

Master d'Université Spécialisé en Ingénierie des Systèmes embarqués

Module: Culture digitale

Chapitre 5: Développement Full-Stack **Application: création d'une page web connectée à** **une base de données**

Année universitaire 2024/2025

Dr. Ing. LASSIOUI Abdellah

Front-End Developer vs Back-End Developer

Aspect	Front-End	Back-End
Ce que c'est	Ce que l'utilisateur voit (l'interface)	Ce qui se passe "derrière" le site (le serveur)
Langages	HTML, CSS, JavaScript	PHP, Python, Node.js, Ruby, Java, ...
Rôle	Afficher les pages, styles, animations	Gérer les bases de données, les comptes...
Exemple	Un formulaire de contact (affichage)	Ce qui se passe quand on clique sur "Envoyer"
Où ça s'exécute	Dans le navigateur web de l'utilisateur	Sur le serveur web (ordinateur distant)

Le full-stack Developer maîtrise à la fois le front-end et le back-end

Introduction au langage PHP

❑ Qu'est-ce que PHP ?

- PHP (Hypertext Preprocessor) est un **langage de script côté serveur**.
- Permet de créer des **pages web dynamiques**.

❑ Structure du code PHP

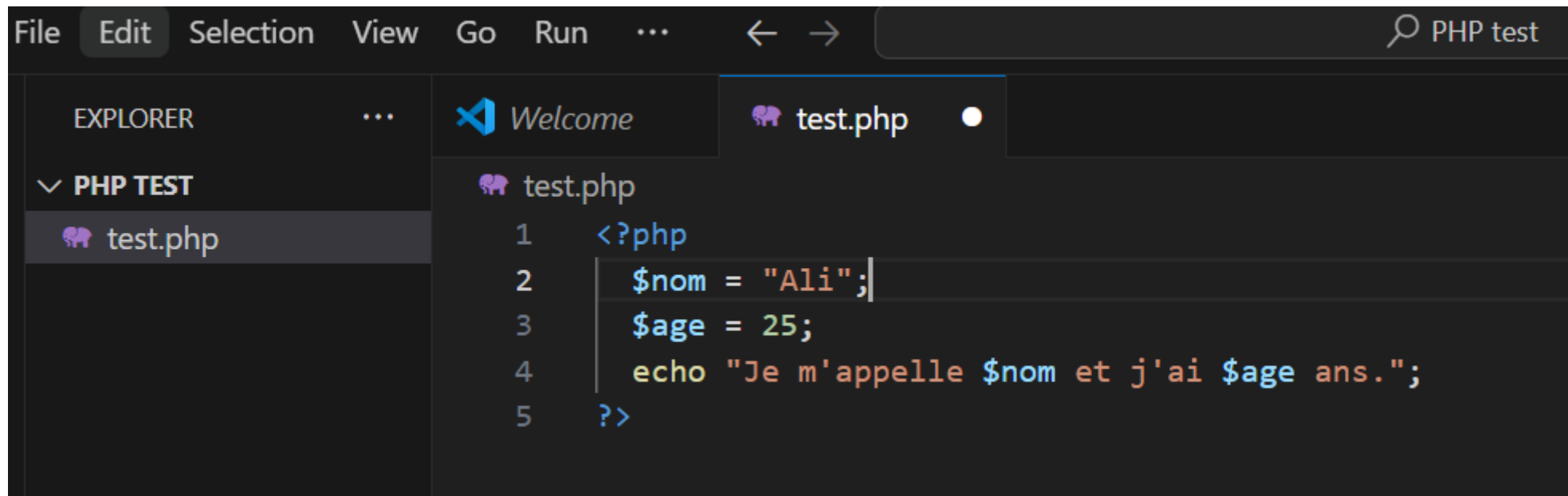
- Un code PHP est placé entre `<?php` et `?>`
- La fonction **echo** sert à afficher une chaîne de caractère dans le navigateur
- Les instructions en php se terminent par un **point virgule**

```
1  <?php
2  echo "Bonjour tout le monde";
3  ?>
```

Introduction au langage PHP

❑ Les variables

- Pour déclarer une variable, on utilise le symbole \$ suivi du nom de la variable
- Le nom de la variable doit **commencer par une lettre ou un underscore (_), sans espace** ni caractère spécial.
- En PHP, **il n'est pas nécessaire de spécifier le type** de la variable.



The screenshot shows a code editor with a dark theme. The menu bar at the top includes File, Edit, Selection, View, Go, Run, and a search icon with the text 'PHP test'. The Explorer panel on the left shows a folder 'PHP TEST' containing a file 'test.php'. The main editor area shows the content of 'test.php' with the following code:

```
1 <?php
2 $nom = "Ali";
3 $age = 25;
4 echo "Je m'appelle $nom et j'ai $age ans.";
5 ?>
```

Introduction au langage PHP

❑ Les opérateurs arithmétiques

- Le tableau suivant résume les différents opérateurs arithmétiques utilisés en langage PHP

Opérateur	Signification	Exemple (\$a = 5, \$b = 2)
+	Addition	$\$a + \$b \rightarrow 7$
-	Soustraction	$\$a - \$b \rightarrow 3$
*	Multiplication	$\$a * \$b \rightarrow 10$
/	Division	$\$a / \$b \rightarrow 2.5$
%	Modulo (reste)	$\$a \% \$b \rightarrow 1$

Introduction au langage PHP

❑ Les opérateurs d'affectation

- Le tableau suivant résume les différents opérateurs d'affectation utilisés en langage PHP

Opérateur	Signification	Exemple
=	Affectation simple	\$a = 5
+=	Ajout et affectation	\$a += 2 (=> 7)
-=	Soustraction et affectation	\$a -= 2 (=> 3)
*=	Multiplication et affectation	\$a *= 2 (=> 10)
/=	Division et affectation	\$a /= 2 (=> 2.5)
%=	Modulo et affectation	\$a %= 2 (=> 1)

Introduction au langage PHP

❑ Les opérateurs de comparaison

- Le tableau suivant résume les différents opérateurs de comparaison utilisés en langage PHP

Opérateur	Signification	Exemple
==	Égal (valeurs égales)	\$a == \$b
===	Identique (valeurs + types)	\$a === \$b
!=	Différent	\$a != \$b
<>	Différent (équivalent à !=)	\$a <> \$b
!==	Pas identique	\$a !== \$b
<	Inférieur	\$a < \$b
>	Supérieur	\$a > \$b
<=	Inférieur ou égal	\$a <= \$b
>=	Supérieur ou égal	\$a >= \$b

Introduction au langage PHP

❑ Les opérateurs de comparaison

- Le tableau suivant résume les différents opérateurs logiques utilisés en langage PHP

Opérateur	Signification	Exemple
&&	ET logique	<code>\$a > 0 && \$b > 0</code>
	Or logique	<code>\$a > 0 \$b > 0</code>
!	NON logique	<code>!(\$a == \$b)</code>

