

CABIN ESCAPES

PROJET Professionnel



par Roizot Victor



- # Présentation

- Qui suis-je?

- Mon parcours.

- Ma reconversion.

- Le monde du bois.





• Présentation du projet

- Site de réservation et contexte : privilégier l'artisanat, local et produit du terroir.

Cahier des charges :

- Coté utilisateur : visite du site, connexion, réservation, activités local, un plan avec les distance et les informations, et avis.
- Coté administrateur : ajout d'une location, blog des activités local à visiter ou déguster.

• Conception

➤ La réflexion : user story

Johanne B :

30 ans, informaticienne, en couple.

Caractère :

Passionné (émotionnel, inactif, secondaire) et réservé.

Motivation :

Recherche un lieu ou passer du temps en amoureux, à la découverte d'un lieu/environnement nouveau et féérique pour un week-end, et revenir régulièrement si le lieu lui plaît.

Aime les plantes et la nature.

Problème :

N'aime pas être au centre de l'attention, est réservée au grand public.

Est connectée, suit l'actualité et est joignable.

Eddy P :

45 ans, graphiste, marié 2 enfants.

Caractère :

Passionné (émotionnel, actif ; secondaire) et optimiste,

Motivation :

Recherche pour un week-end un lieu pour décompresser de la semaine et visiter une région pas trop loin de la maison.

Il a besoin d'un hébergement pour 1 à 2 nuit en rapport avec sa visite pour rester en thème dans cet état de déconnexion, avec une formule tout compris évitent la gestion (petit déjeuner.

Eventuellement pour venir une semaine lors des vacances scolaire et visité davantage ainsi que des monuments historiques.

Problème :

N'a pas de connaissance spécifique à la nature ou lieu à visiter.

La réflexion : Trello

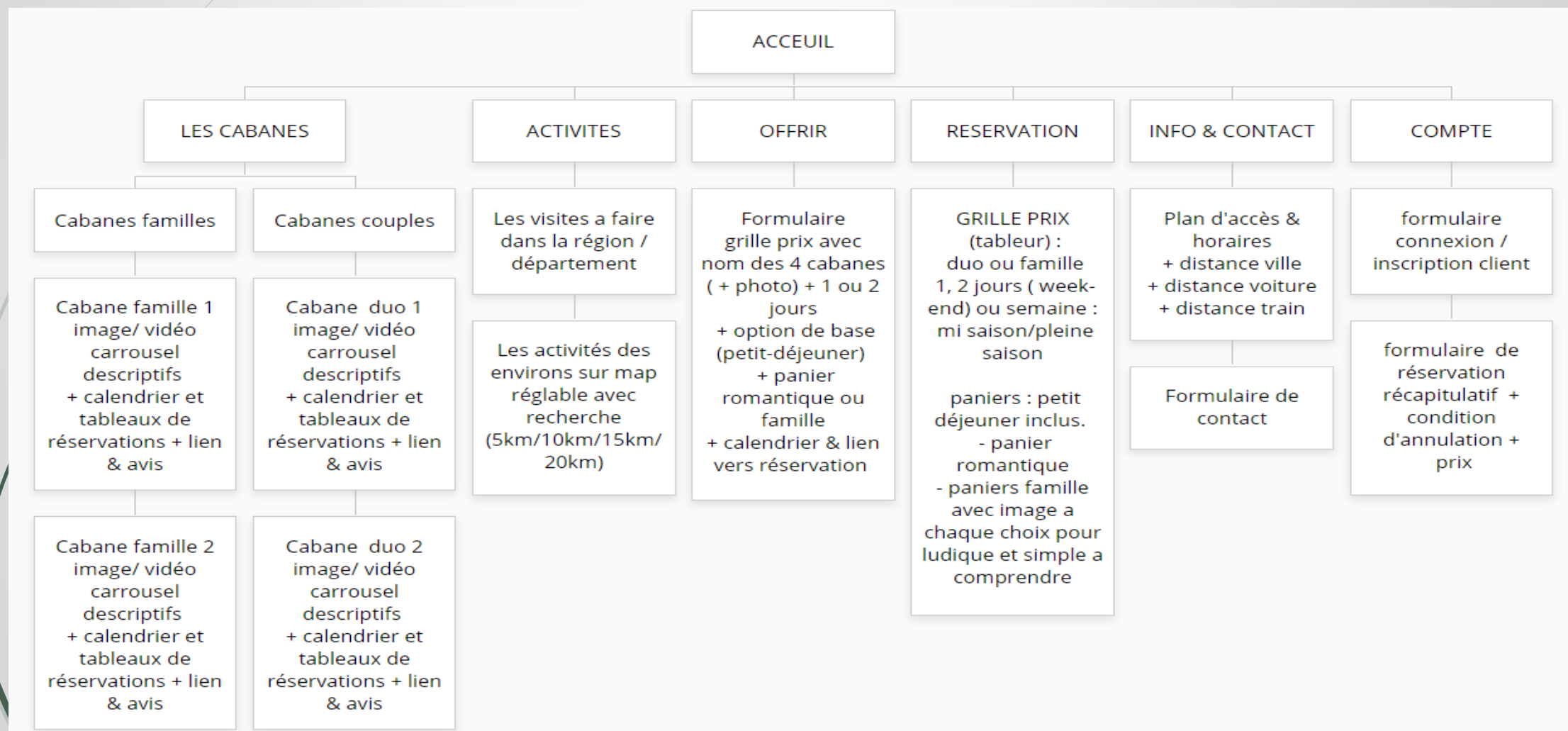
The screenshot shows a Trello board for 'Projet TP' in 'Tableau' view. The board is organized into six columns, each representing a stage in the project workflow. Cards are color-coded to indicate their status: red for 'À faire' (To do), purple for 'À modifier/améliorer' (To be modified/improved), blue for 'À confirmer' (To be confirmed), yellow for 'À corriger' (To be corrected), and green for 'Fini' (Done). The 'reflexion' column includes a due date of '29 sept. 2023' and a card count of 2. The 'maquettage' column has a card count of 2. The 'plan du Site' column has a card count of 6. The 'Back end' column has a card count of 4. The 'document' column has a card count of 1. The board also features a search bar, a 'Parcourir' button, and a 'Partager' button.

