



Rapport de stage

Développement Web Full Stack

Hugo POQUET
BTS SIO – Spécialité SLAM
Année 2022 - 2023



**ÉTABLISSEMENT
SAINT-ADJUTOR**

Sommaire :

| | |
|--|----|
| REMERCIEMENTS | 3 |
| Présentation de l'entreprise | 4 |
| Service d'accueil et moyens informatiques | 5 |
| Mon rôle et ma mission : | 6 |
| GitHub et Trello..... | 7 |
| GitHub : | 7 |
| Trello :..... | 8 |
| Présentation du projet..... | 9 |
| Phase de formation : | 10 |
| Le framework Angular :..... | 11 |
| Mission 1 : Création de la base de données : | 14 |
| Mission 2 : Création d'une page de connexion : | 16 |
| Mission 3 : Création d'une page « Mon compte » : | 20 |
| Mission 4 : Création du panier :..... | 23 |
| Mission 5 : Page de récapitulatif commande : | 27 |
| Mission 6 : Module de paiement : | 34 |
| Compétences acquises..... | 43 |
| Compétences techniques : | 43 |
| Compétences professionnelles et sociales : | 43 |
| Conclusion..... | 44 |
| Annexes..... | 45 |

REMERCIEMENTS

Tout d'abord, je souhaiterais remercier Lucas Schnekenburger, mon tuteur pour ces six semaines de stage riches en expérience, pour son accueil et l'aide qu'il m'a apporté. Il a su être de bon conseil et à l'écoute.

Je tiens ensuite à remercier les enseignants et enseignantes du BTS SIO au lycée Saint-Adjutor à Vernon qui m'ont permis d'acquérir de nombreuses compétences ayant été très utiles durant ce stage.

Enfin j'aimerais remercier mes collègues stagiaires, pour les conditions de travail très agréables et l'entre-aide dont ils ont fait preuve.

Présentation de l'entreprise

Altameos Multimedia est une entreprise située à Evreux qui se spécialise dans les solutions multimédia. Elle a été fondée en 2003 avec l'objectif de fournir aux entreprises de la région des solutions de communication innovantes pour améliorer leur image et leur visibilité sur le marché.

L'entreprise offre une large gamme de services tels que la création de sites internet, le web design, l'hébergement, le référencement, ainsi que des conseils informatiques.

Elle travaille avec de grandes entreprises telles que Vinci, Renault, mais aussi auprès de particuliers.

Altameos Multimedia est uniquement composée de Lucas Schnekenburger, un professionnel passionné et qualifié, travaillant en tant qu'indépendant libéral et proposant les meilleures solutions multimédia à ses clients pour répondre à leurs besoins.



Service d'accueil et moyens informatiques

Dans un premier temps, nous avons été accueillis par notre maître de stage sur Teams qui nous a présenté le projet sur lequel nous allions travailler. Nous avons donc eu comme une sorte de contact avec le client, avec une expression de besoin.

Pour ce qui est du matériel, nous avons utilisé notre matériel personnel étant donné que nous avons réalisé ce stage en télétravail.

| Logiciels | Description |
|-------------------------|--|
| Microsoft Visual Studio | Grâce auquel j'ai pu développer dans tous les langages nécessaires à mes tâches. |
| Wampserver | Qui m'a permis d'effectuer du traitement de base de données, notamment en SQL. |
| Microsoft Teams | Qui nous a permis de faire des points sur l'avancée du projet plusieurs fois par jour en appel audio. |
| Microsoft Word | Qui m'a été bien utile quant à la rédaction de mon rapport de stage. |
| Parsec | Que nous utilisions pour prendre la main sur le poste d'un autre et rendre l'entre-aide plus simple et rapide. |
| GitKraken | Un éditeur nous permettant de faire du versioning pour la mise en commun tout au long du projet. |



Mon rôle et ma mission :

Durant ce stage, j'ai occupé le poste de **Développeur Web** au sein d'une petite équipe de développement.

Ma mission lors de ce stage a été de réaliser un certain nombre de tâches visant à développer des fonctionnalités sur un site web de e-commerce from scratch (en partant de zéro) pour l'entreprise Oslab.

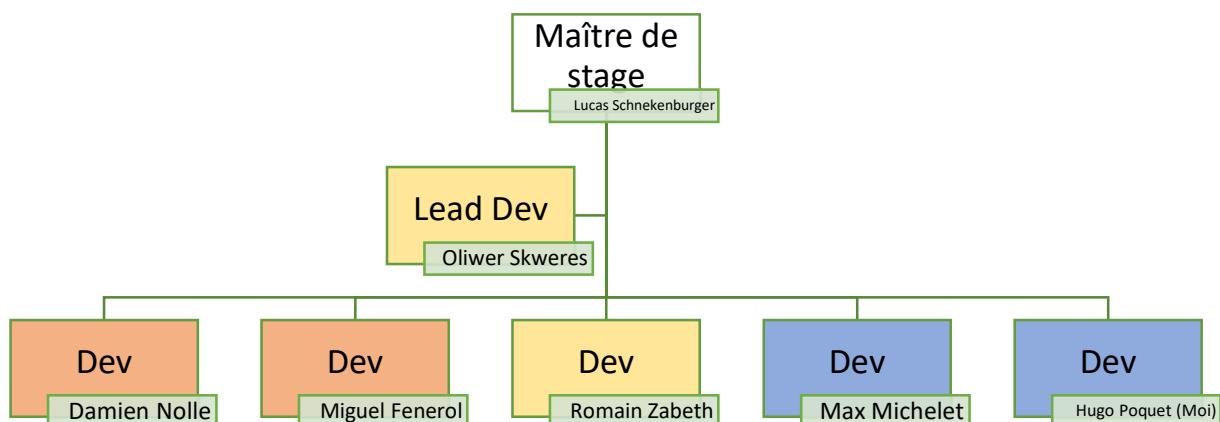
J'ai donc fait en grande partie du code, mais j'ai aussi pu participer aux différentes mises en commun des parties du projet, ainsi qu'à des réunions audios quotidiennes avec notre tuteur de façon à assurer un suivi de nos tâches et nous venir en aide facilement.

J'ai apprécié le travail en petit groupe et l'entre-aide qui en découlait.



Pour la réalisation de notre projet, nous avons suivi l'organisation suivante :

Organigramme du projet :



GitHub et Trello

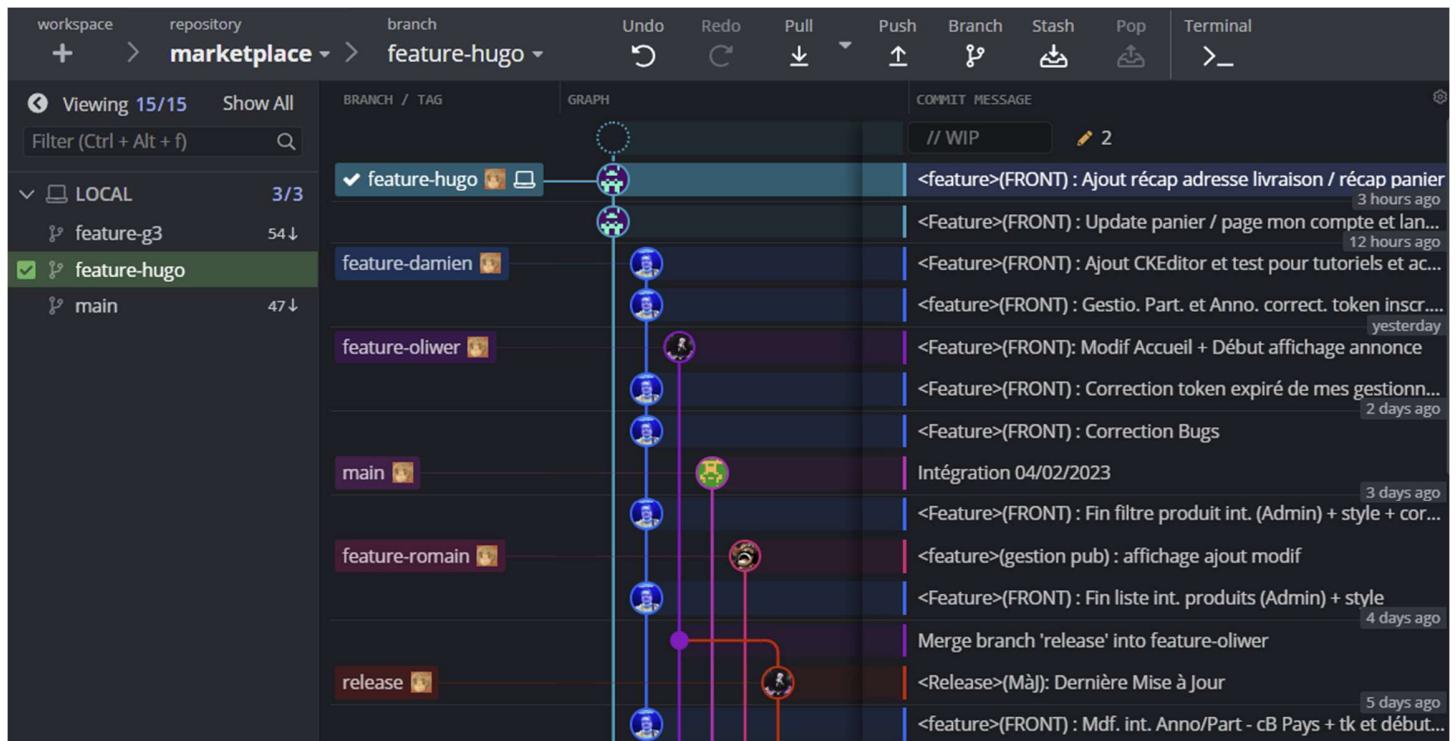
GitHub :

Tout d'abord, GitKraken est un logiciel de versioning lié à GitHub grâce auquel nous avons pu faire toute la mise en commun de notre projet.

Nous avons organisé notre Git avec :



- Une branche « main » : qui contient la version de production de l'application
- Une branche « feature » : par développeur de façon à coder dans sa branche
- Une branche « feature » : par groupe (3 en tout) pour d'abord mettre en commun entre binôme
- Une branche « release » : qui permet de mettre en commun les trois features de groupe



The screenshot shows the GitKraken interface with the following details:

- Workspace:** marketplace
- Repository:** marketplace
- Branch:** feature-hugo
- Toolbar:** Undo, Redo, Pull, Push, Branch, Stash, Pop, Terminal
- Left Sidebar:**
 - Viewing 15/15 Show All
 - Filter (Ctrl + Alt + f)
 - LOCAL (3/3)
 - feature-g3
 - feature-hugo** (selected)
 - main
- Graph View:** Displays the commit history and branches. The main branch is 'main'. Other branches include 'feature-hugo', 'feature-damien', 'feature-oliwer', 'feature-romain', and 'release'. Commits are shown with their messages and timestamps.
- Commit List:**
 - // WIP (2)
 - <feature>(FRONT) : Ajout récap adresse livraison / récap panier (3 hours ago)
 - <Feature>(FRONT) : Update panier / page mon compte et lan... (12 hours ago)
 - <Feature>(FRONT) : Ajout CKEditor et test pour tutoriels et ac... (yesterday)
 - <feature>(FRONT) : Gestio. Part. et Anno. correct. token inscr... (yesterday)
 - <Feature>(FRONT): Modif Accueil + Début affichage annonce (yesterday)
 - <Feature>(FRONT) : Correction token expiré de mes gestionn... (2 days ago)
 - <Feature>(FRONT) : Correction Bugs (2 days ago)
 - Intégration 04/02/2023 (3 days ago)
 - <Feature>(FRONT) : Fin filtre produit int. (Admin) + style + cor... (3 days ago)
 - <feature>(gestion pub) : affichage ajout modif (4 days ago)
 - <Feature>(FRONT) : Fin liste int. produits (Admin) + style (4 days ago)
 - Merge branch 'release' into feature-oliwer (4 days ago)
 - <Release>(MàJ) : Dernière Mise à Jour (5 days ago)
 - <feature>(FRONT) : Mdf. int. Anno/Part - cB Pays + tk et début... (5 days ago)

Trello :

