Énoncé du Problème – Plateforme de Gestion pour l'Institut John Willeit

L'Institut John Willeit est une institution académique en pleine croissance, accueillant chaque année un nombre croissant d'étudiants dans plusieurs filières. Jusqu'à présent, la gestion des inscriptions, des cours, et des données administratives se faisait manuellement, à l'aide de documents papier et de fichiers Excel.

Cette méthode entraîne plusieurs difficultés :

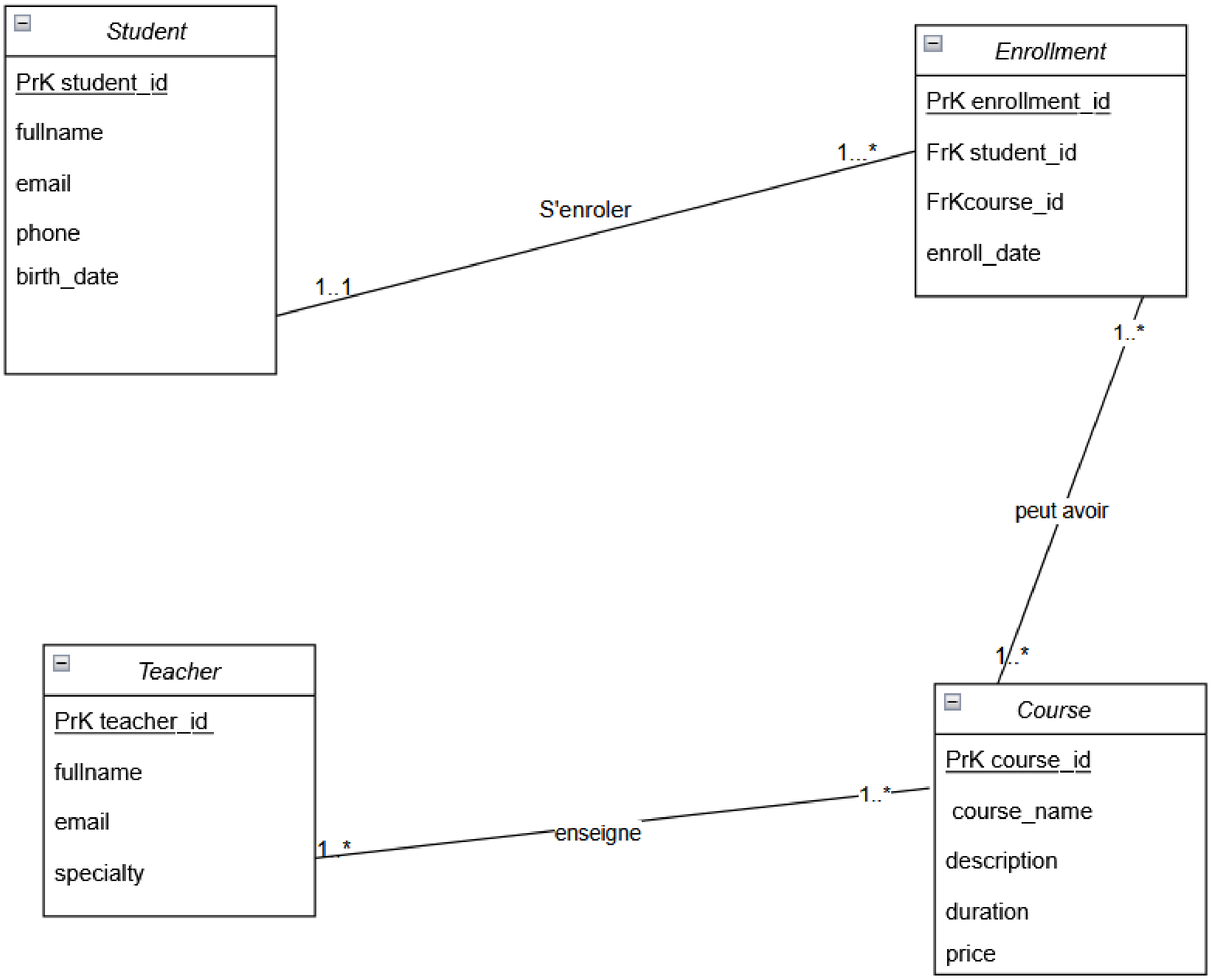
Problèmes rencontrés

1. **Difficulté à gérer les inscriptions** (pertes de données, erreurs humaines).
2. **Absence d'une base de données centralisée** pour stocker les informations des étudiants.
3. **Manque de visibilité** pour le directeur sur les statistiques de l'institut (nombre d'étudiants, cours, etc.).
4. **Impossible d'accéder rapidement aux listes des étudiants ou des cours.**
5. **Aucun tableau de bord** pour suivre les activités de l'école.

Face à ces limites, le directeur de l'Institut John Willeit a contacté un **ingénieur en informatique** pour mettre en place une solution moderne, fiable et accessible.

