



الجامعة الأوروبية بـفاس
EUROMED UNIVERSITY OF FES
UNIVERSITÉ EUROMED DE FÈS

Euro Mediterranean University of Fez

Ecole d'Ingénierie Digitale et
d'Intelligence Artificielle (EIDIA)



École d'Ingénierie Digitale
et d'Intelligence Artificielle

Projet de Fin de Module

Filière : 1ère année cycle ingénieur en cybersécurité

Semestre : 5

Module : Secure Web Programming

Thème :

Développement d'une Application Web de Gestion des Réservations d'un Hôtel

Soutenu le 19 /01/2025

Encadré par :
M. Amamou Ahmed

Préparé par :
M. HAKLANI Zaid
M. JABBAR Idriss

Sommaire :

I) Introduction.....	
II) LES ETAPES DE DEVELOPPEMENT DE L'APPLICATION.....	
2.1. Le developpement de système de connexion	
2.2. Le developpement de la page d'accueil de l'hôtel	
2.3. Le developpement de la page de reservation de l'hôtel	
2.4. Le developpement de la page d'administrateur de l'hôtel	
2.4.1. Le developpement de la page d'administrateur de l'hôtel	
2.4.2. Le developpement de la page d'administrateur de l'hôtel	
2.4.3. Le developpement de la page d'administrateur de l'hôtel	
2.4.4. Le developpement de la page d'administrateur de l'hôtel	
2.4.5. Le developpement de la page d'administrateur de l'hôtel	
III) HOSTING THE WEBSITE WITH APACHE2 ON LINUX.....	
IV) LES DIFFICULTES RENCONTREES LORS DU DEVELOPPEMENT DE L'APPLICATION ET LEURS SOLUTIONS.....	

I) Introduction :

L'objectif de ce projet est de concevoir et de développer une application web dédiée à la gestion des réservations pour un hôtel. Cette solution vise à simplifier la gestion des réservations et des opérations associées grâce à une interface vive et à une gestion fiable des données. Le système sera conçu en utilisant les technologies PHP, HTML, CSS et MySQL.

L'application répondra aux principaux besoins d'un hôtel en intégrant les fonctionnalités suivantes :

Un système de connexion sécurisé : permettant aux administrateurs et aux utilisateurs d'accéder à leurs espaces respectifs.

Gestion des données : possibilité d'ajouter et de supprimer des entrées dans la base de données, incluant les chambres, les profils des clients, les profils des administrateurs et les réservations.

Filtres et recherches avancées : permettant de retrouver rapidement des informations spécifiques pour améliorer l'expérience utilisateur.

Deux plateformes distinctes :

- Espace administrateur : accès à une gestion complète des fonctionnalités en arrière-plan.
- Espace utilisateur classique : accès limité, principalement à la page d'accueil et aux informations générales sur l'hôtel.

De plus, **l'application proposera un tableau de bord interactif** affichant des indicateurs clés, tels que :

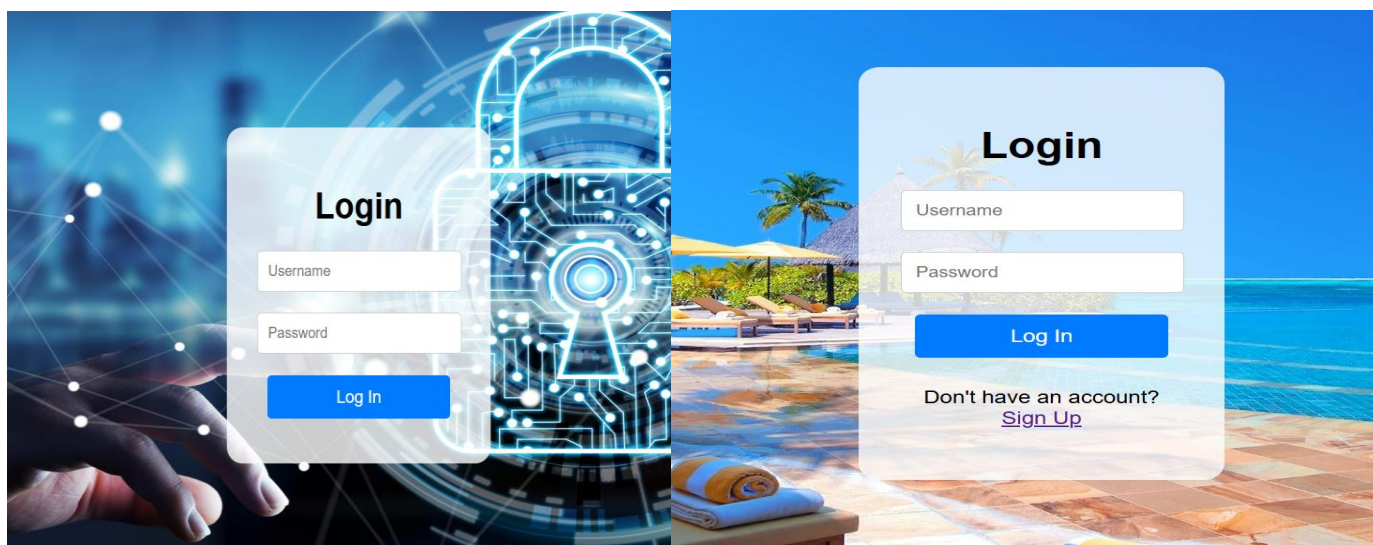
- Le nombre total de chambres disponibles, les réservations en cours et les bénéfices générés côté administrateur, ainsi que les types de chambres, la galerie de l'hôtel et la page de réservation côté client.

