

```

fichier Etudiant.php:
<?php
require_once 'db_connect.php';

class Etudiant {
    private $conn;
    private $table_name = "etudiant_infos";

    public $id;
    public $matricule;
    public $nom;
    public $prenom;
    public $image_path;
    public $classe;
    public $email;
    public $email_parent;
    public $date_naissance;
    public $montant_a_payer;
    public $nom_parent;
    public $solvabilite;

    public function __construct($db) {
        $this->conn = $db;
    }

    public function read() {
        $query = "SELECT * FROM " . $this->table_name . " ORDER BY id
ASC";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
        return $stmt;
    }

    public function readOne() {
        $query = "SELECT * FROM " . $this->table_name . " WHERE matricule
= :matricule LIMIT 0,1";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(":matricule", $this->matricule);
        $stmt->execute();
        return $stmt->fetch(PDO::FETCH_ASSOC);
    }

    public function create() {
        $query = "INSERT INTO " . $this->table_name . " SET
matricule=:matricule, nom=:nom, prenom=:prenom, image_path=:image_path,
classe=:classe, email=:email, email_parent=:email_parent,
date_naissance=:date_naissance, nom_parent=:nom_parent,
montant_a_payer=:montant_a_payer, solvabilite=:solvabilite";
        $stmt = $this->conn->prepare($query);

        $this->matricule=htmlspecialchars(strip_tags($this->matricule));
        $this->nom=htmlspecialchars(strip_tags($this->nom));
        $this->prenom=htmlspecialchars(strip_tags($this->prenom));
    }
}

```

```

        $this->image_path=htmlspecialchars(strip_tags($this-
>image_path));
        $this->classe=htmlspecialchars(strip_tags($this->classe));
        $this->email=htmlspecialchars(strip_tags($this->email));
        $this->email_parent=htmlspecialchars(strip_tags($this-
>email_parent));
        $this->nom_parent=htmlspecialchars(strip_tags($this-
>nom_parent));
        $this->montant_a_payer=htmlspecialchars(strip_tags($this-
>montant_a_payer));
        $this->date_naissance=htmlspecialchars(strip_tags($this-
>date_naissance));
        $this->solvabilite=htmlspecialchars(strip_tags($this-
>solvabilite));

        $stmt->bindParam(":matricule", $this->matricule);
        $stmt->bindParam(":nom", $this->nom);
        $stmt->bindParam(":prenom", $this->prenom);
        $stmt->bindParam(":image_path", $this->image_path);
        $stmt->bindParam(":classe", $this->classe);
        $stmt->bindParam(":email", $this->email);
        $stmt->bindParam(":email_parent", $this->email_parent);
        $stmt->bindParam(":date_naissance", $this->date_naissance);
        $stmt->bindParam(":montant_a_payer", $this->montant_a_payer);
        $stmt->bindParam(":nom_parent", $this->nom_parent);
        $stmt->bindParam(":solvabilite", $this->solvabilite);

        if($stmt->execute()) {
            return true;
        }

        return false;
    }

    public function update() {
        $query = "UPDATE " . $this->table_name . " SET
matricule=:matricule, nom=:nom, prenom=:prenom, image_path=:image_path,
classe=:classe, email=:email, email_parent=:email_parent,
date_naissance=:date_naissance, nom_parent=:nom_parent,
montant_a_payer=:montant_a_payer, solvabilite=:solvabilite WHERE id=:id";
        $stmt = $this->conn->prepare($query);

        $this->matricule=htmlspecialchars(strip_tags($this->matricule));
        $this->nom=htmlspecialchars(strip_tags($this->nom));
        $this->prenom=htmlspecialchars(strip_tags($this->prenom));
        $this->image_path=htmlspecialchars(strip_tags($this-
>image_path));
        $this->classe=htmlspecialchars(strip_tags($this->classe));
        $this->email=htmlspecialchars(strip_tags($this->email));
        $this->email_parent=htmlspecialchars(strip_tags($this-
>email_parent));
        $this->date_naissance=htmlspecialchars(strip_tags($this-
>date_naissance));
    }
}

```

```

        $this->nom_parent=htmlspecialchars(strip_tags($this-
>nom_parent));
        $this->montant_a_payer=htmlspecialchars(strip_tags($this-
>montant_a_payer));
        $this->solvabilite=htmlspecialchars(strip_tags($this-
>solvabilite));
        $this->id=htmlspecialchars(strip_tags($this->id));

        $stmt->bindParam(":matricule", $this->matricule);
        $stmt->bindParam(":nom", $this->nom);
        $stmt->bindParam(":prenom", $this->prenom);
        $stmt->bindParam(":image_path", $this->image_path);
        $stmt->bindParam(":classe", $this->classe);
        $stmt->bindParam(":email", $this->email);
        $stmt->bindParam(":email_parent", $this->email_parent);
        $stmt->bindParam(":date_naissance", $this->date_naissance);
        $stmt->bindParam(":montant_a_payer", $this->montant_a_payer);
        $stmt->bindParam(":nom_parent", $this->nom_parent);
        $stmt->bindParam(":solvabilite", $this->solvabilite);
        $stmt->bindParam(":id", $this->id);

        if($stmt->execute()) {
            return true;
        }

        return false;
    }

    public function delete() {
        $query = "DELETE FROM " . $this->table_name . " WHERE id=:id";
        $stmt = $this->conn->prepare($query);

        $this->id=htmlspecialchars(strip_tags($this->id));

        $stmt->bindParam(":id", $this->id);

        if($stmt->execute()) {
            return true;
        }

        return false;
    }

    public function search($column, $value) {
        $query = "SELECT * FROM " . $this->table_name . " WHERE $column
LIKE :value ORDER BY id ASC";
        $stmt = $this->conn->prepare($query);

        $value = "%$value%";
        $stmt->bindParam(":value", $value);

        $stmt->execute();
        return $stmt;
    }
}

