

École La Plateforme_

Dossier Projet

Post-hit

Titre Professionnel Développeur Web et Web Mobile

Présenté par : Esteban Houppermans

Année : 2025

Table des matières

- 1. Introduction
- 2. Liste des compétences couvertes par le projet
- 3. Cahier des charges
 - Objectifs
 - Public cible
 - User Stories
 - Arborescence du projet
 - MVP
 - Évolutions potentielles

- Wireframes & Maquettes
 - 4. Développement de la partie back-end
 - Technologies et dépendances
 - Architecture
 - Sécurité
 - Base de données
 - 5. Développement de la partie front-end
 - Technologies et dépendances
 - Fonctionnalités
 - Organisation et arborescence
 - 6. Conclusion
-

1. Introduction

Le projet **Post-hit** est un projet fullstack développé dans le cadre de la formation **Développeur Web et Web Mobile** à l'école **La Plateforme**.

Il vise à concevoir une application complète combinant :

- un **frontend moderne** (Angular 20),
- un **backend robuste** (Node.js + Express + MariaDB),
- une **authentification sécurisée** (JWT, Google Auth),
- un **système de paiement intégré** (Stripe),
- une **mise en production professionnelle** (Docker + Nginx).

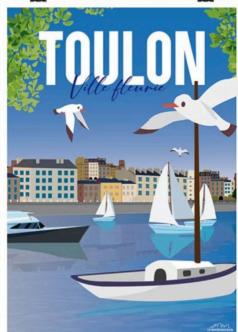
Ce projet illustre la maîtrise de bout en bout du cycle de développement d'une application web moderne : **analyse, conception, développement, sécurisation et déploiement**.

Nouvelle Collection 2025

Posters d'Art en 3D

Découvrez notre collection unique de posters avec prévisualisation 3D interactive.

Transformez votre espace avec des œuvres d'art soigneusement sélectionnées.

[Découvrir la Collection](#)[Voir en 3D](#) [Vue 3D Interactive](#) [Achat Simplifié](#)

500+

Posters Uniques

10k+

Clients Satisfait

24h

Support Réactif

[Minimaliste](#)[Informatique](#)[Meilleures ventes](#)[Éditions limitées](#)[Vintage](#)[Gaming](#)

Black Cat Sushi



Fat Cat Toilet are pooping



John Cena as Chairman Mao



Wanted Monkey D. Luffy

The screenshot shows the Posthit website interface. At the top, there's a navigation bar with the logo "Posthit" and links for "Accueil", "Galerie", "À propos", "Contact", a shopping cart icon, and a "Login" button. Below the navigation, there's a grid of three items:

- Interstellar Black Hole**: An image of a black hole with a bright accretion disk.
- Kermit Meme Wpap Pop Art**: A stylized, colorful version of Kermit the Frog holding a glass.

Below the grid, there are three columns of links:

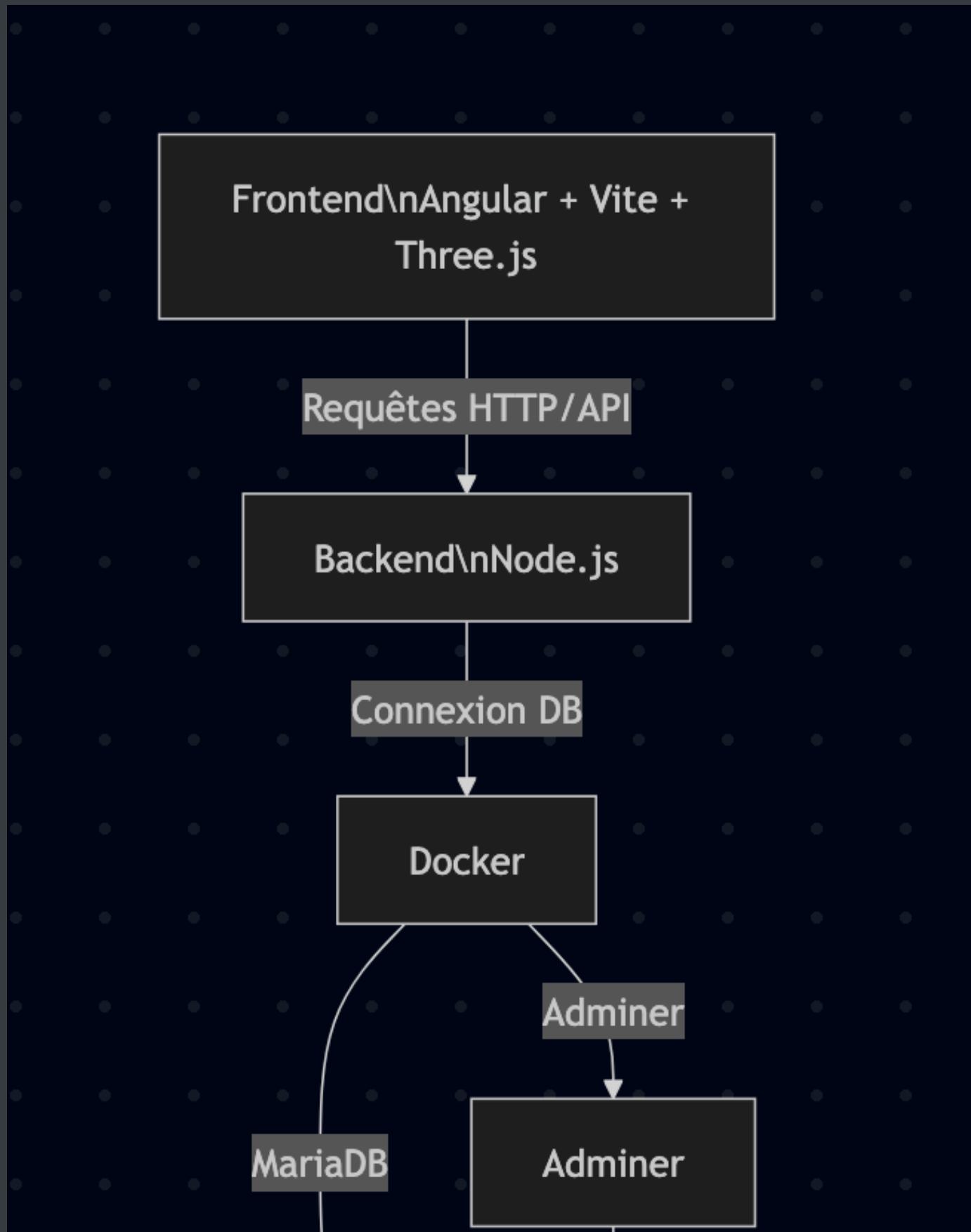
- Information**: Livraison, Paiement sécurisé, CGV, Contact.
- À propos**: Qui sommes-nous ?, Notre mission, Nos valeurs.
- Suivez-nous**: Icons for GitHub, Instagram, and Google.

