

# Cahier de charges pour la création d'un site web « Market place pour les artisans »

Groupe 2 : Aurélie/François/Farah/Mohamed

## Présentation :

L'objectif de notre travail est de développer une application web qui fait office de marketplace pour des artisans qui souhaitent vendre leurs produits.

Les artisans auront des comptes artisans avec lesquels ils posteront leurs produits sous forme d'annonces pour qu'ils soient exposés aux clients qui seront intéressés. Les clients pourront ensuite consulter les annonces pour voir les produits et les informations qui vont avec, et ainsi s'ils sont intéressés ils pourront commander le ou les produits en question après avoir créé un compte.

## Procédé :

- Charte graphique
- Création et récupération de contenu
- Développement
- Maquette du site
- Hébergement de l'application
- Référencement
- Mise à jour
- Statistique des données

## Fonctionnalités :

Définir les grandes fonctionnalités de notre application (4 fonctionnalités à proposer).

- Gestion des produits, recherche des produits, Mise en ligne des annonces.
- Gestion des clients, enregistrement des données clients, enregistrement des options de paiement.
- Gestion des profils des artisans.
- Messagerie de contact client – vendeur.

## Répartition des tâches :

Nous avons donc choisi 4 fonctionnalités, Nous allons nous en attribuer une chacun.

Répartition fonctionnalités :

Gestion des produits : Création du produit, modifier, Recherche des produits, catégories produit, mots clés, spécialité d'artisans ... (Farah)

Mise en ligne des annonces/publicité (Mohamed)

Gestion des utilisateurs : Gestion Compte client/compte artisans (email, nom, prénom, métier) (François)

Messagerie de clients/artisan

Gestion de commandes + paiement des commandes (Aurélie)

## 1. Gestion des utilisateurs (François)

### 1.a. Détails de la fonctionnalité : Gestion des utilisateurs (François)

Les utilisateurs de ce site peuvent consulter les annonces et deviennent clients de ce site dès qu'ils créent un compte utilisateur pour passer une commande. Donc une étape de création de compte utilisateur est nécessaire. On proposera alors à l'utilisateur de remplir un formulaire dans lequel il fournira les informations comme :

-Nom et prénom

-Adresse email

- Mot de passe

-Numéro de téléphone

-Adresse personnelle (adresse, adresse secondaire, code postal, ville)

Proposer à l'internaute une fonctionnalité par exemple trouver sa ville avec un code postal etc.

-Mode de livraison (domicile, dépôt)

-Photo de profil (optionnel)

-Mode de paiement (par carte, virement)

Récupération de ces informations remplies par les utilisateurs.

La connexion par les réseaux sociaux, tel que Facebook, est aussi envisageable.

Une fois le compte validé, le client pourra se connecter via une nouvelle page de connexion sur laquelle il renseignera son adresse email et son mot de passe (Système d'authentification des clients par login et mot de passe). Ensuite il pourra valider sa commande et procéder au paiement.

L'utilisateur a la possibilité de modifier ses informations. Il a également la possibilité de supprimer son compte. Il peut également consulter les paramètres de sécurité et de confidentialité et enfin il pourra nous-contacter.

## L'artisan :

Également, l'artisan doit créer un compte pour pouvoir déposer ses annonces. L'artisan doit remplir un formulaire dans lequel il fournira les informations comme :

- Nom et prénom
- Adresse email \*
- Mot de passe
- Numéro de téléphone
- Adresse personnelle (adresse, adresse secondaire, code postal, ville)
- Mode de livraison (domicile, dépôt)
- Photo de profil

Une fois le compte validé, l'artisan pourra se connecter via une nouvelle page de connexion sur laquelle il renseignera son adresse email et son mot de passe (Système d'authentification des clients par login et mot de passe). Ensuite il pourra déposer ses annonces.

L'artisan a la possibilité de modifier ses informations. Il a également la possibilité de supprimer son compte. Il peut également consulter les paramètres de sécurité et de confidentialité et enfin il pourra nous-contacter.

Nous avons par la suite, travaillé sur la charte graphique du point de vue pages du site et aussi du point de vue composants.

Dans la suite j'explique la charte graphique concernant ma fonctionnalité (gestion des utilisateurs et des artisans).

### 1.b. Carte du site et plan de navigation :

La figure1 montre le plan du site. Il présente les grandes lignes de la structure de notre site. Il contient les pages web que nous voulons créer et les informations qu'elles contiennent.

Concernant ma fonctionnalité : gestion des utilisateurs et des artisans, à partir d'une page d'accueil, l'utilisateur ou l'artisan peut se connecter en cliquant sur l'élément «Login ». Ainsi, une page de connexion s'ouvre pour entrer les identifiants.

Une fois connecté, l'utilisateur est renvoyé vers la page d'accueil qui contient les annonces.

Si l'utilisateur n'a pas de compte, il peut créer un compte via une nouvelle page « création de compte ». Une fois les informations demandées remplies, l'utilisateur valide et une nouvelle page de connexion s'ouvre. L'utilisateur se connecte. Une fois connecté, l'utilisateur est renvoyé vers la page d'accueil qui contient les annonces.

Dans le cas où l'artisan veut se connecter, s'il est connecté il est envoyé vers la page «Ajouts d'annonces ».

La figure2 montre le plan de navigation du point de vue des composants. A partir de la navbar, l'utilisateur peut cliquer sur l'élément « Login ». Ainsi, une page de connexion s'ouvre pour entrer les identifiants.

Une fois connecté, l'utilisateur est renvoyé vers la page d'accueil qui contient les annonces.

Egalement, s'il est connecté, l'utilisateur voit apparaitre un icone dans la navbar. En cliquant sur cet icone, il peut modifier ses informations ou supprimer son compte.

Si l'utilisateur n'a pas de compte, il peut créer un compte via une nouvelle page « création de compte ». Une fois les informations demandées remplies, l'utilisateur valide et une nouvelle page de connexion s'ouvre. L'utilisateur se connecte. Une fois connecté, l'utilisateur est renvoyé vers la page d'accueil qui contient les annonces.