Trello Board: Projet TP

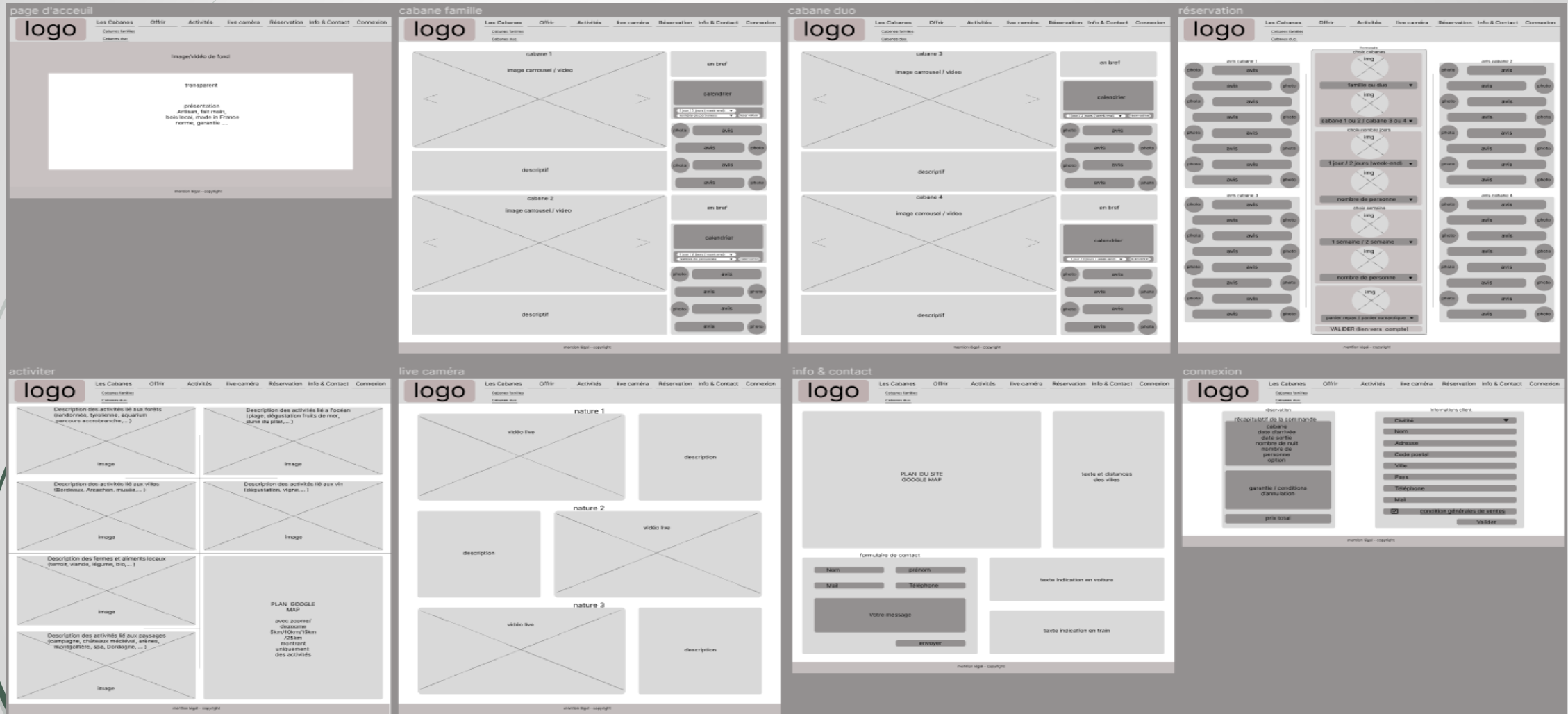
Columns:

- Légende couleur**
 - À faire
 - À modifier/améliorer
 - À confirmer
 - À corriger
 - Fini
 - + Ajouter une carte
- reflexion**
 - regarder la concurrence pour les personas
 - créer le persona (29 sept. 2023) (2)
 - + Ajouter une carte
- maquettage**
 - regarder la concurrence
 - arborescence (1)
 - zoning (2)
 - wireframe
 - prototype
 - + Ajouter une carte
- plan du Site**
 - nav
 - menu 1 - accueil
 - sous menu - cabane en famille
 - sous menu - cabane en duo
 - menu 2- activité
 - menu 3 - offrir
 - menu 4 - réservation
 - menu 5 - Info & contact
 - menu 6 - connexion / inscription
 - + Ajouter une carte
- Back end**
 - base de donnée MCD / MLD (3)
 - 1er CRUD utilisateur
 - 2eme CRUD ajout cabane uniquement
 - 3eme CRUD réservation
 - 4eme CRUD commentaire
 - + Ajouter une carte
- document**
 - résumé du projet (1)
 - dossier professionnel (1)
 - Projet TP (1)
 - + Ajouter une carte