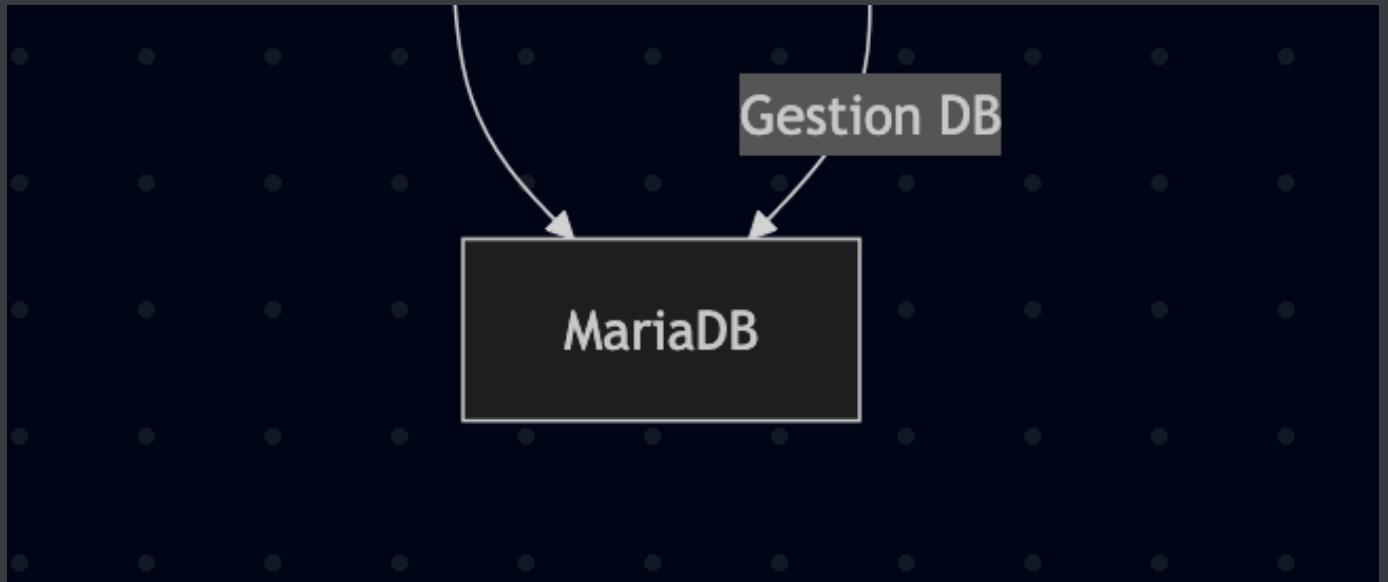
At the bottom center, it says "© Post'Hit. Tous droits réservés."

2. Liste des compétences couvertes par le projet

- Développement front-end avec Angular 20
- Développement back-end avec Node.js / Express
- Mise en place et gestion d'une base de données relationnelle MariaDB
- Création et sécurisation d'API REST
- Intégration d'un système de paiement Stripe
- Authentification sécurisée (JWT, bcrypt, Google OAuth)
- Gestion des rôles et permissions
- Intégration graphique avancée avec **Three.js**

- Déploiement conteneurisé (Docker, Nginx)
- Versionning avec Git/GitHub
- Mise en place d'un environnement de production sécurisé (HTTPS, SSL, .env)





3. Cahier des charges

Objectifs

- Créer une plateforme intuitive et sécurisée.
- Permettre aux utilisateurs de créer un compte et gérer leur profil.
- Faciliter les paiements via Stripe.
- Proposer une interface moderne, responsive et performante.