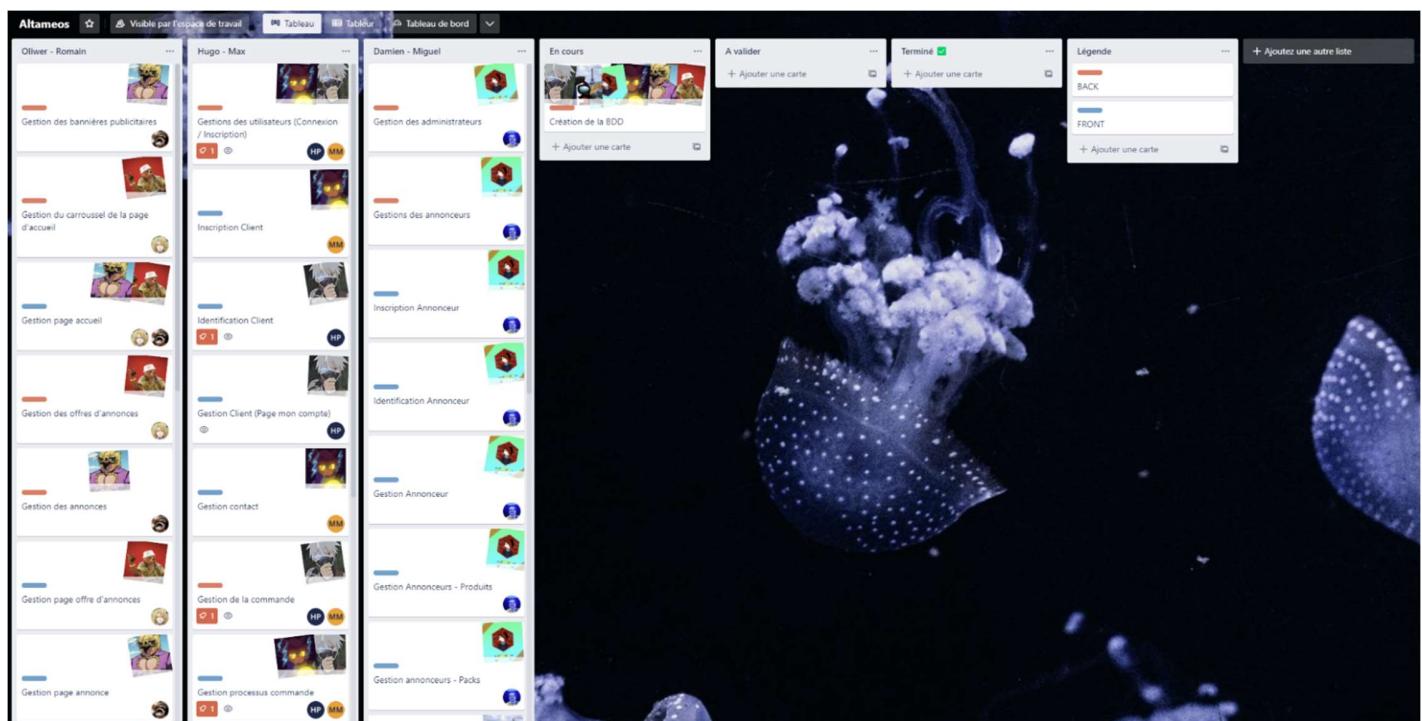
Nous avons ensuite utilisé le site Trello qui nous a servi de Management visuel.

Dans ce Trello, nous avons d'abord découpé toutes les tâches du projet pour se les répartir équitablement.



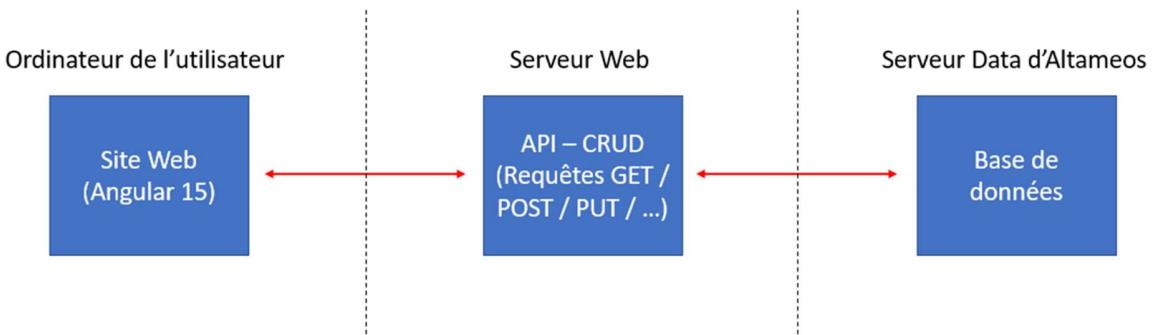
Comme vous pouvez le voir ci-dessous, nous avons organisé notre Trello avec les colonnes suivantes :

- Une colonne par groupe contenant les tâches des membres qui le compose.
- Une colonne « En cours » contenant les tâches en cours de développement.
- Une colonne « À valider » contenant les tâches terminées attendant d'être validées par notre maître de stage.
- Une colonne « Terminé » contenant les tâches terminées et validées par le tuteur.
- Une colonne « Légende » qui définit le code couleur adopté pour les tâches. Nous avons ici choisi de séparer le **front end** et le **back end**.



Présentation du projet

L'architecture du fonctionnement de notre application web suit ce schéma :



Tout d'abord, sur l'ordinateur de l'utilisateur, on retrouve le code html de tous les composants du site, les informations en stockage local, ainsi que les cookies.



Ensuite, le serveur Data accueille la base de données contenant toutes les informations venant alimenter le site en contenu. On y retrouve les données utilisateurs, les différents produits, etc... Cette base de données n'est pas accessible directement depuis l'application web.

C'est pourquoi, nous avons un serveur web faisant tourner une API – CRUD nous permettant d'envoyer des requêtes http GET / POST / PUT / DELETE / PATCH / REQUEST. Nous verrons que certaines informations, comme les mots de passe, voyagent cryptées pour garantir leur confidentialité.



Phase de formation :

Durant la première semaine, nous nous sommes formés au framework Angular afin de commencer le projet en ayant acquis quelques bases sur cette nouvelle technologie.

Je me suis donc formé à partir de la documentation Angular ainsi que de nombreux tutoriels sur Internet, tout en alimentant un petit projet de test de façon à mettre ce que j'apprenais en pratique. Cette phase m'a beaucoup aidé et je suis certain qu'elle a été déterminante pour assurer un meilleur départ sur le projet.



Le framework Angular :

Angular est un framework JavaScript open-source pour le développement d'applications web. Il est basé sur le modèle-vue-contrôleur (MVC) et utilise TypeScript. Il offre une structure pour les applications complexes et des outils pour faciliter le développement. Il permet également la création d'applications à la fois côté client et côté serveur avec Angular Universal.

Dans le cadre de notre projet, nous avons utilisé la version 15 d'Angular.



[Lien vers la documentation](#)

Angular Products

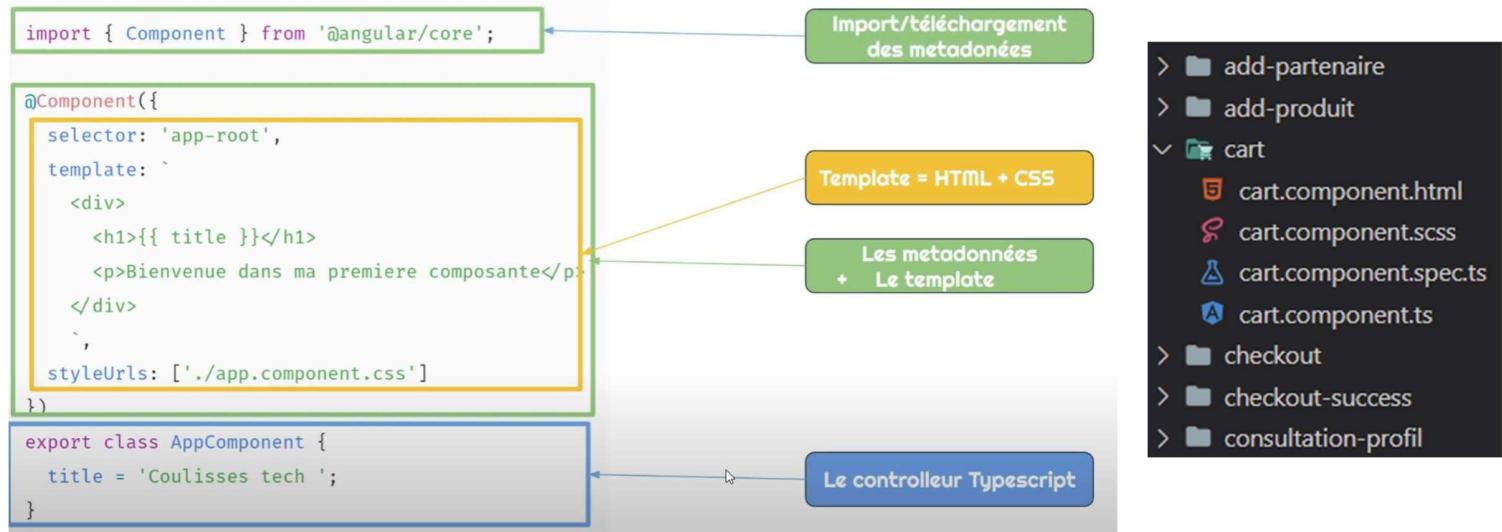


A grid of 18 icons representing various Angular products and tools, arranged in two rows of nine. Each icon is a hexagon with a white symbol inside. The first row includes: Angular (A), Protractor (a protractor), Forms (a form), PWA (a smartphone with a checkmark), Augury (a crescent moon), Language Services (a checkmark inside a circle), Router (a map), Elements (a hexagon), and CDK (a grid). The second row includes: Universal (a document), Karma (a K), Labs (a flask), Compiler (a gear), i18n (a globe), Http (two parallel lines with colons), Material (a document with arrows), Animations (a cube), and CLI (a terminal window).

| | | | | | | | | |
|-----------|------------|-------|----------|--------|-------------------|----------|------------|-----|
| Angular | Protractor | Forms | PWA | Augury | Language Services | Router | Elements | CDK |
| Universal | Karma | Labs | Compiler | i18n | Http | Material | Animations | CLI |

Angular fonctionne avec des composants, représentant les différentes fonctionnalités d'une application.