```

```

}

?>

***** fichier
EtudiantListe.php
<?php
require_once 'classes/db_connect.php';
require_once 'classes/Etudiant.php';

$database = new Database();
$db = $database->getConnection();
$etudiant = new Etudiant($db);
$stmt = $etudiant->read();
$num = $stmt->rowCount();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Dashboard - CampusFlow</title>
    <style>
        /* --- Style général --- */
        body {
            margin: 0;
            font-family: Arial, sans-serif;
            display: flex;
            min-height: 100vh;
            background-color: #f4f4f9;
        }

        /* --- Barre horizontale supérieure --- */
        .top-bar {
            background-color: rgb(108, 108, 108); /* Couleur claire */
            color: white; /* Changer la couleur du texte en blanc */
            width: 100%;
            padding: 10px 20px;
            display: flex;
            align-items: center;
            position: fixed;
            top: 0;
            height: 60px;
            z-index: 1000;
        }

        .top-bar .logo {
            font-size: 1.5em;
            font-weight: bold;
            display: flex;
            align-items: center;
        }
    </style>

```

```
.top-bar .logo img {
    width: 40px;
    height: 100%;
    border-radius: 05%;
    margin-right: 10px;
}

/* --- Barre latérale gauche --- */
.side-bar {
    background-color: #4a4a4a; /* Couleur sombre */
    color: white;
    width: 210px;
    padding: 20px 10px;
    position: fixed;
    top: 0;
    bottom: 0;
    height: 100vh;
    z-index: 1001;
}
.top-bar .logo img {
    margin-left: 260px; /* Ajuste la valeur selon tes besoins */
}
.admin-section {
    text-align: center;
    margin-bottom: 20px;
}
.admin-section .admin-icon {
    width: 60px;
    height: 60px;
    border-radius: 50%;
}
.admin-section .admin-name {
    margin-top: 10px;
    font-weight: bold;
}
.divider {
    border: 0;
    border-top: 1px solid #777;
    margin: 15px 0;
}
.menu {
    list-style: none;
    padding: 0;
}
.menu li {
    margin: 10px 0;
}
```

```
.menu a {
    text-decoration: none;
    color: white;
    display: flex;
    align-items: center;
    padding: 10px;
    border-radius: 5px;
    transition: background-color 0.3s ease;
}

.menu a:hover {
    background-color: #5a5a5a;
}

.menu-icon {
    width: 20px;
    height: 20px;
    margin-right: 10px;
}

.dropdown-menu {
    list-style: none;
    padding-left: 20px;
    display: none;
}

.dropdown-menu a {
    color: #bbb;
}

.dropdown-menu a:hover {
    color: white;
}

/* --- Contenu principal --- */
.main-content {
    margin-left: 250px;
    margin-right: 10px;
    margin-top: 80px;
    padding: 20px;
    flex-grow: 1;
    text-align: center;
}

.main-content img {
    width: 150px; /* Agrandir l'emoji */
    margin-bottom: 20px;
}

.main-content h2 {
    margin: 10px 0;
    font-size: 1.5em;
}
```

```
.main-content p {
    margin-bottom: 20px;
    color: #555;
}

.main-content button {
    padding: 10px 20px;
    background-color: #4a90e2;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.main-content button:hover {
    background-color: #357abd;
}

.main table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    font-size: 14px;
}

.main table th, .main table td {
    padding: 10px;
    text-align: center;
}

.main table th {
    background-color: #66a6ff;
    color: #ffffff;
    position: sticky;
    top: 0;
    z-index: 5;
    border: 1px solid #ddd; /* Ajouter des bordures à l'en-tête */
}

.main table img {
    width: 40px; /* Largeur de l'image */
    height: 40px; /* Hauteur de l'image */
    border-radius: 50%; /* Bordures arrondies pour un cercle */
    object-fit: cover; /* Ajuster l'image pour qu'elle soit bien
contenue */
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /* Optionnel : un
ombrage léger pour un effet esthétique */
}

.main button {
    border: none;
```

```
padding: 5px 10px;
border-radius: 25px;
font-size: 12px;
font-weight: bold;
cursor: pointer;
transition: transform 0.3s, box-shadow 0.3s;
margin: 2px;
}

.main button:hover {
    transform: translateY(-1px);
    box-shadow: 0 3px 8px rgba(0, 0, 0, 0.2);
}

.main .btn-add {
    background: linear-gradient(145deg, #66a6ff, #89f7fe);
    width: 200px;
}

.main .btn-edit {
    background: linear-gradient(145deg, #4caf50, #81c784);
}

.main .btn-delete {
    background: linear-gradient(145deg, #f44336, #e57373);
}

.main .btn-info {
    background: linear-gradient(145deg, #ff9800, #ffb74d);
}

.card {
    background: #ffffff;
    border-radius: 15px;
    padding: 20px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
    margin-top: 20px;
}

/* Modal de détails */
.modal {
    display: none;
    position: fixed;
    z-index: 100;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    overflow: auto;
    background-color: rgba(0,0,0,0.4);
    backdrop-filter: blur(5px);
}

.modal-content {
```

```
background-color: #fefefe;
margin: 10% auto;
padding: 20px;
border-radius: 10px;
width: 80%;
max-width: 600px;
box-shadow: 0 4px 6px rgba(0,0,0,0.1);
animation: slide-down 0.3s ease;
}

@keyframes slide-down {
    from { transform: translateY(-50px); opacity: 0; }
    to { transform: translateY(0); opacity: 1; }
}

.modal-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    border-bottom: 1px solid #ddd;
    padding-bottom: 10px;
}

.modal-header h2 {
    margin: 0;
    color: #333;
}

.close {
    color: #aaa;
    font-size: 28px;
    font-weight: bold;
    cursor: pointer;
}

.close:hover {
    color: #000;
}

.modal-body {
    display: grid;
    grid-template-columns: 1fr 2fr;
    gap: 15px;
    margin-top: 15px;
}

.modal-body img {
    max-width: 200px;
    border-radius: 10px;
    box-shadow: 0 4px 6px rgba(0,0,0,0.1);
}

.modal-body .details {
    display: grid;
```

```
        grid-template-columns: auto 1fr;
        gap: 10px;
    }

.modal-body .details strong {
    color: #666;
}

/* Style du card pour le tableau */
.table-card {
    background-color: #fff; /* Couleur de fond blanche */
    border-radius: 10px; /* Bords arrondis */
    padding: 20px; /* Espace intérieur */
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1); /* Ombre */
    margin: 20px auto; /* Centre le card */
    max-width: 100%; /* Limite la largeur du card */
}

/* Styles pour le tableau */
.main-content table {
    width: 100%; /* Occupe toute la largeur du card */
    border-collapse: collapse; /* Evite les doubles bordures */
    font-size: 12px; /* Reduit la taille de la police */
}

.main-content table th, .main-content table td {
    border: none; /* Enlève les bordures */
    padding: 5px; /* Reduit l'espace intérieur des cellules */
    text-align: center; /* Centre le contenu des cellules */
}

/* Arrondir l'image de l'étudiant */
.main-content table img {
    width: 40px; /* Largeur de l'image */
    height: 40px; /* Hauteur de l'image */
    border-radius: 50%; /* Arrondi l'image */
    object-fit: cover; /* Garde le ratio de l'image */
}

/* Aligner les boutons horizontalement */
.action-buttons {
    display: flex; /* Utilise flexbox pour l'alignement */
    gap: 5px; /* Espace entre les boutons */
}

.add-button-container {
    text-align: left;
    margin-bottom: 20px;
}

/* Reduire la hauteur des lignes et alterner les couleurs */
.main-content table tr:nth-child(even) {
    background-color: #f9f9f9; /* Couleur grise claire pour les lignes paires */
```

```

        }

    .main-content table tr:nth-child(odd) {
        background-color: #ffffff; /* Couleur blanche pour les lignes
impaires */
    }

    /* Style pour le card de recherche */
    .search-card {
        background-color: #fff;
        border-radius: 10px;
        padding: 20px;
        box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
        margin-bottom: 20px;
        display: flex;
        align-items: center;
        justify-content: space-between;
    }

    .search-card select, .search-card input {
        padding: 10px;
        border: 1px solid #ddd;
        border-radius: 5px;
        margin-right: 10px;
    }

    .search-card input {
        flex-grow: 1;
    }