Introduction au langage PHP

❑ Les structures de contrôle

- Les blocs de code sont entre **accolades {}**
- Chaque condition est entre **parenthèses ()**
- L'instruction se termine par un **point-virgule**

❑ Remarques

- On peut mettre un seul else dans une structure if...else.
- Mais tu peux mettre autant de **elseif** que tu veux entre le if et le else.

```
Welcome test.php
test.php
1  <?php
2      if ($age >= 18)
3      {
4          echo "Majeur";
5      }
6      else
7      {
8          echo "Mineur";
9      }
10  ?>
11
```

```
test.php
1  <?php
2      $note = 15;
3
4      if ($note >= 16) {
5          echo "Très bien";
6      } elseif ($note >= 10) {
7          echo "Moyen";
8      } else {
9          echo "Échec";
10     }
11  ?>
12
```

Introduction au langage PHP

❑ Les boucles en php

- Il existe plusieurs types de boucles utilisées en php
- La boucle for
- La boucle while
- La boucle do...while
- La boucle foreach

```
test.php
1  <?php
2  for ($i = 1; $i <= 5; $i++) {
3      echo "Compteur : $i<br>";
4  }
5  ?>
```

```
test.php
1  <?php
2  $i = 1;
3  while ($i <= 5) {
4      echo "Valeur : $i<br>";
5      $i++;
6  }
7  ?>
```

```
test.php
1  <?php
2  $fruits = ["Pomme", "Banane", "Orange"];
3  foreach ($fruits as $fruit) {
4      echo "Fruit : $fruit<br>";
5  }
6  ?>
```

```
test.php
1  <?php
2  $i = 1;
3  do {
4      echo "Valeur : $i<br>";
5      $i++;
6  } while ($i <= 5);
7  ?>
```

Introduction au langage PHP

❑ Les formulaires

- Côté HTML on utilise la balise `<form>` pour créer un formulaire
- `action = "traitement.php"`: indique le fichier php qui va recevoir les données
- `method = "post"`: indique la méthode d'envoi des données

```
<form method="post" action="traitement.php">  
  <input type="text" name="nom">  
  <input type="submit" value="Envoyer">  
</form>
```



```
<?php  
  $nom = $_POST['nom'];  
  echo "Bonjour $nom";  
?>
```

Introduction au langage PHP

❑ PHP et MySQL

- Connexion à la base de données

```
test.php
1  <?php
2  $connexion = mysqli_connect("localhost", "root", "", "ma_base");
3
4  if (!$connexion) {
5      die("Échec de la connexion : " . mysqli_connect_error());
6  }
7  ?>
```

- On crée une variable **connexion** et on utilise la fonction **mysqli_connect** pour accéder à la base de données **ma_base** stockée dans un serveur local (**localhost**) avec un nom d'utilisateur **root** et un **mot de passe vide**
- **die**: est une fonction qui affiche un message et arrête l'exécution du script php

Introduction au langage PHP

❑ PHP et MySQL

- Insertion des données

```
$nom = "Ali";  
$requete = "INSERT INTO utilisateurs (nom) VALUES ('$nom')";  
mysqli_query($connexion, $requete);
```

- Dans ce script php, on crée deux variables: **nom** (chaîne de caractères) et **requete** (requête SQL)
- On utilise la fonction **mysqli_query** pour insérer les données dans la base de données utilisateurs

Introduction au langage PHP

❑ PHP et MySQL

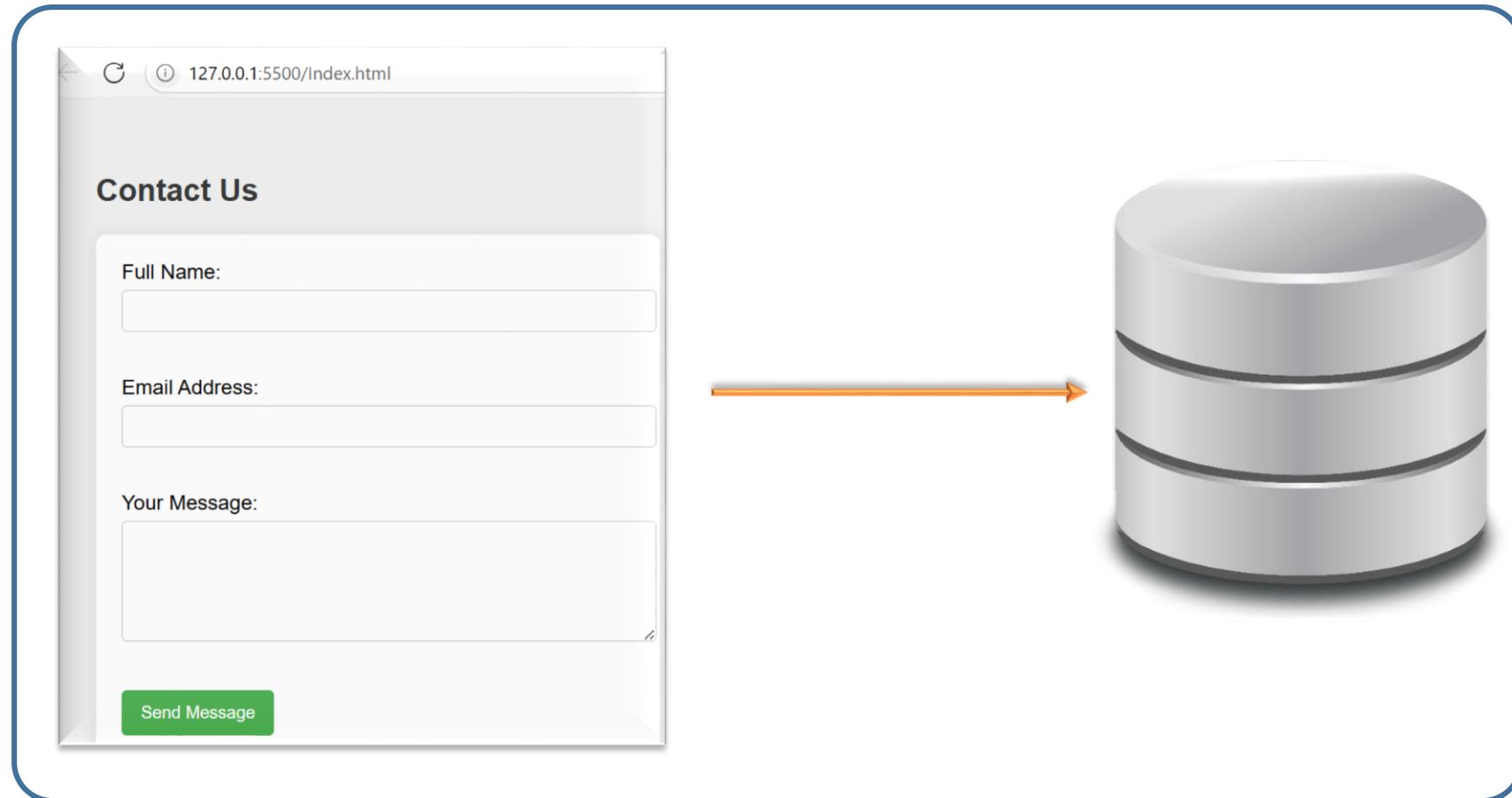
- lecture des données

```
$requete = "SELECT * FROM utilisateurs";  
$resultat = mysqli_query($connexion, $requete);  
  
while ($ligne = mysqli_fetch_assoc($resultat)) {  
    echo "Nom : " . $ligne['nom'] . "<br>";  
}
```