Des graphiques simples permettront également de visualiser les données de manière claire. En résumé, cette application simplifiera les tâches administratives tout en offrant une expérience utilisateur fluide et agréable, adaptée aux besoins d'un hôtel.

II) LES ETAPES DE DEVELOPPEMENT DE L'APLICATION :

2.1. Le developpement de système de connexion : (fichier login.php signup.php et admin_login.php)

Au début, nous avons créé deux plateformes de connexion distinctes : une pour la connexion des administrateurs à leurs espaces de gestion personnels (image à gauche) et une autre pour la connexion des clients sur la page d'accueil de l'hôtel (image à droite). Le code PHP utilisé est le même.



En parallèle, nous avons déjà créé notre base de données SQL, comprenant une table **admin_login** pour les administrateurs et une table **login** pour les clients. Ces tables permettent de stocker les données respectives une fois que les deux plateformes seront connectées à la base de données **reservation_hotel**.

Nous avons ensuite commencé à créer le fichier PHP chargé d'assurer la connexion entre notre plateforme et la base de données.

```

<?php
include('db_connection.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username']; // Get it from the form
    $password = $_POST['password'];

    // Check the username and password
    $sql = "SELECT * FROM login WHERE uname = '$username' AND pass = '$password'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // User exists, redirect to another page
        header( header: "Location: homepage.php");
    } else {
        // User doesn't exist
        echo '<div class="notification">Invalid username or password!</div>';
    }
}
?>

```

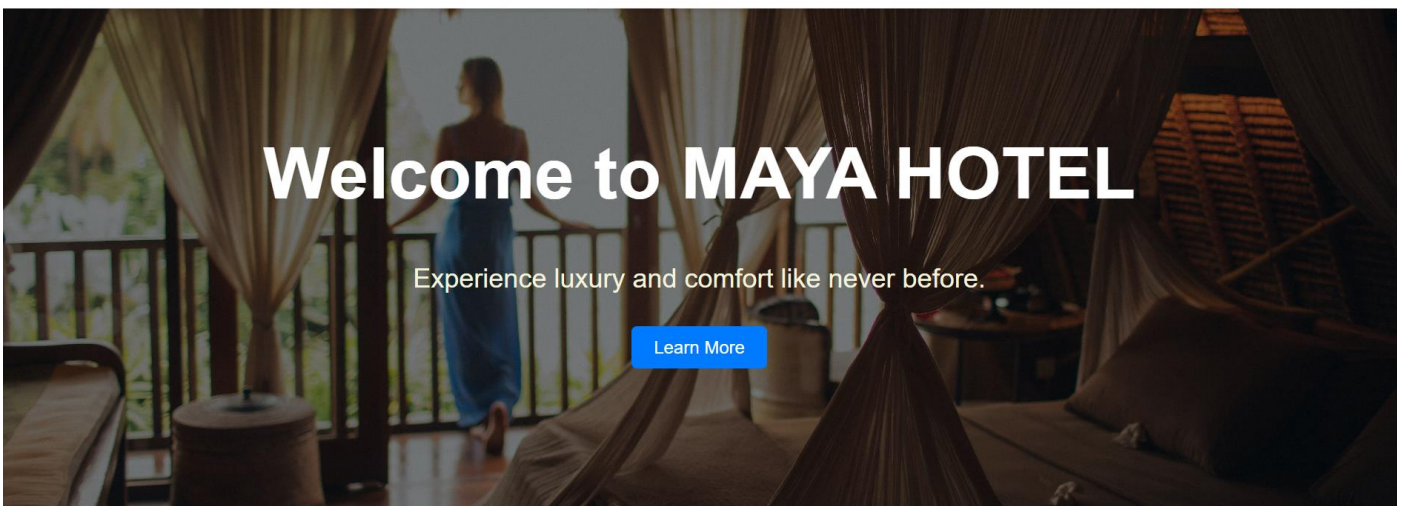
Nous avons directement ajouté, en bas de la page de connexion des utilisateurs classiques, une option permettant aux nouveaux utilisateurs de créer un compte. Une fois inscrits, ils sont automatiquement redirigés vers la page de connexion principale afin d'accéder à la page d'accueil de l'hôtel.

Quant à la page de connexion des administrateurs, elle permet d'accéder à la gestion des données récupérées depuis la page de l'hôtel en arrière-plan.

2.2. Le developpement de la page d'accueil de l'hotel : (fichier homepage.php)

MAYA HOTEL

[Home](#) [About](#) [Chairmen](#) [Gallery](#) [Rooms](#) [Contact Us](#) [Log Out](#)



La page d'accueil de l'hôtel a été entièrement construite en utilisant uniquement le front-end : le CSS pour la mise en forme de l'interface (éléments) et le HTML pour leur disposition.

Cette page comprend une barre de navigation dynamique en haut, divisant la page en sept sections : **HOME**, **ABOUT**, **CHAIRMEN**, **GALLERY**, **ROOMS**, **CONTACT US** et **LOG OUT**.

Cette organisation permet aux clients de filtrer rapidement les informations recherchées et d'améliorer leur expérience utilisateur.

```
<nav class="navbar">
  <div class="navbar-brand">MAYA <span>HOTEL</span></div>
  <ul class="nav-links">
    <li><a href="#home" class="nav-link">Home</a></li>
    <li><a href="#about" class="nav-link">About</a></li>
    <li><a href="#team" class="nav-link">Chairmen</a></li>
    <li><a href="#gallery" class="nav-link">Gallery</a></li>
    <li><a href="#rooms" class="nav-link">Rooms</a></li>
    <li><a href="#contact" class="nav-link">Contact Us</a></li>
    <li><a href="login.php" class="nav-link">Log Out</a></li>
  </ul>
</nav>
```

La quasi-totalité de la page est gérée par le CSS, permettant d'afficher la galerie et les chambres de notre hôtel de manière moderne.