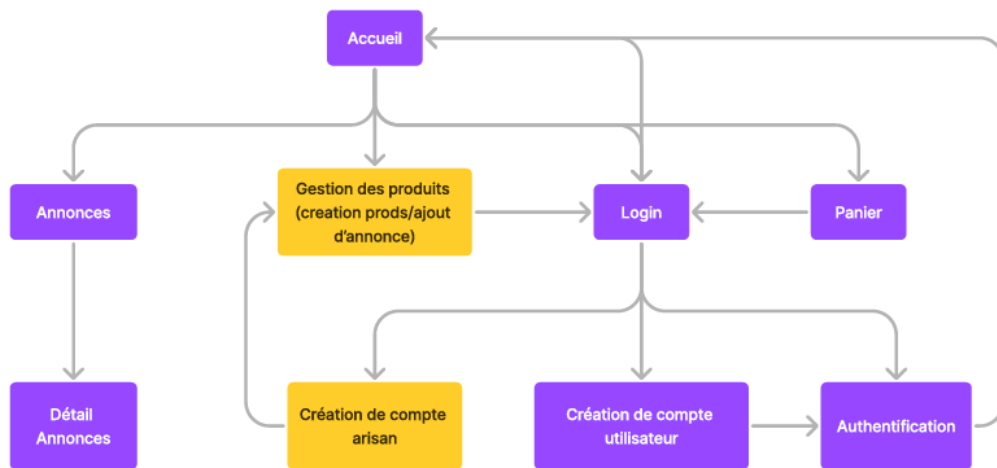


Fig.1. Carte du site. Point de vue des pages.

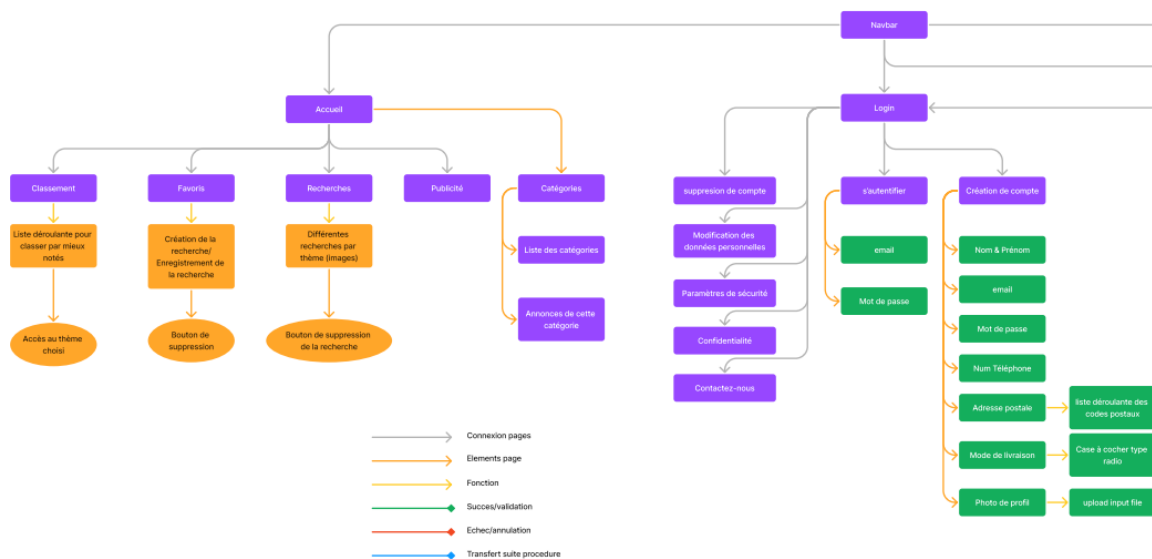


Fig.2.a Plan de navigation. Point de vue des composants.

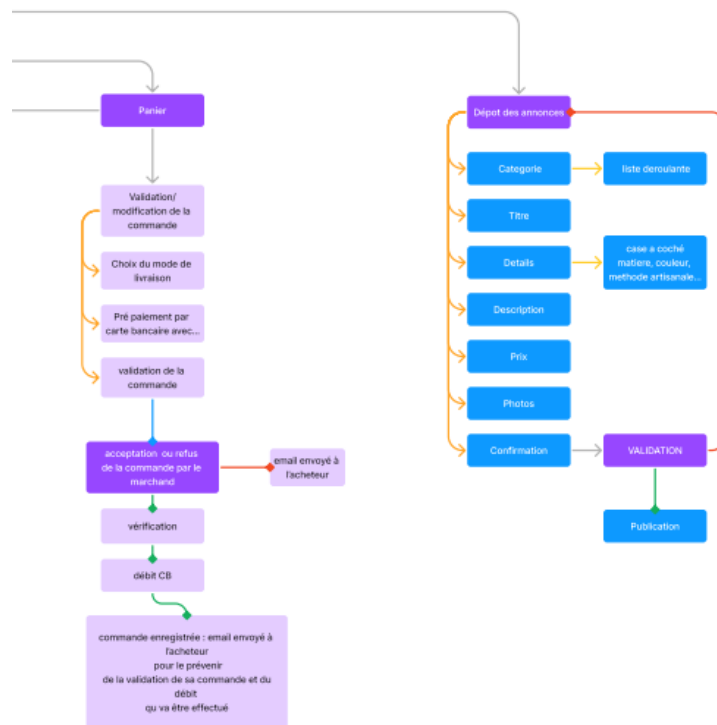


Fig.2.b Plan de navigation. Point de vue des composants.

### 1.c. Maquettage :

Après avoir terminé le plan de navigation, j'ai effectué le maquettage de ma fonctionnalité pour ce site.

Concernant ma fonctionnalité : gestion des utilisateurs et des artisans, à partir d'une page d'accueil, l'utilisateur ou l'artisan peut se connecter en cliquant sur l'élément «Login ». Ainsi, une page de connexion s'ouvre pour entrer les identifiants (Fig.3).

Home icon | **CATÉGORIES** ▾ | Barre de recherche | **Login** | **Panier**

---

**Bonjour, veuillez entrer vos identifiants, sinon créer un compte**

**Adresse email \***

**Mot de passe \***

Mot de passe oublié ?

**Se connecter**

Créer un compte

Fig.3. Page connexion des utilisateurs.

Une fois connecté, l'utilisateur est renvoyé vers la page d'accueil qui contient les annonces.

Si l'utilisateur n'a pas de compte, il peut créer un compte via une nouvelle page « création de compte » (Figure 4).