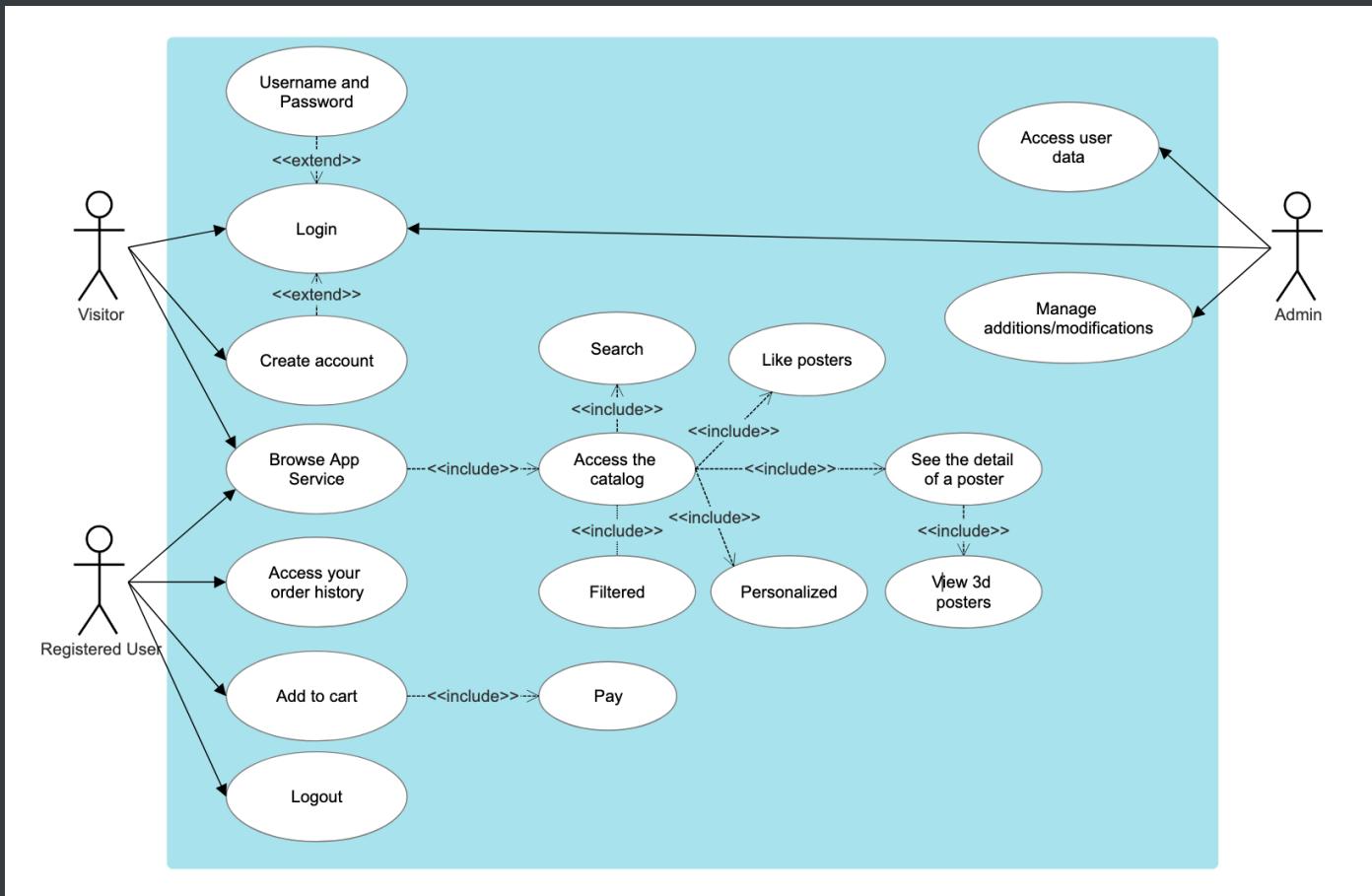
Public cible

Le projet cible les utilisateurs recherchant une application simple d'utilisation, accessible sur **desktop** et **mobile**.

User Stories

- **Visiteur** : consulter du contenu public, créer un compte.
- **Utilisateur connecté** : gérer son profil, interagir avec du contenu, réaliser un paiement.

- **Administrateur** : gérer les utilisateurs et superviser l'application.



Arborescence du projet

Post-hit/

```

├── backend/ # API Node.js / Express
├── front/ # Angular 20
├── nginx_proxy/ # Configuration Nginx
├── docs/ # Use cases, maquettes
└── BDD/ # Export MariaDB

```

```

.
├── backend
│   ├── config
│   ├── controllers
│   ├── models
│   ├── public
│   └── routes

```

```
├── docs
├── front
|   ├── dist
|   |   └── front
|   |       └── browser
|   ├── public
|   |   └── font
|   |       └── Asap
|   └── src
|       ├── app
|       |   ├── about-us
|       |   ├── burger-menu
|       |   ├── button
|       |   ├── cart
|       |   ├── cgv
|       |   ├── contact
|       |   ├── footer
|       |   ├── form
|       |   ├── likest-product
|       |   ├── link
|       |   ├── login
|       |   ├── main
|       |   ├── navbar
|       |   ├── payement
|       |   ├── product-page
|       |   ├── return-btn
|       |   ├── searchbar
|       |   ├── shop
|       |   ├── success
|       |   ├── test3-d
|       |   ├── theme
|       |   └── user-profil
|       └── environments
└── nginx_proxy
    ├── data
    |   └── access
```

```
|   └── custom_ssl
|   └── letsencrypt-acme-challenge
|   └── logs
|   └── nginx
|       ├── dead_host
|       ├── default_host
|       ├── default_www
|       ├── proxy_host
|       ├── redirection_host
|       ├── stream
|       └── temp
└── letsencrypt
```

MVP

- Authentification sécurisée (JWT + bcrypt + Google Auth).
- Gestion des utilisateurs et des rôles.
- Paiement Stripe.
- Interface Angular responsive.