Solution

MCD of our System



Création de la Base de Données

```
CREATE DATABASE john_willeit ;

CREATE DATABASE john_willeit CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

USE john_willeit;
```

Création des Tables (SQL complet)

```
CREATE TABLE students (

    student_id INT AUTO_INCREMENT PRIMARY KEY,

    fullname VARCHAR(150) NOT NULL,

    email VARCHAR(150) UNIQUE NOT NULL,
    phone VARCHAR(20),
    date_of_birth DATE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);



CREATE TABLE courses (
    course_id INT AUTO_INCREMENT PRIMARY KEY,
    course_name VARCHAR(150) NOT NULL,
    description TEXT,
    duration_months INT,
    price DECIMAL(10,2),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TABLE teachers (
    teacher_id INT AUTO_INCREMENT PRIMARY KEY,
    fullname VARCHAR(150) NOT NULL,
    email VARCHAR(150) UNIQUE NOT NULL,
    specialty VARCHAR(150),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TABLE enrollments (
    enrollment_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    course_id INT,
    enrollment_date DATE DEFAULT CURRENT_DATE,
    FOREIGN KEY (student_id) REFERENCES students(student_id) ON DELETE CASCADE,
    FOREIGN KEY (course_id) REFERENCES courses(course_id) ON DELETE CASCADE
);
```

Ajouter quelques données (INSERT)

Étudiants

```
INSERT INTO students (fullname, email, phone, date_of_birth)
VALUES
('Faustin Djakba', 'faustin@example.com', '650000001', '2000-05-06'),
('Sarah Talla', 'sarah@example.com', '650000002', '1999-11-02');
```

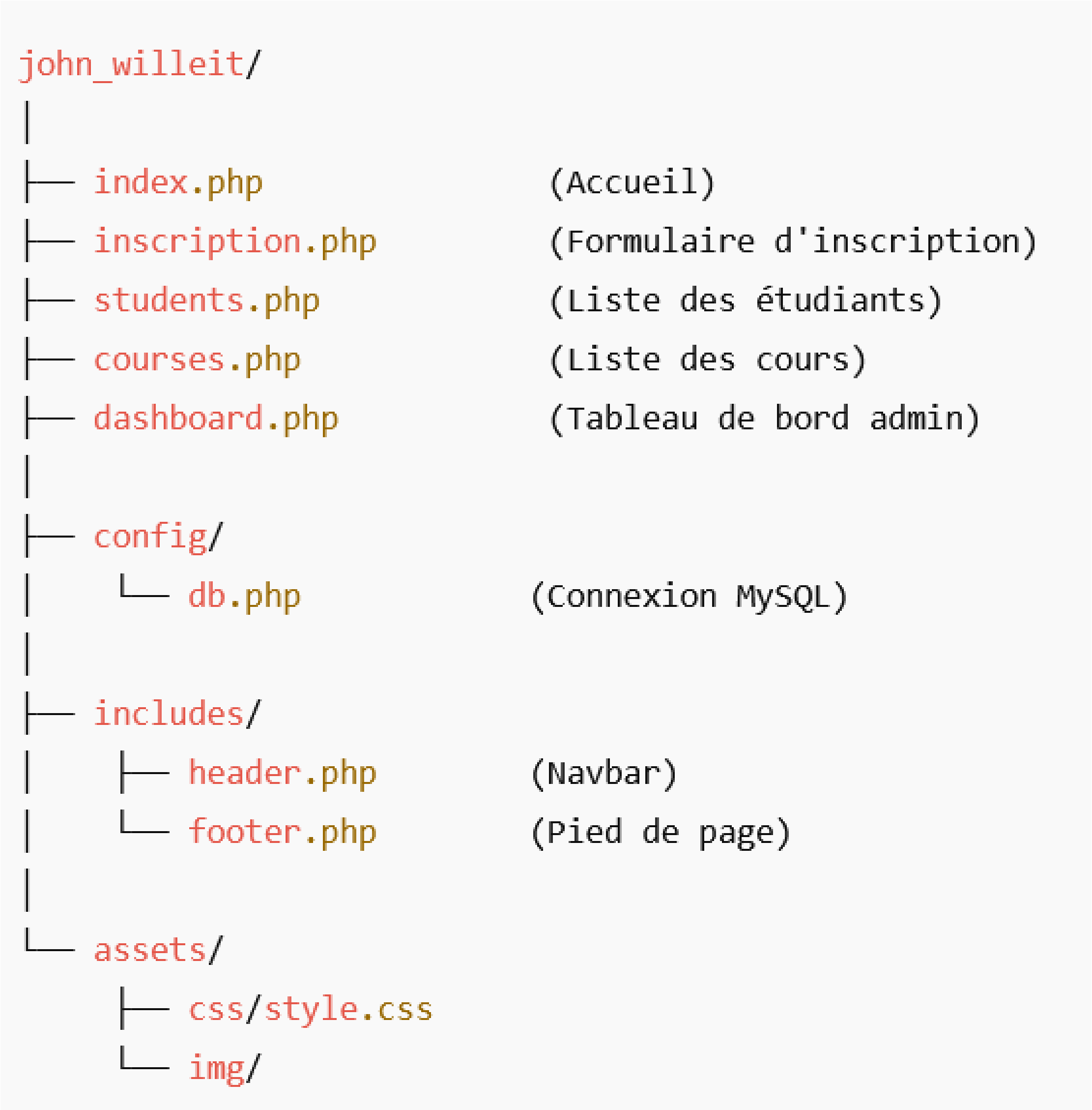
Cours

```
INSERT INTO courses (course_name, description, duration_months, price)
VALUES
('Ethical Hacking', 'Learn penetration testing and cyber defense', 3, 150000),
('Web Development', 'HTML, CSS, JavaScript, PHP', 4, 130000);
```