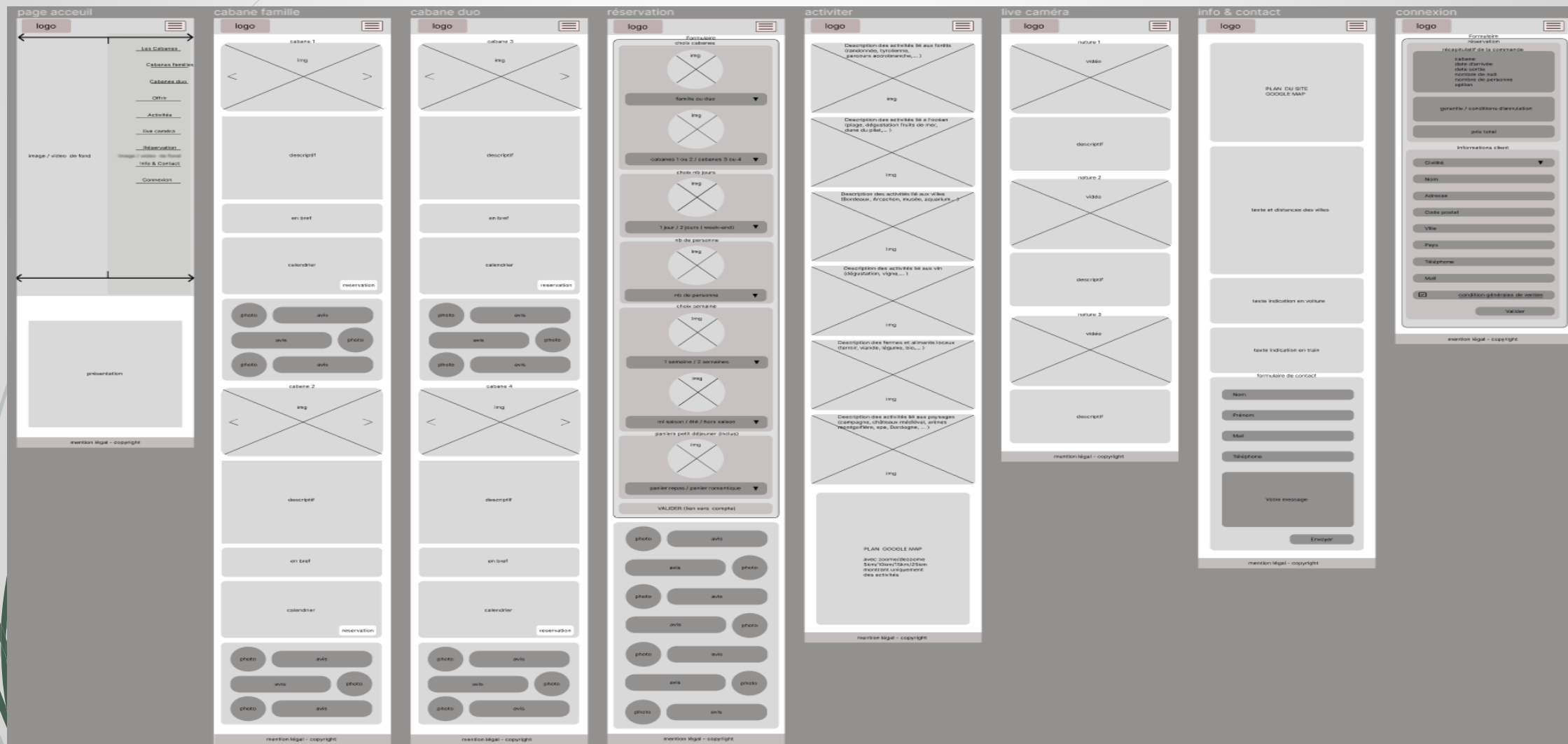
Le maquettage Front : l'arborescence avec Gloomaps :



Le maquettage Front : Figma (en desktop):



Le maquettage Front : Figma (en mobile):

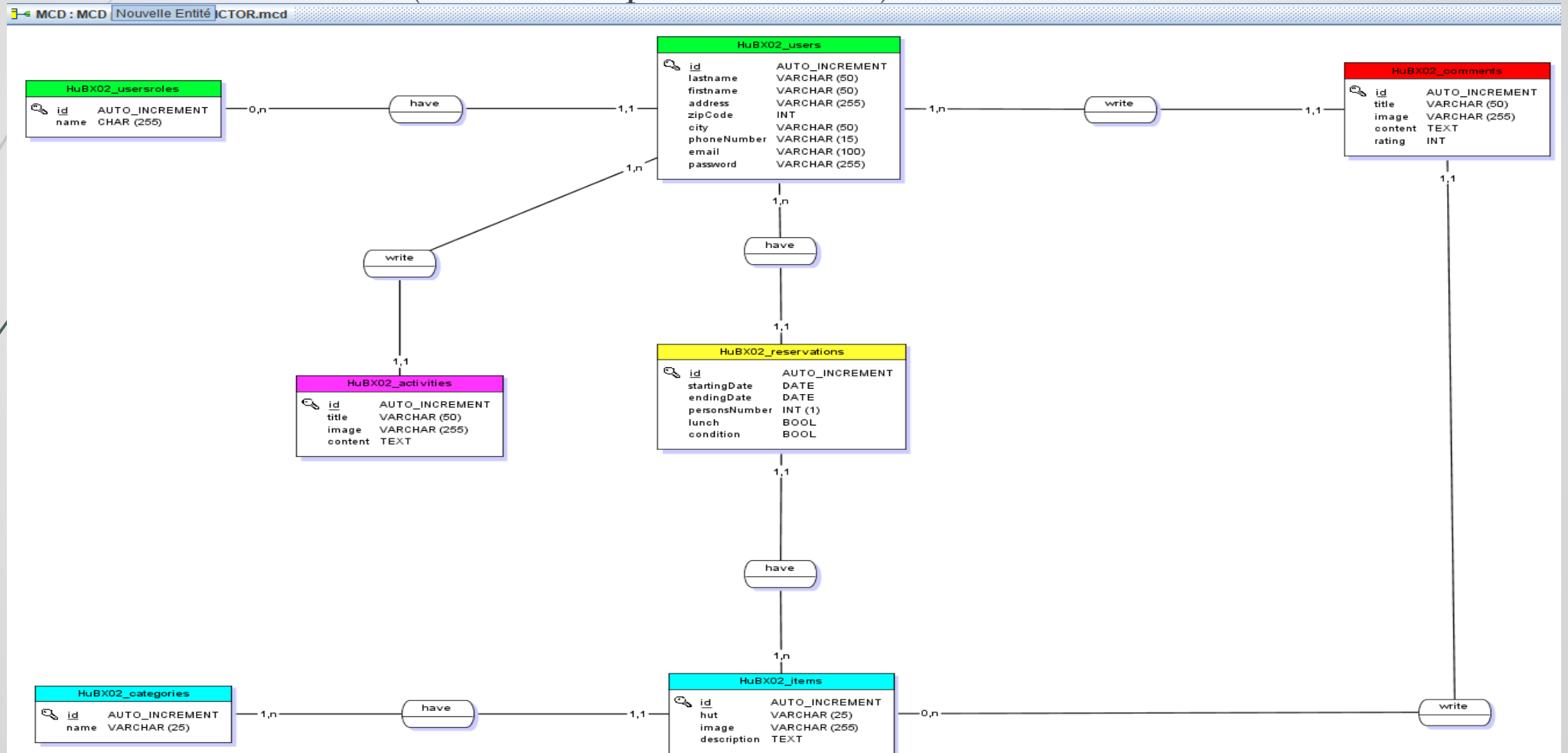


➤ La Réflexion Back avec le dictionnaire de donnée:

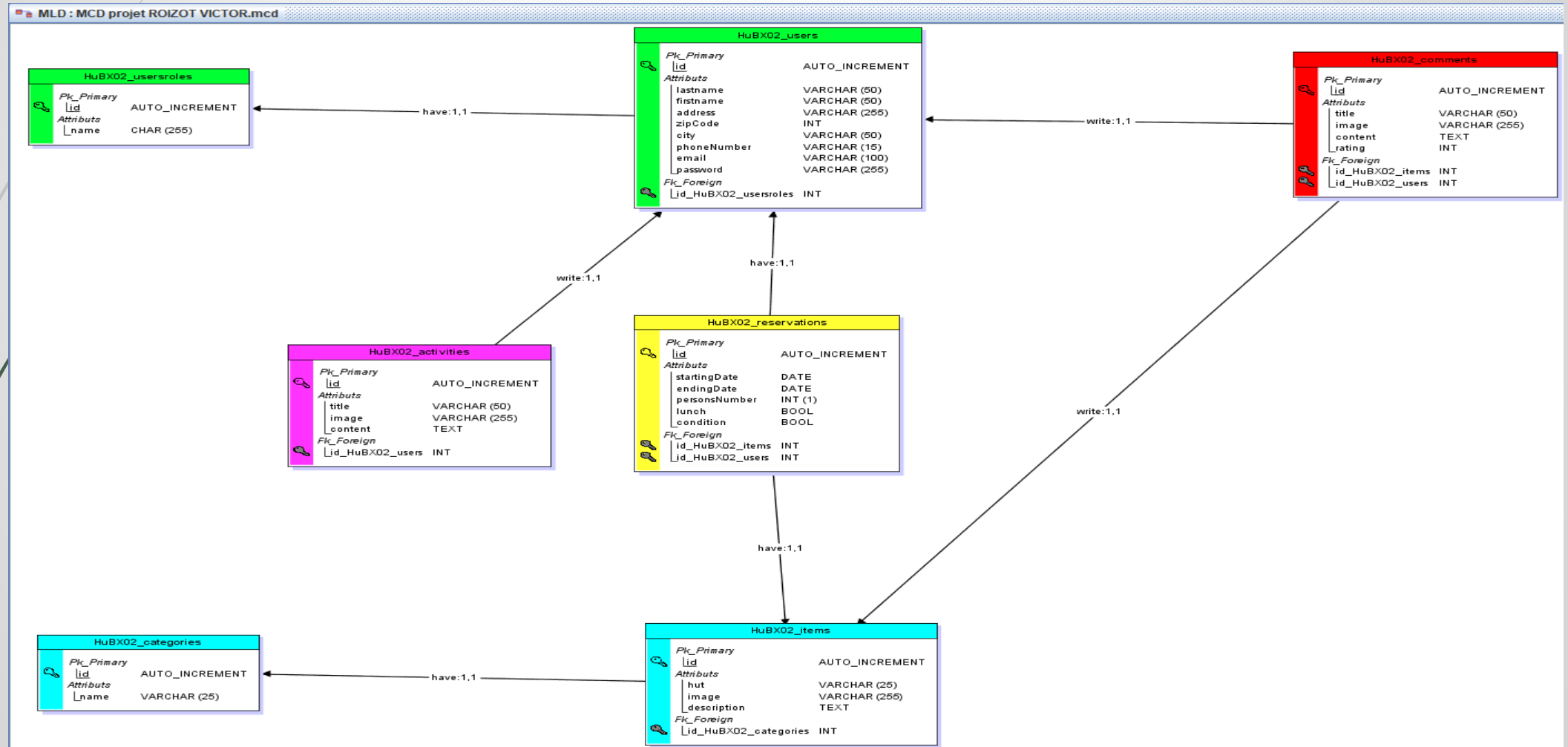
HuBX02_users					
Nom	Nom Fr	Type	Taille	Options	Description
ID		Int		Auto_increment NOT NULL	clé primaire
firstname	nom d'utilisateur	varchar	20	NOT NULL	nom de l'utilisateur qui ne doit pas dépasser 20 caractères et doit être uniquement en lettres, chiffres et caractère spéciaux
lastname	prénom d'utilisateur	varchar	20	NOT NULL	prénom de l'utilisateur qui ne doit pas dépasser 20 caractères et doit être uniquement en lettres, chiffres et caractère spéciaux
address	adresse postal de l'utilisateur	varchar	255	NOT NULL	adresse de l'utilisateur qui ne doit pas dépasser 255 caractères et doit être uniquement en lettres et chiffres
zipCode	code postal de l'utilisateur	int	10	NOT NULL	code postal de l'utilisateur qui ne doit pas dépasser 10 caractères et doit être uniquement en chiffres
city	ville	varchar	50	NOT NULL	ville de l'utilisateur qui ne doit pas dépasser 50 caractères et doit être uniquement en lettres et chiffres
phoneNumber	numéro de téléphone	varchar	15	NOT NULL	numéro de l'utilisateur, doit être uniquement en chiffres et caractère spéciaux
email	adresse mail	varchar	100	UNIQUE NOT NULL	adresse mail de l'utilisateur, ne doit pas dépasser les 100 caractères, peut comporter lettres, chiffres, underscores, tirets, @ et points
password	mot de passe	varchar	255	NOT NULL	mot de passe de l'utilisateur, ne doit pas dépasser les 255 caractères, peut comporter lettres minuscule et majuscule, chiffres, et caractère spéciaux

Le maquettage Back : Jmerisse

Le MCD (Modèle conceptuel de données) :



Le MLD (Modèle Logique de données) : avec les clés étrangères que les cardinalités ont prédéfinies



Le script généré à partir du MCD

```
SQL SQL : MCD projet ROIZOT VICTOR.mcd
#-----
#      Script MySQL.
#-----

#-----
# Table: HuBX02_categories
#-----

CREATE TABLE HuBX02_categories(
  id      Int      Auto_increment NOT NULL ,
  name    Varchar (25) NOT NULL
  ,CONSTRAINT HuBX02_categories_PK PRIMARY KEY (id)
)ENGINE=InnoDB;

#-----
# Table: HuBX02_items
#-----

CREATE TABLE HuBX02_items(
  id              Int      Auto_increment NOT NULL ,
  hut             Varchar (25) NOT NULL ,
  image           Varchar (255) NOT NULL ,
  description      Text NOT NULL ,
  id_HuBX02_categories Int NOT NULL
  ,CONSTRAINT HuBX02_items_PK PRIMARY KEY (id)
  ,CONSTRAINT HuBX02_items_HuBX02_categories_FK FOREIGN KEY (id_HuBX02_categories) REFERENCES HuBX02_categories(id)
)ENGINE=InnoDB;

#-----
# Table: HuBX02_usersroles
#-----

CREATE TABLE HuBX02_usersroles(
  id      Int      Auto_increment NOT NULL ,
  name    Char (255) NOT NULL
  ,CONSTRAINT HuBX02_usersroles_PK PRIMARY KEY (id)
)ENGINE=InnoDB;
```