Header: Home icon, Logo, CATÉGORIES, Barre de recherche, Login, Panier

Form Title: Merci de renseigner les champs suivants

Form Fields:

- Nom et prénom
- Adresse email \*
- Mot de passe \*
- Numéro de téléphone
- Adresse postale
- Mode de livraison
- Photo de profil (with upload button)

Form Button: Créer votre compte

Fig.4. Page création de compte.

Une fois les informations demandées remplies, l'utilisateur valide et une nouvelle page de connexion s'ouvre. L'utilisateur se connecte. Une fois connecté, l'utilisateur est renvoyé vers la page d'accueil qui contient les annonces.

Dans le cas où l'artisan veut se connecter, s'il est connecté il est envoyé vers la page « Ajouts d'annonces ».

Egalement, s'il est connecté, l'utilisateur voit apparaître une icône dans la navbar. En cliquant sur cet icône, il peut modifier ses informations ou supprimer son compte (Figure 5 et Figure 6).

Header: Home icon, Logo, CATÉGORIES, Barre de recherche, User profile icon, Panier

Welcome Message: Bienvenue User xxx

User Actions:

- Modifier vos données personnelles (with edit icon)
- Supprimer votre compte (with delete icon)
- Paramètres d' sécurité
- Confidentialité

Contact Button: Contactez-nous

Fig.5. Page modification de compte.



Fig.6. Page modification de compte.

### 1.c. Réalisation des interfaces utilisateurs statiques web ou web mobile :

Une fois le maquettage terminé, j'ai réalisé les interfaces statiques (IHM) en utilisant HTML3, Bootstrap et CSS3.

J'ai effectué quelques modifications de la maquette :

-J'ai rajouté un élément « Login artisan » dans la barre de navigation. Je pense que séparer l'accès au site pour les clients ou pour les artisans. Le client connecté est redirigé vers la page d'accueil. L'artisan connecté est redirigé vers la page des dépôts des annonces ou produits.

-J'ai rajouté des icônes pour quelques éléments de la barre de navigation.

Dans la suite j'explique la réalisation des différentes pages.

#### 1.c.1.Page « PageConnexion.html»

-HTML : C'est un langage de balisage qui définit la structure du contenu. Un document HTML est une suite d'éléments utilisés pour encadrer différentes parties du contenu afin de les faire apparaître ou se comporter d'une certaine façon. Les balises autour du contenu permettent de transformer une image ou un texte pour obtenir un lien vers une autre page, de créer des titres, etc.

-head : dans un document HTML head est une partie du document qui n'est pas affichée par le navigateur au chargement de la page. Elle contient des informations comme le titre (<title>) de la page, des liens aux CSS (si vous souhaitez composer le

contenu HTML grâce des CSS), des liens aux favicons et des méta-données (auteur du document, mots-clés décrivant le document, etc.).

Par exemple dans mon cas :

```
<link rel="stylesheet" href="/marketplacev2.css">
```

-Le Doctype : La définition de type de document (aussi appelée DTD) spécifie la version HTML utilisée dans le code source d'un site web. De cette façon, un navigateur ou un autre logiciel de lecture peut afficher le contenu du code source car il est capable de détecter le type de document dont il s'agit

```
<!DOCTYPE html>
```

-La déclaration de la langue :

```
<html lang="fr">
```

Afin de garantir la bonne restitution des contenus textuels, une déclaration de langue principale doit être effectuée sur chaque page.

-Barre de navigation :

-Deux navbar ont été réalisées. Une que j'ai faite. C'est une barre horizontale faite avec bootstrap5.



Lors de notre réunion de groupe, on a décidé d'utiliser la navbar que Mohamed avait réalisé.



J'ai utilisé le CSS3 pour travailler le style de toutes les pages. Page de style « marketplacev2.css ».

Pour la connexion du client ou de l'artisan, j'ai fait un formulaire avec bootstrap5 et j'ai amélioré le style avec css3.




Shopart

Categorie

Rechercher un produit ou u

Rechercher

Login Artisan

Login Client

Panier

## Bonjour, bienvenue sur la page de connexion

Veuillez entrer vos identifiants sinon créer un compte

Adresse email

Mot de passe

☐
Rester connecté  
[Mot de passe oublié?](#)

SE CONNECTER

Vous n'avez pas de compte

CRÉER UN COMPTE

-Par exemple, quand le client survole le bouton « se connecter » ou le bouton « créer un compte », le gradient linéaire de couleur s'inverse ainsi que l'espacement entre les lettres s'agrandit. Egalement, j'ai rajouté un effet « shadow ».

```
.btn-2:hover{
  background-image: linear-gradient(to right,rgb(255,69,0),rgb(139,69,19));
  letter-spacing: 0.25em;
  box-shadow: 0 0 35px;
```

Adresse email

Mot de passe

☐
Rester connecté  
[Mot de passe oublié?](#)

SE CONNECTER

Vous n'avez pas de compte

CRÉER UN COMPTE

Une fois le client a rentré les identifiants, et cliqué sur se connecter alors il est redirigé vers la page suivante.



Il pourra alors modifier ou supprimer son compte ou changer de page.

#### 1.c.2. Page « CreationCompteV2.html »

Pour la création de compte du client ou de l'artisan, j'ai fait un formulaire avec bootstrap5 et j'ai amélioré le style avec css3.

J'ai utilisé un élément div « container » et à l'intérieur un élément div « row » pour pouvoir afficher en colonne les champs du formulaire.

```
<div class="container mt-5">
  <form class="row g-3" id="form" method="get"
action="https://www.google.com">
```

J'ai mis dans le formulaire un bouton radio pour spécifier s'il s'agit d'un client ou d'un artisan. Voici le résultat :

☒ Client
 ☐ Artisan

Nom

Prénom

Numéro de Téléphone

XX XX XX XX XX

Adresse email

Confirmez votre email

Mot de passe

Confirmez votre mot de passe

Adresse

Complément d'adresse

Rue, numéro

Bat, Res, Chez.

Code postal

Ville

Pays

Choisir

Ajouter une photo de profil

Choisir un fichier

Aucun fichier choisi

SOUMETTRE

Une fois le client a soumis le formulaire alors il est redirigé vers la page suivante.