Dans un composant, on retrouve son interface en HTML, son contrôleur en TypeScript ainsi que son style en CSS ou SCSS dans notre cas :



Il y a ensuite les services, qui sont des composants sans interface, que l'on utilise à la manière d'une classe :

```

4  @Injectable({
5    providedIn: 'root'
6  })
7  export class AnnoncesService {
8
9    constructor(private http:HttpClient) { }
10
11   baseUrl: string = 'http://localhost/api/';
12
13   getAnnonce(id:any) {
14     return this.http.get(this.baseUrl + 'get-produit-annonce.php?idProduit=' + id)
15   }
16
17   getAnnonces() {
18     return this.http.get(this.baseUrl + 'get-annonces.php')
19   }
20

```

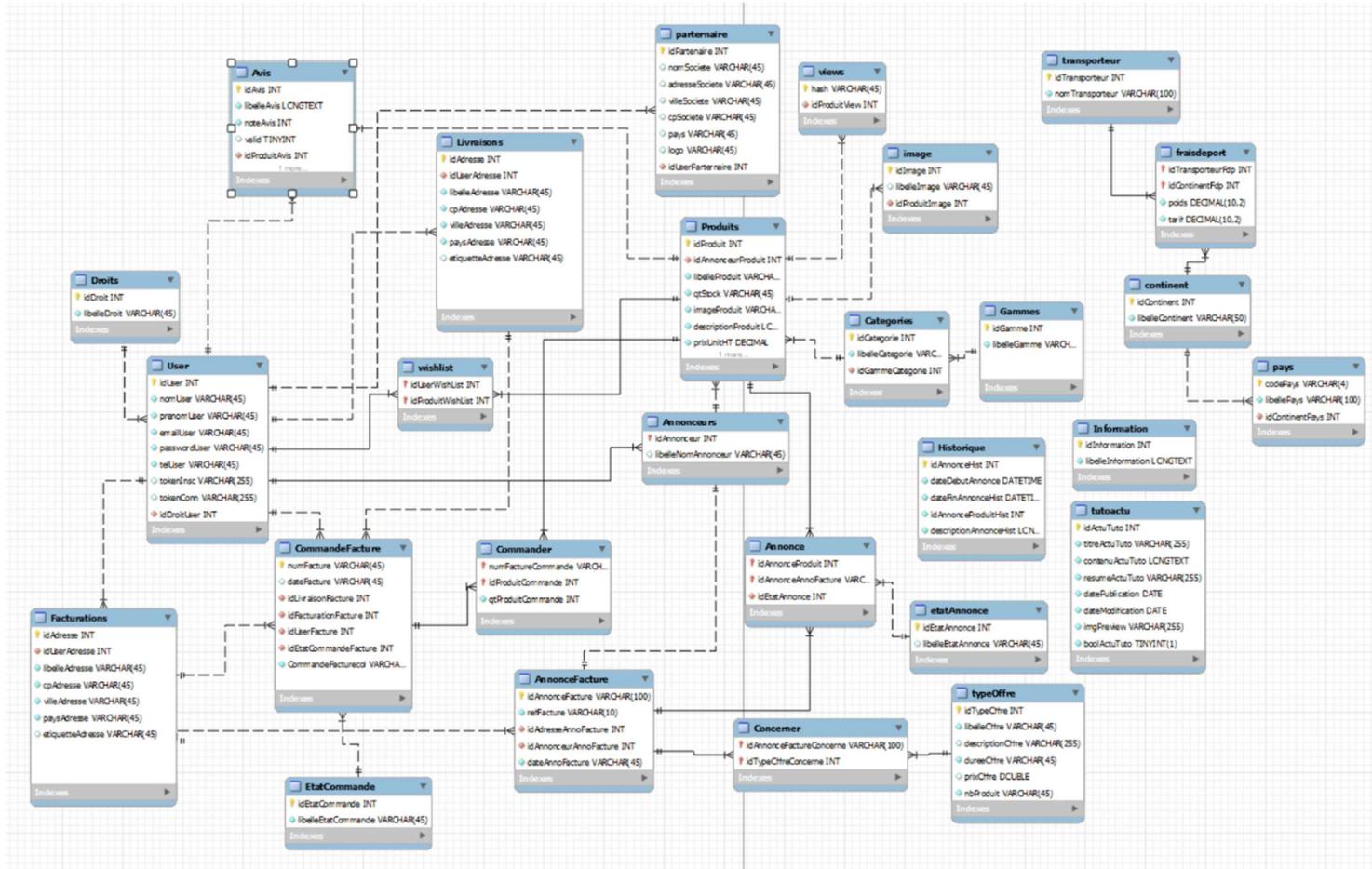
Pour ensuite naviguer entre les composants, entre nos pages, on utilise du routing avec un module comme ci-dessous :

```
52  const routes: Routes = [
53    { path: '', component: HomeComponent },
54    { path: 'login', component: LoginComponent },
55
56    { path: 'register', component: RegisterComponent },
57    { path: 'contact', component: ContactComponent },
58    { path: 'home', component: HomeComponent },
59    { path: 'verify', component: VerifyComponent },
```

On appelle donc ici un composant en fonction d'une route définie dans l'url.

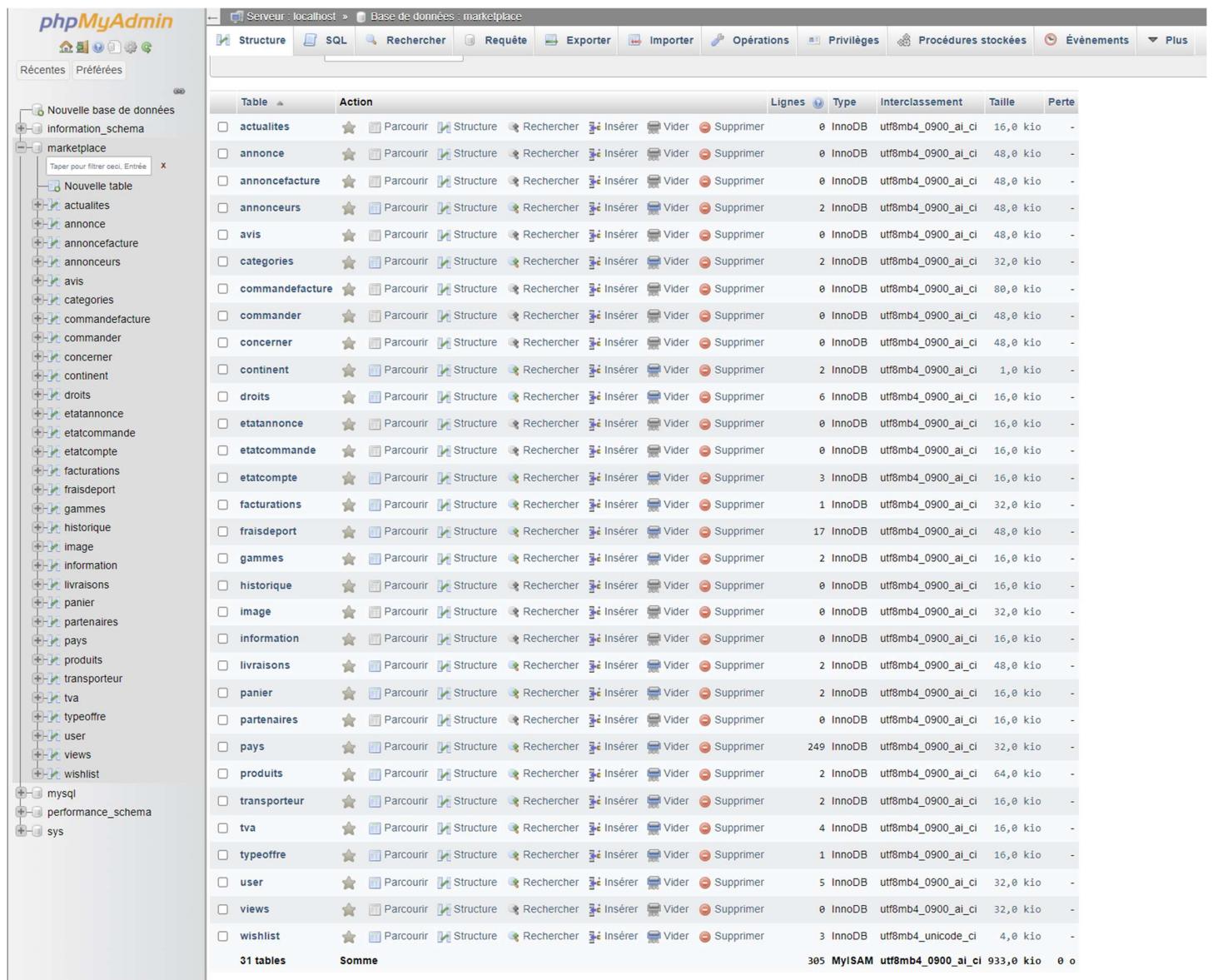
Mission 1 : Création de la base de données :

Pour débuter la création de cette application web, nous avons procédé à la réalisation d'un Schéma Relationnel de base de données sur MySQL Workbench :



Ce schéma a bien-sûr évolué tout au long du stage mais il a cependant été une base solide sur laquelle se reposer pour lancer le projet, puisqu'il a selon moi facilité la compréhension du sujet et de ce qui était attendu par le maître de stage.

Nous avons ensuite été amenés à reproduire cette base de données sur WampServer qui nous a servi de serveur de base de données.



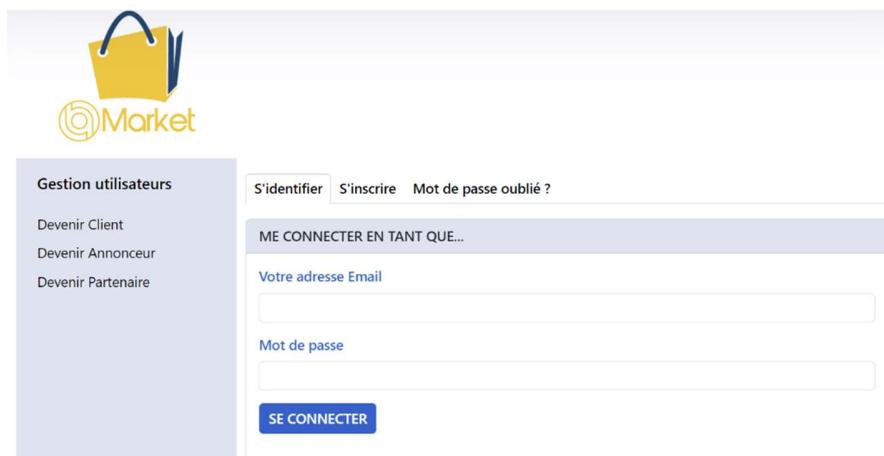
| Table | Action | Lignes | Type | Interclassement | Taille | Perte |
|-----------------|--|--------|--------|--------------------|-----------|-------|
| actualites | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| annonce | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 48,0 kio | - |
| annoncefacture | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 48,0 kio | - |
| annonceurs | Parcourir Structure Rechercher Insérer Vider Supprimer | 2 | InnoDB | utf8mb4_0900_ai_ci | 48,0 kio | - |
| avis | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 48,0 kio | - |
| categories | Parcourir Structure Rechercher Insérer Vider Supprimer | 2 | InnoDB | utf8mb4_0900_ai_ci | 32,0 kio | - |
| commandefacture | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 80,0 kio | - |
| commander | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 48,0 kio | - |
| concerner | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 48,0 kio | - |
| continent | Parcourir Structure Rechercher Insérer Vider Supprimer | 2 | InnoDB | utf8mb4_0900_ai_ci | 1,0 kio | - |
| droits | Parcourir Structure Rechercher Insérer Vider Supprimer | 6 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| etatannonce | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| etatcommande | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| etatcompte | Parcourir Structure Rechercher Insérer Vider Supprimer | 3 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| facturations | Parcourir Structure Rechercher Insérer Vider Supprimer | 1 | InnoDB | utf8mb4_0900_ai_ci | 32,0 kio | - |
| fraisdeport | Parcourir Structure Rechercher Insérer Vider Supprimer | 17 | InnoDB | utf8mb4_0900_ai_ci | 48,0 kio | - |
| gammes | Parcourir Structure Rechercher Insérer Vider Supprimer | 2 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| historique | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| image | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 32,0 kio | - |
| information | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| livraisons | Parcourir Structure Rechercher Insérer Vider Supprimer | 2 | InnoDB | utf8mb4_0900_ai_ci | 48,0 kio | - |
| panier | Parcourir Structure Rechercher Insérer Vider Supprimer | 2 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| partenaires | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| pays | Parcourir Structure Rechercher Insérer Vider Supprimer | 249 | InnoDB | utf8mb4_0900_ai_ci | 32,0 kio | - |
| produits | Parcourir Structure Rechercher Insérer Vider Supprimer | 2 | InnoDB | utf8mb4_0900_ai_ci | 64,0 kio | - |
| transporteur | Parcourir Structure Rechercher Insérer Vider Supprimer | 2 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| tva | Parcourir Structure Rechercher Insérer Vider Supprimer | 4 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| typeoffre | Parcourir Structure Rechercher Insérer Vider Supprimer | 1 | InnoDB | utf8mb4_0900_ai_ci | 16,0 kio | - |
| user | Parcourir Structure Rechercher Insérer Vider Supprimer | 5 | InnoDB | utf8mb4_0900_ai_ci | 32,0 kio | - |
| views | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_0900_ai_ci | 32,0 kio | - |
| wishlist | Parcourir Structure Rechercher Insérer Vider Supprimer | 3 | InnoDB | utf8mb4_unicode_ci | 4,0 kio | - |
| 31 tables | | Somme | | | 933,0 kio | 0 o |

Notre tuteur nous avait fourni un dictionnaire de données de façon à avoir une idée plus précise du contenu de cette base de données, ce qui nous a beaucoup aidé pour la réaliser.