• Présentation du CRUD items en PDO Avec Architecture MVC (Model View Controller)

- Create des articles :
- Model : La classe et les requêtes SQL

```
1  <?php
2  class Items
3  {
4      public int $id;
5      public string $hut;
6      public string $image;
7      public string $description;
8      public int $id_categories;
9      private $pdo;
10
11     public function __construct()
28     public function create()
29     {
30         $sql = 'INSERT INTO `hubx02_items` (`hut`, `image`, `description`, `id_categories`)
31         VALUES (:hut, :image, :description, :id_categories)';
32         $req = $this->pdo->prepare($sql);
33         $req->bindValue(':hut', $this->hut, PDO::PARAM_STR);
34         $req->bindValue(':image', $this->image, PDO::PARAM_STR);
35         $req->bindValue(':description', $this->description, PDO::PARAM_STR);
36         $req->bindValue(':id_categories', $this->id_categories, PDO::PARAM_INT);
37         return $req->execute();
38     }
```

View : Le formulaire et les variables superglobales

```
1 <section>
2   <h1>Ajouter une cabane</h1>
3
4   <?php if (isset($success)) { ?>
5       <p class="success"><?= $success ?></p>
6       <!--Vérification des erreurs liées à l'ajout des cabanes -->
7   <?php } else if (isset($errors['itemAdd'])) { ?>
8       <p class="errors"><?= $errors['itemAdd'] ?></p>
9   <?php } ?>
10
11   <div class="formContainer">
12       <form class="form" action="/ajout-item" method="POST" enctype="multipart/form-data">
13
14           <label for="image">Image de la cabane</label>
15           <input type="file" name="image" id="image">
16           <?php if (isset($errors['image'])) { ?>
17               <p class="errors"><?= $errors['image'] ?></p>
18           <?php } ?>
19
20           <label for="hut">Nom de la cabane</label>
21           <input type="text" name="hut" id="hut">
22           <?php if (isset($errors['hut'])) { ?>
23               <p class="errors"><?= $errors['hut'] ?></p>
24           <?php } ?>
25
```

➤ Controller : vérifications et conditions

Je récupère les informations
de l'administrateur sinon je suis renvoyé
à la connexion

J'instancie

Je nettoie le code et le rend conforme

```
1 <?php
2 require_once '../models/itemsModel.php';
3 require_once '../models/categoriesModel.php';
4 require_once '../utils/regex.php';
5 require_once '../utils/messages.php';
6 require_once '../utils/functions.php';
7 session_start();
8 if (empty($_SESSION['user'])) {
9     header('Location: /connexion');
10    exit();
11 }
12 $item = new Items();
13 $categories = new categories();
14 $categoriesList = $categories->getList();
15 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
16     if (!empty($_POST['hut'])) {
17         if (preg_match($regex['hut'], $_POST['hut'])) {
18             $item->hut = clean($_POST['hut']);
19         } else {
20             $errors['hut'] = ITEM_TITLE_ERROR_INVALID;
21         }
22     } else {
23         $errors['hut'] = ITEM_TITLE_ERROR_EMPTY;
24     }
```


- Les regex filtrer les chaines de caractères, la fonction clean nettoie les chaines de caractères.



```
1 $regex = [  
2     'name' => '/^[A-zÄ-ÿ]{1,}([ \'-]{1}[A-zÄ-ÿ]{1,}){0,}$/',  
3     'address' => '/^([0-9]{1,4}( ?[a-z]{1,})? )?([A-zÄ-ÿ0-9]{2,}([- ]{1}[A-zÄ-ÿ0-9]{1,}){1,})$/',  
4     'zipCode' => '/^(([0-8]{1}[0-9]{1})|(9[0-5]{1}))[0-9]{3}$/',  
5     'city' => '/^[A-zÄ-ÿ]{1,}([ \'-]{1}[A-zÄ-ÿ]{1,}){0,}$/',  
6     'phoneNumber' => '/(^0[1-79]){1}([0-9]{2}){4}$/',  
7     'password' => '/^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*\W){8,}$/',  
8     'hut' => '/^[A-zÄ-ÿ0-9]{1,}([ \'-\',A-zÄ-ÿ0-9\?!\:\\/]{1,})$/',  
9     'description' => '/(<script>|(&lt;script&gt;))/',  
10 ];
```




```
1 function clean($string)  
2 {  
3     $string = trim($string);  
4  
5     $string = strip_tags($string);  
6     return $string;  
7 }
```

Les constantes et le tableau errors :

```
1 $errors = [];  
2 // ITEMS  
3 // hut  
4 define('ITEM_TITLE_ERROR_EMPTY', 'Le titre est requis');  
5 define('ITEM_TITLE_ERROR_INVALID', 'Le titre est invalide. Il ne peut contenir que des lettres, des chiffres, des espaces, des tirets et des apostrophes');  
6  
7 // image  
8 define('ITEM_IMAGE_ERROR_EMPTY', 'L\'image est requise');  
9 define('ITEM_IMAGE_ERROR_INVALID', 'L\'image est invalide');  
10 define('ITEM_IMAGE_ERROR_EXTENSION', 'L\'image est invalide. Elle doit être au format jpg, jpeg, png, gif ou webp');  
11 define('ITEM_IMAGE_ERROR_SIZE', 'L\'image est invalide. Elle doit faire moins de 10Mo');  
12 define('ITEM_IMAGE_ERROR', 'Une erreur est survenue lors de l\'envoi de l\'image');  
13  
14 // description  
15 define('ITEM_DESCRIPTION_ERROR_EMPTY', 'Le contenu est requis');  
16 define('ITEM_DESCRIPTION_ERROR_INVALID', 'Le contenu est invalide. Il ne peut pas contenir de balises script.');
```

```
17  
18 // categories  
19 define('CATEGORIES_ERROR_EMPTY', 'La catégorie est requise');  
20 define('CATEGORIES_ERROR_INVALID', 'La catégorie est invalide');
```