Évolutions potentielles

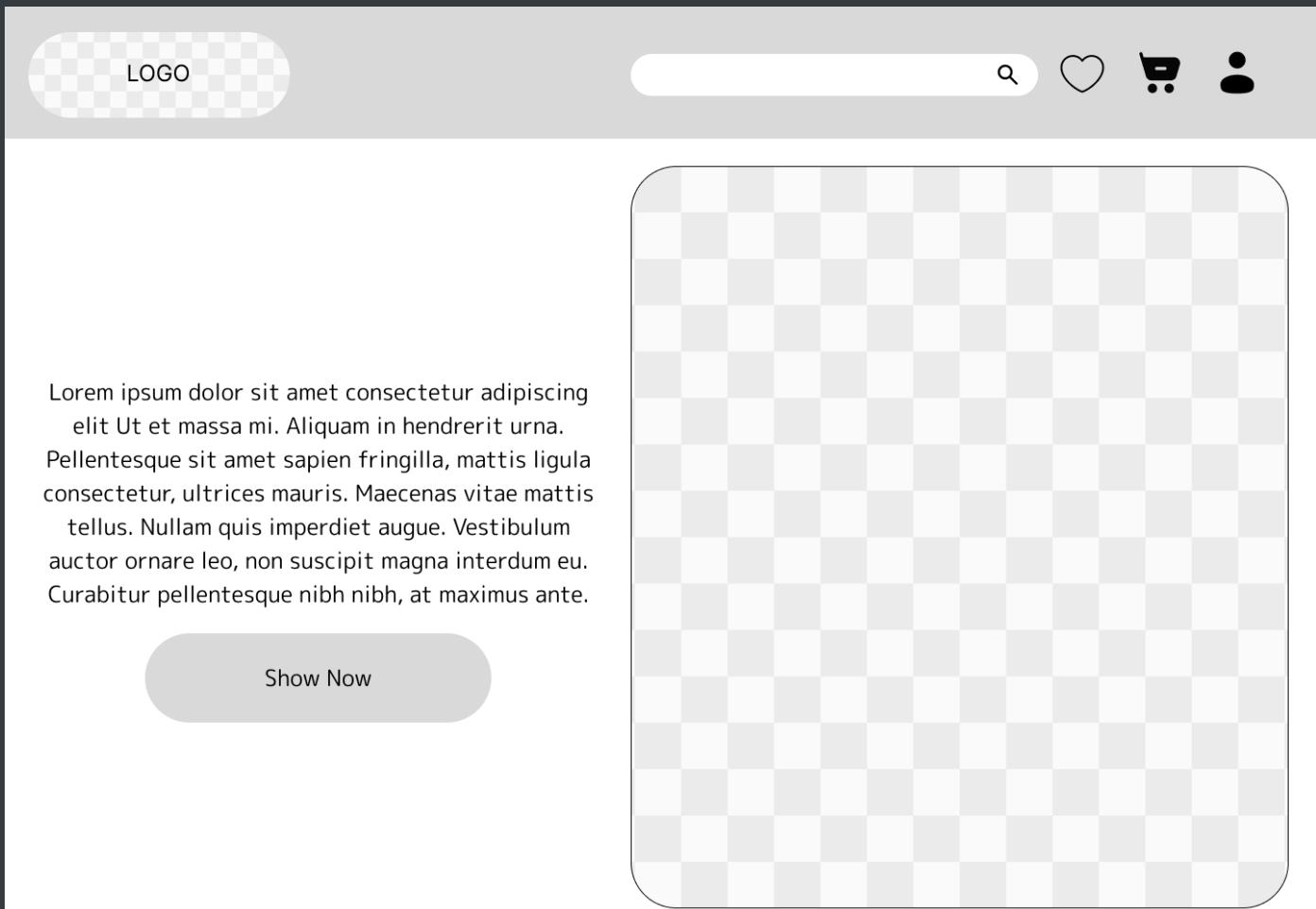
- Messagerie interne.
- Notifications en temps réel.
- Internationalisation (FR/EN).
- Nouvelles intégrations 3D avec Three.js.

Wireframes & Maquettes

Les maquettes (cf. `/docs/maquettes.pdf`) présentent :

- Une **page de connexion**
- Une **page d'accueil**

- Une **page profil**
- Une **page paiement**
- Un **espace admin**



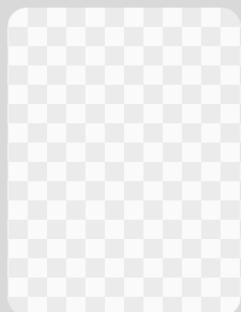
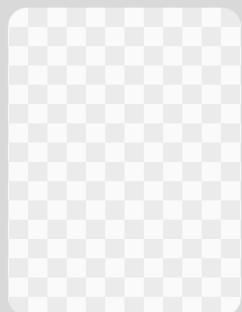
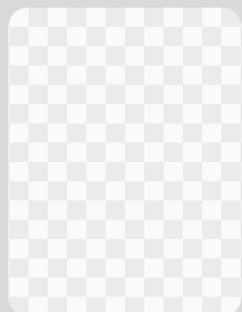
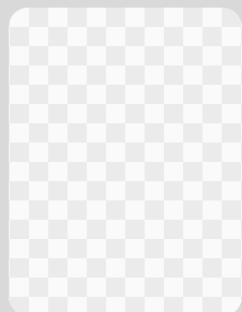
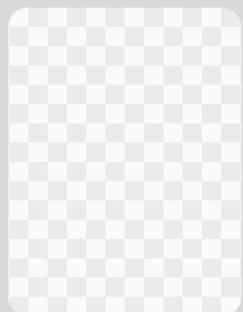
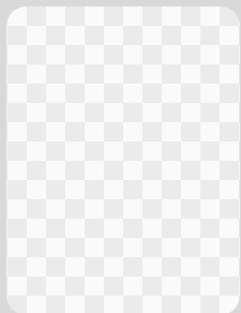
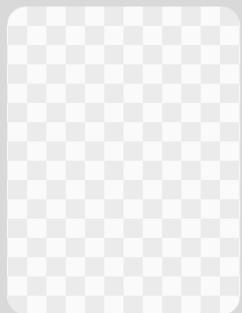
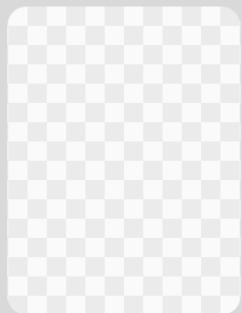
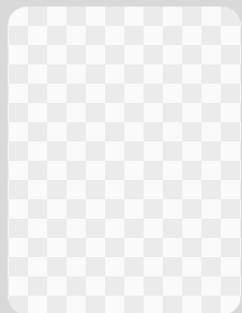
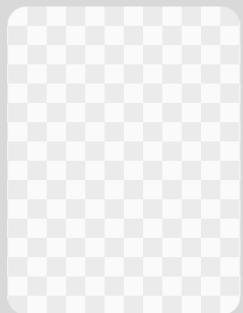
Licensed artworks from top brands