Chaque chambre est présentée avec un bouton **"Book Now"**, qui est lié à la page de réservation de la chambre souhaitée.



Single Room

Starting at MAD450/night

[Book Now](#)

```
<div class="room-card">
  
  <h3><font style="..."> Single Room</font></h3>
  <p>Starting at MAD450/night</p>
  <a href="reservation.php" class="btn">Book Now</a>
</div>
```

2.3. Le developpement de la page de reservation de l'hôtel : (fichier reservation.php)

Sur la page de réservation de l'hôtel, nous avons ajouté une option de retour sur la barre de navigation dynamique, située à l'extrémité droite, au cas où le client changerait d'avis. Le client sera ainsi redirigé vers la page précédente.

MAYA HOTEL

Return

RESERVATION

PERSONAL INFORMATION

Title*

First Name

Last Name

Email

Nationality*

☒ Moroccan
 ☐ Non Moroccan

Phone Number

La page prend la forme d'un grand formulaire, créé pour recueillir les informations nécessaires des clients afin de procéder à leurs réservations. Le formulaire est construit en HTML, tandis que le CSS est utilisé pour rendre la page plus interactive et vivante pour le client. Nous avons également ajouté une simple option **HUMAN VERIFICATION** pour protéger notre site contre les robots et filtrer les demandes réelles des fausses.

Dans ce fichier, nous avons utilisé une programmation back-end en PHP pour assurer la connexion entre la page de réservation et la base de données. Le fichier PHP récupère toutes les données saisies par l'utilisateur et les insère dans la table "**roombook**" de la base de données afin de ne pas les perdre. Il veille également à ce que l'option **HUMAN VERIFICATION** fonctionne correctement.

```
$title = $conn->real_escape_string($_POST['title']);
$fname = $conn->real_escape_string($_POST['fname']);
$lname = $conn->real_escape_string($_POST['lname']);
$email = $conn->real_escape_string($_POST['email']);
$nationality = $conn->real_escape_string($_POST['nation']);
$phone = $conn->real_escape_string($_POST['phone']);
$troom = $conn->real_escape_string($_POST['troom']);
$bed = $conn->real_escape_string($_POST['bed']);
$nroom = $conn->real_escape_string($_POST['nroom']);
$meal = $conn->real_escape_string($_POST['meal']);
$cin = $conn->real_escape_string($_POST['cin']);
$cout = $conn->real_escape_string($_POST['cout']);

// Calculate the number of days between check-in and check-out
$cin_date = new DateTime($cin);
$cout_date = new DateTime($cout);
$interval = $cin_date->diff($cout_date);
$ndays = $interval->days;

// Verify human code
if ($_POST['code1'] !== $_POST['code']) {
    echo "<script>alert('Human verification failed! Please try again.');";
}
```

Une fois l'opération terminée avec succès, une notification informera l'utilisateur et il sera automatiquement redirigé vers la page d'accueil.

2.4. Le developpement de la page d'administrateur de l'hotel : (fichier home.php)

La plateforme des administrateurs de l'hôtel comprend une barre de navigation dynamique en haut et à gauche de la page. Nous avons également créé un tableau qui affiche le statut des réservations effectuées par les clients via la page de réservation. Ce processus est réalisé par un programme PHP qui collecte les informations saisies dans la table "**roombook**" par les clients et les affiche dans le tableau.

Le CSS est utilisé pour rendre l'interaction plus vivante, tandis que le HTML lie chaque élément des deux barres de navigation à leur propre page d'interaction.


```

<div class="panel">
  <table class="table">
    <thead>
      <tr>
        <th></th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td></td>
      </tr>
    </tbody>
  </table>
</div>

<?php
// Fetch all rows from the 'roombook' table
$sql = "SELECT * FROM roombook";
$result = mysqli_query($conn, $sql);

// Check if the query returned any rows
if (mysqli_num_rows($result) > 0) {
  // Loop through each row and print the data
  while ($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>" . $row['id'] . "</td>";
    echo "<td>" . $row['Title'] . "</td>";
    echo "<td>" . $row['FName'] . "</td>";
    echo "<td>" . $row['LName'] . "</td>";
    echo "<td>" . $row['National'] . "</td>";
    echo "<td>" . $row['Phone'] . "</td>";
    echo "<td>" . $row['TRoom'] . "</td>";
    echo "<td>" . $row['NRoom'] . "</td>";
  }
}

```

MAYA HOTEL

Administrator ManagementProfile ManagementRoom Management

Status

Payment

Profit

Logout

Room Booking Status

ID	Title	First Name	Last Name	Nationality	Phone	Room Type	No. of Rooms	Bed Type	Meal Type	Check In	Check Out
1	Mr.	idriess	jabbar	Moroccan	0678517026	Superior Room	2	Single	Room only	2025-01-18	2025-01-19

Commençons par les options offertes à l'administrateur sur la barre en haut :

2.4.1. Le developpement de la page d'administrateur de l'hotel : (fichier adminsetting.php)

Tout d'abord, nous avons l'option "**Administrator Management**". Nous avons créé un fichier PHP qui collecte les données de la table **admin_login** et les affiche dans un tableau, permettant ainsi de voir les informations de tous les profils administrateurs.