Shopart

Categorie

Rechercher un produit

Rechercher

Gérer mon compte

Déconnexion

Panier

CLIENT, CONSULTER LES ANNONCES

ARTISAN, DÉPOSEZ DES ANNONCES

SUPPRIMER VOTRE COMPTE

MODIFIER VOTRE COMPTE

DÉCONNEXION

Il pourra alors modifier ou supprimer son compte ou changer de page.

1.c.3.Page « ModifSuppCompte.html »



S'il souhaite, le client ou l'artisan peut cliquer sur « modifier votre compte » pour modifier ses informations ou cliquer sur supprimer votre compte pour la suppression.

J'ai utilisé le media query dans le css3 pour spécifier l'apparence de l'application pour les écrans de largeur en dessous de 600x.

```
@media only screen and (max-width:600px){
  .container{
    width: 90%;
    position: absolute;
    left: 50%;
    top: 40%;
    padding-bottom: 40px;
  }
}
```

La largeur du container passera de 50% à 90%.

Dans ce travail j'ai défini dans un fichier css le style pour les différents éléments de la page :

```
.container{
  width: 50%;
  height: 50vh
  background-image: linear-gradient(to right,rgb(255,69,0),rgb(255, 153,
35));

  padding: 20px;
  margin-top: 10px;
  border-radius: 5px;
  position: absolute;
  left: 50%;
```

```
    top: 40%;
    transform: translate(-50%,-50%);
}
.form-control{
    border-radius: 0;
    border: none;
    border-bottom: 1px solid rgb(255,69,0);
    font-size: 20px;
}
.form-select{
    border-radius: 0;
    border: none;
    border-bottom: 1px solid rgb(255,69,0);
    font-size: 20px;
}
.form-label{
    font-size: 20px;
    color: white;
}
.btn-primary{
    border-radius: 0;
    background-color: rgb(255,69,0);
    border: none;
}
.form-check-label{
    font-size: 20px;
    color: white;
}
.form-control-file{
    font-size: 20px;
    color: white;
}
.form-control-file{
    font-size: 20px;
    color: white;
}
.col-10{
    color: white;
    padding: 20px;
    margin-left: 30px;
}
.btn-2{
    background-image: linear-gradient(to left,rgb(255,69,0),rgb(139,69,19));
    font-size: 20px;
    color: white;
    letter-spacing: 0.1em;
    transition: 0.5s;
    text-transform: uppercase;
```

```

}
.btn-2:hover{

    background-image: linear-gradient(to right,rgb(255,69,0),rgb(139,69,19));
    letter-spacing: 0.25em;
    box-shadow: 0 0 35px;

}
.ajoutphoto{
    font-size: 20px;
    color: white;
}
.taille{
    font-size: 20px;
    color: white;
}
header {
    width:auto;
    background-size: cover;
}
nav{
    display:flex;
}
.nav-link{
    color: rgb(139,69,19) !important;
    font-size: 22px;
    font-weight: bolder;
    margin: 0 50px;
}
.search{
    color: rgb(139,69,19) !important;
    font-size: 22px;
    font-weight: bolder;
}
h1,h2{
    color: white;
    font-size: 25px;
}
.btn-1:hover{
transform: translateY(-5px);
}

/**media Query**/
@media only screen and (max-width:600px){
    .container{
        width: 90%;
        position: absolute;

```

```

    left: 50%;
    top: 40%;
    padding-bottom: 40px;
  }
}

```

## 1.d Réalisation d'un site web adaptable : dynamique et le publier

Dans cette étape, j'ai contrôlé les saisies utilisateurs avec javascript. Des messages d'erreurs s'affichent sur la page en cas d'erreur de saisie ou de succès de saisie.

J'ai utilisé le POO pour la création des clients et des artisans à partir d'une d'une classe utilisateur.

### 1.1.d Contrôle de saisie lors de de l'authentification de l'utilisateur

#### PageConnexionV3AvecQunit.js

Dans les champs inputs du formulaire dans la page html, j'ai rajouté un identifiant email pour pouvoir le pointer avec javascript en utilisant. J'ai fait la même chose avec le champ input password, form. J'ai également crée une div avec un id='error' pour pouvoir afficher les messages d'erreurs en cas d'erreurs de saisie.

```

const email = document.getElementById('email')
const password = document.getElementById('password')
const form = document.getElementById('form')
const errorElement = document.getElementById('error')

```

J'ai rajouté une icone de type 'eye' qui affiche ou cache le mot de passe :

```

const eyeIcon = document.querySelector(".pass-field i");
eyeIcon.addEventListener("click", () => {
  password.type = password.type === "password" ? "text" : "password";

  eyeIcon.className = `fa-solid fa-eye${password.type === "password" ? "" : "-slash"}`;
});

```

A form, j'ai rajouté un adventlistener qui déclenche au submit une fonction qui traitera les saisies utilisateurs et affichera les messages d'erreurs.

```

function valider(e){
  let messages = []
  if (email.value === '' || email.value == null) {
    messages.push('email is required')
  }

  if (password.value !== '' && password.value.length <= 6) {
    messages.push('Le mot de passe doit contenir au moins 6 caractères')
  }

  if (password.value.length >= 20) {
    messages.push('Le mot de passe doit contenir plus de 20 caractères')
  }
}

```

```

}

if (password.value === 'password') {
  messages.push('Password cannot be password')
}

if (messages.length > 0) {
  e.preventDefault()
  errorElement.innerText = messages.join(', ')
}
}

```

J'ai étudié le cas où l'utilisateur a fourni un champ input email vide, un mot de passe court ou très long et le cas où il rentre password comme mot de passe.