[see all...](#)



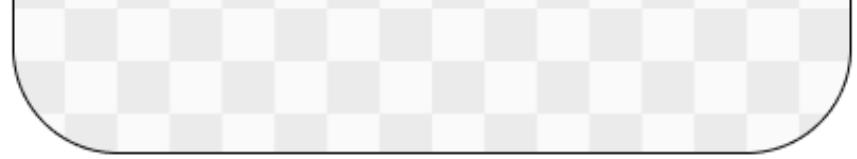
Title Title Title Title Title Title Title

See the hottest picks



LOGO



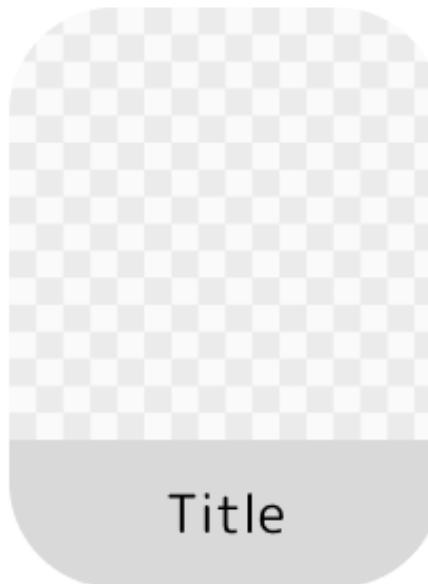


Licensed artworks from top brands

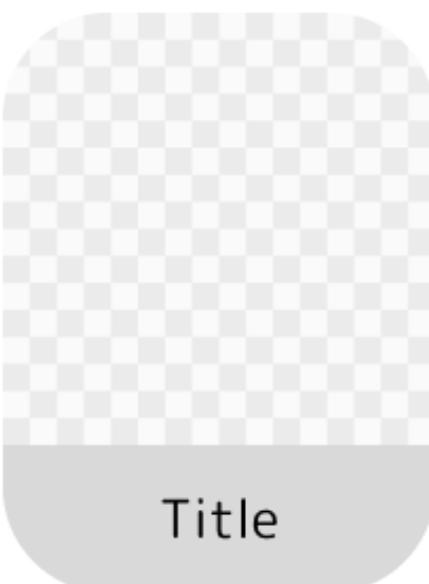
[see all...](#)