Ensuite, deux boutons sont placés en bas de la page pour la création d'un nouveau compte de connexion pour un profil administrateur. (Pour des raisons de sécurité, seul un administrateur existant peut créer un nouveau compte pour un autre administrateur).

```

<?php
include ('db.php');
$sql = "SELECT * FROM 'admin_login'";
$result = mysqli_query($conn, $sql);
while($row = mysqli_fetch_array($result)) {
  $id = $row['id'];
  $us = $row['username'];
  $ps = $row['pass'];
  echo "<tr>
    <td>{$id}</td>
    <td>{$us}</td>
    <td>{$ps}</td>
    <td style='text-align: right;'><a href='adminsettingdel.php?eid={$id}' class='btn'>
  </td>
  </tr>";
}
?>

```

```

<?php
if(isset($_POST['in'])) {
  $newus = $_POST['newus'];
  $newps = $_POST['newps'];
  $newsq = "INSERT INTO admin_login (username, pass) VALUES ('$newus', '$newps')";
  if(mysqli_query($conn, $newsq)) {
    sleep( seconds: 1);
  }
  header( header: "Location: adminsetting.php");
}

```


Le tableau comprend également un bouton **Delete** que nous avons créé et lié au fichier **"adminsettingdel.php"**. Ce fichier est directement connecté à la base de données et a été programmé pour récupérer l'ID d'un profil administrateur spécifique qui n'est plus valide, puis le supprimer.

MAIN MENU

Administrator ManagementProfile ManagementRoom ManagementLogout

ADMINISTRATOR Accounts

ID	User Name	Password	Remove
1	zaid	zaid2004	Delete

Add New Admin

New Administrator NameNew Password

2.4.2. Le developpement de la page d'administrateur de l'hôtel : (usersetting.php usersettingdel.php)

Ces deux fichiers sont des copies similaires des fichiers **adminsetting.php** et **adminsettingdel.php**, mais liés à une table différente (**login** au lieu de **admin_login**).

Ces fichiers permettent la gestion des profils des utilisateurs classiques.

2.4.3. Le developpement de la page d'administrateur de l'hôtel : (setting.php addroom.php roomdel.php)

Nous avons programmé un fichier PHP pour gérer le nombre de chambres disponibles dans notre hôtel, ainsi que leurs types. Ce programme back-end collecte les informations de la table **"room"** et les affiche dans un ordre croissant par date, sous forme de petites bordures décorées avec le CSS, tandis que le HTML est utilisé pour l'insertion.

Une barre de navigation à gauche permet de choisir entre deux options : ajouter une chambre (**addroom.php**) et supprimer une chambre (**roomdel.php**).

Laissant de côté le CSS et le HTML, car ce sont presque les mêmes templates qui se répètent, nous passons à la configuration du back-end.

Le fichier **addroom.php** collecte les données de la table **room** et les affiche dans un tableau classé par ID. Un autre tableau permet à l'utilisateur de choisir un type de chambre et un type de lit à ajouter à la table **room** de la base de données, ce qui permet ensuite d'afficher ces informations sur les autres tables destinées à l'affichage des chambres sur la plateforme.

Room Status
Add Room
Delete Room

ADD NEW ROOM

Please fill in the required informations below to add a new room.

Type Of Room *

Bedding Type

Add New

ROOMS INFORMATION LIST

Room ID	Room Type	Bedding
1	Superior Room	Single

```

<?php
include('db.php');
if(isset($_POST['add'])) {
    $room = $_POST['troom'];
    $bed = $_POST['bed'];
    $place = 'Free';
    $check = "SELECT * FROM room WHERE type = '$room' AND bedding = '$bed'";
    $rs = mysqli_query($conn, $check);
    $data = mysqli_fetch_array($rs, MYSQLI_NUM);
    if($data[0] > 1) {
        echo "<p>Room Already Exists</p>";
    } else {
        $sql = "INSERT INTO `room` (`type`, `bedding`, `place`) VALUES ('$room', '$bed'";
        if(mysqli_query($conn, $sql)) {
            echo "<p>New Room Added Successfully</p>";
        } else {
            echo "<p>A Problem Occurred, Please Check the system.</p>";
        }
    }
}

```

Nous avons programmé le fichier **roomdel.php** pour permettre à l'administrateur de supprimer une chambre qui n'est plus valide. Le fichier PHP parcourt les IDs de la table **room** et offre la possibilité de sélectionner l'ID de la chambre à supprimer. Une fois l'opération effectuée, l'administrateur reçoit une notification indiquant le succès ou l'échec de l'opération.

WARNING! DELETING a ROOM is IRREVERSIBLE

Please fill in the required informations below to delete a room.

Choose a room to delete

Select the Room ID *

Delete Room

Single

Double

Superior Room

```

<?php
if (isset($_POST['del'])) {
    $did = $_POST['id'];
    $sql = "DELETE FROM `room` WHERE id = '$did'";
    if (mysqli_query($conn, $sql)) {
        echo "<script>alert('Room Deleted Successfully');</script>";
        sleep( seconds: 1);
        header( header: "Location: roomdel.php");
    } else {
        echo "<script>alert('Error! Please Check the System');</script>";
        header( header: "Location: home.php");
    }
}

```

Les chambres sont également affichées en bas de la page, et nous avons configuré un bouton de retour au menu principal (**home.php**) pour permettre à l'administrateur d'annuler ses opérations à tout moment.

2.4.4. Le developpement de la page d'administrateur de l'hotel : (payement.php)

En revenant à la barre de navigation à gauche, sous **MAIN MENU**, nous allons nous concentrer uniquement sur les étapes de la création du fichier PHP, puisque le template de la plateforme de l'administrateur reste le même.