- Dans ce script php, on crée deux variables: **requête** (requête SQL) et **resultat** (résultat de la requête SQL)
- On utilise la fonction **mysqli_query** pour récupérer les données à partir la base de données utilisateurs
- On crée une variable **ligne** à l'intérieur d'une boucle **while**. Tant que cette variable est non nulle, on affiche son contenu à travers la fonction **echo**

Projet: Création d'un formulaire avec stockage de données dans une base de données

- L'objectif de cette application est de montrer le processus de développement full-Stack d'une application web
- L'idée est de créer un formulaire en ligne permettant à un utilisateur de saisir des informations comme (le nom, prénom, email, message....) et de collecter ses informations pour les stocker dans une base de données crée par MySQL



Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Technologies à utiliser

▪ Front-End

- **Langage HTML**: pour la création et la structuration de la page web
- **Langage CSS**: pour styliser la page web

▪ Back-End

- **MySQL** : Pour la création de la base de données
- **Langage PHP**: pour la collecte des données des utilisateurs et la connexion avec la base de données

Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Structure du projet

▪ Dossier: Formulaire

- `index.html`: pour la création et la structuration de la page web
- `style.css`: pour styliser la page web
- `submit.php`: script php pour manipuler les données
- `db_config.php`: script php pour se connecter avec la base de données
- `contact_db.db`: base de données stockant les données des utilisateurs

▪ Remarque:

- Tout ces fichiers doivent êtres mis dans le même dossier

Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Étape 1: Création de la base de données

```
CREATE DATABASE contact_db;
```

```
USE contact_db;
```

```
CREATE TABLE messages (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    full_name VARCHAR(100),
```

```
    email VARCHAR(100),
```

```
    message TEXT,
```

```
    submitted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
```

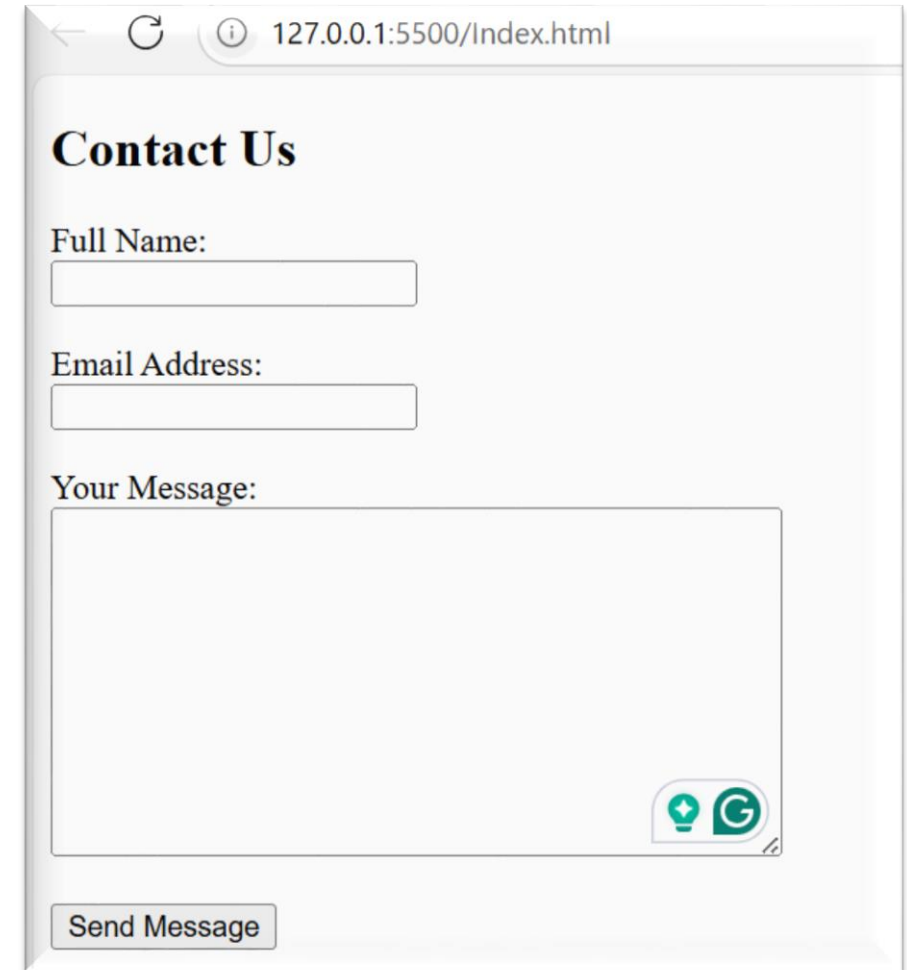
Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Étape 2: Création du formulaire (index.html)

Le formulaire que nous désirons créer est illustré dans la figure suivante.

Il est constitué de champs suivants:

- Un titre **Contact Us**
- Un champ de texte pour saisir le nom et le prénom (**Full Name**)
- Un champ de texte pour saisir l'adresse email (**Email Address**)
- Un champ de texte pour saisir le message (**Your Message**)
- Un bouton pour soumettre le message (**Send Message**)



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5500/Index.html'. The page title is 'Contact Us'. The form contains three input fields: 'Full Name:', 'Email Address:', and 'Your Message:'. The 'Your Message:' field is a large text area. At the bottom right of the form is a small icon with a green plus sign and a green 'G'. Below the form is a 'Send Message' button.

Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Étape 2: Création du formulaire (index.html)

```
<> Index.html > html > body > h2
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Contact Form</title>
6      <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9      <h2>Contact Us</h2>
10     <form action="submit.php" method="POST">
11         <input type="text" name="full_name" placeholder="Full Name" required><br><br>
12         <input type="email" name="email" placeholder="Email Address" required><br><br>
13         <textarea name="message" rows="5" placeholder="Your Message" required></textarea><br><br>
14         <input type="submit" value="Send Message">
15     </form>
16 </body>
17 </html>
```

Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Étape 3: Création d'un code css (style.ht ml)

```
✓ FULL STACK
  db_config.php
  <> Index.html
  # style.css
  submit.php
  view_messages.php

# style.css > body
1  body {
2      font-family: Arial, sans-serif;
3      background-color: #f0f0f0;
4      padding: 30px;
5  }
6  h2 {
7      color: #333;
8  }
9  form {
10     background-color: white;
11     padding: 20px;
12     border-radius: 8px;
13     max-width: 400px;
14     box-shadow: 0 0 10px rgba(0,0,0,0.1);
15 }
16 input[type="text"],
17 input[type="email"],
18 textarea {
19     width: 100%;
20     padding: 8px;
21     margin-top: 5px;
22     margin-bottom: 15px;
23     border: 1px solid #ccc;
24     border-radius: 4px;
```

```
26 input[type="submit"] {
27     background-color: #4CAF50;
28     color: white;
29     padding: 10px 15px;
30     border: none;
31     border-radius: 4px;
32     cursor: pointer;
33 }
34 input[type="submit"]:hover {
35     background-color: #45a049;
36 }
37
```

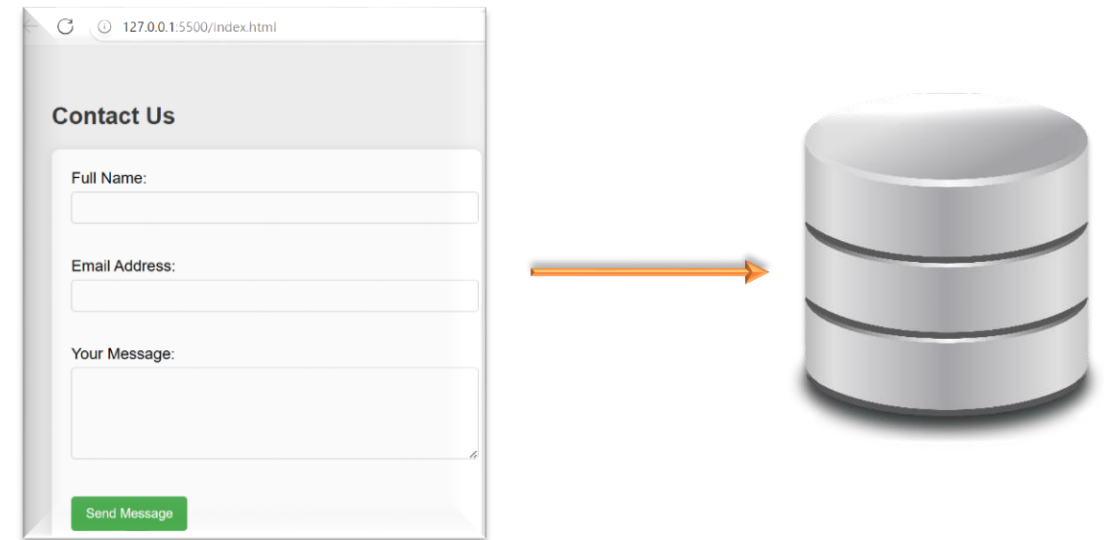
Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Étape 4: Création d'un code PHP (submit.php)

L'objectif de cette étape est de créer un fichier PHP permettant de collecter les données à partir du formulaire et de les stocker dans la base de données

Cette étape est réalisée en trois sous étapes:

- Connexion à la base de données
- Recevoir les données à partir du formulaire en utilisant la méthode POST
- Insérer les données à la table messages de la base de données contact_db



Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Étape 4: connexion à la base de données (db_config.php)

Cette portion

```
db_config.php
1  <?php
2  $host = "localhost";
3  $username = "root";
4  $password = "";
5  $database = "contact_db";
6
7  // Création d'une connexion
8  $conn = new mysqli($host, $username, $password, $database);
9
10 // Test de la connexion
11 if ($conn->connect_error) {
12     die("Connection failed: " . $conn->connect_error);
13 }
14 ?>
```

Projet: Création d'un formulaire avec stockage de données dans une base de données

❑ Étape 4: connexion à la base de données (submit.php)

Ce fichier PHP permet de collecter les données à partir du formulaire, puis il prépare la requête sql après il teste si les données ont été insérées correctement dans la base de données ou non.

Un message de succès ou d'erreur sera affiché par la suite

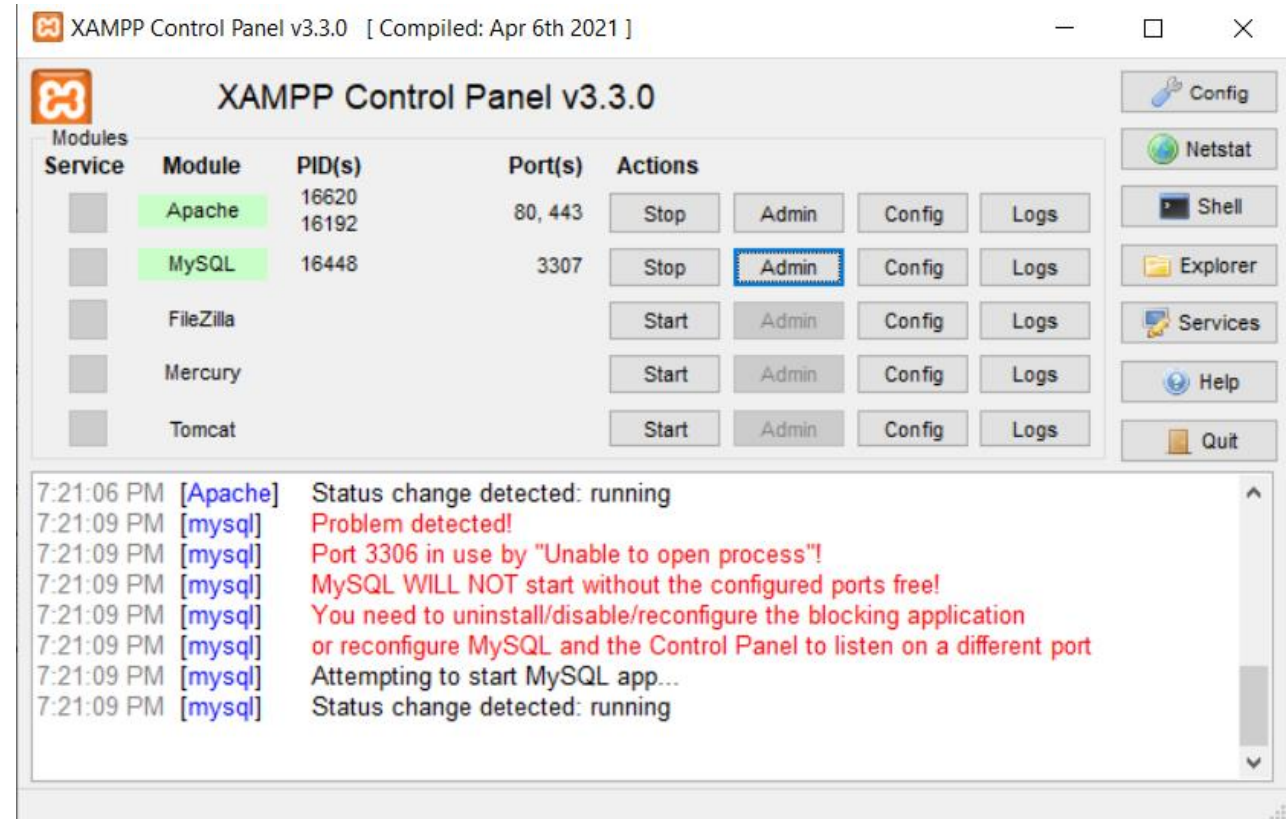
```
submit.php
1  <?php
2  // se connecter à la base de données
3  include 'db_config.php';
4  // récupérer les données et les mettre dans des variables
5      $name = $_POST['full_name'];
6      $email = $_POST['email'];
7      $message = $_POST['message'];
8
9      // créer une variable sous forme de requête sql
10     $sql = "INSERT INTO messages (full_name, email, message)
11           VALUES ('$name', '$email', '$message')";
12
13     // exécuter la requête, puis générer un message ok ou non
14     if (mysqli_query($conn, $sql)) {
15         echo "Thank you! Your message has been sent.";
16     } else {
17         echo "Error: " . mysqli_error($conn);
18     }
19
20     // terminer la connexion avec la base de données
21     mysqli_close($conn);
22     ?>
```