J'ai effectué les test avec Qunit et ça a marché :

```

//Pour les tests unitaires : il faut decommenter la fonction suivante et
commenter la fonction ci-haut

// function valider(arg){
//   let messages = []
//   let code;
//   let retour = {code, messages}
//   if (arg === '' || arg == null) {
//     retour.messages.push("email is required");
//     retour.code = 0;
//   }
//   if (arg !== '' && arg.length < 6 ) {
//     retour.messages.push("Le mot de passe doit contenir au moins 6
caractères");
//     retour.code = 1;
//   }
//   if (arg.length > 20 ) {
//     retour.messages.push("Le mot de passe doit contenir plus de 20
caractères");
//     retour.code = 2;
//   }
//   return retour;
// }
// const {test} = QUnit;
// test('T1 - email is required', assert => {
//   const reponse = valider2("");
//   assert.equal(reponse.code,0);
//   assert.equal(reponse.messages,"email is required");
// });
// test('T2 - mot de passe court', assert => {
//   const reponse = valider2("v");
//   assert.equal(reponse.code,1);

```



```
//      assert.equal(reponse.messages,"Le mot de passe doit contenir au moins 6 caractères");  
// });  
  
// test('T3 - mot de passe long', assert => {  
//     const reponse = valider2("vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv");  
//     assert.equal(reponse.code,2);  
//     assert.equal(reponse.messages,"Le mot de passe doit contenir plus de 20 caractères");  
// });
```

#### 1.1.d saisie de formulaire de création de compte de l'utilisateur

## dataCreationCompte.js et PageCreationCompteV2.js

Dans `dataCreationCompte.js`, j'ai créé une classe :

```
export class Utilisateur
```

Cette classe contient, la fonction constructor, les setters et les getters

```
getComposantHtml(){
```

et

```
toString() {
```

Le code est dans le fichier dataCreationCompte.js.

Ce code permet de créer un utilisateur. Par exemple j'ai créé trois utilisateurs :

```
var user1 = new
Utilisateur("art", "p1", "Sylvain", "Barbay", "0611223344", "s.bar@gmail.com", "s.ba
r@gmail.com", "1111", "1111", "Rue Marechal hoche", "Residence les
totos", "75000", "Paris", "image1.jpg", "", new Date())
var user2 = new
Utilisateur("art", "p2", "Arnauld", "Lefort", "0655663322", "a.lefort@gmail.com", "a
.lefort@gmail.com", "1111", "1111", "Rue Eiffel", "Residence les
camelias", "06000", "Nice", "images-users/pexels-andrea-piacquadio-
774909.jpg", "", new Date())
var user3 = new
Utilisateur("art", "p3", "Robert", "Laffitte", "0615141316", "r.laffitte@gmail.com"
, "r.laffitte@gmail.com", "1111", "1111", "Rue Jean Moulin", "Residence les
cerisiers", "13000", "Paris", "images-users/pexels-anastasiya-gepp-
1462637.jpg", "", new Date())
export let users = [user1, user2, user3];
```

Un client peut créer un compte en remplissant le formulaire suivant :

Un bouton de type radio vérifie s'il est client ou artisan dont voici le code :

```
let checkclient = document.querySelector("#client");
let choix1 = checkclient.checked;
let checkartisan = document.querySelector("#artisan");
let choix2 = checkartisan.checked;

if (choix1){document.querySelector("#message").innerHTML = "Vous êtes un Client";}
else if (choix2){document.querySelector("#message").innerHTML = "Vous êtes un artisan";}
```

Ensuite, un autre code vérifie la saisie des autres champs inputs :

```
var erreur;
var inputs = document.getElementsByTagName("input");
for (var i = 0; i < inputs.length; i++){
    if (inputs[i].value===null || inputs[i].value.length < 2){
        erreur = "Veuillez renseigner tous les champs";
        inputs[i].classList.add("error");
    }
}
if (document.querySelector("#mail1").value !== document.querySelector("#mail2").value){
    erreur = "les adresses emails ne sont pas identiques";
    document.getElementById("mail1").classList.add("error");
}
```

```

        document.getElementById("mail2").classList.add("error");
    }
    if (document.querySelector("#tel").value.length < 10 ||
isNaN(document.querySelector("#tel").value)){
        erreur = "Le numéro de téléphone de contenir 10 chiffres";
        document.getElementById("tel").classList.add("error");
    }
    if (document.querySelector("#passw1").value !==
document.querySelector("#passw2").value){
        erreur = "les mots de passes ne sont pas identiques";
        document.getElementById("passw1").classList.add("error");
        document.getElementById("passw2").classList.add("error");
    }

    if (erreur){
        document.getElementById("message2").innerHTML = erreur;
        e.preventDefault();
        return false;
    }
    else if (!erreur){
        alert('Formulaire envoyé');
        let nom = document.querySelector("#nom").value;
        let prenom1 = document.querySelector("#prenom").value;
        let tel1 = document.querySelector("#tel").value;
        console.log(tel1.length);
        let mail1 = document.querySelector("#mail1").value;
        let mail2 = document.querySelector("#mail2").value;
        let passw1 = document.querySelector("#passw1").value;
        let passw2 = document.querySelector("#passw2").value;
        let adresse1 = document.querySelector("#inputAddress1").value;
        let adresse2 = document.querySelector("#inputAddress2").value;
        // let codepostal1 = document.querySelector("#codepostal").value;
        let codepostal1 = '06000';
        let ville1 = document.querySelector("#zoneSelect2").value + 'code postal';
        let departement1 = document.querySelector("#zoneSelect1").value;
        let photo1 = document.getElementById('photoprofil').files[0].name
        //Les images sont dans assets/images/images-users
        let leType = 'art';
        document.querySelector("#form-user").style.display = "none";
        let user = new data.Utilisateur(leType, "p1",
nom,prenom1,tel1,mail1,mail2,passw1,passw2,adresse1,adresse2,codepostal1,ville
1,departement1,photo1,new Date());
        return user;
    }
}

```

S'il y a un champ vide, un message d'erreur s'affiche ainsi que le style du champ input change. Le changement s'effectue en rajoutant la classe 'error'

```

for (var i = 0; i < inputs.length; i++){

```

```
if (inputs[i].value===null || inputs[i].value.length < 2){  
    erreur = "Veuillez renseigner tous les champs";  
    inputs[i].classList.add("error");  
}  
}
```

Le numéro de tel doit comporter 10 chiffres :

```
if (document.querySelector("#tel").value.length < 10 ||  
isNaN(document.querySelector("#tel").value)){  
    erreur = "Le numéro de téléphone de contenir 10 chiffres";  
    document.getElementById("tel").classList.add("error");  
}
```

## 2.Mise en ligne des annonces (Mohamed):

La page de mise en ligne des annonces sera accessible à partir du menu principal des utilisateurs possédant un compte artisan.

Lorsqu'ils lanceront la procédure de mise en ligne d'annonces, ils seront dirigés vers une page qui contiendra un formulaire de dépôt qu'ils rempliront avec les informations de leur produit à mettre en ligne pour constituer l'annonce.

Le formulaire va demander :

Le choix d'une catégorie (bijouterie, textile, décoration ..)