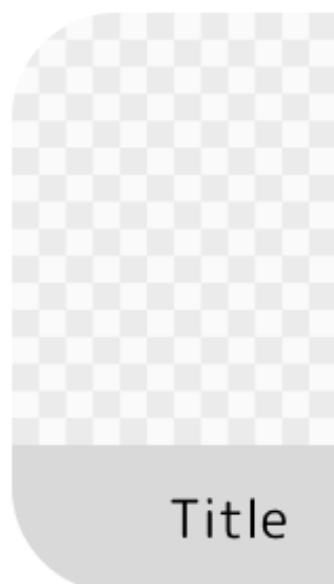
Title



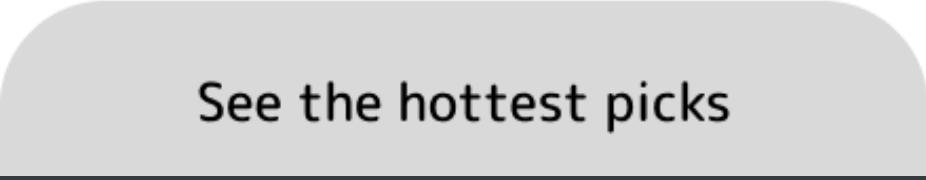
Title



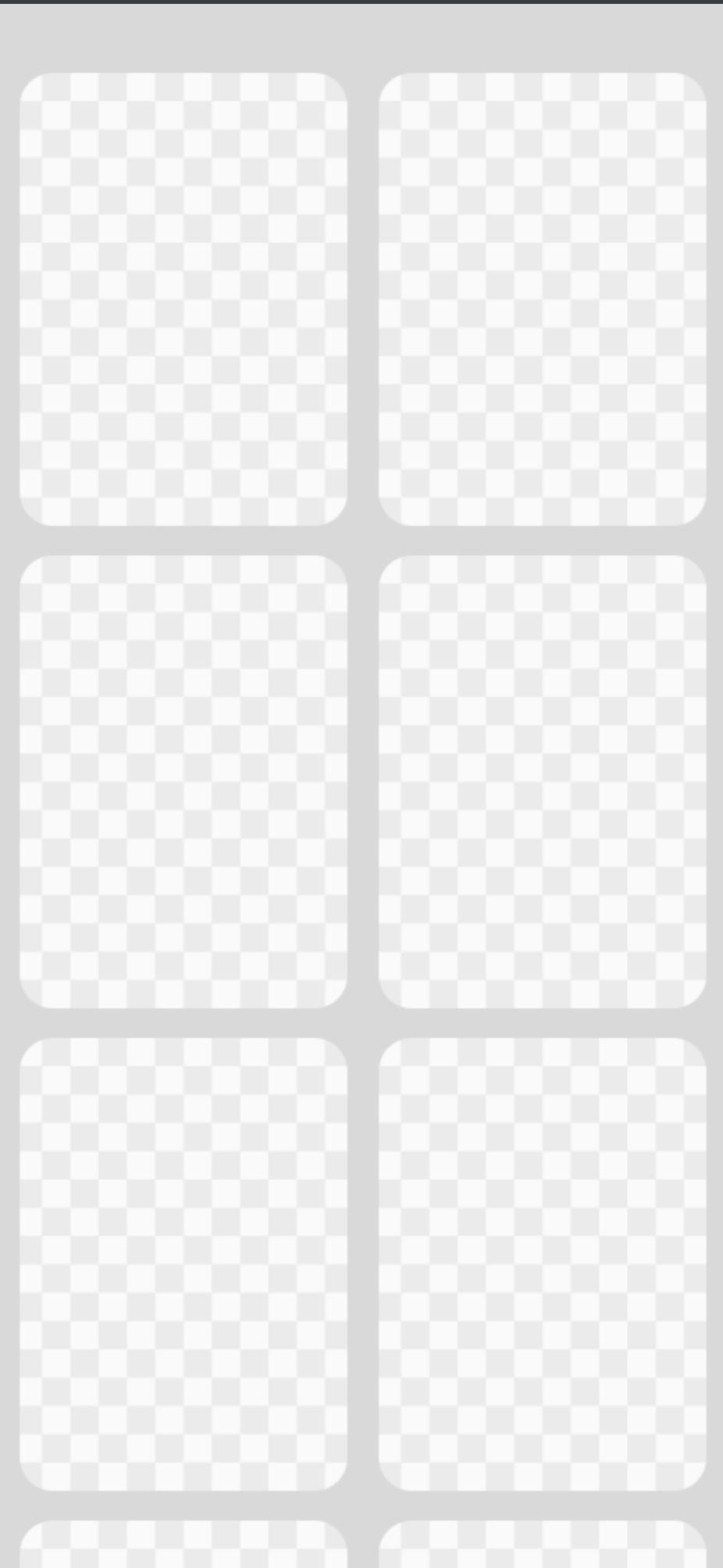
Title



Title



See the hottest picks





CONTACT
CONTACT
CONTACT
CONTACT
CONTACT

CONTACT
CONTACT
CONTACT
CONTACT
CONTACT



LOGO



Email Enter your email

We'll never share your email with anyone else.

Password Enter your password

Use at least 8 characters.

[Forgot Password?](#)[Sign Up](#)[Log In](#)

4. Développement de la partie back-end

Technologies et dépendances

- **Node.js + Express** : serveur et API REST.
- **MariaDB (mysql2)** : base de données relationnelle.
- **bcrypt** : hashage des mots de passe.
- **jsonwebtoken (JWT)** : gestion des sessions.
- **nodemailer** : envoi d'e-mails.
- **stripe** : intégration paiement en ligne.
- **google-auth-library** : authentification sociale.
- **express-fileupload** : upload de fichiers.
- **dotenv, cors, morgan** : configuration, sécurité et logs.

```
{  
  "name": "bdd",  
  "version": "1.0.0",  
  "description": "",  
  "license": "ISC",  
  "author": "",  
  "type": "module",  
  "main": "server.js",  
  ▷ Déboguer  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1",  
    "start": "node server.js",  
    "dev": "nodemon server.js"  
  },  
  "dependencies": {  
    "bcrypt": "^6.0.0",  
    "cors": "^2.8.5",  
    "dotenv": "^16.4.5",  
    "express": "^5.1.0",  
    "express-fileupload": "^1.5.1",  
    "google-auth-library": "^9.0.0",  
    "jsonwebtoken": "^9.0.2",  
    "morgan": "^1.10.1",  
    "mysql2": "^3.14.1",  
    "node-fetch": "^2.7.0",  
    "nodemailer": "^7.0.3",  
    "nodemon": "^3.1.10",  
    "stripe": "^12.18.0"  
  }  
}
```

Architecture

Le backend suit une architecture MVC simplifiée :

- **Routes** : définissent les endpoints de l'API.

- **Contrôleurs** : implémentent la logique métier.
- **Modèles** : gèrent l'accès aux données via MariaDB.
- **Middlewares** : sécurité, validation JWT, gestion des rôles et erreurs.

```
/Users/este./Desktop/v2/Post-hit/backend/config
├── db.js
└── google-auth.js
├── mail.js
└── stripe.js
/Users/este./Desktop/v2/Post-hit/backend/controllers
├── auth.controller.js
├── bestselling.controller.js
├── cart.controller.js
├── categories.controller.js
├── images.controller.js
├── mail.controller.js
├── products.controller.js
└── stripe.controller.js
├── user.controller.js
/Users/este./Desktop/v2/Post-hit/backend/models
├── bestselling.model.js
├── cart.model.js
├── categorie.model.js
├── product.model.js
└── productsize.model.js
└── user.model.js
```

