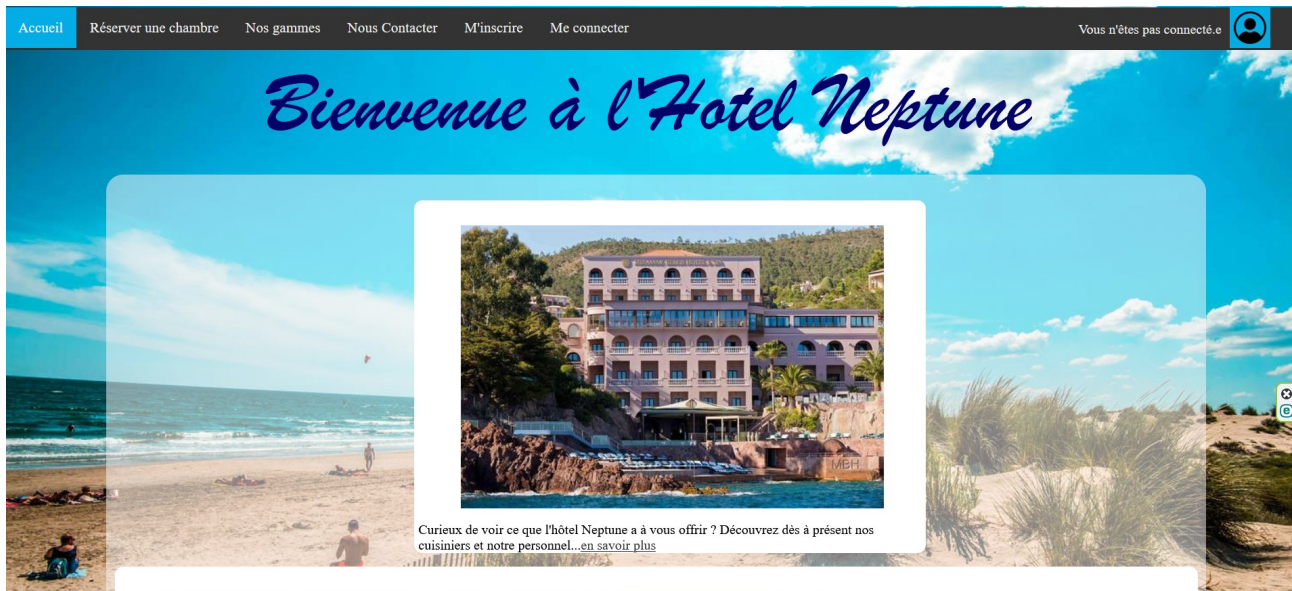


PROJET WEB

HÔTEL NEPTUNE

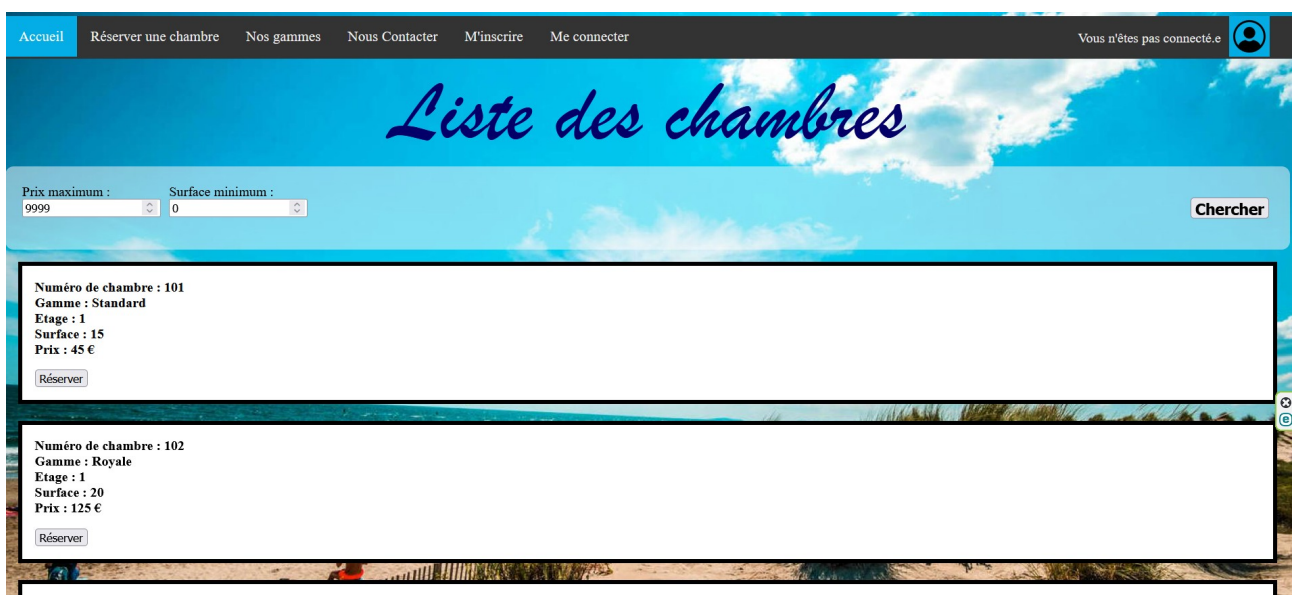
NEMO TORRES

Pour recontextualiser la mission nous considérons que nous travaillons pour un hôtel situé à Palavas, l'hôtel s'appelle l'hôtel Neptune, il joue sur une image d'hôtel en bord de plage ce qui peut être très attractif pour les touristes. Nous avons considéré que l'hôtel n'avait aucun site internet existant mais que nous n'avions pas besoin de nous occuper de la partie hébergement du site.