Un titre : pour prendre les mots-clés les plus explicites et qui décrive le mieux le(s) produit(s) brièvement

Une description détaillée dans laquelle l'artisans décrira les spécificités de son(ses) produit(s) : les dimensions, le(s) matériau(x), les couleurs, les méthodes utilisés pour sa fabrication ...

Des photos du(des) produit(s) avec photo préférée

Description brève

Video (non obligatoire)

Et le prix

Date de création (publication) de l'annonce

stock

Il pourra ensuite choisir de communiquer ou non ses coordonnées (mail / téléphone) pour les clients qui seront intéressés par les produits pour éventuellement le contacter.

L'artisan pourra ensuite valider et il accèdera à une page qui récapitulera les informations saisies pour qu'il se relise et vérifie.

Si tout est bon, l'artisan pourra valider le dépôt de son annonce, ou modifier certaines informations si nécessaire.

A partir de là, l'annonce sera mise en ligne et s'affichera sur la page des annonces qui affichera toutes les annonces triées par catégorie, matériau, couleur etc.

Le relais est passé maintenant à la partie « gestion des produit ».

### 3.Gestion des produits (Farah):

L'affichage de la page :

L'entête : contiendra le logo de l'entreprise et les options de navigations (avec une icône barre de recherche de produits par artisans, couleur, ..... ) qui sera la même pour toutes les autres pages du site.

Dans mon contenu principal :

On y trouve des icones de tris dans le but de faciliter la recherche pour le client et cela sera afficher comme suit :

- \* Prix (moins de 20euros/plus de20 /des barres de recherche prix min et max).

- \*Vendeurs (meilleur vendeur \*\*\*\*\*/ tous les vendeurs).

- \*Trier par (plus cher/moins cher/meilleurs avis/les plus récents).

- \* Ensuite j'aurais toutes les images des produits avec en dessous un titre/une description en bref du produit/le prix/et des étoiles pour les avis) les produits seront listés les uns à la suite des autres, avec une photo, le titre le prix et les étoiles.

-Dès qu'on choisit un produit, une nouvelle page s'affiche avec des images en (carrousel) et une petite vidéo si possible, taille/ dimension du produit/ nombre de produits disponible /prix/...

Dans la barre latérale :

On pourra voir de la publicité des artisans pour leurs nouveaux produits à venir

En bas de la page on peut voir les dernières recherches effectuées :

L'utilisateur peut, s'il est connecté, enregistrer sa recherche pour la relancer plus tard.

Il aura aussi accès à la liste des recherches qu'il a mémorisé.

Il pourra supprimer une de ses recherches au besoin.

A partir de la liste ses recherches, l'utilisateur peut en choisir une et la modifier avant de la Relancer.

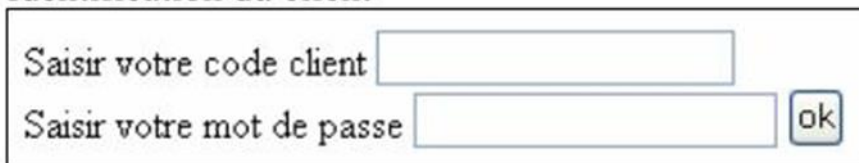
#### 4. Fonctionnalité Gestion de commandes (Aurelie):

La gestion des commandes joue un rôle extrêmement important dans la notoriété de l'entreprise. En effet la gestion rapide des commandes est un critère de satisfaction client.

La gestion de commandes concerne plusieurs étapes : le passage de la commande, l'enregistrement de la commande, l'exécution de la commande, la livraison du colis et le retour du colis éventuellement.

\_ Tout d'abord le client va naviguer sur la marketplace à la recherche de son article grâce à la barre de recherche, le client verra donc une page avec une barre de recherche ainsi que les produits classés par catégories. Quand il aura repéré ce qu'il souhaite il pourra le mettre dans son panier. Cependant avant de pouvoir commander, le client va devoir s'identifier. Il sera alors sur une page d'identification qui lui permettra d'accéder à son espace privé. Le Client verra donc à l'écran ce moyen d'identification :

##### Identification du client




Saisir votre code client

Saisir votre mot de passe

\_ Une fois qu'il sera identifié le client pourra alors valider son panier d'achat. Il sera sur la page panier :

##### PANIER

Produit	Prix	Quantité	Sous-total
 Coaching et Formation Divi	80,00€	<input type="text" value="1"/>	80,00€
<div>Code promo <input type="text"/></div> <div><input type="button" value="Appliquer le code promo"/></div> <div><input type="button" value="Mettre à jour le panier"/></div>			

##### TOTAL PANIER

Sous-total	80,00€
Total	80,00€

Le client pourra alors valider la commande qui est dans son panier et arrivera sur une page de commande avec détails de facturation, récapitulatif de la commande, choix du mode de livraison, carte de crédit:

## Détails de facturation

Prénom *	Nom *
<input type="text" value="Julien"/>	<input type="text" value="GUIARD"/>
Nom de l'entreprise (facultatif)	
<input type="text" value="Maintenance WP"/>	
URL de votre site (facultatif)	
<input type="text" value="www.maintenance-wp.fr"/>	
Pays *	
<input type="text" value="France"/>	
Numéro et nom de rue *	
<input type="text" value="35 rue Olivier de Serres"/>	
<input type="text" value="Appartement, bureau, etc. (optionnel)"/>	
Ville *	
<input type="text" value="PARIS"/>	
Code postal *	
<input type="text" value="75015"/>	
Téléphone (facultatif)	
<input type="text" value="0675390386"/>	
Adresse e-mail *	
<input type="text" value="julien@maintenance-wp.fr"/>	



## Informations complémentaires

Notes de commande (facultatif)

Vous avez des informations complémentaires à nous laisser sur votre bug, les problèmes rencontrés ou encore votre site web? Laissez nous un message ici.