```

</style>

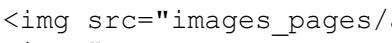
</head>

<body>

<!-- Barre latérale gauche -->

<aside class="side-bar">

<div class="admin-section">



<p class="admin-name">Administrateur</p>

</div>

<hr class="divider">

<ul class="menu">



<p>Etudiant</p>

<ul class="dropdown-menu">

Liste des

Etudiants

Ajouter un

étudiant


```

        <a href="#" class="dropdown-toggle">
             Finances
        </a>
        <ul class="dropdown-menu">
            <li><a href="VersementListe.php">Liste des
versements</a></li>
            <li><a href="versement.php">Effectuer un
versement</a></li>
        </ul>
    </li>
    <li>
        <a href="graph.php">
             Graphiques
        </a>
    </li>
    <li>
        <a href="settings.php">
             Paramètres
        </a>
    </li>
    <li>
        <a href="logout.php">
             Déconnexion
        </a>
    </li>
</ul>
</aside>


<main class="main-content">
    <div class="search-card">
        <select id="search-column">
            <option value="id">ID</option>
            <option value="matricule">Matricule</option>
            <option value="nom">Nom</option>
            <option value="prenom">Prénom</option>
            <option value="classe">Classe</option>
            <option value="email">Email</option>
            <option value="nom_parent">Nom Parent</option>
            <option value="email_parent">Email Parent</option>
            <option value="date_naissance">Date de Naissance</option>
            <option value="solvabilite">Solubilité</option>
        </select>
        <input type="text" id="search-value"
placeholder="Rechercher..." onkeyup="filterTable()">
    </div>
    <div class="table-card">
        <div class="add-button-container">
            <button class="btn-add"
onclick="window.location.href='ajouter_etudiant.php'">

```

```

        Ajouter un Étudiant
    </button>
</div>
<table id="student-table">
    <tr style="background-color: #ffffff;">
        <th>ID</th>
        <th>Matricule</th>
        <th>Image</th>
        <th>Nom</th>
        <th>Prénom</th>
        <th>Classe</th>
        <th>Email</th>
        <th>Nom Parent</th>
        <th>Email Parent</th>
        <th>Date de Naissance</th>
        <th>Montant à payer</th>
        <th>Solubilité</th>
        <th>Actions</th>
    </tr>
<tbody id="student-table-body">
    <?php
    if ($num > 0) {
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            extract($row);
    ?>
        <tr>
            <td><?php echo $id; ?></td>
            <td><?php echo $matricule; ?></td>
            <td>
                <img src='photos/<?php echo
$image_path; ?>' alt='Photo'>
            </td>
            <td><?php echo $nom; ?></td>
            <td><?php echo $prenom; ?></td>
            <td><?php echo $classe; ?></td>
            <td><?php echo $email; ?></td>
            <td><?php echo $nom_parent; ?></td>
            <td><?php echo $email_parent; ?></td>
            <td><?php echo $date_naissance; ?></td>
            <td><?php echo $montant_a_payer; ?></td>
            <td><?php echo $solvabilite; ?></td>
            <td class="action-buttons">
                <a
                    href="modifier_etudiant.php?id=<?php echo $id; ?>">
                    
                </a>
                <a
                    href="supprimer_etudiant.php?id=<?php echo $id; ?>" onclick="return
confirm('Êtes-vous sûr de vouloir supprimer cet étudiant?');">
                    
                </a>
            </td>
        </tr>
    }
}