Test du projet

Outils à installer

XAMPP est un environnement de développement web gratuit et open-source qui regroupe plusieurs outils essentiels pour créer et tester localement des sites web. Il inclut principalement :

- ❑ **Apache** (serveur web)
- ❑ **MySQL** ou MariaDB (système de gestion de base de données)
- ❑ **PHP** (langage de script côté serveur)
- ❑



Utilisation principale : XAMPP permet aux développeurs de simuler un serveur web local sur leur ordinateur pour développer, tester et déboguer des applications web sans avoir besoin d'un serveur en ligne.

Outils à installer (suite)

❑ **Apache** (ou **Apache HTTP Server**) est un **logiciel** installé sur un ordinateur (appelé **serveur**) qui **écoute les requêtes HTTP** envoyées par les navigateurs (comme Chrome, Firefox) et **répond avec le contenu demandé**, souvent une page HTML.

Il agit comme un **intermédiaire** entre l'utilisateur et les fichiers du site web:

- ❑ **reçoit les requêtes** des navigateurs
- ❑ **trouve les fichiers demandés** (HTML, PHP, images...)
- ❑ **exécute les scripts si besoin (PHP)**
- ❑ **renvoie la réponse** au navigateur

Installation de XAMPP (suite)

❑ XAMPP inclut également d'autres outils utiles pour les développeurs d'applications

Web notamment: [FileZilla](#), [Mercury](#) et [Tomcat](#).

Outil	Fonction principale
FileZilla	Transfert de fichiers via FTP
Mercury	Serveur pour envoyer/recevoir des e-mails
Tomcat	Hébergement d'applications web Java

Configuration de XAMPP

Étape 1 : Modifier le port dans XAMPP

- ❑ Ouvre XAMPP Control Panel.
- ❑ Clique sur le bouton “Config” à côté de MySQL.
- ❑ Choisis “my.ini” → un fichier s’ouvre dans un éditeur de texte.
- ❑ Rechercher les lignes suivantes :

port=3306 et remplace les par

port=3307

The image shows the XAMPP Control Panel v3.3.0 window. The 'MySQL' service is selected, and the 'Config' button is highlighted with a red arrow. A second red arrow points from the 'Config' button to a Notepad window titled 'my - Notepad'. The Notepad window displays the contents of the 'my.ini' file. Two sections of the file are highlighted with red boxes: the first box contains the lines '# password = your_password' and 'port=3307', and the second box contains the lines '[mysqld]' and 'port=3307'. The Notepad window also shows a log of events in the background, including 'Tomcat Started' and 'Make sure you check the "/>

Configuration de XAMPP

Étape 2 : Modifier le port dans le fichier de configuration de PHPMyAdmin

- ❑ Allez dans le dossier d'installation de XAMPP

C:\xampp\phpMyAdmin\

- ❑ Ouvrez le fichier : **config.inc.php**

- ❑ Trouvez la ligne :

`$cfg['Servers'][$i]['USER'] = '';`

- ❑ Remplacez comme indiqué dans la capture d'écran

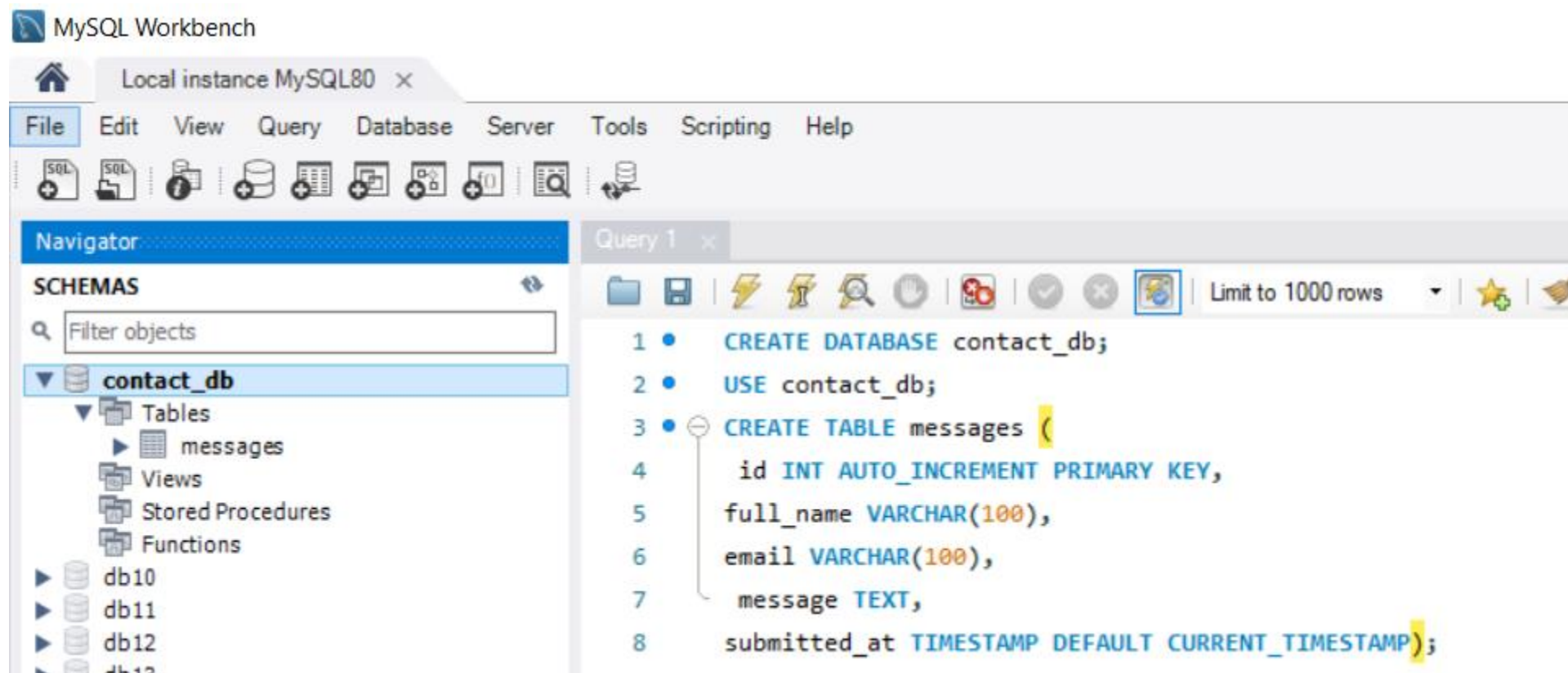
```
17
18  /* Authentication type and info */
19  $cfg['Servers'][$i]['auth_type'] = 'config';
20  $cfg['Servers'][$i]['user'] = 'LASSIOUI';
21  $cfg['Servers'][$i]['password'] = ' ';
22  $cfg['Servers'][$i]['extension'] = 'mysqli';
23  $cfg['Servers'][$i]['AllowNoPassword'] = true;
24  $cfg['Lang'] = '';
25
26  /* Bind to the localhost ipv4 address and tcp */
27  $cfg['Servers'][$i]['host'] = '127.0.0.1';
```