Le fichier PHP est programmé de manière à collecter certains paramètres de la table **roombook**, nécessaires pour calculer la facture, tels que le nombre de jours, etc., puis les insérer dans la table **payment**. Une fois l'opération effectuée, il les affiche dans une table contenant une colonne nommée **Total**.

Payment Details

ID	Full Name	Room Type	Bed Type	Check In	Check Out	No. of Rooms	Meal Type	Total	Print
1	idriiss jabbar	Superior Room	Single	2025-01-18	2025-01-19	2	Room only	MAD1300	Print

```

<table class="table">
    $re = $stmt->get_result();

    if ($re->num_rows > 0) {
        while ($row = mysqli_fetch_array($re)) {
            $id = $row['id'];
            echo "<tr>
                <td>{$row['id']}</td>
                <td>{$row['title']} {$row['fname']} {$row['lname']}</td>
                <td>{$row['troom']}</td>
                <td>{$row['tbed']}</td>
                <td>{$row['cin']}</td>
                <td>{$row['cout']}</td>
                <td>{$row['nroom']}</td>
                <td>{$row['meal']}</td>
                <td>MAD{$row['fintot']}</td>
                <td style='...'><a href='print.php?pid=$id' class='btn'>Print</a></td>
            </tr>";
        }
    }

    $insertSql = "INSERT INTO payment (id, fname, lname,
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

    $stmt = $conn->prepare($insertSql);
    $stmt->bind_param(
        types: "issssisssdi",
        &var1: $row['id'], // ID
        &....: $row['FName'], // First name
        $row['LName'], // Last name
        $row['TRoom'], // Room type
        $row['Bed'], // Bed type
        $row['NRoom'], // Number of rooms
        $row['cin'], // Check-in
        $row['cout'], // Check-out
    );
    $stmt->execute();
}

```

```
// Fetch data from roombook table
$sql = "SELECT id, FName, LName, TRoom, Bed, cin, cout, NRoom, Meal, nodays FROM roombook";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Loop through each row
    while ($row = $result->fetch_assoc()) {
        // Check if the record already exists in the payment table using the unique id
        $checkSql = "SELECT id FROM payment WHERE id = ?";
        $checkStmt = $conn->prepare($checkSql);
        $checkStmt->bind_param( types: "i", &var1: $row['id']);
        $checkStmt->execute();
        $checkStmt->store_result();

        if ($checkStmt->num_rows > 0) {
            // Skip the insertion if the record already exists
            continue;
        }
    }
}
```

Cette colonne affiche le total de la facture du client en MAD, calculé selon des tarifs définis dans le code PHP. Nous avons également ajouté un bouton **Print** lié au fichier **print.php**. En cliquant sur ce bouton, l'utilisateur est redirigé vers la page **invoice**.


Sur cette page, un code PHP collecte les données nécessaires de la table **payment**, comme le total, etc., et les affiche dans un tableau sous forme de facture à imprimer.

```
<th>Item</th>
<th>No. of Days</th>
<th>Rate</th>
<th>Quantity</th>
<th>Price</th>
</tr>
<thead>
<tbody>
</tbody>
</table>

<td><?php echo htmlspecialchars($meal); ?></td>
<td><?php echo htmlspecialchars($noofdays); ?></td>
<td><?php echo number_format( num: $fintot / $noofdays, deci
<td>1</td>
<td><?php echo number_format($fintot, decimals: 2); ?></td>
```

INVOICE

MAYA HOTEL,
DOWNHILL Road,
MOROCCO,
MOROCCO
(+212) 678517026



Date

19/01/2025

Item	No. of Days	Rate	Quantity	Price
Room only	1	1,300.00	1	1,300.00
Total				1,300.00

Cette facture contient également des informations sur l'hôtel, ainsi qu'un logo de l'hôtel, affiché avec le HTML et organisé avec le CSS.

2.4.5. Le developpement de la page d'administrateur de l'hotel : (profit.php)

Le fichier **profit.php** est un petit programme qui collecte certaines informations nécessaires de la table **payment** à l'aide d'un code PHP et les affiche dans un tableau, servant de rappel des réservations effectuées ainsi que du montant gagné à partir de celles-ci. Sa principale fonctionnalité est l'affichage de la somme totale des profits générés par les réservations en temps réel.

Profit Details

ID	Full Name
1	idriss jabbar

Total Profit: MAD1,300.00

```
$query = "SELECT * FROM payment";
$result = mysqli_query($conn, $query);
$tot = 0; // Initialize total profit

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_array($result)) {
        $profit = $row['fintot'];
        $tot += $profit;
    }
}

<tr>
    <td><?php echo $row['id']; ?></td>
    <td><?php echo $row['title'] . ' ' . $row['fname'] .
    <td><?php echo $row['cin']; ?></td>
    <td><?php echo $row['cout']; ?></td>
    <td>MAD<?php echo number_format($row['fintot'], 2,
</tr>
<?php
```

IV) HOSTING THE WEBSITE WITH APACHE2 ON LINUX :

Dans cette partie, nous allons partager les étapes principales pour hoster un website sur un moteur de recherche come firefox.

En premier, nous allons déposer notre dossier qui contient tous les fichiers PHP sur notre machine virtuelle.

Déposer le dossier contenant tous les fichiers sur Ubuntu sur Home par exemple

Nous alons ensuite installer php-mbstring avec la commande **#sudo apt install php-mbstring**

Installer web server apache2 avec la commande **#sudo apt install apache2 php php-gd php-zip php-mysql mysql-server**

On tape la commande **cd /etc/apache2/sites-available/**

Using vim to edit text file within the terminal. On l'installe avec :

#sudo apt install vim

We want to change the default website file to tell it the path of our website wich is home and the name of our folder

On tape **# sudo vim 000-default.conf**

Find document root and edit it with, **/home/www/G_reserrv/utilisateur** (path of your folder)

Then click on **escape key + :wq** wich stand for write and quite.