```

```

        <a href="#" onclick="showStudentDetails(
            '<?php echo $id; ?>',
            '<?php echo $matricule; ?>',
            '<?php echo $image_path; ?>',
            '<?php echo $nom; ?>',
            '<?php echo $prenom; ?>',
            '<?php echo $classe; ?>',
            '<?php echo $email; ?>',
            '<?php echo $date_naissance; ?>',
            '<?php echo $nom_parent; ?>',
            '<?php echo $email_parent; ?>',
            '<?php echo $solvabilite; ?>'
        )">
            
        </a>
    </td>
</tr>
<?php
}
} else {
?>
<tr>
    <td colspan="12">Aucun étudiant trouvé</td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</main>

<!-- Modal pour les détails de l'étudiant --&gt;
&lt;div id="studentModal" class="modal"&gt;
    &lt;div class="modal-content"&gt;
        &lt;div class="modal-header"&gt;
            &lt;h2&gt;Détails de l'Étudiant&lt;/h2&gt;
            &lt;span class="close" onclick="closeModal()">&times;</span>
        </div>
        <div class="modal-body">
            <img id="modalStudentImage" src="" alt="Photo de
l'étudiant">
            <div class="details">
                <strong>Nom:</strong> <span id="modalNom"></span>
                <strong>Prénom:</strong> <span
id="modalPrenom"></span>
                <strong>Matricule:</strong> <span
id="modalMatricule"></span>
                <strong>Classe:</strong> <span
id="modalClasse"></span>
                <strong>Email:</strong> <span id="modalEmail"></span>
                <strong>Date de Naissance:</strong> <span
id="modalDateNaissance"></span>
            </div>
        </div>
    </div>
</div>

```

```

                <strong>Nom Parent:</strong> <span
id="modalNomParent"></span>
                <strong>Email Parent:</strong> <span
id="modalEmailParent"></span>
                <strong>Solvabilite:</strong> <span
id="modalsolvabilite"></span>
            </div>
        </div>
    </div>

<script>
    // Gestion des menus déroulants
    document.querySelectorAll('.dropdown-toggle').forEach(toggle => {
        toggle.addEventListener('click', (e) => {
            e.preventDefault();
            const menu = toggle.nextElementSibling;
            menu.style.display = menu.style.display === 'block' ?
'none' : 'block';
        });
    });

    // Redirection vers la page des étudiants
    function redirectToEtudiants() {
        window.location.href = "EtudiantListe.php";
    }

    function showStudentDetails(id, matricule, imagePath, nom,
prenom, classe, email, dateNaissance, nomParent, emailParent,
solvabilite) {
        document.getElementById('modalStudentImage').src = 'photos/' +
imagePath;
        document.getElementById('modalNom').textContent = nom;
        document.getElementById('modalPrenom').textContent = prenom;
        document.getElementById('modalMatricule').textContent =
matricule;
        document.getElementById('modalClasse').textContent = classe;
        document.getElementById('modalEmail').textContent = email;
        document.getElementById('modalDateNaissance').textContent =
dateNaissance;
        document.getElementById('modalNomParent').textContent =
nomParent;
        document.getElementById('modalEmailParent').textContent =
emailParent;
        document.getElementById('modalsolvabilite').textContent =
solvabilite;

        document.getElementById('studentModal').style.display =
'block';
    }

    function closeModal() {
        document.getElementById('studentModal').style.display =
'none';

```

```

        }

        // Fermer le modal si on clique en dehors
        window.onclick = function(event) {
            const modal = document.getElementById('studentModal');
            if (event.target == modal) {
                modal.style.display = 'none';
            }
        }

        // Fonction pour filtrer le tableau en temps réel
        function filterTable() {
            const column = document.getElementById('search-
column').value;
            const value = document.getElementById('search-value').value;

            const xhr = new XMLHttpRequest();
            xhr.open('GET', 'search.php?column=' + column + '&value=' +
value, true);
            xhr.onload = function() {
                if (this.status == 200) {
                    const response = JSON.parse(this.responseText);
                    let output = '';
                    if (response.records) {
                        response.records.forEach(record => {
                            output += `
                                <tr>
                                    <td>${record.id}</td>
                                    <td>${record.matricule}</td>
                                    <td><img
src='${record.image_path}' alt='Photo'></td>
                                    <td>${record.nom}</td>
                                    <td>${record.prenom}</td>
                                    <td>${record.classe}</td>
                                    <td>${record.email}</td>
                                    <td>${record.nom_parent}</td>
                                    <td>${record.email_parent}</td>
                                    <td>${record.date_naissance}</td>
                                    <td>${record.montant_a_payer}</td>
                                    <td>${record.solvabilite}</td>
                                    <td class="action-buttons">
                                        <a
href="modifier_etudiant.php?id=${record.id}">
                                            
                                        </a>
                                        <a
href="supprimer_etudiant.php?id=${record.id}" onclick="return
confirm('Êtes-vous sûr de vouloir supprimer cet étudiant?');">
                                            
                                        </a>
                                        <a href="#" onclick="showStudentDetails(
```

```

```

 '${record.id}',
 '${record.matricule}',
 '${record.image_path}',
 '${record.nom}',
 '${record.prenom}',
 '${record.classe}',
 '${record.email}',
 '${record.date_naissance}',
 '${record.nom_parent}',
 '${record.email_parent}',
 '${record.solvabilite}'
) ">