Enseignants

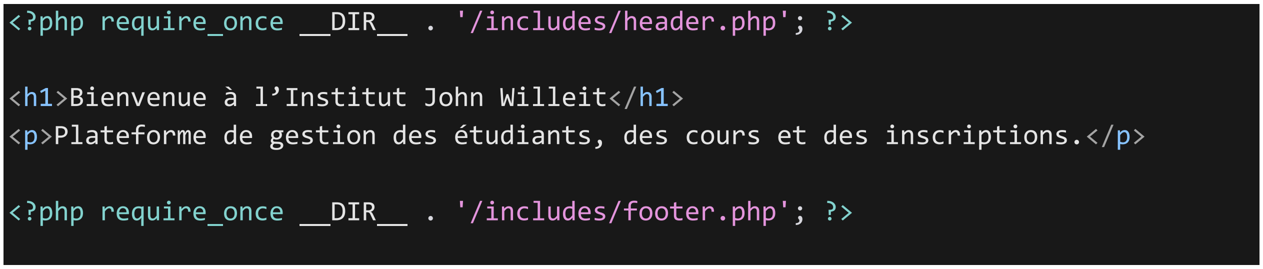
```
INSERT INTO teachers (fullname, email, specialty)
VALUES
('Daniel Augot', 'daniel@institut.com', 'Cybersecurity'),
('John Willeit', 'john@institut.com', 'Programming');
```

1-Voici la Structure



```
john_willeit/
│
├── index.php               (Accueil)
├── inscription.php         (Formulaire d'inscription)
├── students.php            (Liste des étudiants)
├── courses.php             (Liste des cours)
├── dashboard.php           (Tableau de bord admin)
│
├── config/
│    └── db.php             (Connexion MySQL)
│
├── includes/
│    ├── header.php         (Navbar)
│    └── footer.php         (Pied de page)
│
└── assets/
     ├── css/style.css
     └── img/
```