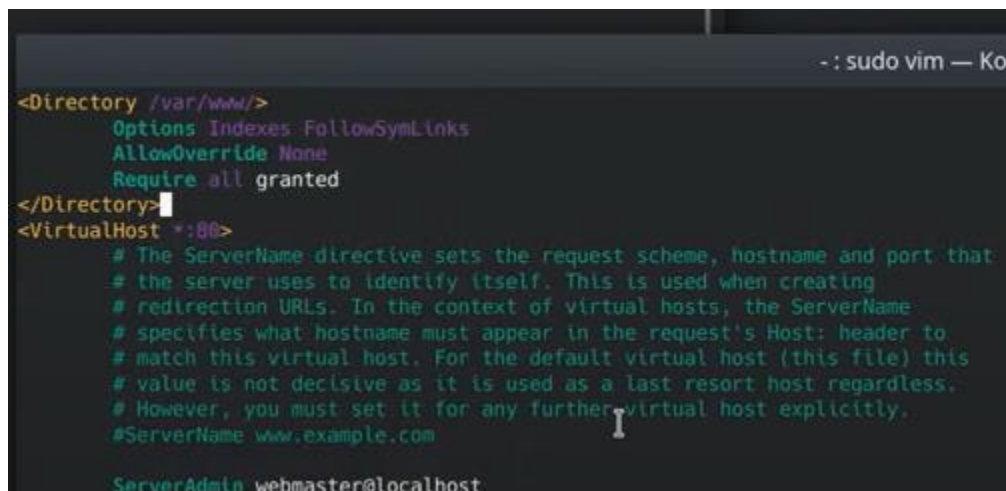
In the terminal, while opening the **/etc/apache2/ path**. Wirte **#vim appache2.conf**

We need to search the directory word, so write **espace key + /directory**

Copy the last part

comeback inside sites-available path and open the old file with #sudo vim 000-default.conf

Past the part you just copied in the begining like this :



```
- : sudo vim — Kor
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com
    ServerAdmin webmaster@localhost
```

And write the path your write earlier on document root with adding a slash in the end of the path.

Restart apache2 web server with **#sudo service apache2 restart**

Open FireFox browser and write localhost, your website must appear in front of you, now all that it is left is to give it your domaine.

Write on the browser phpmyadmin.net and download php my admin.

Cut the folder and paste the file inside your website folder at home with php files. Rename it phpmyadmin.

Come back to the terminal now, and write **#sudo mysql -u root**

Write **#use mysql;**

Now write:

```
mysql> create user 'mysqllocal'@'localhost' identified by 'localpasswd';
Query OK, 0 rows affected (0.04 sec)

mysql> select user, authentication_string from user;
+-----+-----+
| user                | authentication_string |
+-----+-----+
| mysql.infoschema    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| mysql.session       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| mysql.sys           | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| mysqllocal          | $A$005$Q!kyffZ-      |
| root                | E[xYY.dlFK5.lQUV7E5YlgqVWKIsLpo/GNh/Da.K4MuecWI4 |
+-----+-----+
6 rows in set (0.00 sec)

mysql> grant all privileges on *.* to 'mysqllocal'@'localhost';
Query OK, 0 rows affected (0.04 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.18-0ubuntu0.19.10.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

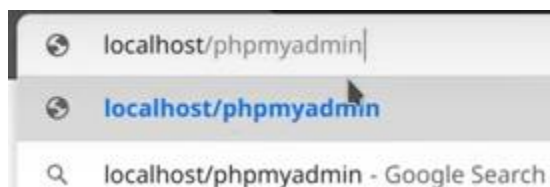
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> alter user 'mysqllocal'@'localhost' identified with mysql_native_password by 'localpasswd';
Query OK, 0 rows affected (0.04 sec)
```

And write exit to exit.

Now write:



And enter the root and pass you had chosen.

Once inside your database, to eliminate the warnings down the site.




```

/etc/apache2/sites-available$ sudo apt install php-mbstring
Done

... Done
packages will be installed:

es will be installed:
string
alled, 0 to remove and 0 not upgraded.
chives.
539 kB of additional disk space will be used.
[Y/n]
ubuntu.com/ubuntu eoan-updates/universe amd64 php7.3-mbstring amd64 7
ubuntu.com/ubuntu eoan/universe amd64 php-mbstring all 2:7.3+69ubuntu
8 kB/s)
selected package php7.3-mbstring.
5549 files and directories currently installed.)
php7.3-mbstring_7.3.11-0ubuntu0.19.10.1_amd64.deb ...
g (7.3.11-0ubuntu0.19.10.1) ...
selected package php-mbstring.
php-mbstring_2%3a7.3+69ubuntu2_all.deb ...
2:7.3+69ubuntu2) ...
ng (7.3.11-0ubuntu0.19.10.1) ...

c/php/7.3/mods-available/mbstring.ini with new version
2:7.3+69ubuntu2) ...
libapache2-mod-php7.3 (7.3.11-0ubuntu0.19.10.1) ...
/etc/apache2/sites-available$ sudo service apache2 restart
/etc/apache2/sites-available$

```

Write those commands.

And create an empty file named tmp inside your phpmyadmin folder.

Open that file path on the terminal and write : **#sudo chmod -R 777 tmp/**

The warnings must have disappeared by now on php my admin.

Let's go to the part when we host our local website using apache.