 </td>
 </tr>
 `;
}
} else {
 output += `<tr><td
colspan="12">${response.message}</td></tr>`;
}
document.getElementById('student-table-
body').innerHTML = output;
}
}
xhr.send();
}
</script>
</body>
</html>
```

```

*****fichier ajouter_etudiant.php
<?php
require_once 'classes/db_connect.php';
require_once 'classes/Etudiant.php';
require_once 'extractor_pdf/fpdf.php';
require 'phpmailer/src/Exception.php';
require 'phpmailer/src/PHPMailer.php';
require 'phpmailer/src/SMTP.php';

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

$database = new Database();
$db = $database->getConnection();
$etudiant = new Etudiant($db);

$error = "";
$success = false;

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 $nom = $_POST['nom'];
 $prenom = $_POST['prenom'];
 $classe = $_POST['classe'];
 $email = $_POST['email'];
 $date_naissance = $_POST['date_naissance'];
 $matricule = $_POST['matricule'];
 $solvabilite = $_POST['solvabilite'];
 $nom_parent = $_POST['nom_parent'];
 $email_parent = $_POST['email_parent'];
}
```

```

$prenom = $_POST['prenom'];
$classe = $_POST['classe'];
$email = $_POST['email'];
$email_parent = $_POST['email_parent'];
$nom_parent = $_POST['nom_parent'];
$date_naissance = $_POST['date_naissance'];
$image_path = $_FILES['image']['name'];

// Déterminer le montant à payer en fonction de la classe
$montant_a_payer = 0;
if ($classe === 'B1') {
 $montant_a_payer = 1000000;
} elseif ($classe === 'B2') {
 $montant_a_payer = 2000000;
} elseif ($classe === 'B3') {
 $montant_a_payer = 3000000;
}

// Validation des champs
if (empty($nom) || empty($prenom) || empty($classe) || empty($email)
|| empty($email_parent) || empty($date_naissance) || empty($image_path))
{
 $error = "Tous les champs sont obligatoires.";
} elseif (strtotime($date_naissance) > strtotime('2011-12-31')) {
 $error = "La date de naissance ne doit pas être supérieure à
2011.";
} else {
 // Générer le matricule
 $matricule = generateMatricule($db, $classe);

 // Déplacer l'image téléchargée
 $target_dir = "photos/";
 $target_file = $target_dir . basename($image_path);
 if (move_uploaded_file($_FILES['image']['tmp_name'],
$target_file)) {
 // Insérer les données dans la table etudiant_infos
 $etudiant->nom = $nom;
 $etudiant->prenom = $prenom;
 $etudiant->matricule = $matricule;
 $etudiant->image_path = $image_path;
 $etudiant->classe = $classe;
 $etudiant->email = $email;
 $etudiant->email_parent = $email_parent;
 $etudiant->nom_parent = $nom_parent;
 $etudiant->date_naissance = $date_naissance;
 $etudiant->montant_a_payer = $montant_a_payer;
 $etudiant->solvabilite = 'INSOLVABLE';

 if ($etudiant->create()) {
 // Insérer les données dans la table student_compte
 $stmt = $db->prepare("INSERT INTO student_compte
(matricule, password) VALUES (:matricule, :password)");
 $matriculeVar = $matricule; // Utiliser une variable pour
passer par référence
 }
 }
}

```

```

 $stmt->bindParam(':matricule', $matriculeVar);
 $stmt->bindParam(':password', $matriculeVar);
 $stmt->execute();

 // Générer la carte étudiant
 $pdf_path = generateStudentCard($matricule, $nom,
$prenom, $classe, $date_naissance, $image_path);

 // Envoyer l'email
 sendEmail($email_parent, $email, "Carte d'Etudiant",
"Recevez ci-joint la carte d'étudiant pour l'étudiant : $nom $prenom,
dont le matricule est le suivant : $matricule", $pdf_path);

 $success = true;
 header("Location: EtudiantListe.php");
 exit();
 } else {
 $error = "Erreur lors de l'ajout de l'étudiant.";
 }
} else {
 $error = "Erreur lors du téléchargement de l'image.";
}
}

function generateMatricule($db, $classe) {
 $year = date('y');
 $prefix = $year . $classe;
 $stmt = $db->prepare("SELECT matricule FROM etudiant_infos WHERE
matricule LIKE :prefix ORDER BY matricule DESC LIMIT 1");
 $prefixVar = $prefix . '%'; // Utiliser une variable pour passer par
référence
 $stmt->bindParam(':prefix', $prefixVar);
 $stmt->execute();
 $row = $stmt->fetch(PDO::FETCH_ASSOC);

 if ($row) {
 $lastMatricule = $row['matricule'];
 $lastNumber = intval(substr($lastMatricule, -3));
 $newNumber = sprintf('%03d', $lastNumber + 1);
 } else {
 $newNumber = '001';
 }

 return $prefix . $newNumber;
}

function generateStudentCard($matricule, $nom, $prenom, $classe,
$date_naissance, $image_path) {
 // Initialiser FPDF
 $pdf = new FPDF('P', 'mm', 'A4');
 $pdf->AddPage();
 $pdf->SetMargins(10, 10, 10);
}

```

```

// Style de la carte étudiant
$pdf->SetFillColor(240, 240, 255); // Fond clair
$pdf->SetDrawColor(200, 200, 200); // Bordure douce
$pdf->SetTextColor(50, 50, 50);

// Conteneur de la carte
$pdf->SetXY(20, 40);
$pdf->SetFont('Arial', 'B', 18);
$pdf->Cell(170, 10, 'Carte Etudiant', 0, 1, 'C', true);
$pdf->Ln(10);