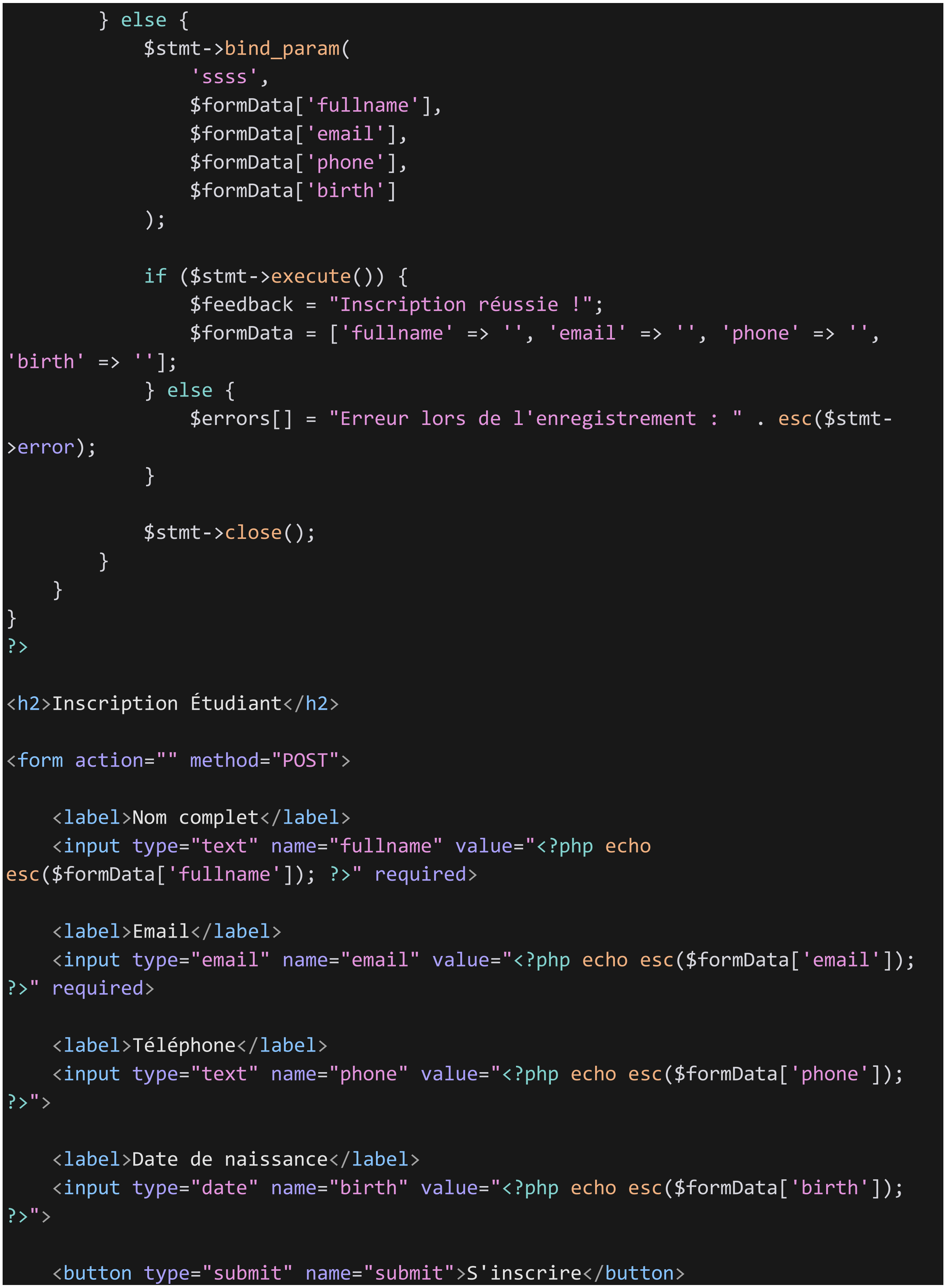
index.php



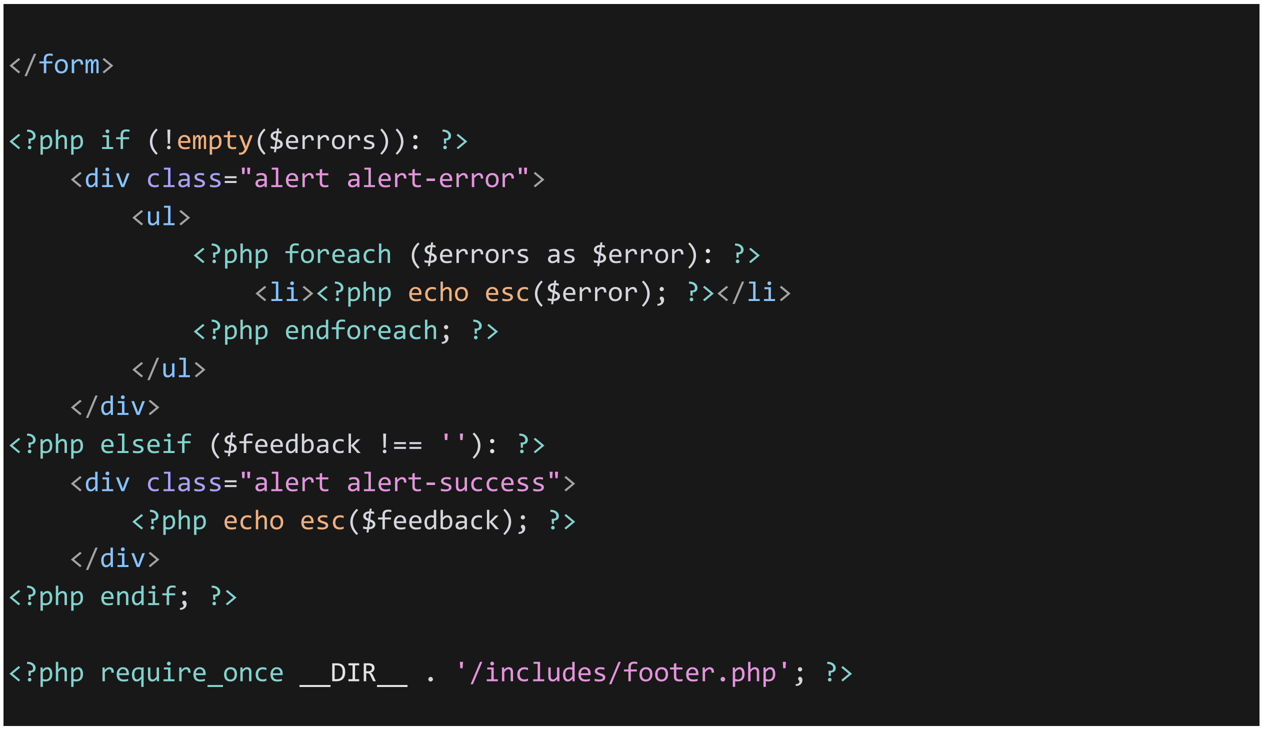
```
<?php require_once __DIR__ . '/includes/header.php'; ?>

<h1>Bienvenue à l’Institut John Willeit</h1>
<p>Plateforme de gestion des étudiants, des cours et des inscriptions.</p>

<?php require_once __DIR__ . '/includes/footer.php'; ?>
```

inscription.php

```php
<?php
require_once __DIR__ . '/includes/header.php';
require_once __DIR__ . '/includes/config/db.php';
require_once __DIR__ . '/includes/helpers.php';

$formData = [
    'fullname' => '',
    'email' => '',
    'phone' => '',
    'birth' => '',
];
$errors = [];
$feedback = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $formData['fullname'] = post_value('fullname') ?? '';
    $formData['email'] = post_value('email') ?? '';
    $formData['phone'] = post_value('phone') ?? '';
    $formData['birth'] = post_value('birth') ?? '';

    if ($formData['fullname'] === '') {
        $errors[] = "Le nom complet est requis.";
    }

    if (!filter_var($formData['email'], FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Email invalide.";
    }

    if ($formData['birth'] !== '' && !preg_match('/^\d{4}-\d{2}-\d{2}$/',
$formData['birth'])) {
        $errors[] = "La date de naissance doit être au format AAAA-MM-JJ.";
    }

    if (empty($errors)) {
        $stmt = $conn->prepare(
            "INSERT INTO students (fullname, email, phone, date_of_birth) VALUES
(?, ?, ?, ?)"
        );

        if ($stmt === false) {
            $errors[] = "Impossible de préparer la requête : " . esc($conn-
>error);
```

