

# Mise en place d'une architecture full stack

Chapellier Logan

Vanneste Lucas

S4 DACS



UNIVERSITÉ  
DE LORRAINE



nancy

Charlemagne  
Informatique

# Sommaire

- Présentation des fonctionnalités de l'application
- Présentation des différents Dockers
- Description de la sécurité globale (code,docker,ports)
- Problèmes et améliorations possible
- Bonus (Dimension économique)
- Démonstration de l'application

# Présentation des fonctionnalités

**Wishlist Application**

**Registration**

Username

Password

**Register**

**Login**

Username

Password

**Login**

Page d'accueil

**Wishlist Application**

**My Wishlist**

Smartphone - 799.99 (phone, technology)

**Lists Shared with Me**

Smartwatch - 299.99

Item Name

0

Keywords

**Add**

**Share My Wishlist**

user2  
user3  
user4  
user5  
...

**Share with Selected Users**

Page une fois connecté

# Docker frontend

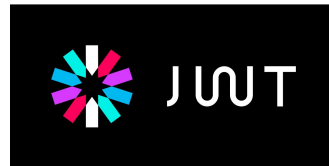
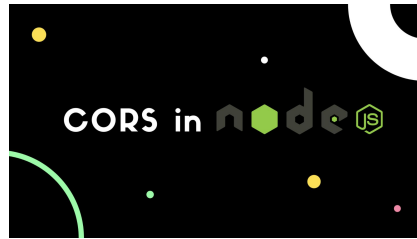


Vue.js :  
Framework JavaScript pour faciliter la création  
et la maintenance de l'interface utilisateur

Vuex :  
Bibliothèque de gestion d'état pour les  
applications Vue.js

Axios :  
Client HTTP pour communiquer avec le  
backend de manière asynchrone et efficace

# Docker backend



Node.js et Express.js :

Technologie pour la création d'API robuste et rapide.  
Express.js offre une architecture légère et flexible pour le développement.

PostgreSQL :

Système de gestion de base de données utilisé avec la partie client pour se connecter à la base de données

Bcrypt :

Bibliothèque de hachage de mots de passe pour protéger les mots de passe des utilisateurs

JSON Web Tokens (JWT) :

Permet de créer des jetons d'authentification sécurisés et de leur attribuer une durée

CORS (Cross-Origin Resource Sharing) :

Permet de sécuriser l'accès aux ressources du serveur backend en autorisant uniquement les domaines de confiance à interagir avec l'API

# Docker base de données



PostgreSQL :

Simplifie le déploiement et la gestion de la base de données. Il devient plus facile de déployer la base de données de manière cohérente et reproductible sur différents environnements.

# Docker script d'insertion



Python :

Il offre une variété de bibliothèques pour interagir avec les bases de données, ce qui en fait un choix naturel pour le script d'insertion de données.

# Sécurité au niveau du code



JSON Web Tokens (JWT) :

Stockage de clé privée pour la génération des tokens dans une variable d'environnement

CORS (Cross-Origin Resource Sharing) :

Filtrage des types de requêtes API autorisé

Bcrypt :

Utilisation de la fonction de hachage de Bcrypt avec un cost de 10



# Sécurité des réseaux docker

```
networks:  
  frontend-backend-net:  
    driver: bridge  
  backend-db-net:  
    driver: bridge  
  db-script-net:  
    driver: bridge
```

Exemple avec le backend :

```
backend:  
  build: ./backend  
  ports:  
    - "5000:5000"  
  depends_on:  
    - db  
  networks:  
    - frontend-backend-net  
    - backend-db-net
```

# Sécurité au niveau gestion des ports

- Frontend : port 8085
- Backend : port 5000
- Base de données : port 5432
- Insertion Script : pas de port spécifié

Composant Source	Composant Cible	Type de Communication	Protocole	Ports Utilisés	Description
Frontend	Backend	Requêtes API	HTTP/HTTPS	5000	Le frontend envoie des requêtes API au backend pour gérer l'authentification, les wishlists, etc.
Backend	Base de Données	Requêtes SQL	TCP/IP	5432	Le backend interagit avec la base de données PostgreSQL pour lire et écrire des données.
Script d'Insertion	Base de Données	Requêtes SQL	TCP/IP	5432	Le script d'insertion envoie des requêtes SQL pour insérer des données initiales dans la base
Utilisateur	Frontend	Navigation Web	HTTP/HTTPS	8085	Les utilisateurs accèdent à l'interface de l'application via le frontend.

# Problème et amélioration possible

## Problème rencontré

- Compréhension des dockerfiles
- Partage de wishlist

## Amélioration possible

- Faire des authentications constante
- Possibilité de supprimer des listes ou des articles
- Amélioration du style du site
- Supprimé le partage d'une wishlist

# Estimation des coûts de développement

Catégorie	Coût (en €)
Développeurs	3000
Outils Logiciels	500
Frais Généraux	600
Serveurs	2400
Bases de Données	600
Services Cloud Additionnels	1000
Maintenance	600
Support Client	1000
Total	9700

Plus-Value pour l'entreprise :

Amélioration de l'expérience client :

Potentiel augmentation du nombres d'utilisateur et par conséquent des bénéfices

Ajout de fonctionnalité bénéfique au site

# Démonstration de l'application