// Ajout de l'image (photo de profil)
if (!empty($image_path) && file_exists('photos/' . $image_path)) {
 $pdf->Image('photos/' . $image_path, 85, 55, 40, 40); // Centré
 $pdf->Ln(50);
}

// Informations de l'étudiant
$pdf->SetFont('Arial', '', 14);
$pdf->Cell(0, 10, 'Matricule: ' . $matricule, 0, 1, 'C');
$pdf->Cell(0, 10, 'Nom: ' . $nom, 0, 1, 'C');
$pdf->Cell(0, 10, 'Prenom: ' . $prenom, 0, 1, 'C');
$pdf->Cell(0, 10, 'Classe: ' . $classe, 0, 1, 'C');
$pdf->Cell(0, 10, 'Date de Naissance: ' . $date_naissance, 0, 1,
'C');

// Ajout du logo de Keyce Informatique
$pdf->Image('images/keyce.jpeg', 15, 10, 30, 30);

// Ajout d'une bordure autour de la carte
$pdf->SetXY(15, 35);
$pdf->SetLineWidth(0.5);
$pdf->Rect(15, 35, 180, 160, 'D');

// Sortie PDF dans le sous-dossier "cartes"
$pdf_path = 'cartes/carte_etudiant_' . $matricule . '.pdf';
$pdf->Output('F', $pdf_path);

return $pdf_path;
}

function sendEmail($to, $cc, $subject, $body, $attachment) {
 $mail = new PHPMailer(true);

 try {
 //Server settings
 $mail->SMTPDebug = 0; // Enable verbose
debug output
 $mail->isSMTP(); // Set mailer to use
SMTP
 $mail->Host = 'smtp.gmail.com'; // Specify main and
backup SMTP servers
 $mail->SMTPAuth = true; // Enable SMTP
authentication
 }
}

```

```

 $mail->Username = 'gaza45palestine@gmail.com'; // SMTP username
 $mail->Password = 'tira vtly vbec schk'; // SMTP password
(app-specific password)
 $mail->SMTPSecure = 'tls'; // Enable TLS
encryption, `ssl` also accepted
 $mail->Port = 587; // TCP port to connect
to

 //Recipients
$mail->setFrom('gaza45palestine@gmail.com', 'FREE PALESTINE');
$mail->addAddress($to); // Add a recipient
$mail->addCC($cc); // Add a CC recipient

 // Attachments
$mail->addAttachment($attachment); // Add attachments

 // Content
$mail->isHTML(true); // Set email format to
HTML
 $mail->Subject = $subject;
$mail->Body = $body;

 $mail->send();
} catch (Exception $e) {
 echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-
scale=1.0">
 <title>Ajouter un Étudiant</title>
 <style>
 /* Fond avec un dégradé doux */
 body {
 margin: 0;
 font-family: 'Arial', sans-serif;
 background: linear-gradient(145deg, #fce4ec, #e3f2fd);
 color: #333;
 display: flex;
 justify-content: center;
 align-items: center;
 height: 100vh;
 }

 /* Conteneur du formulaire */
.form-container {
 background: #ffffff;
 border-radius: 15px;
 box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);

```

```
padding: 30px;
width: 400px;
text-align: center;
}

.form-container h2 {
margin-bottom: 20px;
font-size: 24px;
color: #333;
}

.form-container .form-group {
margin-bottom: 15px;
text-align: left;
}

.form-container .form-group label {
display: block;
font-size: 14px;
margin-bottom: 5px;
color: #555;
}

.form-container .form-group input,
.form-container .form-group select {
width: 100%;
padding: 10px;
border: 1px solid #ddd;
border-radius: 5px;
font-size: 14px;
box-sizing: border-box;
}

.form-container .form-group input[type="file"] {
border: none;
}

.form-container .form-group button {
background: linear-gradient(145deg, #64b5f6, #f06292);
color: #fff;
border: none;
padding: 10px 20px;
border-radius: 25px;
font-size: 14px;
cursor: pointer;
transition: all 0.3s;
width: 100%;
}

.form-container .form-group button:hover {
background: linear-gradient(145deg, #42a5f5, #ec407a);
transform: translateY(-2px);
box-shadow: 0 5px 15px rgba(64, 115, 255, 0.2);
}
```

```

.form-container .error {
 color: #f44336;
 margin-bottom: 15px;
 font-size: 14px;
 text-align: center;
}

.form-container img {
 margin-top: 10px;
 width: 100px;
 height: 100px;
 border-radius: 50%;
 display: block;
 margin-left: auto;
 margin-right: auto;
}
</style>
</head>
<body>
<div class="form-container">
 <h2>Ajouter un Étudiant</h2>
 <?php if ($error): ?>
 <div class="error"><?php echo $error; ?></div>
 <?php endif; ?>
 <form method="POST" enctype="multipart/form-data">
 <div class="form-group">
 <label for="nom">Nom</label>
 <input type="text" id="nom" name="nom" required>
 </div>
 <div class="form-group">
 <label for="prenom">Prénom</label>
 <input type="text" id="prenom" name="prenom" required>
 </div>
 <div class="form-group">
 <label for="classe">Classe</label>
 <select id="classe" name="classe" required>
 <option value="B1">B1</option>
 <option value="B2">B2</option>
 <option value="B3">B3</option>
 </select>
 </div>
 <div class="form-group">
 <label for="email">Email</label>
 <input type="email" id="email" name="email" required>
 </div>
 <div class="form-group">
 <label for="nom_parent">Nom Parent</label>
 <input type="text" id="nom_parent" name="nom_parent"
required>
 </div>
 <div class="form-group">
 <label for="email_parent">Email Parent</label>

```