## Votre commande

Produit	Total
Maintenance Business × 1	49,00€ / mois
<b>Sous-total</b>	49,00€
<b>TVA</b>	9,80€
<b>Total</b>	58,80€
<b>Totaux Récurrents</b>	
<b>Sous-total</b>	49,00€ / mois
<b>TVA</b>	9,80€ / mois
<b>Total récurrent</b>	58,80€ / mois 1er renouvellement : 21/05/2019

Carte de crédit  

Payer par carte de crédit

Numéro de carte \*

Date d'expiration \*

Cryptogramme visuel \*

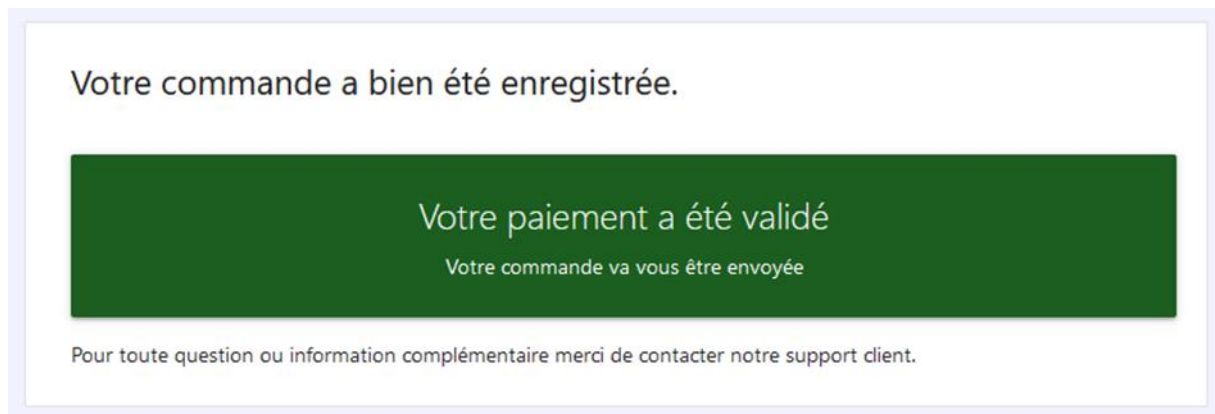
Vos données personnelles seront utilisées pour améliorer votre expérience sur le site et uniquement sur Maintenance WP.  
En savoir plus sur notre [politique de confidentialité](#).

☐ J'ai lu et j'accepte les [conditions générales](#) \*

**SOUSCRIRE**

Une fois le paiement effectué, et le paiement validé, l'utilisateur verra :





\_ Dès lors qu'un acheteur valide une commande sur la Marketplace, celle-ci vérifie les informations de paiement, et informe-le ou les vendeurs (en cas de multi-panier), qu'une commande est en attente d'acceptation.

Le marchand doit alors accepter ou refuser d'honorer la commande notifiée.

Une fois que le vendeur a vérifié le détail de la commande ainsi que la disponibilité du produit, il convient d'accepter celle-ci, l'acheteur sera alors débité de son achat.

\_ Un collaborateur ou un système logiciel vérifie que le paiement est correctement passé et que les informations de livraison sont dûment renseignées. Si tout est en ordre, la commande est enregistrée.

\_ Une confirmation de commande sera alors envoyée par email au client ainsi qu'une facture, le stock sera mis à jour.

\_ Une fois enregistrée, la commande est traitée. À ce stade, la [gestion d'entrepôt](#) prend le relai. Les produits commandés par le client sont collectés et emballés. Le colis est ensuite affranchi et expédié. Le client reçoit une notification d'expédition par e-mail, avec un lien de redirection vers une plateforme de suivi du colis.

\_ Le colis est livré soit en relai colis ou à domicile, selon le mode de livraison choisi au moment du passage de la commande. Le lendemain du jour de la livraison est le point de départ du délai de rétractation légal de quatorze jours.

\_ Le service client peut éventuellement envoyer une enquête de satisfaction par e-mail. Cela lui permet d'évaluer, et d'améliorer, sa gestion de commande.

\_ Un client peut renvoyer un produit, soit en exerçant son droit de rétractation, soit en application d'une garantie légale ou commerciale. Il est judicieux d'intégrer l'étape du retour produit, hypothétique, dans la gestion de commande.

## Javascript :

### EXPLICATIONS DE MON CODE

La page la plus importante pour ma partie est la page panier qui est une page très importante pour un site e-commerce. Elle permet aux utilisateurs de visualiser les produits qu'ils ont ajoutés à leur panier, de modifier les quantités et de supprimer des produits.

Sur la page panier, c'est en codant que j'ai choisi de faire des modifications par rapport au maquettage pour différentes raisons : plus logique selon moi, ou plus accessible pour ma réalisation pour le moment. Par exemple, au maquettage j'avais prévu un bouton « mettre à jour le panier » mais en codant je me suis rendue compte que cela était trop vague et que ce serait plus clair pour l'utilisateur d'avoir un bouton « supprimer un produit » et un bouton « modifier la quantité » à la place d'un seul « mettre à jour le panier ». J'ai également modifié « poursuivez vos achats » en « ajouter un produit ».

J'ai ajouté un bouton « valider le panier » ce qui permet de diriger sur une autre page, la page de livraison. J'ai également modifié l'ordre de mes pages et j'ai regroupé la page facturation et récapitulatif de commande car cela me semblait plus clair pour l'utilisateur. J'ai visualisé le parcours d'achat de l'utilisateur ce qui m'a permis de prendre de nouvelles décisions et de faire les modifications en codant afin de lui rendre son parcours le plus agréable et simple possible.

Pour les éléments de codage voici des détails sur mon travail :

Sur la page panier avant l'utilisation de bootstrap:

- J'ai utilisé la balise `<table>` pour afficher les produits du panier. Cette balise est adaptée à la présentation de données sous forme de tableau.
- J'ai utilisé les balises `<tr>` et `<td>` pour créer les lignes et les colonnes du tableau. Ces balises sont les balises standard pour créer des tableaux HTML.
- J'ai utilisé les attributs `id` et `name` pour identifier les éléments du tableau. Ces attributs permettent d'accéder aux éléments du tableau à partir du code JavaScript.
- J'ai voulu utiliser la balise `<strong>` pour mettre en avant le « Montant Total »
- J'ai voulu utiliser `onclick= window.location.href` pour rediriger vers une autre page.

Après l'utilisation de Bootstrap la page panier est beaucoup plus moderne et la classe `btn` a été appliquée aux boutons pour leur donner un style.

Je n'ai pas réussi à trouver comment faire le calcul des sous totaux, je reste sur zéro, je dois faire évoluer cela ainsi que l'affichage de la tva.

Je suis contente d'avoir réussi à faire fonctionner mes liens entre chaque page.