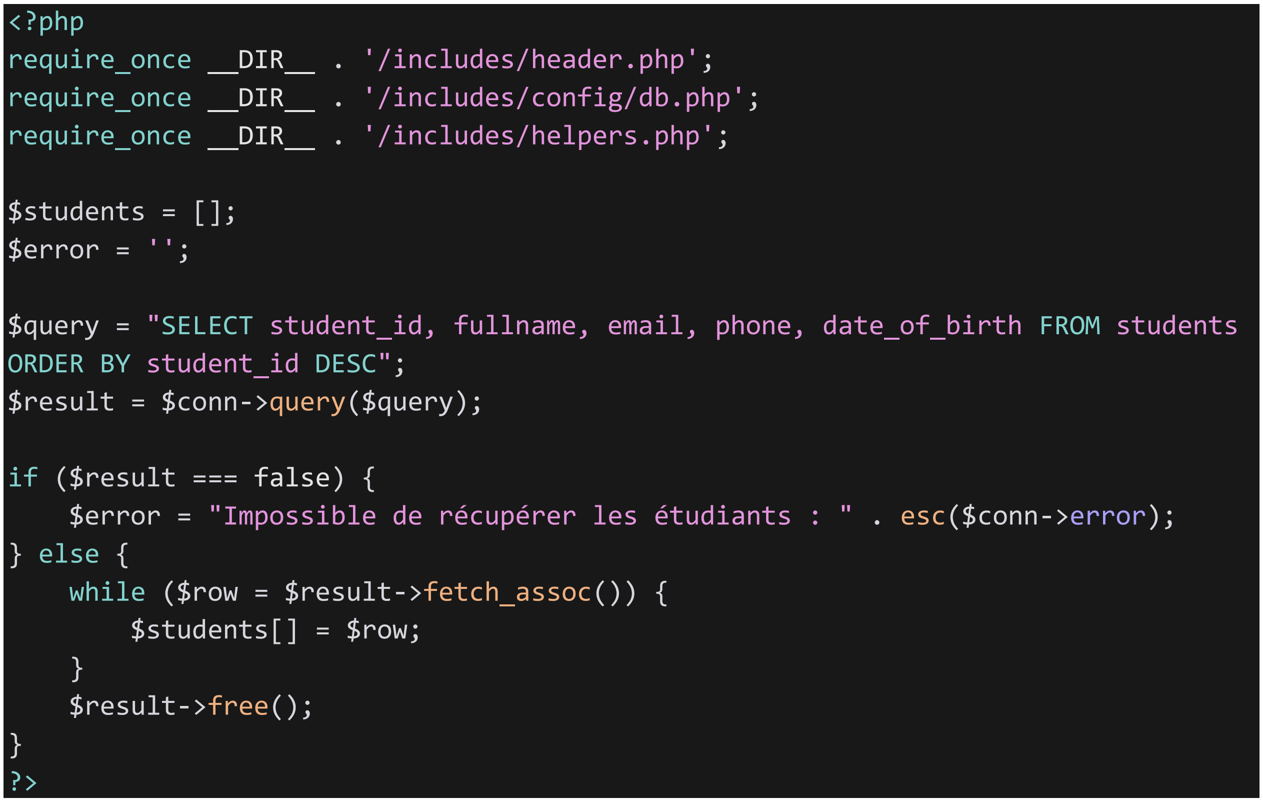


```
        } else {
            $stmt->bind_param(
                'ssss',
                $formData['fullname'],
                $formData['email'],
                $formData['phone'],
                $formData['birth']
            );

            if ($stmt->execute()) {
                $feedback = "Inscription réussie !";
                $formData = ['fullname' => '', 'email' => '', 'phone' => '',
'birth' => ''];
            } else {
                $errors[] = "Erreur lors de l'enregistrement : " . esc($stmt-
>error);
            }

            $stmt->close();
        }
    }
}
?>

<h2>Inscription Étudiant</h2>

<form action="" method="POST">

    <label>Nom complet</label>
    <input type="text" name="fullname" value="<?php echo
esc($formData['fullname']); ?>" required>

    <label>Email</label>
    <input type="email" name="email" value="<?php echo esc($formData['email']);
?>" required>

    <label>Téléphone</label>
    <input type="text" name="phone" value="<?php echo esc($formData['phone']);
?>">

    <label>Date de naissance</label>
    <input type="date" name="birth" value="<?php echo esc($formData['birth']);
?>">

    <button type="submit" name="submit">S'inscrire</button>
```



```
</form>

<?php if (!empty($errors)): ?>
    <div class="alert alert-error">
        <ul>
            <?php foreach ($errors as $error): ?>
                <li><?php echo esc($error); ?></li>
            <?php endforeach; ?>
        </ul>
    </div>
<?php elseif ($feedback !== ''): ?>
    <div class="alert alert-success">
        <?php echo esc($feedback); ?>
    </div>
<?php endif; ?>

<?php require_once __DIR__ . '/includes/footer.php'; ?>
```

students.php



```
<?php
require_once __DIR__ . '/includes/header.php';
require_once __DIR__ . '/includes/config/db.php';
require_once __DIR__ . '/includes/helpers.php';

$students = [];
$error = '';

$query = "SELECT student_id, fullname, email, phone, date_of_birth FROM students
ORDER BY student_id DESC";
$result = $conn->query($query);

if ($result === false) {
    $error = "Impossible de récupérer les étudiants : " . esc($conn->error);
} else {
    while ($row = $result->fetch_assoc()) {
        $students[] = $row;
    }
    $result->free();
}
?>
```

```
<h2>Liste des Étudiants</h2>

<?php if ($error !== ''): ?>
    <div class="alert alert-error">
        <?php echo esc($error); ?>
    </div>
<?php elseif (empty($students)): ?>
    <p>Aucun étudiant pour le moment.</p>
<?php else: ?>
    <table class="data-table">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nom complet</th>
                <th>Email</th>
                <th>Téléphone</th>
                <th>Date de naissance</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($students as $student): ?>
                <tr>
                    <td><?php echo esc((string)$student['student_id']); ?></td>
                    <td><?php echo esc($student['fullname']); ?></td>
                    <td><?php echo esc($student['email']); ?></td>
                    <td><?php echo esc($student['phone']); ?></td>
                    <td><?php echo esc($student['date_of_birth']); ?></td>
                </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
<?php endif; ?>

<?php require_once __DIR__ . '/includes/footer.php'; ?>
```