- 
- Les différentes pages php contenant les vérifications, les requêtes, les messages d'erreurs sont inclus une fois grâce à la commande `require_once`.



```
1  <?php
2  require_once '../models/itemsModel.php';
3  require_once '../models/categoriesModel.php';
4  require_once '../utils/regex.php';
5  require_once '../utils/messages.php';
6  require_once '../utils/functions.php';
```

➡ Read : Méthode dans model pour la liste et un article.

```
1 public function getById()
2 {
3     $sql = 'SELECT `hut`, `image`, `description`, `name` AS `categorie`, `id_categories`
4           FROM `hubx02_items` AS `i`
5           INNER JOIN `hubx02_categories` AS `c` ON `id_categories` = `c`.`id`
6           WHERE `i`.`id` = :id';
7     $req = $this->pdo->prepare($sql);
8     $req->bindValue(':id', $this->id, PDO::PARAM_INT);
9     $req->execute();
10    return $req->fetch(PDO::FETCH_OBJ);
11 }
12
13 public function getList()
14 {
15     $sql = 'SELECT `i`.`id`, `hut`, `image`, SUBSTR(`description`, 1, 500) AS `description`, `name` AS `categorie`
16           FROM `hubx02_items` AS `i`
17           INNER JOIN `hubx02_categories` AS `c` ON `id_categories` = `c`.`id`
18 ORDER BY `i`.`id` ASC';
19     $req = $this->pdo->query($sql);
20     return $req->fetchAll(PDO::FETCH_OBJ);
21 }
```

➡ Controller : La liste et l'article.



```
1 <?php
2 require_once '../models/itemsModel.php';
3
4 session_start();
5 $item = new Items();
6 $itemsList = $item->getList();
7
8 require_once '../views/parts/header.php';
9 require_once '../views/items/itemList.php';
10 require_once '../views/parts/footer.php';
```



```
1 <?php
2 require_once '../models/itemsModel.php';
3 session_start();
4
5 $item = new Items();
6 $item->id = $_GET['id'];
7 if ($item->checkIfExists() == 0) {
8     header('Location: /list-cabanes');
9     exit;
10 }
11 $itemDetails = $item->getById();
12
13 require_once '../views/parts/header.php';
14 require_once '../views/items/viewItem.php';
15 require_once '../views/parts/footer.php';
```

View de la cabane choisi et de la liste :

```
1  <section>
2    <h1>cabanes</h1>
3    <div>
4      <?php foreach ($itemsList as $i) { ?>
5        <div>
6          
7          <div>
8            <h2><?=$i->hut ?></h2>
9            <p>
10              <b>Catégorie :</b> <?=$i->categorie ?><br>
11            </p>
12            <p>
13              <?= strip_tags($i->description) ?>...
14            </p>
15            <a class=lien href="/cabane-<?=$i->id ?>">Lire la suite</a>
16          </div>
17        </div>
18      <?php } ?>
19    </div>
20  </section>
```

```
1  <section>
2    <h1>cabanes</h1>
3    <div>
4      
5      <h1><?=$itemDetails->hut ?></h1>
6      <p>
7        <b>Catégorie :</b> <?=$itemDetails->categorie ?><br>
8      </p>
9      <p>
10        <?= strip_tags($itemDetails->description) ?>
11      </p>
12    </div>
13  </section>
```

➤ Update méthode dans model :

Nous préparons la méthode avec la ligne `$req = $this->pdo->prepare($sql) ;`

Elle sera ensuite exécutée avec la ligne `return $req->execute() ;`

Une fois que les marqueurs nominatifs seront remplis depuis le formulaire.

```
1 public function updateItem()
2 {
3     $sql = 'UPDATE `hubx02_items` SET `hut` = :hut, `image` = :image, `description` = :description, `id_categories` = :id_categories WHERE `id` = :id';
4     $req = $this->pdo->prepare($sql);
5     $req->bindValue(':hut', $this->hut, PDO::PARAM_STR);
6     $req->bindValue(':image', $this->image, PDO::PARAM_STR);
7     $req->bindValue(':description', $this->description, PDO::PARAM_STR);
8     $req->bindValue(':id_categories', $this->id_categories, PDO::PARAM_INT);
9     $req->bindValue(':id', $this->id, PDO::PARAM_INT);
10    return $req->execute();
```




plusieurs éléments nous garantissent une plus grande sécurité contre les injections SQL :

Par exemple sur la ligne `$req->bindValue('hut' , $this->hut, PDO::PARAM_STR)`

`$req->bindValue()` permet de vérifier le type de donnée.

Il est suivi de `PDO::PARAM_STR` qui indique que le type de donnée est une chaîne de caractère (string).

- **Controller :** Nous vérifions si l'administrateur a les droits, avant d'instancier plusieurs éléments et vérifier les conditions liées à la sécurité comme lors de l'étape de la création, avec les regex, la fonction clean, Sinon des messages d'erreur seront affichés.

```
1  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
2      if (isset($_POST['updateItem'])) {
3          if (!empty($_POST['hut'])) {
4              if (preg_match($regex['hut'], $_POST['hut'])) {
5                  $item->hut = clean($_POST['hut']);
6              } else {
7                  $errors['hut'] = ITEM_TITLE_ERROR_INVALID;
8              }
9          } else {
10             $errors['hut'] = ITEM_TITLE_ERROR_EMPTY;
11         }
12     }
13 }
```

```
16  if ($_SESSION['user']['id_usersRoles'] != 255) {
17      header('Location: /list-cabane');
18      exit;
19  }
20
21  $item = new Items();
22  $categories = new categories();
23  $user = new Users;
24  $user->id = $_SESSION['user']['id'];
25  $item->id = $_GET['id'];
26  $imageUploaded = true;
27
28  if ($item->checkIfExists() == 0) {
29      header('Location: /list-cabanes');
30      exit;
31  }
32
33  $itemDetails = $item->getById();
34  $userAccount = $user->getById();
35  $categoriesList = $categories->getList();
```

- View : un formulaire pour remplacer les données.