Mission 2 : Création d'une page de connexion :

Ma première tâche pour ce stage a été de réaliser un formulaire de connexion afin de permettre l'authentification d'utilisateurs enregistrés en base de données. J'ai donc commencé par réaliser le front-end (la partie interface utilisateur) à l'aide de la bibliothèque Angular Materials, qui permet d'importer facilement des modules tels que des formulaires ou des tableaux par exemple.

Angular Materials nous a permis de tous utiliser le même format d'interface sur nos tâches pour que l'application entière soit développée avec la même technologie.



Après avoir réalisé le front-end (côté client), j'ai dû m'occuper du back-end (côté serveur / API). Durant cette phase, nous avons utilisé le CRUD, qui est une API php permettant de manipuler une base de données et récupérer les données sous Angular.

J'ai donc réalisé plusieurs requêtes sql côté serveur pour procéder aux vérifications nécessaires à la connexion. Ces requêtes sont appelées par l'interface via des requêtes http grâce auxquelles on peut utiliser les méthodes GET, POST, PUT, DELETE, PATCH ou encore REQUEST.

```

22 try {
23     $sql = "SELECT idUser, nomUser, prenomUser, emailUser, passwordUser, telUser, tokenInsc, tokenConn, libelleDroit
24         FROM user U
25             INNER JOIN droits D ON D.idDroit = U.idDroitUser
26             WHERE emailUser = '" . $_GET['email'] . "' AND passwordUser = '" . $_GET['password'] . "' AND tokenInsc = '0'";
27     $stmt = $conn->prepare($sql);
28     $stmt->execute();
29     if($stmt->rowCount() > 0) {
30         $data = null;
31         $data = $stmt->fetch(PDO::FETCH_ASSOC);
32
33         echo json_encode([
34             'success' => 1,
35             'data' => $data
36         ]);

```

Suite à cela, j'ai mis en place un token d'authentification qui se génère côté client grâce au module Firebase. J'ai ensuite dû mettre en place des vérifications liées à ce token telles que sa date d'expiration.

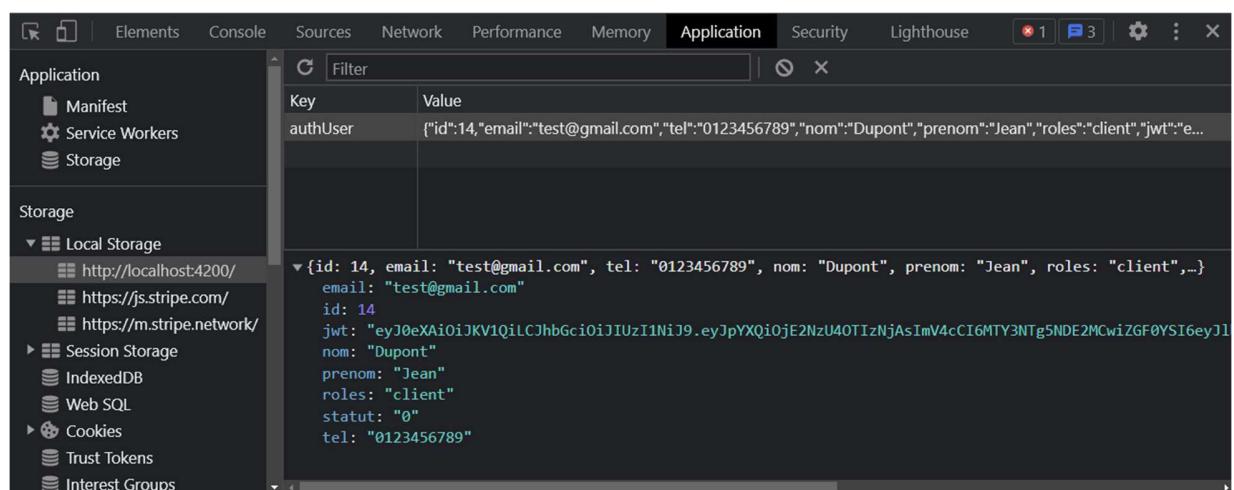
```

21 // Encodage d'un token JWT
22 $secretKey = "-----";
23 $issuedAt = time();
24 // $expirationTime = $issuedAt + (60); // 1 minute
25 $expirationTime = $issuedAt + (60 * 30); // 30 minutes
26 // $expirationTime = $issuedAt + (60 * 60); // 1 heure
27 // $expirationTime = $issuedAt + (60 * 60 * 24); // 1 jour
28 // $expirationTime = $issuedAt + (60 * 60 * 24 * 7); // 1 semaine
29 // $expirationTime = $issuedAt + (60 * 60 * 24 * 7 * 4); // 1 mois
30
31 $payload = array(
32     "iat" => $issuedAt,
33     "exp" => $expirationTime,
34     "data" => array(
35         "email" => $_GET['email'],
36     )
37 );
38
39 // Token JWT
40 $jwt = JWT::encode($payload, $secretKey, 'HS256');
41

```

J'enregistre ensuite mon token crypté en base de données pour venir vérifier que le token que j'ai envoyé côté client n'a pas été modifié via une deuxième requête http.

Une fois toutes les vérifications effectuées, je stocke temporairement les données utilisateur dans le stockage local du navigateur du client, et ce jusqu'à ce que le token expire, donnant lieu à la déconnexion de l'utilisateur.



| Key | Value |
|----------|--|
| authUser | {"id":14,"email":"test@gmail.com","tel":"0123456789","nom":"Dupont","prenom":"Jean","roles":"client","jwt":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NzU4OTIxNjAsImV4cCI6MTY3NTg5NDE2MCwiZGF0YSI6eyJl..."} <pre>{ id: 14 email: "test@gmail.com" tel: "0123456789" nom: "Dupont" prenom: "Jean" roles: "client" statut: "0" jwt: "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NzU4OTIxNjAsImV4cCI6MTY3NTg5NDE2MCwiZGF0YSI6eyJl..." }</pre> |

Ces informations sont enregistrées dans le stockage local de l'utilisateur depuis un service d'authentification en TypeScript.

```
//Vérification token valide ou non
if (validation == true) {
    // Ajout des informations de l'utilisateur dans le localStorage
    localStorage.setItem(
        'authUser',
        JSON.stringify({
            id: appUser.idUser,
            email: appUser.emailUser,
            tel: appUser.telUser,
            nom: appUser.nomUser,
            prenom: appUser.prenomUser,
            roles: appUser.libelleDroit,
            jwt: token,
            statut: appUser.tokenInsc,
        })
    );
    window.location.reload();
} else {
    // Token invalide
    alert('Token invalide, veuillez vous reconnecter');

    this.logout().subscribe({
        next: () => {
            window.location.reload();
        },
    });
}
```

On peut voir ici que, le token étant créé et validé, on initialise l'item « authUser » dans le stockage local. Ce token est donc, comme vu précédemment, crypté et possède des informations de l'utilisateur ainsi qu'une date d'expiration.

Cette date d'expiration est vérifiée à chaque action sur le site comme un changement de page par exemple. Cette vérification est essentielle car elle permet de garantir une certaine sécurité en demandant à l'utilisateur de se reconnecter régulièrement. Cela permet d'assurer de façon plus fiable l'identité de l'utilisateur et garantir la confidentialité de ses données.

```
// Vérification du token
// Vérification du token enregistré dans la base de données
this.http
  .get(this.baseUrl + 'tokenCheck.php?email=' + user.email)
  .subscribe((data: any) => {
    // Vérification de l'expiration du token
    this.http
      .get(this.baseUrl + 'tokenExpirationCheck.php?token=' + token)
      .subscribe((data2: any) => {
        if (data.success == 1) {
          // Token enregistré dans la base de données
          let tokenDb = data.data.tokenConn;

          // Décryptage du token
          tokenDb = atob(tokenDb);

          // Comparaison token en bdd et token généré
          if (tokenDb == token) {
            // Si le token n'a pas expiré et qu'il n'y a pas d'erreur
            if (data2.error != 1 && data2.hasExpired != 1) {
              // Token valide
              validation = true;
            } else {
              // Token invalide
              validation = false;
            }
          } else {
            // Token invalide
            validation = false;
          }
        } else {
          // Token invalide
          validation = false;
        }
      })
    // Vérification token valide ou non
    if (validation == true) {
      // Tout est ok on peut continuer
    } else {
      // Token invalide
      //alert('Token invalide, veuillez vous reconnecter');

      this.logout().subscribe({
        next: () => {
          this.router.navigateByUrl('/login');
        },
      });
      console.log('Token invalide, veuillez vous reconnecter');
      window.location.reload();
    }
  });
});
```

De plus, pour vérifier les données entrées par l'utilisateur, je procède au cryptage du mot de passe pour le comparer avec celui en base de données une fois haché.

Mission 3 : Crédation d'une page « Mon compte » :

Une fois l'authentification d'un utilisateur réalisée, j'ai créé une page de consultation de son profil dans laquelle l'utilisateur peut retrouver les informations qu'il a entré lors de son inscription, ces informations pouvant être modifiées.

The screenshot shows a user profile page for a marketplace client. The top navigation bar includes links for ACCUEIL, OSMARKET, ANNONCEUR, PARTENAIRE, CONTACT, and WISHLIST. A 'CONNECTÉ(E) EN TANT QUE MARKETPLACE CLIENT' message is displayed. The main content area is titled 'Profil de Marketplace Client'. Below it, there are tabs for 'PROFIL', 'MODIFIER : NOM / PRÉNOM / TÉLÉPHONE', 'GESTION DU MOT DE PASSE', 'INFORMATIONS DE LIVRAISON', and 'INFOI >'. The 'PROFIL' tab is selected. The profile information table contains the following data:

| | |
|--------------------------|------------------|
| Nom : | Client |
| Prénom : | Marketplace |
| Mail : | client@gmail.com |
| Numéro de téléphone : | 443534234 |
| Rôle sur le site : | client |
| Statut du compte : | Vérifié |
| Solde de points crédit : | 0.00 |

A shopping cart icon with a '10' notification is visible on the right side of the profile table.

The screenshot shows a password change form for a user named Jean Dupont. The top navigation bar includes links for Profil, Modifier : nom / prénom / téléphone, Gestion du mot de passe, Informations de livraison, and Historique des transactions. The 'Modifier : nom / prénom / téléphone' tab is selected. The form fields are as follows:

- Entrez un nom: Dupont
- Entrez un prénom: Jean
- Entrez un numéro de téléphone: 0123456789
- Enregistrer button

Il a aussi dans cette page la possibilité de changer son mot de passe.

The screenshot shows a password change form for a user named Jean Dupont. The top navigation bar includes links for Profil, Modifier : nom / prénom / téléphone, Gestion du mot de passe, Informations de livraison, and Historique des transactions. The 'Gestion du mot de passe' tab is selected. The form fields are as follows:

- Nouveau mot de passe : (input field)
- Confirmez le nouveau mot de passe : (input field)
- Enregistrer button

Il est aussi possible, sur l'onglet « Informations de livraison », de venir ajouter ou modifier les informations relatives à son adresse de livraison, informations qui peuvent être utilisées lors de la validation d'une commande.

Profil de Jean Dupont

Profil Modifier : nom / prénom / téléphone Gestion du mot de passe **Informations de livraison** Historique des transactions

- Adresse : 1 rue de test
- Code postal : 75000
- Ville : Paris
- Pays : France
- Etiquette adresse :

Ajouter ou modifier une adresse de livraison

Gestion de l'adresse de livraison

Nouvelle adresse :
1 rue de test

Nouveau code postal :
75000

Nouvelle ville :
Paris

Nouveau pays :

Nouvelle étiquette :

Les formulaires comme celui-ci-dessus ont été créés avec l'aide d'Angular Material. C'était d'une part relativement simple à intégrer, mais cela nous a aussi permis de tous utiliser la même mise en forme sur nos pages.