courses.php

```
<?php
require_once __DIR__ . '/includes/header.php';
require_once __DIR__ . '/includes/config/db.php';
require_once __DIR__ . '/includes/helpers.php';
```

```
$courses = [];
$error = '';

$query = "SELECT course_id, course_name, description, duration_months, price FROM
courses ORDER BY course_id DESC";
$result = $conn->query($query);

if ($result === false) {
    $error = "Impossible de récupérer les cours : " . esc($conn->error);
} else {
    while ($row = $result->fetch_assoc()) {
        $courses[] = $row;
    }
    $result->free();
}
?>

<h2>Liste des Cours</h2>

<?php if ($error !== ''): ?>
    <div class="alert alert-error">
        <?php echo esc($error); ?>
    </div>
<?php elseif (empty($courses)): ?>
    <p>Aucun cours enregistré pour le moment.</p>
<?php else: ?>
    <table class="data-table">
        <thead>
            <tr>
                <th>ID</th>
                <th>Cours</th>
                <th>Description</th>
                <th>Durée</th>
                <th>Prix</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($courses as $course): ?>
                <tr>
                    <td><?php echo esc((string)$course['course_id']); ?></td>
                    <td><?php echo esc($course['course_name']); ?></td>
                    <td><?php echo esc($course['description']); ?></td>
                    <td><?php echo esc((string)$course['duration_months']); ?>
mois</td>
```

```
                    <td><?php echo esc(number_format((float)$course['price'], 0,
',', ' ')); ?> FCFA</td>
                </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
<?php endif; ?>

<?php require_once __DIR__ . '/includes/footer.php'; ?>
```

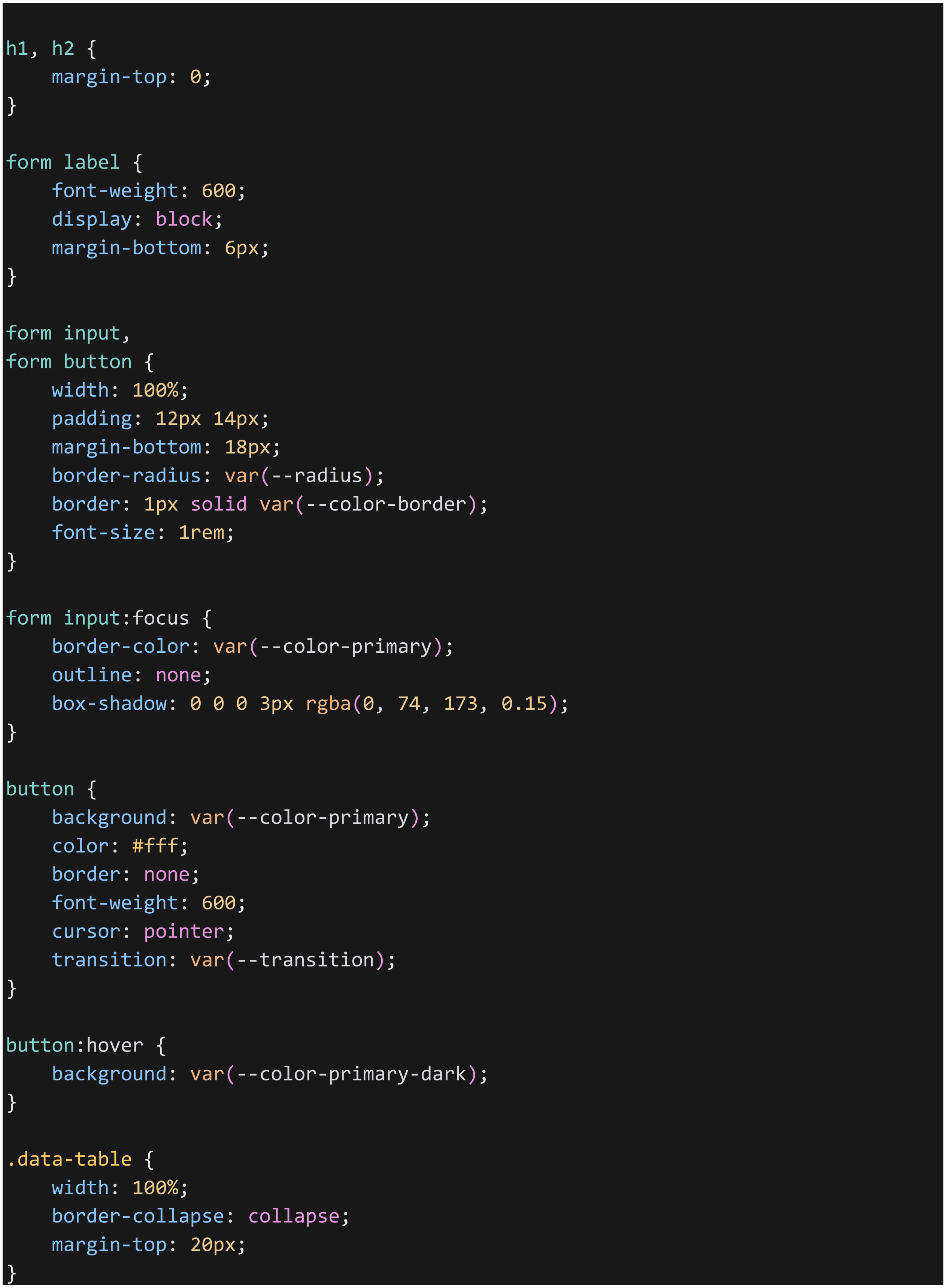
dashboard.php

```
<?php
require_once __DIR__ . '/includes/header.php';
require_once __DIR__ . '/includes/config/db.php';
require_once __DIR__ . '/includes/helpers.php';

$stats = [
    'students' => 0,
    'courses' => 0,
];
$error = '';

$queries = [
    'students' => "SELECT COUNT(*) AS total FROM students",
    'courses' => "SELECT COUNT(*) AS total FROM courses",
];

foreach ($queries as $key => $sql) {
    $result = $conn->query($sql);
    if ($result === false) {
        $error = "Impossible de calculer les statistiques : " . esc($conn-
>error);
        break;
    }

    $data = $result->fetch_assoc();
    $stats[$key] = (int)($data['total'] ?? 0);
    $result->free();
}
```

```
?>

<h2>Dashboard Administrateur</h2>

<?php if ($error !== ''): ?>
    <div class="alert alert-error">
        <?php echo esc($error); ?>
    </div>
<?php else: ?>
    <div class="dashboard-cards">
        <div class="card">
            <p>Total Étudiants</p>
            <strong><?php echo esc((string)$stats['students']); ?></strong>
        </div>
        <div class="card">
            <p>Total Cours</p>
            <strong><?php echo esc((string)$stats['courses']); ?></strong>
        </div>
    </div>
<?php endif; ?>

<?php require_once __DIR__ . '/includes/footer.php'; ?>
```