Je fait revenir les informations déjà existantes depuis la base de données avec les variables \$itemDetails.

```
8      <?php if (isset($errors['updateItem'])) { ?>
9          <p class="errors"><?= $errors['updateItem'] ?></p>
10     <?php } ?>
11     <div class="formContainer">
12         <form class="form" action="/modifier-item-<?= $item->id ?>" method="post" enctype="multipart/form-data">
13
14             <label for="image">Image de la cabane</label>
15             <input type="file" name="image" id="image">
16             <?php if (isset($errors['image'])) { ?>
17                 <p class="errors"><?= $errors['image'] ?></p>
18             <?php } ?>
19
20             <label for="hut">Nom de la cabane</label>
21             <input type="text" name="hut" id="hut" value="<?= $itemDetails->hut ?>">
22             <?php if (isset($errors['hut'])) { ?>
23                 <p class="errors"><?= $errors['hut'] ?></p>
24             <?php } ?>
```

- 
- Delete méthode et requête dans model :



```
1 public function delete()
2 {
3     $sql = 'DELETE FROM `hubx02_items` WHERE `id` = :id';
4     $req = $this->pdo->prepare($sql);
5     $req->bindValue(':id', $this->id, PDO::PARAM_INT);
6     return $req->execute();
7 }
```

- Je rajoute la suppression sous le formulaire de modification dans view.
Je fais confirmer la suppression par une fenêtre modale afin de confirmer et éviter une action humaine liée à un mauvais geste.

```
1 <div class="DivDelete">
2     <div class="containerDelete">
3         <h2>Supprimer la cabane</h2>
4
5         <button class="submitDelete" id="openModalBtn">Supprimer</button>
6     </div>
7 </div>
8
9 <div id="modalContainer">
10     <div id="modal">
11         <span id="closeModalBtn">&times;</span>
12         <p id="modalText">Êtes-vous sûr de vouloir supprimer la cabane ?</p>
13         <form class="FormBtnDelete" action="/modifier-item-<?=$item->id ?>" method="POST">
14             <button class="btnDelete" type="submit" name="delete">Supprimer</button>
15         </form>
16     </div>
17 </div>
```

- Dans le contrôleur, la suppression se situe sous la modification de l'article
- La fonction unlink() supprimera l'image qui est stockée dans le dossier assets/img/items et non dans la base de données.

```
1  if (isset($_POST['delete'])) {
2      if ($item->delete()) {
3          unlink('../assets/img/items/' . $itemDetails->image);
4          header('Location: /list-cabane');
5          exit;
6      }
7  }
8
9  $itemDetails = $item->getById();
10
11  require_once '../views/parts/header.php';
12  require_once '../views/items/updateItem.php';
13  require_once '../views/parts/footer.php';
```


- # Présentation du jeu d'essai :

- Je teste mon CRUD d'ajout de la cabane avec les messages d'erreur en desktop.

The screenshot shows a web application interface for adding a cabin. The page has a dark wood-grain background. At the top, there is a navigation bar with links: Accueil, Cabanes, Dupont, Admin, and Déconnexion. The main heading is 'Ajouter une cabane'. The form contains the following fields and error messages:

- Image de la cabane:** A file upload button labeled 'Choisir un fichier' and 'Aucun fichier choisi'. Below it, a red error message reads 'L'image est requise'.
- Nom de la cabane:** A text input field. Below it, a red error message reads 'Le titre est requis'.
- Catégorie de la cabane:** A dropdown menu with the placeholder text 'Choisissez une catégorie'. Below it, a red error message reads 'La catégorie est requise'.
- Description de la cabane:** A rich text editor with a toolbar (File, Edit, View, Insert, Format) and a 'Paragraph' dropdown. Below the editor, a red error message reads 'Le contenu est requis'.

At the bottom of the form is a 'Créer' button. The footer of the page contains the text 'Cabin Escapes' and 'Copyright 2024'.

Et en version mobile pour le succès



The image shows a mobile application interface for adding a new cabin. The background is a dark wood texture. At the top, there's a header with a small logo on the left and a hamburger menu icon on the right. The main title is "Ajouter une cabane" in white. Below it, a green message says "La cabane a bien été ajouté". The form consists of several sections: "Image de la cabane" with a file picker button labeled "Choisir un fichier" and "Aucun fichier choisi"; "Nom de la cabane" with a text input field; "Catégorie de la cabane" with a dropdown menu showing "Choisissez une catégorie"; and "Description de la cabane" with a rich text editor. The rich text editor has a menu bar with "File", "Edit", "View", "Insert", and "Format", and a toolbar with undo, redo, paragraph, bold, italic, bulleted list, numbered list, and a more options menu. The text area is empty. At the bottom of the form is a large "Créer" button. The footer contains the text "Cabin Escapes" and "Copyright 2024".

Ajouter une cabane

La cabane a bien été ajouté

Image de la cabane

Choisir un fichier Aucun fichier choisi

Nom de la cabane

Catégorie de la cabane

Choisissez une catégorie

Description de la cabane

File Edit View Insert Format


↶ ↷ Paragraph B I ☰ ☷ ☹ ☹ ☹ ☹ ...

Créer

Cabin Escapes


Copyright 2024

- La partie visible de la liste des articles à louer depuis l'onglet Cabane en format ordinateur.



AccueilCabanesDupontAdminDéconnexion

Les cabanes




La cabane du Cerf

Catégorie : Famille

La cabane du Cerf est située en début de forêt à proximité du domaine, et permet d'être au plus près de la Nature. Cette cabane dans la foree et en rodin s'intègre tellement bien dans son environnement que vous ne l'apercevrez qu'au dernier moment. Émerveillement garanti pour vos enfants lorsqu'ils découvriront l'endroit où il s'endormiront le soir ! Dès votre arrivé...

[Lire la suite](#)



La cabane des lapins


Catégorie : Couple

La cabane Lapin se trouve au coeur d'une clairière, bien cachée au milieu de la forêt, afin d'assurer une tranquillité maximale à ses occupants. Émerveillement garanti lorsque vous la découvrirez, au dernier moment ! Depuis le pas de la porte, vous aurez pour seul horizon les arbres environnants et pour seuls compagnons les écureuils et autres petits oiseaux. La cabane Lapin est idéale pour une escapade en amoureux, ...


[Lire la suite](#)

Cabin EscapesCopyright 2024

- Et la visualisation détaillée d'une cabane après avoir cliqué sur « lire la suite ».