Voici un morceau du code html de cette page, on peut voir qu'un formulaire est déclaré et qu'il exécute un évènement lors de sa soumission.

```

1  <mat-tab-group>
2    <mat-tab label="Gestion de l'adresse de livraison">
3      <form [formGroup]="addressFormGroup" (ngSubmit)="updateAddress()">
4        <div class="mb-3">
5          <label class="form-label">Nouvelle adresse :</label>
6          <input type="text" formControlName="newAdresse" class="form-control" />
7        </div>
8
9        <div class="mb-3">
10         <label class="form-label">Nouveau code postal :</label>
11         <input
12           type="text"
13           formControlName="newCodePostal"
14           class="form-control"
15           />
16       </div>

```

Cet évènement est géré par le fichier TypeScript du même composant sous la forme d'une fonction.

```

37  updateAddress() {
38    this.http
39      .get(
40        this.baseUrl +
41          'updateDeliveryAddress.php?idUser=' +
42            this.authService.authenticatedUser!.id +
43            '&libelleAdresse=' +
44            this.addressFormGroup.value.newAdresse +
45            '&cpAdresse=' +
46            this.addressFormGroup.value.newCodePostal +
47            '&villeAdresse=' +
48            this.addressFormGroup.value.newVille +
49            '&codePaysAdresse=' +
50            this.addressFormGroup.value.newPays +
51            '&etiquetteAdresse=' +
52            this.addressFormGroup.value.newEtiquette
53      )
54      .subscribe((data: any) => {
55        if (data.success == 1) {
56          alert('Adresse de livraison mise à jour');
57        } else {
58          alert("Erreur lors de la mise à jour de l'adresse de livraison");
59        }
60      });
61  }
62 }
```

On peut voir ici qu'une requête http est exécutée avec comme paramètres, l'identifiant utilisateur, l'adresse, la ville, le code du pays, ainsi qu'une étiquette. Ces informations sont envoyées dans une méthode GET qui s'exécute dans l'API CRUD.

```

31  $sql = "UPDATE livraisons
32  SET libelleAdresse = '" . $_GET['libelleAdresse'] . "' ,
33  cpAdresse = '" . $_GET['cpAdresse'] . "' ,
34  villeAdresse = '" . $_GET['villeAdresse'] . "' ,
35  codePaysAdresse = '" . $_GET['codePaysAdresse'] . "' ,
36  etiquetteAdresse = '" . $_GET['etiquetteAdresse'] . "' ,
37  WHERE idUserAdresse = '" . $_GET['idUser'] . "' ;
38
39  $stmt = $conn->prepare($sql);
40  $stmt->execute();
```

Je précise qu'il est possible de modifier ses informations de facturation de la même manière dans l'onglet « Information de facturation », ces deux adresses pouvant être différentes.

Mission 4 : Crédation du panier :

La tâche suivante a consisté à réaliser une page contenant le panier de l'utilisateur.

J'ai donc modifié la page « Liste des produits » créée par mon collègue Oliwer Skweres pour rendre le bouton « Ajouter au panier » fonctionnel. Je vérifie donc que mon produit n'est pas déjà dans mon panier en base de données pour enfin l'ajouter et l'afficher sur la page.

Dans le panier, j'ai la possibilité de modifier la quantité d'un produit, de le retirer du panier, ou simplement de passer une commande. Lorsque j'augmente la quantité, une vérification du stock se fait en arrière-plan.

Toutes les actions réalisées sur le panier sont envoyées côté serveur, sont vérifiées pour enfin être enregistrées en base de données, pour assurer un certain niveau de sécurité et assurer l'intégrité du panier du client, de sorte que s'il le modifie, tout soit pris en compte et enregistré.

| No. | Numéro Produit | Nom | Prix unitaire HT | TVA | Poids à l'unité (en g) | Quantité | Retirer du panier |
|-----|----------------|----------|------------------|---------|------------------------|---------------------------------|--------------------|
| 1 | 9 | Iphone X | 600.00 € | 20.00 % | 500.20 g | - 1 + | trash |
| 2 | 10 | Clavier | 60.00 € | 5.50 % | 700.00 g | - 2 + | trash |

Sous-total HT : 720 €
Poids total : 1.9 kg
Sous-total TTC : 846.6 €

Passer la commande

Les fonctionnalités dynamiques sur cette page sont connectées en temps réel à l'API CRUD. C'est-à-dire que toute action, comme augmenter ou diminuer la quantité et retirer l'article du panier, sont enregistrées en base de données de façon à sauvegarder automatiquement toute action de l'utilisateur.

En effet, j'ai créé une table panier qui contient toutes les informations d'un panier client. Cette table sert donc à charger le contenu d'un panier au chargement de la page mais aussi à sauvegarder les différentes modifications de l'utilisateur.

| | # | Nom | Type | Interclassement | Attributs | Null | Valeur par défaut | Commentaires | Extra | Action |
|---|---|------------------------|------|-----------------|-----------|------|-------------------|--------------|---|--------|
| □ | 1 | idUser | int | | | Non | Aucun(e) | |  Modifier  Supprimer  Plus | |
| □ | 2 | idProduitPanier | int | | | Non | Aucun(e) | |  Modifier  Supprimer  Plus | |
| □ | 3 | qtProduit | int | | | Non | Aucun(e) | |  Modifier  Supprimer  Plus | |

| idUser | idProduitPanier | qtProduit |
|---------------|------------------------|------------------|
| 14 | 9 | 1 |
| 1 | 9 | 1 |
| 14 | 10 | 1 |

```
try {
    $sql = "SELECT idUser, idProduitPanier
            FROM panier
            WHERE idUser = '" . $_GET['idUser'] . "' AND idProduitPanier = '" . $_GET['idProduct'] . "'";
    $stmt = $conn->prepare($sql);
    $stmt->execute();
    if($stmt->rowCount() > 0) {
        $sql = "UPDATE panier
                SET qtProduit = '" . $_GET['quantity'] . "'"
                WHERE idUser = '" . $_GET['idUser'] . "' AND idProduitPanier = '" . $_GET['idProduct'] . "'";
        $stmt = $conn->prepare($sql);
        $stmt->execute();

        echo json_encode([
            'success' => 1,
            'message' => "La quantité a été modifiée."
        ]);
    }
}
```

```
try {
    $sql = "SELECT idUser, idProduitPanier
            FROM panier
            WHERE idUser = '" . $_GET['idUser'] . "' AND idProduitPanier = '" . $_GET['idProduct'] . "'";
    $stmt = $conn->prepare($sql);
    $stmt->execute();
    if($stmt->rowCount() > 0) {
        $sql = "DELETE FROM panier
                WHERE idUser = '" . $_GET['idUser'] . "' AND idProduitPanier = '" . $_GET['idProduct'] . "'";
        $stmt = $conn->prepare($sql);
        $stmt->execute();

        echo json_encode([
            'success' => 1,
            'message' => "Le produit a été supprimé du panier."
        ]);
    }
}
```

De plus, j'ai fait en sorte que lorsque l'on augmente la quantité pour un produit, une vérification du stock se lance.

De cette façon, on plafonne la quantité pouvant être demandée à la quantité en stock.

En plus de cela, ce paramètre ne peut pas dépasser la valeur 99, pour éviter toute erreur potentielle.

```
add(element: PeriodicElement) {
  if (element.quantity >= 99) {
    alert('You have reached the maximum quantity');
  } else {
    this.http
      .get(this.baseUrl + 'stockQtCheck.php?idProduct=' + element.numProd)
      .subscribe((data: any) => {
        if (data.success == 1) {
          if (element.quantity + 1 > data.data.qtStock) {
            alert('Vous avez atteint la quantité maximale en stock');
          } else {
            element.quantity += 1;
            this.sousTotalHT = Math.round(this.sousTotalHTCalc() * 100) / 100;
            this.sousTotalTTC =
              Math.round(this.sousTotalTTCCalc() * 100) / 100;
            this.poidsTotal = Math.round(this.poidsTotalCalc() * 100) / 100;
            this.poidsTotal = this.poidsTotal! / 1000;
            this.poidsTotal = Math.round(this.poidsTotal * 100) / 100;
            console.log('add :', element.position);
            this.http
              .get(
                this.baseUrl +
                  'updateQuantityCart.php?idUser=' +
                  this.authService.authenticatedUser!.id +
                  '&idProduct=' +
                  element.numProd +
                  '&quantity=' +
                  element.quantity
              )
              .subscribe((data2: any) => {
                if (data2.success == 0) {
                  alert(data2.message);
                }
              });
            }
          }
        }
      });
}
```

Enfin, une vérification du stock par rapport à la quantité demandée pour chaque produit est effectuée au chargement de la page.

```
this.http
.get(
  this.baseUrl +
    'stockQtCheckCart.php?idUser=' +
    this.authService.authenticatedUser!.id
)
.subscribe((data: any) => {
  if (data.success == 1 && data.modif == 1) {
    alert(
      'La quantité de certains produits va être modifiée car le stock était insuffisant.'
    );
    window.location.reload();
  } else if (data.error == 1) {
    alert(data.message);
  } else if (data.success == 1 && data.modif == 0) {
    // Tout est ok
  }
});
```

En effet, une requête http est envoyée, et pour chaque produit, si sa quantité demandée dépasse sa quantité en stock, alors la quantité demandée prend la valeur maximum, c'est-à-dire la valeur du stock.

De plus, si le stock est à 0, alors on retire l'article du panier.

Cette vérification permet à l'utilisateur d'être informé d'une potentielle rupture de stock dès son arrivée dans le panier.

```
22 $success = 0;
23 $modif = 0;
24 $error = 0;
25
26 $sql = "SELECT idUser, idProduitPanier, qtProduit
27   FROM panier
28   WHERE idUser = '" . $_GET['idUser'] . "'";
29 $stmt = $conn->prepare($sql);
30 $stmt->execute();
31
32 if($stmt->rowCount() > 0) {
33   $data = null;
34   $data = $stmt->fetchAll(PDO::FETCH_ASSOC);
35
36   foreach($data as $row) {
37     $idProduit = $row['idProduitPanier'];
38     $qtProduitDemandee = $row['qtProduit'];
39
40     $sql = "SELECT qtStock
41       FROM produits
42       WHERE idProduit = '" . $idProduit . "'";
43     $stmt = $conn->prepare($sql);
44     $stmt->execute();
45
46     $qtStock = 0;
47     $qtStock = $stmt->fetch(PDO::FETCH_ASSOC)[ 'qtStock'];
```

```
if($stmt->rowCount() > 0) {
  if ($qtStock < $qtProduitDemandee) {
    if ($qtStock == 0){
      $sql = "DELETE FROM panier
              WHERE idProduitPanier = '" . $idProduit . "'"
              AND idUser = '" . $_GET['idUser'] . "'";
      $stmt = $conn->prepare($sql);
      $stmt->execute();

      if($stmt->rowCount() > 0) {
        $success = 1;
        $modif = 1;
      } else {
        $error = 1;
      }
    } else {
      $sql = "UPDATE panier
              SET qtProduit = " . $qtStock . "
              WHERE idProduitPanier = '" . $idProduit . "'"
              AND idUser = '" . $_GET['idUser'] . "'";
      $stmt = $conn->prepare($sql);
      $stmt->execute();

      if($stmt->rowCount() > 0) {
        $success = 1;
        $modif = 1;
      } else {
        $error = 1;
      }
    }
  }
}
```

Mission 5 : Page de récapitulatif commande :

Cette page est affichée lorsque l'utilisateur valide son panier en appuyant sur « Passer la commande » (comme montré ci-dessus). Elle présente à l'utilisateur toutes les informations nécessaires à la commande qu'il s'apprête à passer comme les informations concernant son adresse de livraison. Il peut la renseigner si ce n'est pas encore fait ou bien venir la modifier.

Ensuite, il a un aperçu détaillé du contenu de son panier pour y vérifier les articles, leurs prix, la quantité de produits, etc...

C'est alors qu'on affiche les frais de port pour cette commande en fonction du transporteur choisi.

Enfin, le total de la commande incluant les frais de port est affiché et on propose à l'utilisateur de passer au paiement.

The screenshot shows the checkout process on the altameos Market website. The top navigation bar includes links for ACCUEIL, PRODUITS, ANNONCEUR, PARTENAIRE, a search bar, and user account information (Jean Dupont, Se déconnecter).