assets/css/style.css

```
:root {
    --color-primary: #004aad;
    --color-primary-dark: #00327a;
    --color-accent: #00bfa5;
    --color-bg: #f5f7fb;
    --color-border: #e2e6f0;
    --color-text: #1f2a37;
    --radius: 10px;
    --shadow: 0 10px 25px rgba(15, 23, 42, 0.08);
    --transition: 0.2s ease;
}

* {
    box-sizing: border-box;
}

body {
    font-family: "Segoe UI", Arial, sans-serif;
    background: var(--color-bg);
    color: var(--color-text);
    margin: 0;
    line-height: 1.5;
}

.navbar {
    background: #fff;
    box-shadow: var(--shadow);
    padding: 18px 10%;
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
    align-items: center;
}

.navbar-brand {
    font-weight: 600;
    font-size: 1.1rem;
    color: var(--color-primary);
}

.navbar-links a {
    color: var(--color-text);
    margin-left: 18px;
    text-decoration: none;
    font-weight: 500;
    padding-bottom: 4px;
    border-bottom: 2px solid transparent;
    transition: var(--transition);
}

.navbar-links a:hover {
    border-bottom-color: var(--color-accent);
    color: var(--color-primary);
}

.container {
    width: min(100%, 960px);
    margin: 40px auto;
    background: #fff;
    padding: 32px;
    border-radius: var(--radius);
    box-shadow: var(--shadow);
}
```



```css
h1, h2 {
    margin-top: 0;
}

form label {
    font-weight: 600;
    display: block;
    margin-bottom: 6px;
}

form input,
form button {
    width: 100%;
    padding: 12px 14px;
    margin-bottom: 18px;
    border-radius: var(--radius);
    border: 1px solid var(--color-border);
    font-size: 1rem;
}

form input:focus {
    border-color: var(--color-primary);
    outline: none;
    box-shadow: 0 0 0 3px rgba(0, 74, 173, 0.15);
}

button {
    background: var(--color-primary);
    color: #fff;
    border: none;
    font-weight: 600;
    cursor: pointer;
    transition: var(--transition);
}

button:hover {
    background: var(--color-primary-dark);
}

.data-table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}
```

```css
.data-table th,
.data-table td {
    padding: 14px;
    border-bottom: 1px solid var(--color-border);
    text-align: left;
}

.data-table th {
    background: #f8fafc;
    font-size: 0.95rem;
    letter-spacing: 0.03em;
    text-transform: uppercase;
}

.data-table tr:hover td {
    background: #f1f5f9;
}

.alert {
    border-radius: var(--radius);
    padding: 16px 18px;
    margin-top: 20px;
    border: 1px solid;
}

.alert ul {
    margin: 0;
    padding-left: 18px;
}

.alert-success {
    background: #ecfdf5;
    border-color: #a7f3d0;
    color: #065f46;
}

.alert-error {
    background: #fef2f2;
    border-color: #fecaca;
    color: #991b1b;
}

.dashboard-cards {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    gap: 20px;
    margin-top: 30px;
}

.card {
    padding: 24px;
    border-radius: var(--radius);
    background: linear-gradient(135deg, var(--color-primary), var(--color-accent));
    color: #fff;
    box-shadow: var(--shadow);
}

.card strong {
    font-size: 2rem;
    display: block;
    margin-top: 8px;
}

footer {
    text-align: center;
    margin: 40px 0;
    color: #94a3b8;
}

@media (max-width: 640px) {
    .navbar {
        padding: 18px 20px;
    }

    .navbar-links {
        width: 100%;
        margin-top: 12px;
        display: flex;
        flex-wrap: wrap;
        gap: 10px;
    }

    .container {
        padding: 24px 20px;
    }
}
```

config/db.php

```php
<?php
declare(strict_types=1);

$host = "localhost";
$user = "root";
$pass = "";
$dbname = "john_willeit";

$conn = new mysqli($host, $user, $pass, $dbname);

if ($conn->connect_error) {
    die("Connexion à la base de données impossible : " . $conn->connect_error);
}

$conn->set_charset('utf8mb4');
?>
```