Write : **#sudo vim /etc/hosts**

Then :

```

127.0.0.1    localhost
127.0.1.1    leosilver-laptop
127.0.0.1    mysi

```

next to the ip address with added of localhost, write the name of your domain like mayahotel.com, so that when we write our website name on the browser it redirects us internally to this ip address which is localhost. **Escape key+ :wq**

Now go back to :

```

/etc/apache2/sites-available$ sudo cp 000-default.conf mysite.me.conf
/etc/apache2/sites-available$ ls
-ssl.conf mysite.me.conf
/etc/apache2/sites-available$ sudo vim mysite.me.conf

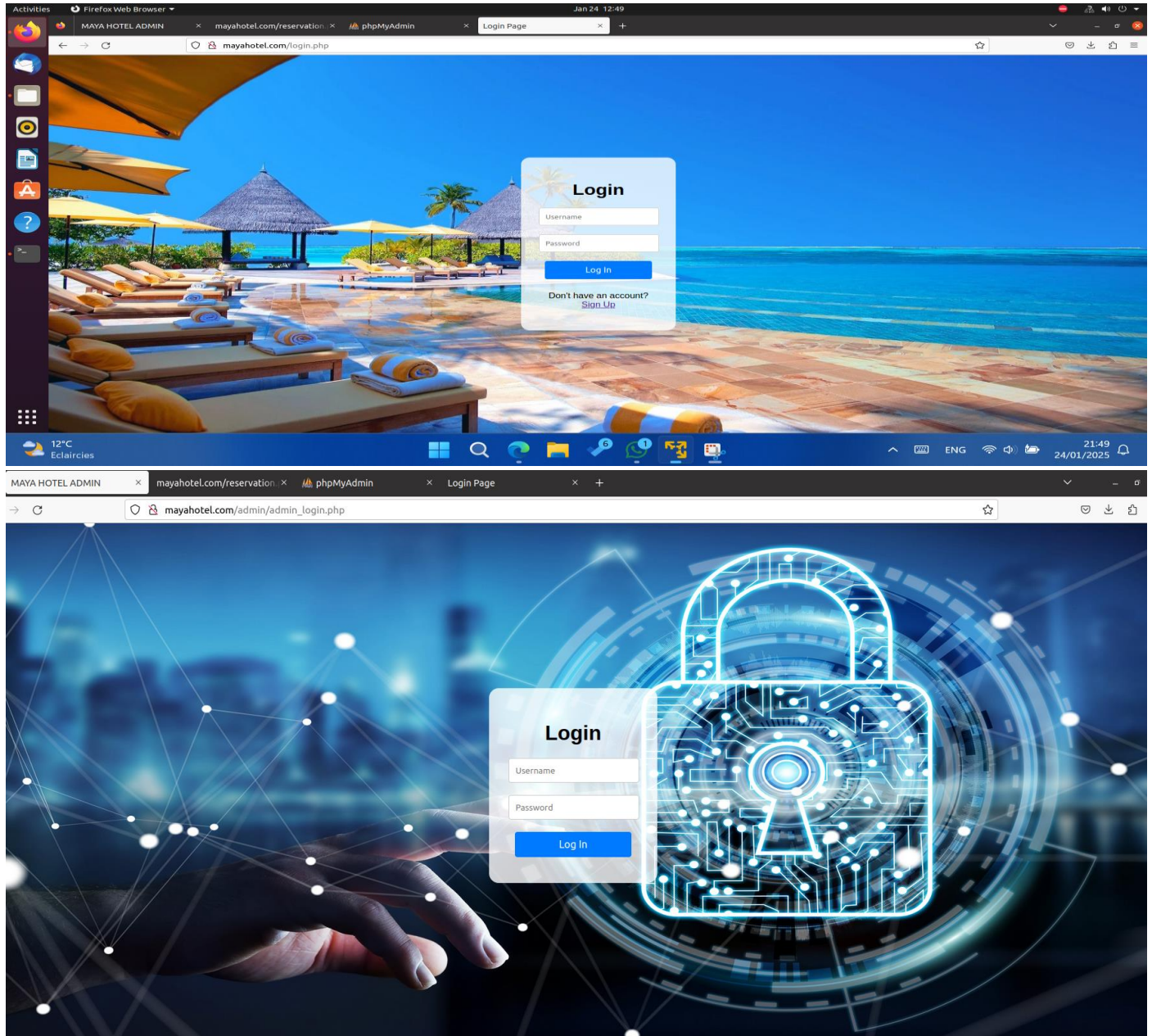
```

Instead of mysite.me.conf, you should write the name you gave to your website.

And we need to write the server name to tell the the domain name identified for this folder. :wq

```
sudo a2ensite mysite.me.conf  
sudo service apache2 restart
```

After those two commands, your website must be able to appear correctly.



III) LES DIFFICULTES RENCONTREES LORS DU DEVELOPPEMENT DE L'APPLICATION ET LEURS SOLUTIONS :

Au cours de notre travail, nous avons rencontré diverses lacunes, tant du côté de la gestion des intervalles serrés de temps que du côté de l'informatique et de la programmation.

Étant donné que c'était notre premier projet en développement web, nous manquions d'expérience. De plus, nous n'avions pas beaucoup de connaissances et de techniques pour accomplir nos tâches de manière efficace, ce qui nous a fait perdre du temps et de l'énergie, surtout en ce qui concerne les différentes formes de requêtes utilisables en PHP, chacune ayant son propre cas d'utilisation spécifique.