Récapitulatif de l'adresse de livraison

- Adresse : 1 rue de test
- Code postal : 75000
- Ville : Paris
- Pays : France
- Etiquette adresse :

[Modifier](#)

Récapitulatif du panier

| No. | Nom | Prix unitaire HT | TVA | Poids à l'unité (en g) | Quantité |
|-----|----------|------------------|---------|------------------------|----------|
| 1 | Iphone X | 600.00 € | 20.00 % | 500.20 g | 4 |
| 2 | Clavier | 60.00 € | 5.50 % | 700.00 g | 3 |

[Modifier](#)

Frais de port
Veuillez choisir un transporteur :

DPD

Montant des frais de port : 10.00 €

Paiement
Sous-total HT : 2580 €
Poids total : 4.1 kg
Sous-total TTC : 3079.9 € (frais de port inclus)

[Passer au paiement](#)

Sur cette interface, le bouton « Passer au paiement » n'est disponible que si l'adresse de livraison est bien renseignée, si le panier n'est pas vide, et si le choix du transporteur a été effectué. Sous ces conditions, la procédure de paiement peut avoir lieu.

De plus, les informations concernant le montant à payer ne sont affichées que lorsqu'un transporteur a été choisi.

Pour ce qui est des calculs de frais de port, j'ai proposé à mon tuteur l'idée d'intégrer une api de transporteur tel qu'UPS de façon à faciliter cette tâche et nous permettre d'avoir accès à tout un tas d'autres services comme le suivi des colis par exemple. Cependant, sous certaines contraintes, j'ai dû créer une table « fraisdeport » et gérer mes propres tarifs en fonction d'un transporteur, d'une destination et d'un poids de colis.

Voici la table « fraisdeport » ainsi que quelques exemples de tarifs factices :

Le poids étant en kg et le prix en €.

Côté template, une fonction de calcul des frais de port est exécutée à chaque changement de la sélection d'un transporteur.

```

99   <mat-card-content>
100  <form [formGroup]="transporteurFormGroup">
101    <p>Veuillez choisir un transporteur :</p>
102    <mat-form-field class="input">
103      <mat-select
104        formControlName="newTransporteur"
105        class="form-control"
106        (selectionChange)="fraisDePortCalc()"
107      >
108        <mat-option
109          *ngFor="let unTransporteur of transporteur"
110          [value]="unTransporteur.idTransporteur"
111          >{{ unTransporteur.nomTransporteur }}</mat-option>
112      >
113    </mat-select>
114  </mat-form-field>
115 </form>
116 <p *ngIf="fraisDePortChoisi == true">
117   Montant des frais de port : {{ fraisDePort }} €
118 </p>
119 </mat-card-content>

184  fraisDePortCalc() {
185    this.http
186    .get(
187      this.baseUrl +
188        'calculFraisDePort.php?idTransporteur=' +
189        this.transporteurFormGroup.value.newTransporteur +
190        '&poidsTotal=' +
191        this.poidsTotal +
192        '&codePays=' +
193        this.codePays
194    )
195    .subscribe((data: any) => {
196      if (data.success == 1) {
197        this.fraisDePort = data.data.tarif;
198        this.sousTotalTTC = +this.sousTotalTTCCalc() + +this.fraisDePort;
199        this.fraisDePortChoisi = true;
200      } else {
201        this.fraisDePort = 0;
202        alert(data.message);
203        window.location.reload();
204      }
205    });
206  }
207}
208

```

On retrouve cette fonction dans le fichier TypeScript qui va donc chercher le tarif en base de données grâce à une requête http. Le sous-total est enfin calculé en fonction de ce tarif retourné.

C'est alors que nous allons pouvoir introduire la prochaine mission qui concerne la procédure de paiement.

Tout d'abord, après click du bouton « Passer au paiement », une nouvelle vérification du stock est lancée et les produits que l'utilisateur a dans son panier sont « réservés » dans le sens où la quantité qu'il demande est décrémentée du stock.

```
<button mat-raised-button color="primary" (click)="reservationProduits()">  
| Passer au paiement  
</button>
```

```
141 |   reservationProduits() {  
142 |     this.http  
143 |       .get(  
144 |         this.baseUrl +  
145 |           'stockQtCheck2.php?idUser=' +  
146 |             this.authService.authenticatedUser!.id  
147 |       )  
148 |         .subscribe((data: any) => {  
149 |           if (data.success == 1) {  
150 |             // Stock ok  
151 |             // On réserve les produits  
152 |             this.http  
153 |               .get(  
154 |                 this.baseUrl +  
155 |                   'reservationProduits.php?idUser=' +  
156 |                     this.authService.authenticatedUser!.id  
157 |               )  
158 |                 .subscribe((data2: any) => {  
159 |                   if (data2.success == 1) {  
160 |                     // Les produits ont été réservés  
161 |                     // On lance le timer  
162 |                     alert('Les produits ont été réservés');  
163 |                     this.timer();  
164 |                   } else {  
165 |                     alert('Erreur lors de la réservation des produits');  
166 |                     this.router.navigate(['/cart']);  
167 |                   }  
168 |                 });  
169 |               } else {  
170 |                 alert('Le stock est insuffisant pour un ou plusieurs produits');  
171 |                 this.router.navigate(['/cart']);  
172 |               }  
173 |             );  
174 |         }
```

On procède donc dans un premier temps à la deuxième vérification du stock de chaque produit dans le panier :

```
22 $erreurStock = false;
23
24 $sql = "SELECT idUser, idProduitPanier, qtProduit
25     |     FROM panier
26     |     WHERE idUser = '" . $_GET['idUser'] . "'";
27 $stmt = $conn->prepare($sql);
28 $stmt->execute();
29
30 if($stmt->rowCount() > 0) {
31     $data = null;
32     $data = $stmt->fetchAll(PDO::FETCH_ASSOC);
33
34     foreach($data as $row) {
35         $idProduit = $row['idProduitPanier'];
36         $qtProduitDemandee = $row['qtProduit'];
37
38         $sql = "SELECT qtStock
39             |     FROM produits
40             |     WHERE idProduit = '" . $idProduit . "'";
41         $stmt = $conn->prepare($sql);
42         $stmt->execute();
43
44         $qtStock = 0;
45         $qtStock = $stmt->fetch(PDO::FETCH_ASSOC)['qtStock'];
46
47         if($stmt->rowCount() > 0) {
48             if ($qtStock < $qtProduitDemandee) {
49                 $erreurStock = true;
50             } else {
51                 // Tout est ok
52             }
53         } else {
54             $erreurStock = true;
55         }
56     }
```

On procède ensuite à la réservation des produits pour l'utilisateur :

```
22 $erreurReserv = false;
23
24 $sql = "SELECT idUser, idProduitPanier, qtProduit
25     |     FROM panier
26     |     WHERE idUser = '" . $_GET['idUser'] . "'";
27 $stmt = $conn->prepare($sql);
28 $stmt->execute();
29
30 if($stmt->rowCount() > 0) {
31     $data = null;
32     $data = $stmt->fetchAll(PDO::FETCH_ASSOC);
33
34     foreach($data as $row) {
35         $idProduit = $row['idProduitPanier'];
36         $qtProduitDemandee = $row['qtProduit'];
37
38         $sql = "UPDATE produits
39             |             SET qtStock = qtStock - " . $qtProduitDemandee . "
40             |             WHERE idProduit = '" . $idProduit . "'";
41         $stmt = $conn->prepare($sql);
42         $stmt->execute();
43
44         if($stmt->rowCount() > 0) {
45             $erreurReserv = false;
46         } else {
47             $erreurReserv = true;
48         }
49     }
```

Mais cette réservation ne doit pas être définitive et, dans le cas où le client ne va pas au bout de la procédure de paiement, il faut pouvoir l'annuler. C'est pourquoi un compte à rebours de trois minutes est lancé dans un premier temps, laissant le temps à l'utilisateur de procéder à son paiement.

```

176  timer() {
177    setTimeout(() => {
178      // Faire ici une condition qui vérifie si le paiement a été effectué
179      // Si oui, on ne fait rien
180      // Si non, on remet en stock
181      // Pour vérifier si le paiement a été effectué, on peut vérifier si le panier est vide
182      // Etant donné que le panier sera vidé une fois le paiement effectué
183
184      alert('Session expirée, vous allez être redirigé vers le panier');
185      this.http
186        .get(
187          this.baseUrl +
188            'remiseStockProduits.php?idUser=' +
189            this.authService.authenticatedUser!.id
190        )
191        .subscribe((data: any) => {});
192      this.router.navigate(['/cart']);
193    }, 180000); // Redirection vers le panier après 3 minutes
194  }

```

Si le paiement n'a pas été effectué à la fin du compte à rebours, alors la procédure est annulée et les produits sont remis en stock.

```

22  $sql = "SELECT idUser, idProduitPanier, qtProduit
23    |   FROM panier
24    |   WHERE idUser = '" . $_GET['idUser'] . "'";
25  $stmt = $conn->prepare($sql);
26  $stmt->execute();
27
28 if($stmt->rowCount() > 0) {
29   $data = null;
30   $data = $stmt->fetchAll(PDO::FETCH_ASSOC);
31
32   foreach($data as $row) {
33     $idProduit = $row['idProduitPanier'];
34     $qtProduitDemandee = $row['qtProduit'];
35
36     $sql = "UPDATE produits
37           |   SET qtStock = qtStock + " . $qtProduitDemandee . "
38           |   WHERE idProduit = '" . $idProduit . "'";
39     $stmt = $conn->prepare($sql);
40     $stmt->execute();
41   }

```

Cette réservation est importante car, dans le cas où deux utilisateurs lancent la procédure de paiement quasi-simultanément, il faut pouvoir les prévenir que la quantité qu'ils demandent dépasse le stock. De cette façon, l'utilisateur n'a pas de mauvaise surprise une fois avoir payé car ses produits lui auront été réservés.

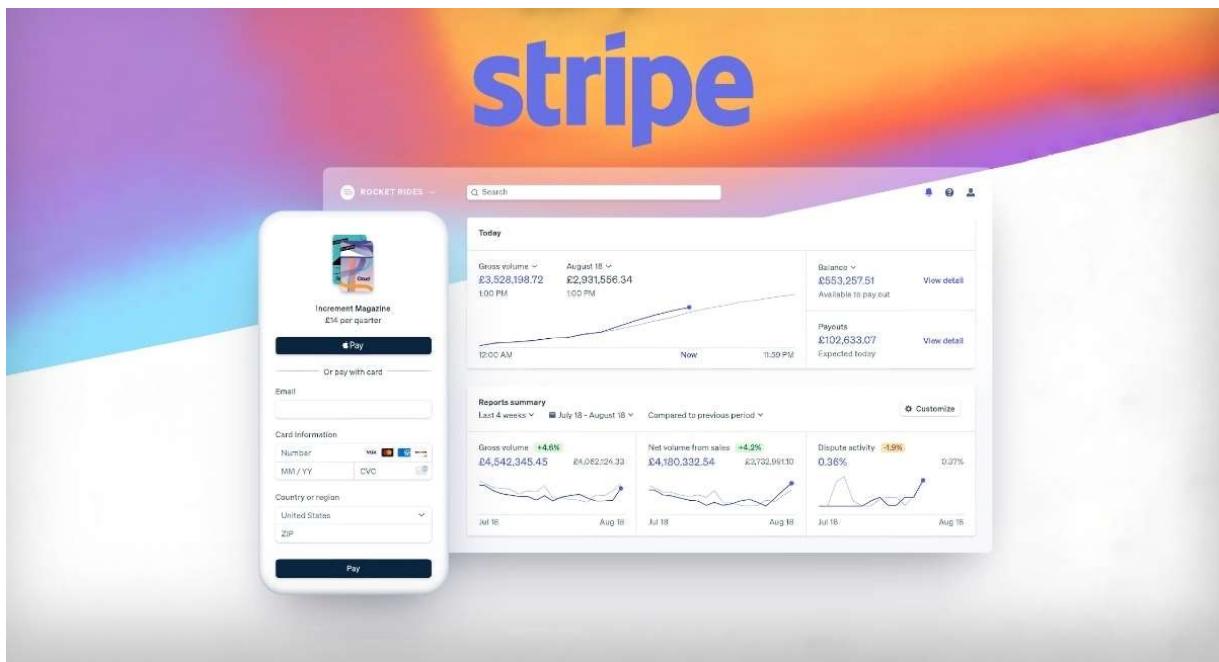
Mission 6 : Module de paiement :

Cette tâche est en cours de développement. Mais je vais toutefois l'introduire.