[Accueil](#) [Cabanes](#) [Dupont](#) [Admin](#) [Déconnexion](#)



La cabane du Cerf

Catégorie : Famille

La cabane du Cerf est située en début de forêt à proximité du domaine, et permet d'être au plus près de la Nature. Cette cabane dans la forêt et en rodin s'intègre tellement bien dans son environnement que vous ne l'apercevrez qu'au dernier moment. Émerveillement garanti pour vos enfants lorsqu'ils découvriront l'endroit où il s'endormiront le soir ! Dès votre arrivée vers votre nid pour la nuit, nous vous lançons un petit défi : trouver les 10 petits cerf en bois qui se sont cachés dans votre cabane. Certains sont sculptés, d'autres dessinés ou gravés, etc. . Toute la famille pourra se lancer dans cette chasse ! Cette cabane peut accueillir jusqu'à 6 personnes, grâce à deux grands lits doubles et deux petits lits simples. Il est également possible d'y installer votre lit parapluie pour les tout-petits. La cabane du Cerf est équipée d'une grande table, de chaises et de petits transats pour les enfants. Ainsi, vous pourrez prendre votre petit-déjeuner face aux autres hôtes de ces bois : écureuils, pic-vert, geais, etc. Le petit-déjeuner est déposé le matin au pied de votre cabane par « la fée clochette ». Nous vous proposons pour le dîner des paniers repas à base de produits locaux en option. Vous pouvez réserver votre panier repas jusqu'à 24h avant votre arrivée. Le WIFI est aussi disponible si vous souhaitez rester connecté. Autre sujet d'étonnement et de découvertes : les toilettes sèches ! Mais on vous laisse le soin de venir tester sur place ces toilettes écolos-rigolotes. Un chauffage électrique permet au besoin de faire de l'intérieur de votre cabane un cocon chaud et douillet, même durant les fraîches nuits d'automne et d'hiver. Sur place, les 10 hectares de forêt seront un fabuleux terrain d'exploration et de jeux pour petits et grands.

[modifier](#)

si l'utilisateur est l'administrateur il pourra cliquer sur le bouton « modifier ma cabane » pour modifier l'article ou la supprimer comme ci-joint en format mobile.

Modifier ma cabane

Votre cabane a bien été mis à jour

Image de la cabane

Choisir un fichier | Aucun fichier choisi

Nom de la cabane

La cabane du Cerf

Catégorie de la cabane

Famille

Description de la cabane

File Edit View Insert Format

↶ ↷ Paragraph ...

La cabane du Cerf est située en début de forêt à proximité du domaine, et permet d'être au plus près de la Nature. Cette cabane dans la foree et en rodin s'intègre tellement bien dans son environnement que vous ne l'apercevrez qu'au dernier moment. Émerveillement garanti pour vos enfants lorsqu'ils découvriront l'endroit où il s'endormiront le soir ! Dès votre arrivé vers votre nid pour la nuit, nous vous lançons un petit défi : trouver les 10 petits cerf en bois qui se sont cachés dans votre cabane. Certains sont sculptés, d'autres dessinés ou gravés, etc. . Toute

p tiny

Modifier

Lorem ipsum dolor sit amet consectetur, adipisicing elit, Fuga in possimus libero fugit maiores veritatis, alias repellendus minima molestias aliquam! Cumque accusantium quod velit nobis aliquam porro a error, eius magni veritatis dolore quaerat inventore nemo ut aut corporis optio possimus quibusdam pariatur voluptas id iure libero tenetur! Illum ullam officia quo perferendis iusto nam, nobis totam! Labore, dolor reiciendis et deleniti itaque temporibus enim unde, deserunt porro totam dignissimos sint perspiciatis corporis ea quibusdam ex quo expedita cumque? Nulla ab provident aperiam tempore, deleniti dolore laudantium veritatis

Êtes-vous sûr de vouloir supprimer la cabane ?

Supprimer

Supprimer la cabane

Supprimer

Cabin Escapes
Copyright 2024

• veille sur les vulnérabilités de sécurité :

- En plus des méthodes évoqué afin protéger mon (Les préfixes HuBX02, les variables superglobales \$_POST, les fonction trim() et strip_tags(), le bindValue(), Le PDO: :PARAM_STR, les regex, le tableau d'erreur...)

Nous pouvons rajouter la fonction filter_var() pour l'email qui a la même fonctionnalité que le regex ou encore password_hash() pour hasher et chiffrer le mot de passe.



```
1 (filter_var($_POST['email'], FILTER_VALIDATE_EMAIL))  
2     $user->password = password_hash($_POST['password'], PASSWORD_DEFAULT);
```

Je vais également évoquer le fichier .htaccess : il permet de configurer les différentes pages et répertoires en réécrivant l'adresse url pour plus de sécurité.



recherche effectuée à partir d'un site anglophone

- j'ai recherché des informations pour ajouter une image de l'items dans mon dossier assets sans pour autant écraser un autre fichier qui aurait le même nom ou si la photo est utilisée deux fois.

Definition and Usage

The `uniqid()` function generates a unique ID based on the microtime (the current time in microseconds).

Que j'ai traduit par :

Définition et usage

La fonction `uniqid()` génère un identifiant unique basé sur le micro temps (l'heure actuelle en micro secondes).



• Synthèse et conclusion

Mon projet a commencé sein d'un cocon familial à quelque mois de la formation et que j'ai continué à faire murir au centre.

Mon projet n'est pour l'heure pas encore fini, en effet je souhaite ajouter une partie réservation avec un calendrier, un CRUD avec des commentaires, une page d'activités et découvertes de la région accompagnée d'une carte dynamique, ainsi qu'une dernière page contenant un formulaire de contact, une carte et des informations sur le lieu.

Mais je n'en reste pas moins fier de pouvoir présenter mon projet et montrer les compétences que j'ai acquis durant cette étape enrichissante.

J'ai hâte de poursuivre mon apprentissage et continuer à évoluer en alternance si possible, sinon m'orienter davantage sur le Back-End.



• Remerciement

Tout d'abord, je tiens à remercier Mme Margot CORBIN sans qui je n'aurais pas pu intégrer l'école de LA MANU dans cette formation de développeur web et web mobile.

Ensuite je remercie plus particulièrement et très sincèrement Johanne BADIN et Eddy PHILIPPO, mes formateurs, qui m'ont appris bien plus que de la connaissance, toujours à l'écoute et dans l'aide, sans qui ce projet n'aurait jamais été aussi abouti.

Je remercie également ma famille qui m'a tous encourager dans ma reconversion professionnelle et plus particulièrement ma femme qui m'a soutenu et a su me libérer partiellement de mes tâches quotidiennes et de la charge des enfants afin de me consacrer pleinement à ma formation.

Pour finir, j'adresse mes derniers remerciements à ma promo et aux autres apprenants, avec qui nous avons passé des bons moments comme des moments difficile mais toujours dans l'entraide et la bonne humeur.