Voici un aperçu de la page de garde de l'hôtel. Nous avons choisi des couleurs qui parlerons le plus avec l'image que renvoie l'hôtel Neptune.

Dans notre cahier des charges nous devons absolument mettre en place une base de donnée qui a servi a stocké nos chambres et nos clients. Voici par exemple la page des chambres qui recherche les infos de la BDD pour les afficher sur notre HTML/CSS :



k?php

```
require_once __DIR__ . DIRECTORY_SEPARATOR . '../activate_session.php';
require_once __DIR__ . DIRECTORY_SEPARATOR . '../database.php';

$request = $database->prepare('SELECT * FROM rooms r WHERE r.dispo = 1');
$request->execute();

if (empty($rooms = $request->fetchAll())) {
    echo("Vous n'avez aucune chambre.");
}

usort(array: &$rooms, callback: function ($a, $b): float|int {
    return $a['num_ch'] - $b['num_ch'];
});

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_POST['search'])) {
        $request = $database->prepare("SELECT *
        FROM rooms r
        WHERE r.prix < :max_prix AND r.surface > :min_surface");
        $request->execute([
            'max_prix'=>$_POST['prix'],
            'min_surface'=>$_POST['min_surface']
        ]);
        $rooms = $request->fetchAll();
    } else if (isset($_POST['book'])) {
        header(header: 'Location: reservation/book_room.php?id=' . $_POST['book']);
    }
}
```

Ce code permet l'appel de différentes chambres qui sont compris entre deux float qui définissent le prix minimal et le prix maximal ce qui crée une requête qui cible les chambres qui vont être affichées.



Nous avons ensuite créé une page administrateur à laquelle on se connecte en entrant les identifiants d'un utilisateur précis qu'on considère comme un administrateur.

Voici par exemple le code d'une page admin qui permet d'afficher la liste des utilisateurs de l'hôtel.

```
<?php

require_once __DIR__ . DIRECTORY_SEPARATOR . '../../activate_session.php';
require_once __DIR__ . DIRECTORY_SEPARATOR . 'check_admin.php';

$request = $database->prepare('SELECT * FROM users u');
$request->execute();

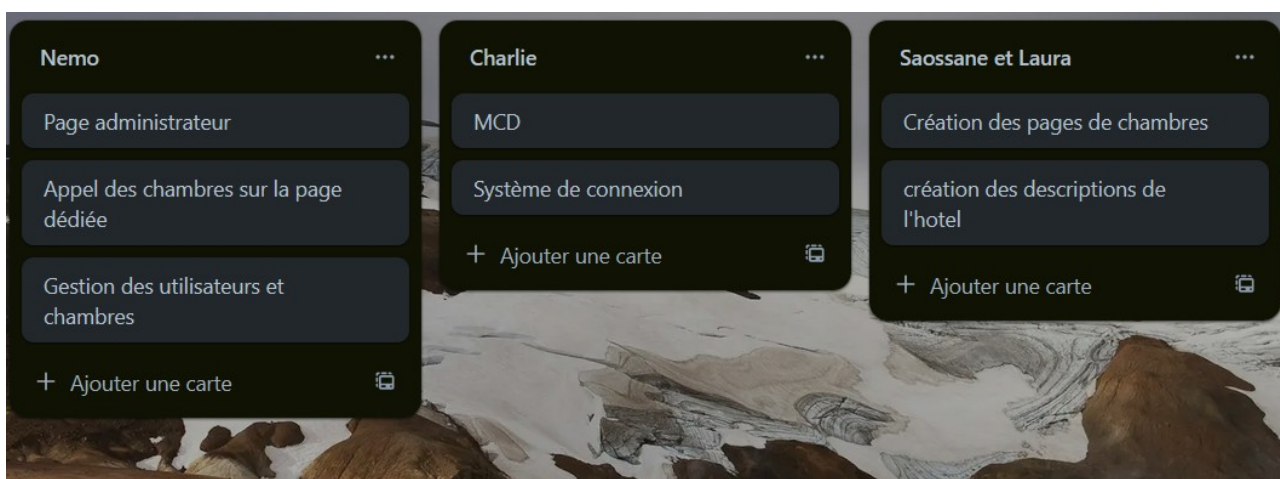
if (empty($users = $request->fetchAll())) {
    echo("Vous n'avez aucun utilisateur.");
}

usort(array: &$users, callback: function ($a, $b): float|int {
    return $a['id'] - $b['id'];
});

?>
<!DOCTYPE html>
<html lang="fr">

<head>
    <link rel="stylesheet" href="styles/pageadmin.css" />
    <link rel="stylesheet" href="styles/rooms.css" />
    <title> Neptune Hotel </title>
    <link rel="shortcut icon" href="images/Logo-Neptune-avec-rond-e1607450857665.ico" />
</head>
```

Gérer le patrimoine informatique	Répondre aux incidents et aux demandes d'assistance et d'évolution	Travailler en mode projet	Organiser son développement professionnel
Nous avons cherché des programmes tels que Dbeaver pour administrer des bases de données. Nous avons aussi mis en place un système de branche sur Github pour assurer la sauvegarde de la solution web.	Nous avons suivi un cahier des charges précis comme la nécessité de mettre une page administrateur ou une base de données.	Un Git a été mis en place pour travailler chacun de son côté je met aussi à disposition un trello représentant l'organisation du groupe.	Connecter PHP, SQL et HTML/CSS était quelque chose de nouveau à ce moment là, nous avons donc recherché les techniques pour appeler la base de données, préparer des requêtes ou supprimer des instances d'une table.



<https://github.com/FunsizedWrath/hotel-neptune>