

# Analyse de site Netflix

## 1. Frontend (ce que l'utilisateur voit)

### Éléments visibles :

- **Barre de navigation** : logo Netflix, bouton de connexion, menu.
- **Bannière principale** : un film ou une série mis en avant.
- **Vignettes de films/séries** classées par catégories : "Tendances", "Populaires", "Nouveautés", etc.
- **Lecteur vidéo** : avec contrôle du son, sous-titres, etc.
- **Effets visuels & animations** : carrousels, survol dynamique (hover), transitions.
- **Responsivité** : adaptabilité à différentes tailles d'écran (mobile, tablette, desktop).

### Technologies frontend utilisées :

Fonctionnalité	Technologie probable
Interface utilisateur	<b>HTML, CSS, JavaScript</b>
Framework frontend	<b>React.js</b> (Netflix l'utilise)
Gestion des états	<b>Redux</b> (ou React Context API)
Animations et effets dynamiques	<b>CSS3 animations, Framer Motion</b> ou lib JS
Adaptation multi-écrans (responsive)	<b>Media queries, Flexbox/Grid</b>
Contenu internationalisé	<b>i18n</b> avec React Intl ou autre

## 2. Backend (ce que l'utilisateur ne voit pas)

### Actions côté serveur :

- **Authentification & gestion de compte** : login, mot de passe, profil utilisateur.
- **Streaming vidéo** : lecture à la demande (VoD), ajustement qualité (adaptive streaming).

- **Recommandations personnalisées** : selon historique de visionnage.
- **Chargement dynamique du contenu** : API REST/GraphQL pour afficher les titres.
- **Système de paiement** : pour les abonnements.
- **Sécurité** : protection des données, gestion des droits d'accès.

## Technologies backend utilisées :

Fonctionnalité	Technologie probable
Serveur d'application	<b>Node.js, Java</b> ou <b>Python</b> (Netflix utilise Java et Node.js)
API de contenu	<b>GraphQL</b> ou <b>REST API</b>
Base de données	<b>MySQL, PostgreSQL, Cassandra, DynamoDB</b>
Recommandations & machine learning	<b>Python, Apache Spark, TensorFlow</b>
Streaming vidéo	<b>Amazon Web Services (AWS), Open Connect</b> (CDN maison Netflix)
Authentification	<b>OAuth2, JWT</b>
Paiement	Intégration de services comme <b>Stripe, Braintree</b> , etc.

## Communication entre frontend & backend :

- Le **frontend** interagit avec le **backend** via des **requêtes HTTP/HTTPS** (AJAX, Fetch API).
- Ces requêtes permettent :
  - de récupérer les films à afficher,
  - de mettre à jour un profil utilisateur,
  - de démarrer un stream vidéo,
  - ou encore de noter un film/série.