Nous allons donc intégrer un module de paiement externe à notre application en passant par le module Stripe.



Stripe est une infrastructure de paiement pour le commerce en ligne et propose un logiciel ainsi qu'une API à intégrer à une application web de façon à pouvoir accepter des paiements, effectuer des virements et gérer ses activités en ligne.

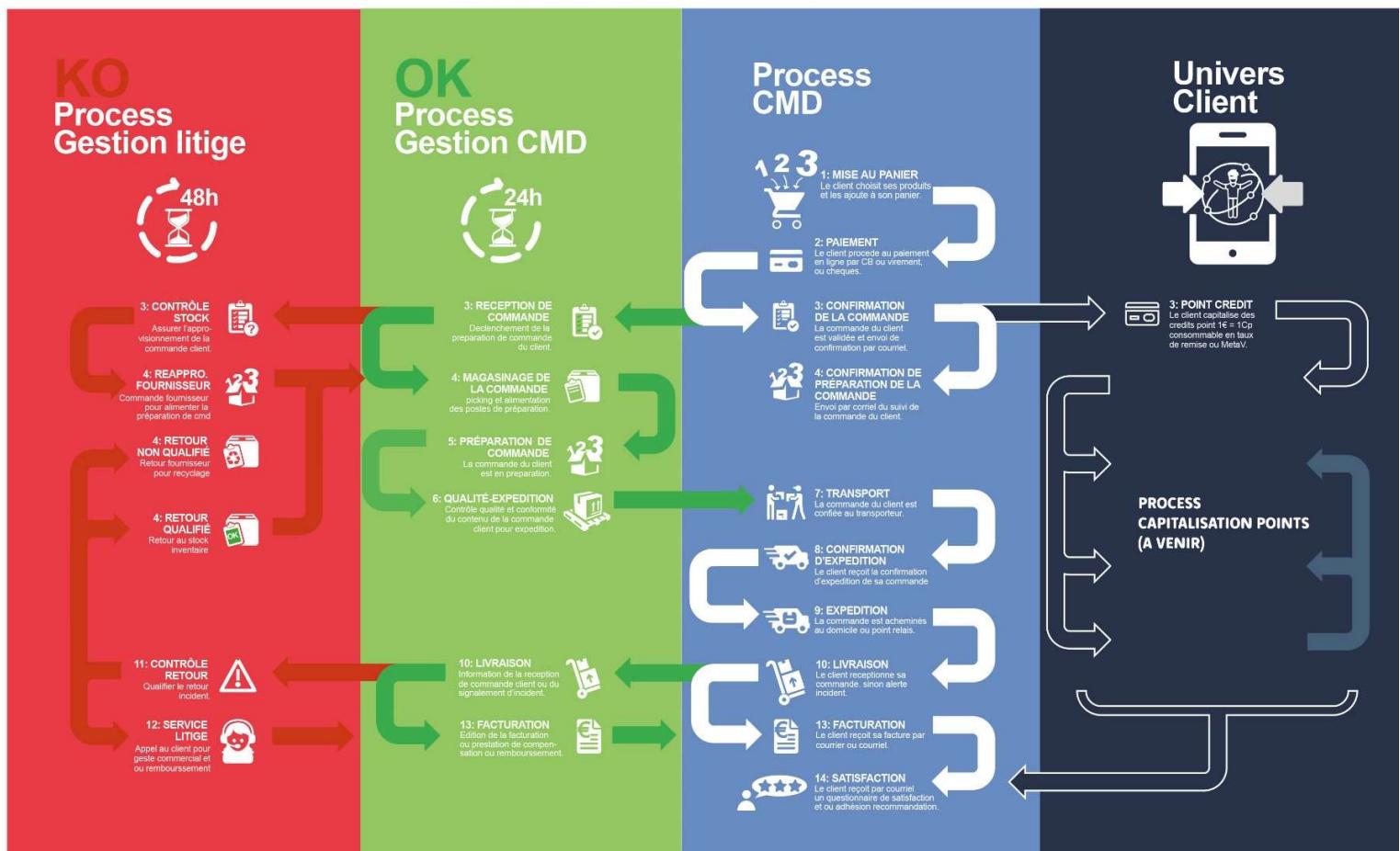


Stripe fournit également une documentation pour faciliter l'intégration de son API, accessible depuis le site internet officiel.

[Lien vers la documentation de l'API Stripe](#)

Dans un premier temps, mon tuteur de stage m'a fourni un schéma de circuit de commande que vous pouvez retrouver ci-dessous :

Notre Process CMD EMarket



Je n'ai malheureusement pas pu traiter tous les cas notamment celui d'un retour d'une commande pour des raisons de temps. Cela fait partie des améliorations à apporter au site.

Quant au module de paiement, après click du bouton « passer au paiement » dans le récapitulatif de commande, une nouvelle session de paiement est créée. Lorsque cette session de paiement a été créée, on redirige l'utilisateur vers une page côté serveur qui contient notre formulaire de paiement.

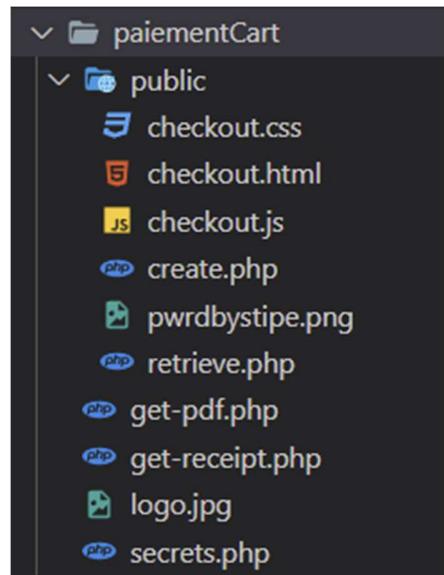
```
payForCart(idFact: any, idTransporteur: any, codePays: any) {
  if (!localStorage.getItem('authUser')) {
    window.location.href = 'http://localhost:4200/login';
  } else {
    let uid = JSON.parse(localStorage.getItem('authUser')!).id;
    let date = new Date();
    let dateString =
      date.getFullYear() +
      '-' +
      (date.getMonth() + 1) +
      '-' +
      date.getDate() +
      ' ' +
      date.getHours() +
      ':' +
      date.getMinutes() +
      ':' +
      date.getSeconds();
    let ciphertext = MD5(
      uid.toString() + idFact.toString() + dateString
    ).toString();

    this.http
      .get(
        this.baseUrl +
        'new-session-paiement.php?session=' +
        ciphertext +
        '&user=' +
        uid
      )
      .subscribe((data) => {
        uid = MD5(uid.toString()).toString();
        idFact = SHA256(idFact.toString()).toString();
        window.location.href =
          'http://localhost/api/paiementCart/public/checkout.html?session=' +
          ciphertext +
          '&idFact=' +
          idFact +
          '&uid=' +
          uid +
          '&idTransporteur=' +
          idTransporteur +
          '&codePays=' +
          codePays;
      });
  }
}
```

Etant donné que je ne peux pas passer le prix total de la commande dans l'url pour éviter qu'il puisse être modifié, je passe l'id du transporteur ainsi que le pays de livraison pour recalculer les frais de port derrière et obtenir le total à payer de l'autre côté.

Nous arrivons ensuite sur le formulaire de paiement, géré côté serveur avec les fichiers checkout.

Les informations bancaires entrées par l'utilisateur sont vérifiées et une intention de paiement est créée si tout est correct.



Paiement

420.4 €
E-mail
test@test.com

Carte bancaire EPS giropay Bancontact

Numéro de carte Date d'expiration CVC

4242 4242 4242 4242 04 / 24 424 139

Pays
France

Confirmer le paiement

Après avoir confirmé le paiement, vous serez redirigé.

Powered by [stripe](#)

```
161 $database = new Operations();
162 $conn = $database->dbConnection();
163 $sql = "SELECT idUserSession FROM session_paiement WHERE hashSession ='". $hash."'";
164 $stmt = $conn->prepare($sql);
165 $stmt->execute();
166 $data = $stmt->fetch(PDO::FETCH_ASSOC);
167 //Check if $uid is present in $data. $uid is crypted in md5 and the datas returned aren't crypted
168 if($uid == md5($data['idUserSession'])) {
169
170     $uid = $data['idUserSession'];
171     $sql = "SELECT idAdresse, libelleAdresse, cpAdresse, villeAdresse, paysAdresse FROM facturations WHERE idUserAdresse ='".$uid."'";
172     $stmt = $conn->prepare($sql);
173     $stmt->execute();
174     $data = $stmt->fetch(PDO::FETCH_ASSOC);
175     // Create a PaymentIntent with amount and currency
176     $paymentIntent = \Stripe\PaymentIntent::create([
177         'amount' => calculateOrderAmount() + calculateOrderShippingFees($codePays, $idTransporteur, calculateOrderWeight()),
178         'currency' => 'eur',
179         'metadata' => [
180             'userid' => $uid,
181             'idFact' => $data['idAdresse'],
182             'adresse' => $data['libelleAdresse'],
183             'cp' => $data['cpAdresse'],
184             'ville' => $data['villeAdresse'],
185             'pays' => $data['paysAdresse'],
186             'weight' => calculateOrderWeight(),
187             'shippingFees' => calculateOrderShippingFees($codePays, $idTransporteur, calculateOrderWeight()),
188         ],
189         'automatic_payment_methods' => [
190             'enabled' => true,
191         ],
192     ]);
193
194     $output = [
195         'clientSecret' => $paymentIntent->client_secret,
196         'prix' => (calculateOrderAmount() + calculateOrderShippingFees($codePays, $idTransporteur, calculateOrderWeight()))
197     ];

```

Une fois les informations de paiement confirmées et valides, l'intention de paiement est créée dans le fichier « create.php » et contient tout ce dont j'ai besoin pour générer une facture (montant de la commande, id de l'utilisateur, adresse de facturation, le poids total de la commande ainsi que le montant des frais de port).

Nous sommes ensuite redirigés vers la page de facturation qui génère un numéro de commande, un récapitulatif concernant les informations client ainsi que son adresse de facturation, et qui génère une facture téléchargeable au format pdf.



The screenshot shows a confirmation page from OSMarket. At the top, there's a navigation bar with links for ACCUEIL, OSMARKET, ANNONCEUR, PARTENAIRE, CONTACT, and WISHLIST. To the right, there are links for 'CONNECTÉ(E) EN TANT QUE' and 'MARKETPLACE CLIENT'. The main content area has a green header bar with the text 'MERCI POUR VOTRE ACHAT !'. Below it, a message states 'La référence de votre commande est : #FDD035EB70' and provides a link to view more commands. A section titled 'Recapitulatif' lists the following information:

| | |
|-------------------------|--------------------------|
| Adresse : 1 rue de test | Prénom : Client |
| Code postal : 75000 | Nom : Marketplace |
| Ville : Paris | Email : client@gmail.com |
| Pays : France | |

At the bottom of the page, there's a link to download the invoice: 'Téléchargez votre facture ici: #FDD035EB70'. A blue button at the bottom left says 'Retour vers la page d'accueil'.

A l'arrivée sur cette page, on lance plusieurs scripts. Le premier est le script qui génère la facture en pdf, le deuxième vide le panier de l'utilisateur en base de données, le troisième met à jour le solde de points crédit de l'utilisateur (ceci est une demande de mon tuteur qui souhaitait mettre en place un système de points de fidélité pour de futures améliorations) les points sont ajoutés de cette façon : 1 point = 1€, ce ratio pouvant être modifié en base de données Enfin, le dernier script envoie un mail de facturation à l'adresse mail renseignée sur le formulaire de paiement, mail contenant la facture en pdf.