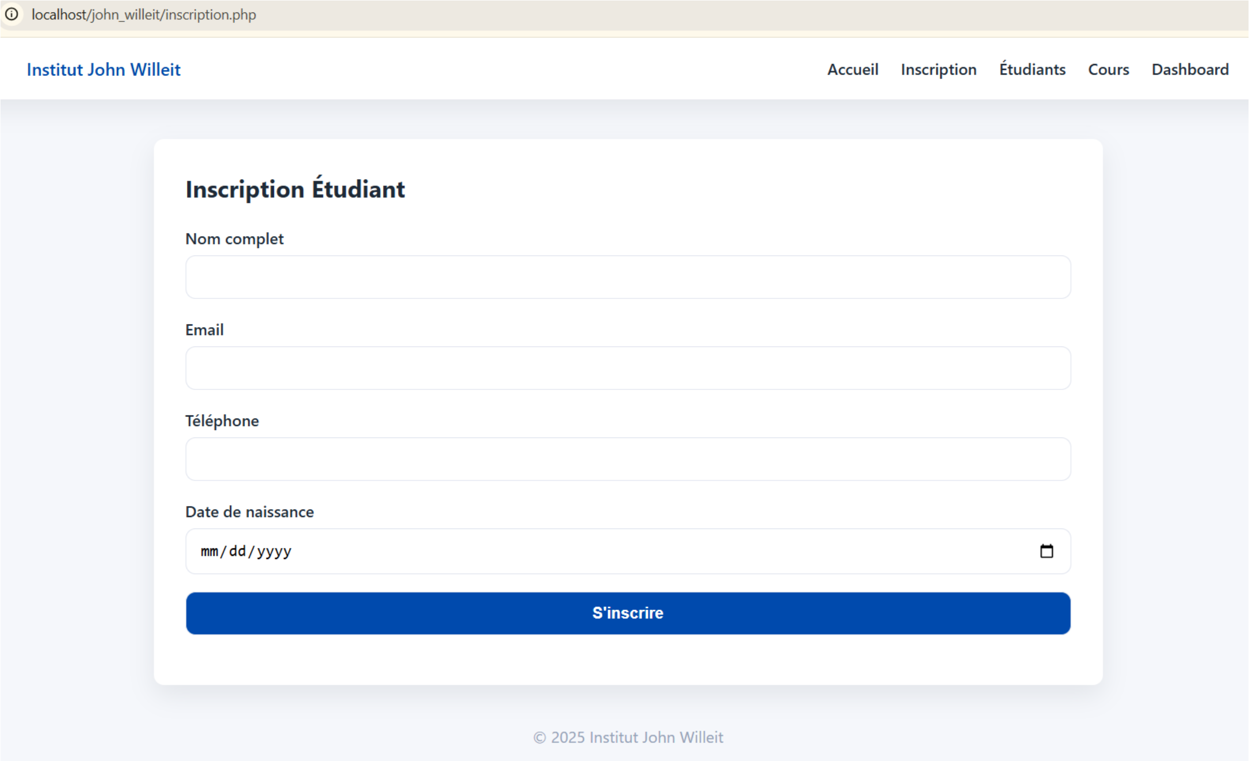
includes/header.php

```html
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Institut John Willeit</title>
    <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>

<nav class="navbar">
    <a href="index.php">Accueil</a>
    <a href="inscription.php">Inscription</a>
    <a href="students.php">Liste Étudiants</a>
    <a href="courses.php">Liste Cours</a>
    <a href="dashboard.php">Dashboard Admin</a>
</nav>

<div class="container">
```

includes/footer.php

```
</div> <!-- end container -->

<footer>
    <p>© 2025 Institut John Willeit</p>
</footer>

</body>
</html>
```

IF everything went ok, you would see something like this



Inscription Étudiant

Nom complet

Email

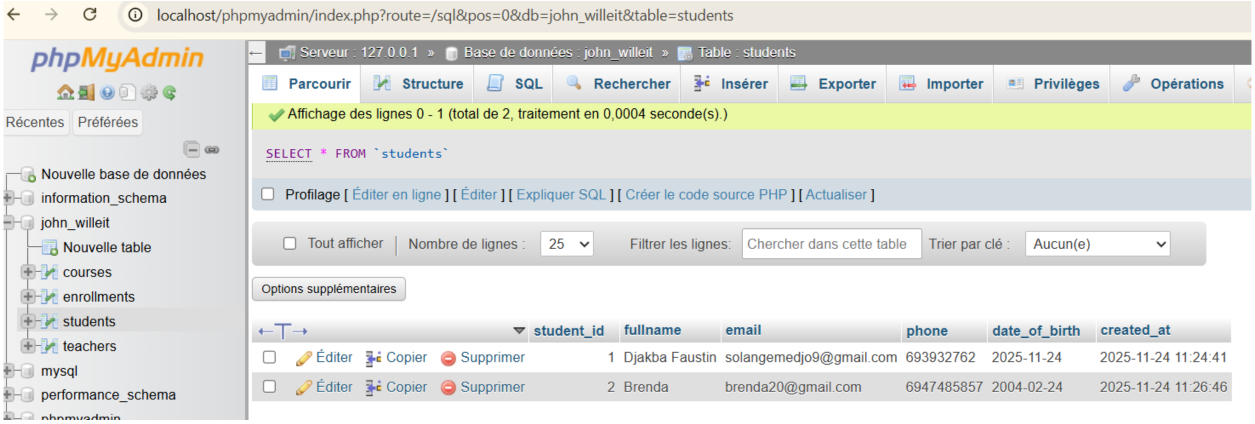
Téléphone

Date de naissance

mm/dd/yyyy

© 2025 Institut John Willeit

For database



✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0004 seconde(s).)

```
SELECT * FROM `students`
```

□ Tout afficher | Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

				student_id	fullname	email	phone	date_of_birth	created_at
□	Éditer	Copier	Supprimer	1	Djakba Faustin	solangemedjo9@gmail.com	693932762	2025-11-24	2025-11-24 11:24:41
□	Éditer	Copier	Supprimer	2	Brenda	brenda20@gmail.com	6947485857	2004-02-24	2025-11-24 11:26:46