Capture d'écran

Sécurité

- Hashage systématique des mots de passe (**bcrypt**).
- Authentification via **JWT** (stockés en cookies HttpOnly).
- Authentification sociale via **Google OAuth**.
- Gestion des rôles (user / admin).

```

export const signUp = async (req, res) => {
  try {
    const { username, email, password } = req.body;
    if (!username || !email || !password) return res.status(400).json("Champs manquants");
    if (!StrongPassword.test(password)) return res.status(400).json("Mot de passe non sécurisé");

    const hashedPassword = await bcrypt.hash(password, 10);
    const user = await userModel.addUser(username, email, hashedPassword);
    if (!user) return res.status(409).json("Nom d'utilisateur déjà pris");
    res.status(201).json(`Utilisateur ${username} créé avec succès`);
  } catch (err) {
    res.status(500).json("Erreur serveur");
  }
};

```

Base de données

La base MariaDB contient plusieurs tables :

- `user` (`id, username, email, password`)
- `bestselling` (`id, product_name, product_price, product_theme, product_desc`)
- `cart` (`id, user_id, product_id, quantity, created_at`)
- `categorie` (`id, categorie_name`)
- `product` (`id, product_name, product_price, product_theme, product_desc`)
- `productsize` (`id, product_id, product_height, product_width`)

	Table	Moteur?	Interclassement?	Longueur des données?	Longueur de l'index?	Espace inutilisé?	Incrément automatique?	Lignes?	Commentaire?
□	bestselling	InnoDB	utf8mb4_ucs2_ci	16,384	0	0	31	~ 18	
□	cart	InnoDB	utf8mb4_ucs2_ci	16,384	0	0	105	~ 13	
□	categorie	InnoDB	utf8mb4_ucs2_ci	16,384	0	0	16	~ 8	
□	product	InnoDB	utf8mb4_ucs2_ci	16,384	0	0	84	~ 72	
□	productsize	InnoDB	utf8mb4_ucs2_ci	16,384	0	0	3	~ 1	
□	user	InnoDB	utf8mb4_ucs2_ci	16,384	16,384	0	31	~ 4	
6 au total		InnoDB	utf8mb4_ucs2_ci	98,304	16,384	0			

Exemple de table (User):

Colonne	Type	Commentaire
id	int(11) <i>Incrément automatique</i>	
username	varchar(255) <i>NULL</i>	
email	varchar(255) <i>NULL</i>	
password	varchar(255) <i>NULL</i>	

Contenus de la table user :

Modification	id	username	email	password
modifier	1	admin	admin	admin
modifier	26	estebanlebg	esteban.h0207@gmail.com	\$2b\$10\$Y7OzVSomJLY6GEkWAala4ueYoDqeWV0fGBf/kDfFuoJSP4G2Z0tG6
modifier	28	tay	0xhex4@gmail.com	\$2b\$10\$MaS1/Ii9nW1OCZUN0p4Xguz9v4Mb01C.jTu6lwu1CBnFhm/2wuSpy
modifier	29	test	lenij14414@asimarif.com	\$2b\$10\$9kw1Ggajm4yrRDKMc/wHouHjtmvPj5wTL6CTW1WZdtYDWcdGET8.
modifier	30	namoux	namoux_123@gmail.com	\$2b\$10\$3ImMIZJEv0cCe7yqaNHrUubGV66CC0Iva8Ebit5Zwv1JD1D4Y3kvO

5. Développement de la partie front-end

Technologies et dépendances

- **Angular 20** : framework principal.
- **ngx-stripe** : intégration Stripe.
- **RxJS, zone.js, tslib** : réactivité et gestion des flux.
- **Three.js & three-stdlib** : animations et 3D.
- **Vite** : build rapide et optimisé.
- **Vercel Analytics** : suivi des performances.



```
"name": "front",
"version": "0.0.0",
▷Déboguer
"scripts": {
  "ng": "ng",
  "start": "ng serve",
  "build": "ng build",
  "watch": "ng build --watch --configuration development",
  "test": "ng test"
},
"private": true,
"dependencies": {
  "@angular/common": "^20.0.0",
  "@angular/compiler": "^20.0.0",
  "@angular/core": "^20.0.0",
  "@angular/forms": "^20.0.0",
  "@angular/platform-browser": "^20.0.0",
  "@angular/router": "^20.0.0",
  "@vercel/analytics": "^1.5.0",
  "@vercel/speed-insights": "^1.2.0",
  "dotenv": "^16.5.0",
  "ngx-stripe": "^20.7.0",
  "platform-browser-dynamic": "^0.0.1-security",
  "rxjs": "~7.8.0",
  "three": "^0.178.0",
  "three-stdlib": "^2.36.0",
  "tslib": "^2.3.0",
  "zone.js": "~0.15.0"
},
"devDependencies": {
  "@angular/build": "^20.0.0",
  "@angular/cli": "^20.0.0",
  "@angular/compiler-cli": "^20.0.0",
  "@types/jasmine": "~5.1.0",
  "@types/three": "^0.177.0",
  "jasmine-core": "~5.7.0",
  "karma": "~6.4.0",
  "karma-chrome-launcher": "~3.2.0",
  "karma-coverage": "~2.2.0"
}
```

```
    "karma-coverage": "~2.0.0",
    "karma-jasmine": "~5.1.0",
    "karma-jasmine-html-reporter": "~2.1.0",
    "typescript": "~5.8.2",
    "vite": "^6.3.5"
}
}
```