```

 <input type="email" id="email_parent" name="email_parent"
required>
 </div>
 <div class="form-group">
 <label for="date_naissance">Date de Naissance</label>
 <input type="date" id="date_naissance"
name="date_naissance" required>
 </div>
 <div class="form-group">
 <label for="image">Photo</label>
 <input type="file" id="image" name="image"
accept="image/*" required>

 </div>
 <div class="form-group">
 <button type="submit">Ajouter Étudiant</button>
 </div>
</form>
</div>

<script>
 // Prévisualisation de l'image
 document.getElementById('image').addEventListener('change',
function(event) {
 const [file] = event.target.files;
 if (file) {
 const preview = document.getElementById('imagePreview');
 preview.src = URL.createObjectURL(file);
 preview.style.display = 'block';
 }
});
</script>
</body>
</html>

*****modifier_et
udiant.php
<?php
require_once 'classes/db_connect.php';
require_once 'classes/Etudiant.php';

$database = new Database();
$db = $database->getConnection();
$etudiant = new Etudiant($db);

$error = "";
$id = isset($_GET['id']) ? $_GET['id'] : die('ID de l\'étudiant non
fourni.');

// Récupérer les informations de l'étudiant
$stmt = $db->prepare("SELECT * FROM etudiant_infos WHERE id = :id");
$stmt->bindParam(':id', $id);
$stmt->execute();

```

```

$row = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$row) {
 die('Étudiant non trouvé.');
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 $matricule = $_POST['matricule'];
 $nom = $_POST['nom'];
 $prenom = $_POST['prenom'];
 $classe = $_POST['classe'];
 $email = $_POST['email'];
 $email_parent = $_POST['email_parent'];
 $nom_parent = $_POST['nom_parent'];
 $date_naissance = $_POST['date_naissance'];
 $montant_a_payer = $_POST['montant_a_payer'];
 $image_path = $_POST['image_path'];
 $solvabilite = $_POST['solvabilite'];

 // Validation des champs
 if (empty($nom) || empty($prenom) || empty($classe) || empty($email)
 || empty($email_parent) || empty($date_naissance)) {
 $error = "Tous les champs sont obligatoires.";
 } elseif (strtotime($date_naissance) > strtotime('2011-12-31')) {
 $error = "La date de naissance ne doit pas être supérieure à
2011.";
 } else {
 // Mettre à jour les informations de l'étudiant
 $etudiant->matricule = $matricule;
 $etudiant->id = $id;
 $etudiant->nom = $nom;
 $etudiant->prenom = $prenom;
 $etudiant->classe = $classe;
 $etudiant->email = $email;
 $etudiant->email_parent = $email_parent;
 $etudiant->date_naissance = $date_naissance;
 $etudiant->nom_parent = $nom_parent;
 $etudiant->montant_a_payer = $montant_a_payer;
 $etudiant->image_path = $image_path;
 $etudiant->solvabilite = $solvabilite;

 if ($etudiant->update()) {
 header("Location: EtudiantListe.php");
 exit();
 } else {
 $error = "Erreur lors de la mise à jour de l'étudiant.";
 }
 }
}

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

```

```
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Modifier un Étudiant</title>
 <style>
 * {
 margin: 0;
 padding: 0;
 box-sizing: border-box;
 }

 body {
 font-family: 'Arial', sans-serif;
 background: linear-gradient(145deg, #89f7fe, #ff7fff);
 display: flex;
 justify-content: center;
 align-items: center;
 min-height: 100vh;
 padding: 20px;
 }

 .form-card {
 background: white;
 border-radius: 20px;
 box-shadow: 0 10px 25px rgba(0,0,0,0.1);
 padding: 40px;
 width: 100%;
 max-width: 500px;
 transition: transform 0.3s ease;
 }

 .form-title {
 text-align: center;
 margin-bottom: 30px;
 color: #333;
 font-size: 24px;
 }

 .form-error {
 background-color: #ffddcc;
 color: #ff4d4d;
 padding: 15px;
 border-radius: 8px;
 margin-bottom: 20px;
 text-align: center;
 }

 .form-group {
 margin-bottom: 20px;
 position: relative;
 }

 .form-group label {
```

```
 display: block;
 margin-bottom: 8px;
 color: #555;
 font-weight: bold;
 }

.form-group input,
.form-group select {
 width: 100%;
 padding: 12px 15px;
 border: 1px solid #ddd;
 border-radius: 8px;
 font-size: 16px;
 transition: all 0.3s ease;
}

.form-group input:focus,
.form-group select:focus {
 outline: none;
 border-color: #89f7fe;
 transform: translateY(-5px);
 box-shadow: 0 5px 15px rgba(137, 247, 254, 0.2);
}

.form-buttons {
 display: flex;
 justify-content: space-between;
 margin-top: 30px;
}

.btn {
 padding: 12px 25px;
 border: none;
 border-radius: 25px;
 font-size: 16px;
 cursor: pointer;
 transition: all 0.3s ease;
}

.btn-cancel {
 background-color: #f1f1f1;
 color: #333;
}

.btn-confirm {
 background: linear-gradient(145deg, #89f7fe, #ff7fff);
 color: white;
}

.btn:hover {
 transform: translateY(-3px);
 box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

</style>
```

```

</head>
<body>
 <div class="form-card">
 <h2 class="form-title">Modifier un Étudiant</h2>

 <?php if ($error) : ?>
 <div class="form-error">
 <?php echo htmlspecialchars($error); ?>
 </div>
 <?php endif; ?>