L'utilisateur peut donc télécharger sa facture comme ci-dessous, dans laquelle il retrouve la date de la commande, le numéro de facture, l'adresse de l'entreprise, son adresse de facturation, un récapitulatif des produits qu'il a commandés, ainsi que le prix total de sa commande.

2023-03-04

Facture

N°FDD035EB70

Altameos
9 Bis rue du Puits Carré
27000 Evreux
France

Client
1 rue de test
75000 Paris
France

Vous trouverez ci dessous le récapitulatif de votre commande:

| Produit | Quantite | Prix Unitaire HT | TVA |
|--------------|----------|------------------|---------|
| Produit test | 4 | 43.00 EUR | 20.00 % |
| Sauvage | 5 | 34.00 EUR | 20.00 % |

Frais de port 10 EUR

Prix Total TTC 420.4 EUR

L'utilisateur peut alors retrouver dans ses informations de profil, dans l'onglet « Historique des commandes », toutes les commandes qu'il a passé sur le site, y retrouver chacune de ses factures, et consulter l'état de la commande, c'est-à-dire si elle est en préparation, en transit, livrée, etc...

The screenshot shows the 'Profil de Marketplace Client' section of the OSMarket website. At the top, there are tabs for 'GÉRATION DU MOT DE PASSE', 'INFORMATIONS DE LIVRAISON', 'INFORMATIONS DE FACTURATION', and 'HISTORIQUE DES COMMANDES'. The 'HISTORIQUE DES COMMANDES' tab is selected. Below it, a table lists four orders with columns for 'No. de facture', 'Date de la facture', 'Prix total', 'Etat de la commande', and 'Lien vers la facture en pdf'. The last row of the table is highlighted in yellow.

| No. de facture | Date de la facture | Prix total | Etat de la commande | Lien vers la facture en pdf |
|----------------|--------------------|------------|---------------------|--------------------------------|
| #4A2CAE212D | 2023-02-18 | 6514.70 € | En transit | 4A2CAE212D.pdf |
| #C8BE1361A6 | 2023-02-18 | 1042.00 € | Livrée | C8BE1361A6.pdf |
| #D2CA60915D | 2023-02-18 | 6514.70 € | Livrée | D2CA60915D.pdf |
| #FDD035EB70 | 2023-03-04 | 420.40 € | Préparation | FDD035EB70.pdf |

De plus, dans l'onglet « Profil », il peut constater que son solde de points crédit a augmenté.

The screenshot shows two side-by-side 'Profil de Marketplace Client' sections. Both sections have a 'PROFIL' tab selected and a 'MODIFIER : NOM / PRÉNOM / TÉLÉPHONE' button. The left section shows a credit balance of 0.00, while the right section shows a credit balance of 420.40. An arrow points from the left section to the right section, indicating the transfer of credit.

| Nom : | Client |
|--------------------------|------------------|
| Prénom : | Marketplace |
| Mail : | client@gmail.com |
| Numéro de téléphone : | 443534234 |
| Rôle sur le site : | client |
| Statut du compte : | Vérifié |
| Solde de points crédit : | 0.00 |

| Nom : | Client |
|--------------------------|------------------|
| Prénom : | Marketplace |
| Mail : | client@gmail.com |
| Numéro de téléphone : | 443534234 |
| Rôle sur le site : | client |
| Statut du compte : | Vérifié |
| Solde de points crédit : | 420.40 |

Enfin, l'utilisateur reçoit un mail de confirmation de commande à l'adresse mail renseignée sur le formulaire de paiement dans lequel il retrouve une nouvelle fois sa facture en pdf.

```

De: "Altameos.fr" <no-reply@altameos.fr>
Date: 17 février 2023 à 11:33:32 UTC+1
À: [REDACTED]
Objet: Altameos.fr - Confirmation de commande

Bonjour,

Votre commande a été prise en compte et est en cours de préparation.

Vous pouvez télécharger votre facture en cliquant sur le lien suivant :

#33483CD2ED

```

Cet envoi de mail se fait à l'aide de PHPMailer qui est une bibliothèque logicielle d'envoi d'e-mails en PHP. Voici le code côté serveur qui génère et envoie le mail :

```

8  use PHPMailer\PHPMailer\PHPMailer;
9  use PHPMailer\PHPMailer\Exception;
10
11 require 'src\Exception.php';
12 require 'src\PHPMailer.php';
13 require 'src\SMTP.php';
14
15
16
17 $mail = new PHPMailer(true);
18 try{
19     //Server settings
20     $mail->SMTPDebug = 0;
21     $mail->isSMTP();
22     $mail->Host = 'frweb7.pulseheberg.net';
23     $mail->SMTPAuth = true;
24     $mail->SMTPSecure = 'ssl';
25     $mail->Username = 'altameos@dantech.fr';
26     $mail->Password = 'jJMqCgjZSL7xCY';
27     $mail->Port = '465';
28
29     //Recipients
30     $mail->setFrom('no-reply@altameos.fr', 'Altameos.fr');
31     $mail->addAddress($_GET['email']);
32     // $mail->addReplyTo('')
33     // $mail->addCC('')
34     // $mail->addBCC('')
35
36     //Attachments
37     // $mail->addAttachment('/var/tmp/file.tar.gz');
38     // $mail->addAttachment('/tmp/image.jpg', 'new.jpg');
39
40     //Content
41     $mail->isHTML(true);
42     $mail->Subject = 'Altameos.fr - Confirmation de commande';
43
44     $mail->Body = 'Bonjour, <br><br> Votre commande a été prise en compte et est en cours de préparation. <br>
45     <br> Vous pouvez télécharger votre facture en cliquant sur le lien suivant : <br><br>
46     <a href="http://localhost/api/facturesCartPayment/' . $_GET['numFacture'] . '.pdf">#' . $_GET['numFacture'] . '</a>';
47
48     $mail->AltBody = 'Bonjour, Votre commande a été prise en compte et est en cours de préparation.
49     Vous pouvez télécharger votre facture en cliquant sur le lien suivant : http://localhost/api/facturesCartPayment/' . \$\_GET\['numFacture'\] . '.pdf';
50
51     $mail->send();
52     echo 'Message has been sent';
53 }
54 catch(Exception $e){
55     echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
56 }

```

Compétences acquises

Compétences techniques :

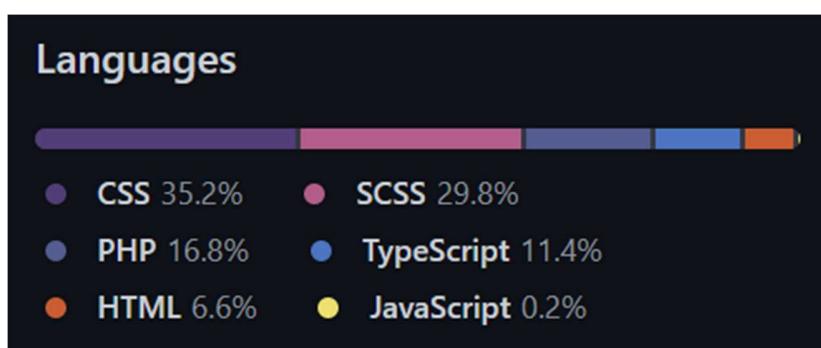
L'application web dont j'ai participé à concevoir durant ce stage, m'a fait découvrir le framework Angular.

Angular repose sur des composants dont l'affichage est développé en HTML.

Ces composants sont dynamisés grâce à une gestion en TypeScript, un sur-ensemble syntaxique strict de JavaScript.

La connexion entre la base de données et l'application web se fait grâce à une API CRUD en PHP qui permet d'exécuter des requêtes http.

Le design de l'interface a été réalisé principalement en SCSS ainsi que du CSS avec Angular Material et bootstrap.



Compétences professionnelles et sociales :

- Une certaine ponctualité nous a été exigée
- Expérience nouvelle pour moi du télétravail
- Travail en groupe sur un grand projet
- Réunions quotidiennes en appel sur Teams pour débriefer de notre avancée

Conclusion

Ce stage a donc été très formateur pour moi, d'un point de vue technique, j'ai dû me former au framework Angular, qui était nouveau pour moi, pour réaliser une application web complète from scratch (en partant de zéro). J'ai pu toucher à des fonctionnalités que je n'avais pas l'habitude de développer notamment le module de paiement ce que j'ai trouvé très intéressant et utile pour mon expérience que je pourrais valoriser dans de futurs projets.

Mais il a aussi été formateur d'un point de vue professionnel et social, puisque j'ai fait l'expérience particulière du télétravail et ai participé à une vie de groupe tout au long du stage. J'ai d'ailleurs beaucoup apprécié le travail en groupe, l'ambiance et le dynamisme que cela apportait.

Je suis toutefois légèrement frustré de ne pas avoir pu rendre un travail complet en tout point dans les délais, puisqu'il reste encore quelques détails à régler.

Ce stage m'a permis de mettre en lumière des points d'amélioration, notamment d'apprendre à simplifier mon approche des problématiques.

Pour finir, je suis très satisfait de mon stage, de l'expérience professionnelle que j'ai pu en tirer et de ce que j'ai appris en programmation web.

Annexes

Quelques commandes simples sont mises à notre disposition sous Angular.

En voici quelques-unes :

- Installer Angular :

```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned  
npm install -g @angular/cli
```

```
PS C:\WINDOWS\system32> Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned  
Modification de la stratégie d'exécution  
La stratégie d'exécution permet de vous prémunir contre les scripts que vous jugez non fiables. En modifiant la  
stratégie d'exécution, vous vous exposez aux risques de sécurité décrits dans la rubrique d'aide  
about_Execution_Policies à l'adresse https://go.microsoft.com/fwlink/?LinkID=135170. Voulez-vous modifier la st  
ratégie  
d'exécution ?  
[O] Oui [T] Oui pour tout [N] Non [U] Non pour tout [S] Suspendre [?] Aide (la valeur par défaut est « N »)  
) : T  
PS C:\WINDOWS\system32> npm install -g @angular/cli  
npm WARN deprecated @npmcli/move-file@2.0.1: This functionality has been moved to @npmcli/fs  
[ ] | reify:minipass-fetch: http fetch GET 200 https://registry.npmjs.org/wcwidth/-/wcwidth-1
```

- Créer un nouveau projet Angular :

```
ng new [Nom de l'application]
```

```
PS C:\Users\dnoll\Desktop> ng new test-angular  
? Would you like to share pseudonymous usage data about this project with the Angular Team  
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more  
details and how to change this setting, see https://angular.io/analytics. (y/N) y
```

- Lancer le serveur local pour faire fonctionner l'application :

```
ng serve --open
```

```
PS C:\wamp64\www\marketplace> ng serve --open
Node.js version v19.4.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please
see https://nodejs.org/en/about/releases/.
✓ Browser application bundle generation complete.

Initial Chunk Files | Names      | Raw Size
vendor.js           | vendor     | 7.06 MB |
main.js             | main       | 956.04 kB |
styles.css, styles.js | styles    | 380.32 kB |
polyfills.js        | polyfills  | 314.28 kB |
runtime.js          | runtime    | 6.52 kB |

| Initial Total | 8.68 MB

Build at: 2023-02-08T21:37:27.099Z - Hash: 4ae969d3d1f9bbbbf - Time: 12792ms
```

- Créer un nouveau composant :

```
ng generate component [Nom du composant]
```

```
PS C:\wamp64\www\marketplace> ng generate component test
Node.js version v19.4.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please
see https://nodejs.org/en/about/releases/.
CREATE src/app/test/test.component.html (19 bytes)
CREATE src/app/test/test.component.spec.ts (585 bytes)
CREATE src/app/test/test.component.ts (195 bytes)
CREATE src/app/test/test.component.scss (0 bytes)
UPDATE src/app/app.module.ts (11146 bytes)
PS C:\wamp64\www\marketplace>
```



- Créer un nouveau service

ng generate service [Nom du service]

```
PS C:\wamp64\www\marketplace> ng generate service test
Node.js version v19.4.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please
see https://nodejs.org/en/about/releases/.
CREATE src/app/test.service.spec.ts (347 bytes)
CREATE src/app/test.service.ts (133 bytes)
PS C:\wamp64\www\marketplace>
```

```
⠇ test.service.spec.ts      U
⠇ test.service.ts          U
```