Fonctionnalités

- Authentification (login, signup, Google Auth).
- Gestion du profil utilisateur.
- Paiement via Stripe Checkout.
- Affichage du contenu (produits).
- Interface responsive et moderne.
- Intégration 3D via Three.js.

```
<div class="login-container">
  <form (ngSubmit)="onLogin()" [FormGroup]="loginForm">
    <h1>Connexion</h1>
    <label for="email">Adresse e-mail</label>
    <input type="email" id="email" formControlName="email" required autocomplete="email">
    <div *ngIf="loginForm.get('email')?.invalid && loginForm.get('email')?.touched" class="error-message">
      Veuillez entrer une adresse e-mail valide.
    </div>

    <label for="password">Mot de passe</label>
    <input type="password" id="password" formControlName="password" required autocomplete="current-password">
    <div *ngIf="loginForm.get('password')?.invalid && loginForm.get('password')?.touched" class="error-message">
      Le mot de passe est requis.
    </div>

    <button type="submit" [disabled]="loginForm.invalid || loading">{{ loading ? 'Connexion...' : 'Se connecter' }}</button>
    <a class="forgot-link" (click)="onForgotPassword()">Mot de passe oublié ?</a>
  </form>
</div>
```



Connexion

Email

Entrez votre email

Mot de passe

Entrez votre mot de passe

Se connecter

Pas encore de compte ? [S'inscrire](#)

Animals

Minimaliste

Informatique

Meilleures ventes

Éditions limitées

Vintage

Prix croissant

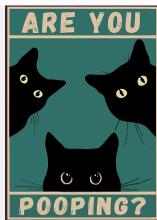
Prix décroissant

Nom A-Z

Nom Z-A



Black Cat Sushi



Fat Cat Toilet are pooping



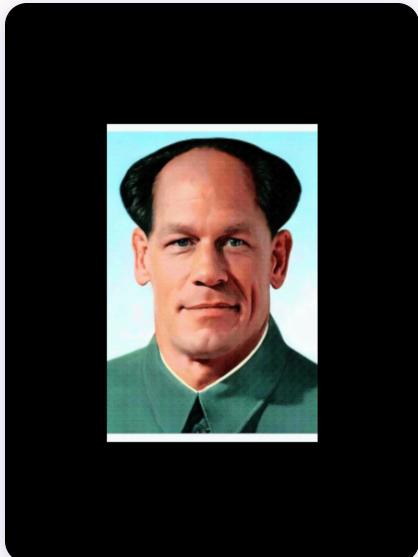
John Cena as Chairman Mao



Wanted Monkey D. Luffy



In



Sans nom

Change Color :

Aucune description disponible

Prix non disponible

ADD to Cart



Cart

2 article(s)
Order total: 93.00€

Shit Happens Funny Poop
Hilarious Funny Bathroom Sign. Shit Happens! Funny Angry Poop Character. Perfect for hanging up in your bathroom. Bathroom home decor.
44€

Ocean Dreams
Illustration sous-marine avec dauphins et récifs colorés.
49€

Empty your cart **Payer**

Posthit [Accueil](#) [Galerie](#) [À propos](#) [Contact](#) [Panier](#) [Pay](#) [Se déconnecter](#)

Adresse e-mail :

Message :

Envoyer

Posthit
16 Rue Albert, 83000 Toulon
[Agrandir le plan](#)

Notre-Dame-de-la-Seds, New School, Iacos-Toulon, Port de Toulon, Av. de la République, Stade Mayol, FNAC Toulon, UFR Faculté de Droit, INTERSPORT TOULON MAYOL.

Données cartographiques Conditions d'utilisation Signaler une erreur cartographique

Information
Livraison
Paiement sécurisé

À propos
[Qui sommes-nous ?](#)
[Notre mission](#)

Suivez-nous

Figure 16 : Capture d'écran du paiement Stripe.

Figure 17 : Capture d'écran d'une animation Three.js.

Organisation et arborescence

L'application est structurée de manière modulaire :

- **Composants** : LoginComponent, ProfileComponent, PaymentComponent, AdminDashboardComponent.
- **Services** : AuthService, ApiService, PaymentService.

```
.
```

- | └── about-us
- | └── burger-menu
- | └── button
- | └── cart
- | └── cgv
- | └── contact
- | └── footer
- | └── form
- | └── likest-product
- | └── link
- | └── login
- | └── main
- | └── navbar
- | └── payement
- | └── product-page
- | └── return-btn
- | └── searcbar
- | └── shop
- | └── success
- | └── test3-d
- | └── theme
- └── user-profil

6. Conclusion

Le projet **Post-hit** m'a permis de travailler sur toutes les étapes du développement d'une application web complète :

- **Backend** : création d'API REST sécurisées, gestion de la BDD, intégration de Stripe et Google Auth.
- **Frontend** : développement d'une interface Angular moderne, responsive et intégrant du 3D.

Ce projet illustre ma capacité à **concevoir, développer et déployer** une application web moderne tout en respectant les **bonnes pratiques de sécurité, de performance et de maintenabilité**.

Il représente un jalon important dans mon apprentissage et démontre une montée en compétence sur l'ensemble de la stack **fullstack JS moderne**.

The screenshot shows the Posthit website homepage. At the top, there's a navigation bar with the logo "Posthit" in pink, followed by links for "Accueil", "Galerie", "À propos", "Contact", a shopping cart icon, and a "Login" button. Below the navigation, three blue callout boxes highlight statistics: "500+ Posters Uniques", "10k+ Clients Satisfais", and "24h Support Réactif". A horizontal row of category filters includes "Minimaliste", "Informatique", "Meilleures ventes", "Éditions limitées", "Vintage", and "Gaming". The main content area displays a grid of four posters. The first poster is a black cat eating sushi with the caption "Black Cat Sushi". The second poster features two black cats with the text "ARE YOU POOPING?". The third poster shows John Cena dressed as Chairman Mao with the caption "John Cena as Chairman Mao". The fourth poster is a wanted poster for Monkey D. Luffy from One Piece with the caption "Wanted Monkey D. Luffy". Each poster has a small preview image at the bottom.