 <form method="POST">
 <div class="form-group">
 <label for="matricule">Matricule</label>
 <input type="text" id="matricule" name="matricule"
 value="<?php echo
 htmlspecialchars($row['matricule']); ?>" readonly>
 </div>
 <div class="form-group">
 <label for="nom">Nom</label>
 <input type="text" id="nom" name="nom"
 value="<?php echo htmlspecialchars($row['nom']);
 ?>" required>
 </div>
 <div class="form-group">
 <label for="prenom">Prénom</label>
 <input type="text" id="prenom" name="prenom"
 value="<?php echo
 htmlspecialchars($row['prenom']); ?>" required>
 </div>
 <div class="form-group">
 <label for="classe">Classe</label>
 <select id="classe" name="classe" required>
 <option value="B1" <?php if ($row['classe'] === 'B1')
echo 'selected'; ?>>B1</option>
 <option value="B2" <?php if ($row['classe'] === 'B2')
echo 'selected'; ?>>B2</option>
 <option value="B3" <?php if ($row['classe'] === 'B3')
echo 'selected'; ?>>B3</option>
 </select>
 </div>
 <div class="form-group">
 <label for="email">Email</label>
 <input type="email" id="email" name="email"
 value="<?php echo htmlspecialchars($row['email']);
 ?>" required>
 </div>
 <div class="form-group">
 <label for="nom_parent">Nom Parent</label>
 <input type="text" id="nom_parent" name="nom_parent"
 value="<?php echo
 htmlspecialchars($row['nom_parent']); ?>" required>
 </div>
 <div class="form-group">

```

```

 <label for="email_parent">Email Parent</label>
 <input type="email" id="email_parent" name="email_parent"
 value="<?php echo
htmlspecialchars($row['email_parent']); ?>" required>
 </div>
 <div class="form-group">
 <label for="montant_a_payer">Montant à payer</label>
 <input type="text" id="montant_a_payer"
 name="montant_a_payer"
 value="<?php echo
htmlspecialchars($row['montant_a_payer']); ?>" required>
 </div>
 <div class="form-group">
 <label for="solvabilite">Solvabilité</label>
 <input type="text" id="solvabilite" name="solvabilite"
 value="<?php echo
htmlspecialchars($row['solvabilite']); ?>" required>
 </div>
 <div class="form-group">
 <label for="date_naissance">Date de Naissance</label>
 <input type="date" id="date_naissance"
 name="date_naissance"
 value="<?php echo
htmlspecialchars($row['date_naissance']); ?>" required>
 </div>
 <div class="form-group">
 <label for="image_path">Chemin de l'image</label>
 <input type="text" id="image_path" name="image_path"
 value="<?php echo
htmlspecialchars($row['image_path']); ?>" required>
 </div>
 <div class="form-buttons">
 <button type="button" class="btn btn-cancel"
 onclick="window.location.href='EtudiantListe.php'">Annuler</button>
 <button type="submit" class="btn btn-confirm">Confirmer</button>
 </div>
</div>

<script>
 // Form input focus animation
 const inputs = document.querySelectorAll('.form-group input,
.form-group select');

 inputs.forEach(input => {
 input.addEventListener('focus', function() {
 this.parentElement.style.transform = 'translateY(-5px)';
 });

 input.addEventListener('blur', function() {
 this.parentElement.style.transform = 'translateY(0)';
 });
 });

```

```

 });
 </script>
</body>
</html>

*****search
.php
<?php
require_once 'classes/db_connect.php';
require_once 'classes/Etudiant.php';

$database = new Database();
$db = $database->getConnection();
$etudiant = new Etudiant($db);

if (isset($_GET['column']) && isset($_GET['value'])) {
 $column = $_GET['column'];
 $value = $_GET['value'];
 $stmt = $etudiant->search($column, $value);
 $num = $stmt->rowCount();

 if ($num > 0) {
 $etudiants_arr = array();
 $etudiants_arr["records"] = array();

 while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
 extract($row);
 $etudiant_item = array(
 "id" => $id,
 "matricule" => $matricule,
 "image_path" => $image_path,
 "nom" => $nom,
 "prenom" => $prenom,
 "classe" => $classe,
 "email" => $email,
 "nom_parent" => $nom_parent,
 "email_parent" => $email_parent,
 "date_naissance" => $date_naissance,
 "montant_a_payer" => $montant_a_payer,
 "solvabilite" => $solvabilite,
);
 array_push($etudiants_arr["records"], $etudiant_item);
 }
 echo json_encode($etudiants_arr);
 } else {
 echo json_encode(array("message" => "Aucun étudiant trouvé."));
 }
} else {
 echo json_encode(array("message" => "Paramètres manquants."));
}
?>
```

```

*****supprimer_etudian
t.php
<?php
require_once 'classes/db_connect.php';
require_once 'classes/Etudiant.php';

$database = new Database();
$db = $database->getConnection();
$etudiant = new Etudiant($db);

if ($_SERVER['REQUEST_METHOD'] === 'GET' && isset($_GET['id'])) {
 $etudiant->id = $_GET['id'];

 if ($etudiant->delete()) {
 // Supprimer également l'entrée correspondante dans la table
 student_compte
 $stmt = $db->prepare("DELETE FROM student_compte WHERE matricule
= :matricule");
 $stmt->bindParam(':matricule', $etudiant->matricule);
 $stmt->execute();

 header("Location: EtudiantListe.php");
 exit();
 } else {
 echo "Erreur lors de la suppression de l'étudiant.";
 }
} else {
 echo "ID de l'étudiant non fourni.";
}
?>
*****db_connect.php
<?php
class Database {
 private $host = 'localhost';
 private $db_name = 'ecole';
 private $username = 'root';
 private $password = '';
 public $conn;

 public function getConnection() {
 $this->conn = null;
 try {
 $this->conn = new PDO("mysql:host=" . $this->host .
";dbname=" . $this->db_name, $this->username, $this->password);
 $this->conn->exec("set names utf8");
 } catch(PDOException $exception) {
 echo "Connection error: " . $exception->getMessage();
 }
 return $this->conn;
 }
}
?>
```