L'utilisation de Bootstrap m'a permis de moderniser chaque page même les plus simples comme vous pouvez voir ici :

AVANT juste avec HTML :

## Félicitations, votre commande est enregistrée !

Votre commande sera débitée uniquement après validation de la commande par l'artisan.

Vous recevrez un e-mail de confirmation avec les détails de votre commande qui indiquera si celle-ci est validée ou non.

[Retour à l'accueil](#)

APRES AVEC BOOTSTRAP :

## Félicitations, votre commande est enregistrée !

Votre commande sera débitée uniquement après validation de la commande par l'artisan.

Vous recevrez un e-mail de confirmation avec les détails de votre commande qui indiquera si celle-ci est validée ou non.

[Retour à l'accueil](#)

### JAVASCRIPT :

Page panier.js :

```
let total = 0; // Déclaration et initialisation de la variable à 0
const table = document.getElementById("table-id");
const tauxTVA = 0.2;

const panier = [
  {
    produit: "Nom produit 1",
    image: "image1.jpg",
    quantité: 2,
    prix: 50,
    "sous-total": 2 * 50, // Calcul du sous-total
  },
  {
    produit: "Nom produit 2",
    image: "image2.jpg",
    quantité: 1,
    prix: 40,
    "sous-total": 1 * 40, // Calcul du sous-total
  },
];

// Calcul du total
function formaterPrix(prix) {
  // Fonction pour formater le prix avec deux chiffres après la virgule
  return prix.toFixed(2);
}

// Ajout d'une variable pour stocker le montant total HT
let montantTotalHT = 0;
```

J'ai réalisé ce code JavaScript pour calculer le total HT, le total TTC et pour afficher un tableau des produits du panier avec le prix formaté.

J'ai d'abord déclaré des variables:

- total: variable pour stocker le total (initialisée à 0)
- table: variable pour stocker l'élément HTML du tableau (table-id)
- tauxTVA: variable pour le taux de TVA (0.2 pour une TVA de 20%)
- panier: tableau contenant les produits du panier (nom, image, quantité, prix, sous-total)

J'ai utilisé la fonction formaterPrix:

- Cette fonction formate le prix avec deux chiffres après la virgule.
- Elle vérifie si le prix est un nombre valide avant de le formater.

Calcul du total HT:

- La variable montantTotalHT est initialisée à 0.
- Une boucle for parcourt chaque produit du panier.
- Pour chaque produit, le prix formaté est multiplié par la quantité et ajouté à montantTotalHT.

Affichage du total HT:

- La valeur de montantTotalHT est affichée dans l'élément HTML montanttotalht.

J'ai utilisé la fonction calculerMontantTTC:

- Cette fonction calcule le montant total TTC en additionnant le montant total HT et la TVA (calculée avec le taux de TVA).

Calcul du total TTC:

- Le montant total TTC est calculé en appelant la fonction calculerMontantTTC avec montantTotalHT et tauxTVA.

Affichage des totaux:

- Le total HT et le total TTC sont affichés dans la console.

Affichage du tableau:

- Le tableau des produits du panier est affiché avec la fonction console.table.

Pour javascript ligne 11 et 18 j'ai rajouté des guillemets à « sous-total » alors que pour produit je n'en avais pas besoin, car sinon le tiret entre sous et total était interprété comme un moins.

## LES TESTS :

```
itjs > ...
QUnit.test("Le sous-total est calculé correctement pour chaque produit", function() {
  const produit1 = {
    produit: "Nom produit 1",
    quantité: 2,
    prix: 50,
  };
  const produit2 = {
    produit: "Nom produit 2",
    quantité: 1,
    prix: 40,
  };

  equal(produit1.sous-total, 100);
  equal(produit2.sous-total, 40);
});
```

Ce test QUnit vérifie si le sous-total est calculé correctement pour chaque produit :

- Le nom du test est "Le sous-total est calculé correctement pour chaque produit". Il s'agit d'une description concise de ce que le test va vérifier.
- La fonction function contient le code du test.
- Deux objets produit sont définis, chacun avec les propriétés produit, quantité et prix.
- Ces objets représentent des exemples de produits dans un panier.
- equal est une fonction d'assertion fournie par QUnit pour comparer deux valeurs.
- La première valeur est produit1.sous-total et produit2.sous-total, qui sont les valeurs réelles du sous-total calculées à partir des données.
- La deuxième valeur est 100 et 40, qui sont les valeurs attendues du sous-total pour chaque produit.

En résumé, ce test vérifie que le sous-total est calculé correctement pour deux produits différents avec des quantités et des prix différents.

Si les deux assertions sont vraies, le test est réussi. Si l'une des assertions est fausse, le test échoue et QUnit affiche un message d'erreur.

```
QUnit.test("Test qui vérifie et supprime les caractères spéciaux écrits par l'utilisateur", function() {
  $("#prix-unitaire-{{produit.id}}").val("10,2%");
  $("#calculerMontantTotalHT").change();
  equal($("#prix-unitaire-{{produit.id}}").val(), "10.2");
});
```

Ce test vérifie et supprime les caractères spéciaux du taux de TVA avant de procéder au calcul et affiche uniquement le nombre. Voici comment il fonctionne :

**1. Nom du test:** test("Les caractères spéciaux sont supprimés du taux de TVA", function() { ... })

Ce nom donne une indication claire de ce que le test va vérifier.

**2. Configuration du test:**

```
$("#prix-unitaire-{{produit.id}}").val("10,2%");
```

Cette ligne simule la saisie d'un utilisateur qui entre un taux de TVA contenant un caractère spécial (la virgule) : "10,2%".

**3. Déclenchement du calcul:**

```
$("#calculerMontantTotalHT").change();
```

Cette ligne simule une action de l'utilisateur, comme un clic sur un bouton, qui déclenche le calcul du montant total HT.

**4. Vérification du résultat:**

```
equal($("#prix-unitaire-{{produit.id}}").val(), "10.2");
```

Cette ligne compare la valeur finale du taux de TVA affiché à l'écran avec la valeur attendue "10.2". Si les deux valeurs sont identiques, le test est réussi.

En résumé, ce test s'assure que la page ne conserve pas les caractères spéciaux saisis par l'utilisateur dans le taux de TVA et affiche uniquement la valeur numérique du taux de TVA après le calcul.