  - YouTube car vous connaissez, mais pas adapté
  - Vimeo
  - Wistia

ça ne sert à rien de réinventer la roue

library, frameworks, CMS, API externes, services...

5. Paiement

Il y a 4 types de systèmes de paiements

- Tout faire soi-même en collaboration avec une banque et leurs APIs
  - Rare sont ceux qui font ça maintenant
- Utiliser une API de paiement avec wallet / porte-monnaie
  - Mangopay par exemple
  - ça c'est idéal pour les sites de type marketplace, ou système avec plusieurs tiers
- Utiliser une API de paiement "direct"
  - Stripe, Lemonsqueezy, Paddle, Stancer, PayPal
- Utiliser une service de paiement externe par lien
  - PayPlug, PayPal, Stancer, Lydia
  - Là, Le client paye sur un site externe et va envoyer à votre app des infos via des webhooks par exemple

Vous pouvez

- Trouver des services en tapant “payment gateway”

- Pour des pays où Stipe et autre ne sont pas disponibles, vous pouvez taper “payment gateway [votre pays]” ou “api de paiement [votre pays]”

- Chercher des implémentations dans votre framework or CMS plutôt que de tout réimplémenter vous même (sauf si c’est pour apprendre)

Comment on gère le paiement ?

Chaque système va avoir une implémentation différente.

Mais le point commun entre tous, c’est de savoir :

- faire une modélisation qui peut s’adapter à plusieurs système de paiement
- avoir des webhooks qui vont aider à connecter les différents systèmes

Exemples de modélisation

https://github.com/laravel/cashier-stripe/tree/15.x/database/migrations

https://github.com/laravel/cashier-paddle/tree/2.x/database/migrations

6. Déploiement

En fonction des projets

- CI / CD
- Hébergement web
  - Mutualisé / Mutualisé boosté
  - VPS (mutualisé mais plus cloisonné et flexible)
  - Serveur dédié
  - Serveur cloud (scalable)
  - Serverless (aussi du Cloud mais d’une manière spécifique)
  - BaaS (backend as a service)
- Séparer les services
  - App sur un serveur
  - Fichiers sur un autre
  - Stream sur un autre… etc

Pour me suivre

- Instagram
  - LeDevUltime
- Tiktok
  - LeDevUltime
- Youtube
  - LeDevUltime

Merci !

Bientôt en live...

Créer un clone de Netflix en 5h
Tout ça, de A à Z en partant de 0

Petite pause...

Créer un clone de Netflix en 5h
Tout ça, de A à Z en partant de 0