- ❑ Le nom d'utilisateur « user » ainsi que le mot de passe doivent être les mêmes que ceux utilisés lors de l'installation de MySQL Workbench.
- ❑ Cela permet d'accéder à la base de données Library_DB créée lors du chapitre précédent

Création de la base de données

Pour créer la base de données, vous pouvez utiliser deux methods:

❑ **Méthode 1: en utilisant le SGBD MySQL comme le montre la figure suivante:**



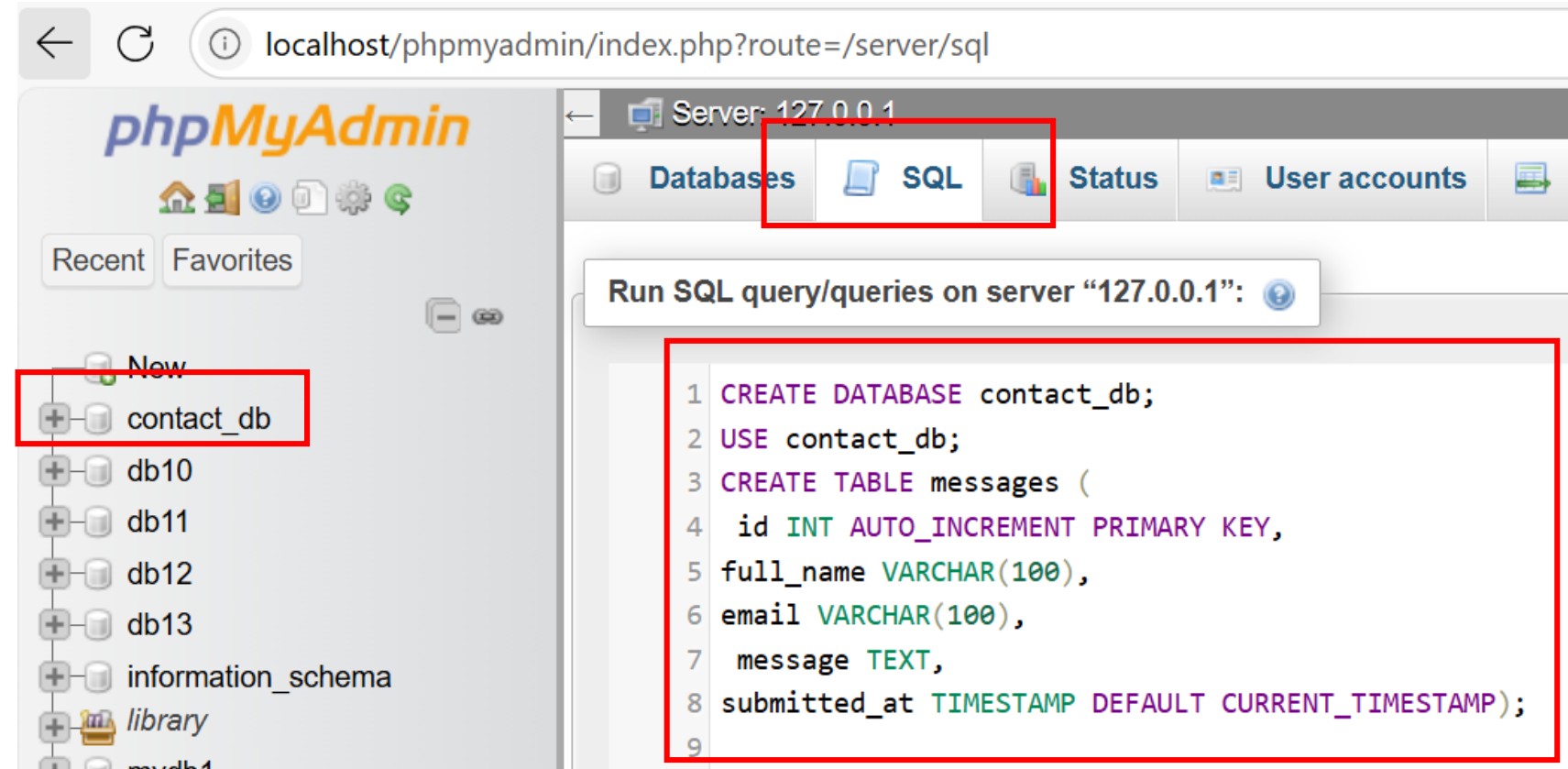
Création de la base de données

Pour créer la base de données, vous pouvez utiliser deux methods:

❑ **Méthode 2: en utilisant l'onglet SQL de XAMPP comme le montre la figure suivante:**

La figure suivante illustre la création de la base de données dans phpMyAdmin

- Sélectionner l'onglet **SQL**
- Taper le code SQL
- Cliquer sur **Go** pour exécuter le code
- La base de données sera créer et s'affiche à gauche



Configuration de XAMPP

Mettre tous les fichiers dans le dossier suivant:

C:\xampp\htdocs\Form

← → ▾ ↑ > This PC > OS (C:) > xampp > htdocs > Form					
Program Files Program Files (x86) ProgramData Recovery Ross-Tech ti tidcs	Name	Date modified	Type	Size	
	db_config	6/19/2025 10:08 PM	PHP Source File	1 KB	
	Index	6/18/2025 9:50 PM	Microsoft Edge HT...	1 KB	
	style	6/17/2025 12:19 PM	Cascading Style Sh...	1 KB	
	submit	6/19/2025 9:52 PM	PHP Source File	1 KB	
	view_messages	6/17/2025 12:34 PM	PHP Source File	2 KB	

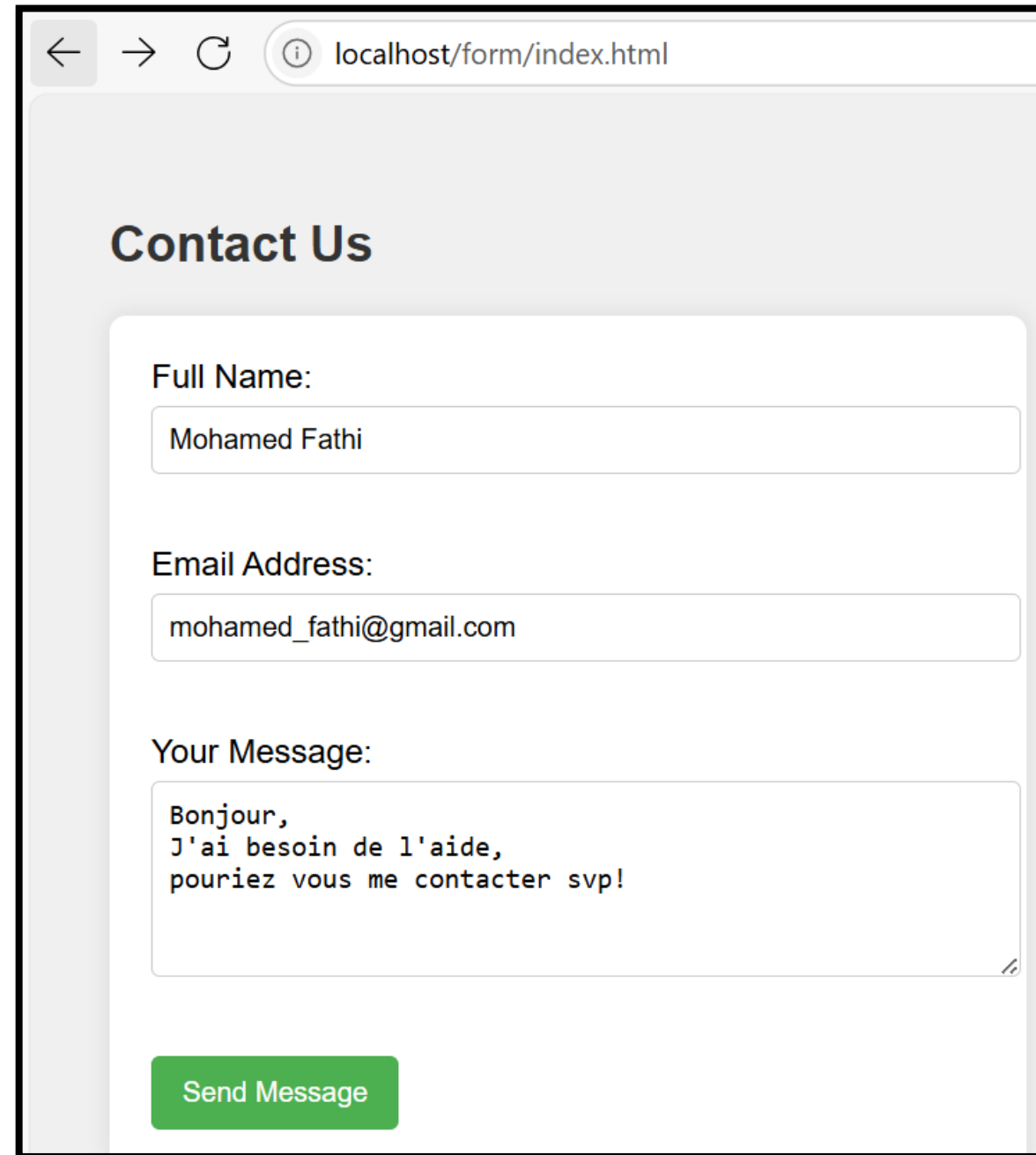
Test de l'application

Ouvrir le navigateur et taper l'adresse suivante:

<http://localhost/form/index.html>

Remplir les champs de la page web et appuyer sur le bouton **Send Message**

Si tout marche bien, un message de succès sera affiché



The screenshot shows a web browser window with the address bar displaying 'localhost/form/index.html'. The page has a light gray background and a white form titled 'Contact Us'. The form contains three input fields: 'Full Name' with the value 'Mohamed Fathi', 'Email Address' with the value 'mohamed_fathi@gmail.com', and 'Your Message' with the text 'Bonjour, J'ai besoin de l'aide, pouriez vous me contacter svp!'. A green 'Send Message' button is at the bottom of the form.

← → ↻ ⓘ localhost/form/index.html

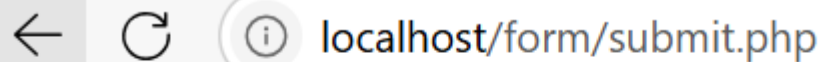
Contact Us

Full Name:

Email Address:

Your Message:

Send Message



The screenshot shows a web browser window with the address bar displaying 'localhost/form/submit.php'. The page content is a simple white box with the text 'Thank you! Your message has been sent.'.