Nous avons rencontré un problème majeur lors de la liaison de nos codes PHP pour effectuer l'insertion ou la collecte des données avec notre base de données. Voici des exemples spécifiques des problèmes rencontrés :

Connexion à la base de données :

Nous avons rencontré des difficultés pour configurer une connexion stable à la base de données **MySQL**, principalement en raison d'une mauvaise configuration initiale des paramètres de connexion (**serveur, nom d'utilisateur, mot de passe, nom de la base**). La solution consistait à vérifier et à utiliser les bonnes informations de connexion, tout en ajoutant des messages d'erreur clairs pour diagnostiquer facilement les problèmes de connexion.

Mise en page générale et impression :

Obtenir une mise en page propre et adaptée à l'impression, avec une taille fixe pour le contenu, a nécessité plusieurs ajustements. Les propriétés comme **box-shadow, padding et overflow** ont été testées à plusieurs reprises avant de parvenir à un résultat satisfaisant.

Gestion des doublons dans la table payment :

Problème : Les données du tableau roombook étaient insérées plusieurs fois dans le tableau payment, créant des doublons à cause de l'absence de vérification préalable avant l'insertion.

La solution consistait à mettre en place un contrôle avec une requête **SELECT** pour vérifier si un enregistrement avec le même identifiant unique (**id**) existe déjà dans la table **payment**. Si l'enregistrement existe, l'insertion est alors ignorée.

Problèmes de compatibilité PHP/HTML :

L'intégration des variables PHP (**\$meal, \$noofdays, \$fintot**) dans les tableaux a été un défi, car il fallait s'assurer que les valeurs s'affichent correctement et respectent le format requis (par exemple : l'affichage des nombres avec deux décimales). Toute erreur dans le code PHP causait des dysfonctionnements dans l'affichage.

Cohérence du design et styles CSS :

Nous avons rencontré des difficultés pour appliquer un style cohérent à tous les éléments, notamment les tableaux **inventory** et **balance**. Les différences dans l'alignement des colonnes et le format des bordures m'ont obligé à ajuster manuellement les styles CSS des templates, tels que les paddings, les marges et les alignements.

Calcul des coûts :

Les calculs des coûts (loyer de la chambre, coût du lit, coût des repas) d'un client à un autre étaient initialement absents, car les valeurs n'étaient pas correctement lues à partir de la table **roombook**.

Ordre et gestion des styles :

Avec un grand nombre de styles spécifiques (par exemple, pour les en-têtes, les tableaux ou les sections **header** et **article**), nous avons eu des difficultés à organiser le CSS de manière claire. Cela a conduit à plusieurs itérations pour supprimer ou simplifier des styles inutiles.

Redirection infinie vers login.php :

Lorsque j'ai ajouté une redirection directe vers **login.php** dans le fichier **homepage.php** pour m'assurer que l'utilisateur passe par la page de connexion avant d'accéder à la page d'accueil, cela a causé une boucle infinie. Même après avoir essayé de me connecter et d'accéder à **homepage.php**, le site restait bloqué sur **login.php**.

Pour résoudre ce problème, j'ai utilisé une vérification de session (`$_SESSION['user_logged_in']`) pour distinguer les utilisateurs connectés des utilisateurs non connectés. Cela a permis de rediriger les utilisateurs non connectés vers **login.php**, tout en laissant les utilisateurs connectés accéder à **homepage.php**.

Gestion des sessions :

Je ne savais pas comment démarrer ou manipuler correctement les sessions pour suivre l'état de connexion de l'utilisateur. Pour cela, j'ai utilisé la fonction `session_start()` au début de chaque fichier nécessitant une vérification d'authentification. J'ai également défini un indicateur (`$_SESSION['user_logged_in']`) une fois que l'utilisateur s'est authentifié avec succès.

Gestion des erreurs de connexion :

Au départ, aucun message d'erreur n'était affiché lorsque l'utilisateur saisisait des identifiants incorrects. Cela rendait difficile pour l'utilisateur de comprendre pourquoi la connexion échouait.

Solution : Nous avons implémenté une notification visuelle qui affiche en premier plan un message en cas d'échec de la connexion, tel que "Nom d'utilisateur ou mot de passe incorrect"

Affichage dynamique des données :

Au départ, les données de réservation ne s'affichaient pas sur la page "**Room Booking**". Cela était dû à l'absence de mécanisme pour récupérer dynamiquement les informations de la base de données en fonction de l'action de l'utilisateur. Pour résoudre ce problème, nous avons mis en place un système basé sur des paramètres **GET** (`?page=bookroom&rid=...`) afin de charger dynamiquement les détails de réservation.

La barre de navigation masquait le contenu de la page :

La barre de navigation était définie avec la position fixe (**position: fixed;**), ce qui la plaçait au-dessus des autres éléments sans ajuster l'espace en haut de la page. Pour résoudre cela, j'ai ajouté un espace (**padding-top**) au corps principal de la page pour compenser la hauteur de la barre de navigation. Cela a permis de maintenir la visibilité du contenu sous la barre fixe.

V) CONCLUSION :

Au cours du développement de cette plateforme, nous avons surmonté plusieurs défis techniques liés à la gestion des données, à la conception de l'interface utilisateur / administrateur et à la configuration du backend. La mise en place de la connexion à la base de données MySQL a nécessité une attention particulière pour garantir les échanges des informations. De plus, la gestion des sessions utilisateurs et des redirections a permis d'assurer une navigation sécurisée. En matière de design, les ajustements des styles CSS ont été essentiels pour garantir une expérience utilisateur interactive et agréable. Les difficultés rencontrées (bien que nombreuses) ont été résolues grâce à des solutions ciblées. Le dernier qui nous a simplifié les codes et optimiser les processus. En fin de compte, ce projet nous a permis d'acquérir une expérience précieuse en développement web et d'améliorer nos compétences en design d'interfaces et en programmation HTML, CSS et PHP.