← ↻ ⓘ localhost/form/submit.php

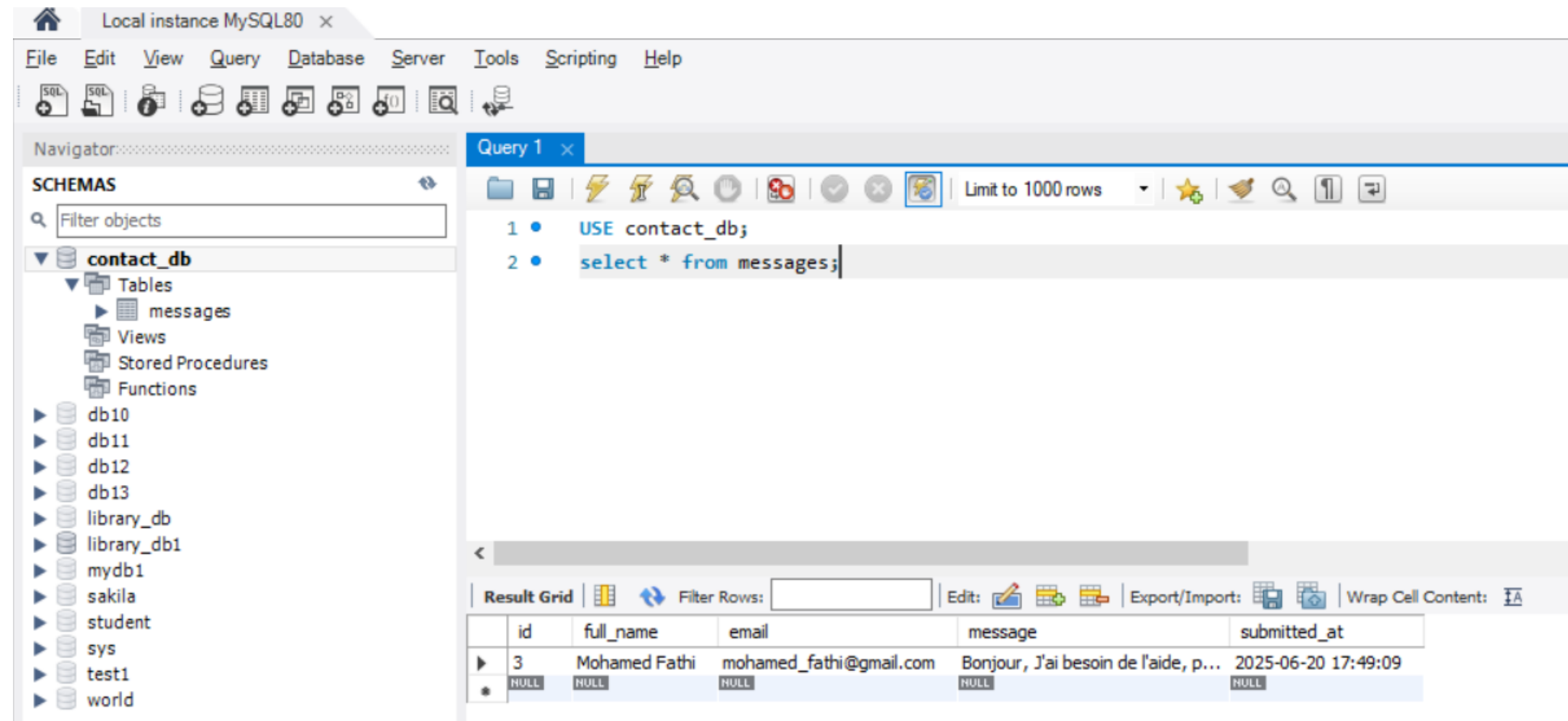
Thank you! Your message has been sent.

Test de l'application

Appliquer la requette suivante dans MySQL et visualiser le contenu de la base de données

Use contact_db;

Select * from messages;



The screenshot shows the MySQL Workbench interface for a local instance of MySQL 8.0. The 'Navigator' pane on the left displays the 'SCHEMAS' tree, with 'contact_db' expanded to show its 'Tables' (including 'messages'), 'Views', 'Stored Procedures', and 'Functions'. The 'Query' editor in the center contains two lines of SQL: 'USE contact_db;' and 'select * from messages;'. The 'Result Grid' at the bottom displays the query results in a table format. The table has five columns: 'id', 'full_name', 'email', 'message', and 'submitted_at'. The first row shows a message with id 3, full_name 'Mohamed Fathi', email 'mohamed_fathi@gmail.com', message 'Bonjour, J'ai besoin de l'aide, p...', and submitted_at '2025-06-20 17:49:09'. A second row is partially visible with NULL values.

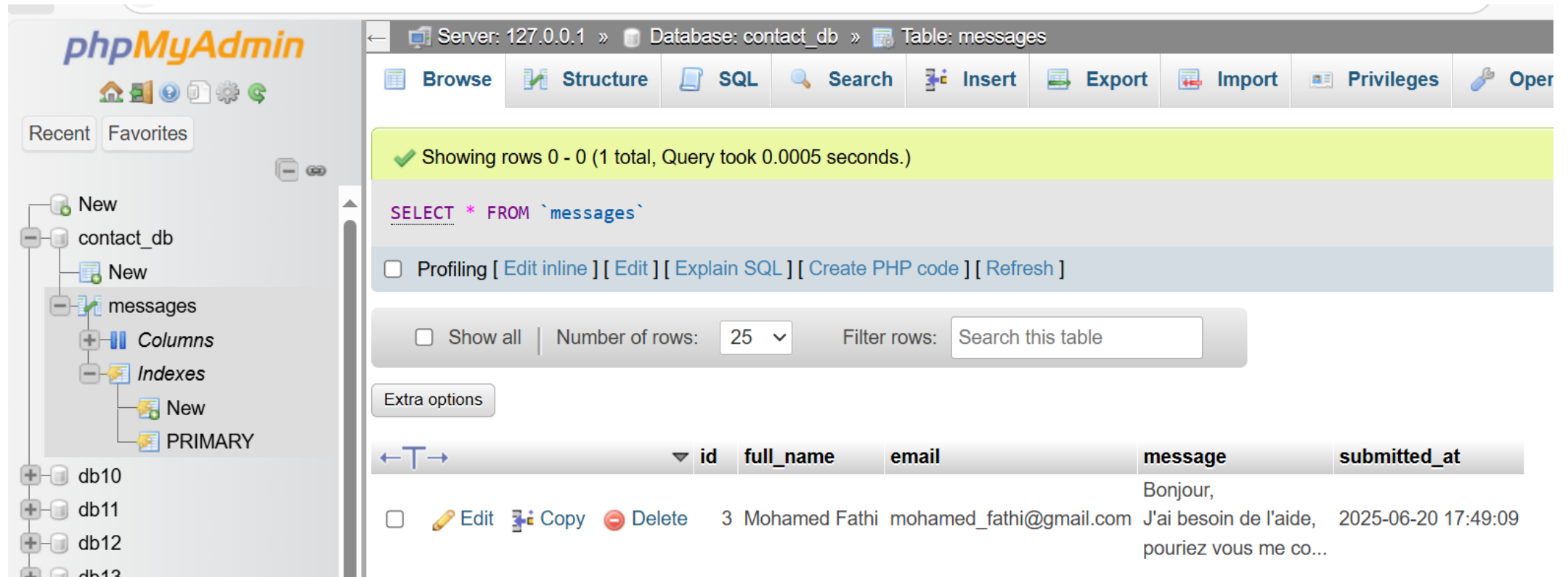
	id	full_name	email	message	submitted_at
▶	3	Mohamed Fathi	mohamed_fathi@gmail.com	Bonjour, J'ai besoin de l'aide, p...	2025-06-20 17:49:09
*	NULL	NULL	NULL	NULL	NULL

Test de l'application

Appliquer la requette suivante dans SQL et visualiser le contenu de la base de données

Use contact_db;

Select * from messages;



The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, showing 'contact_db' with a 'messages' table. The main panel displays the SQL query results for the 'messages' table. The query executed is 'SELECT * FROM `messages`'. The results show one row with the following data:

id	full_name	email	message	submitted_at
3	Mohamed Fathi	mohamed_fathi@gmail.com	Bonjour, J'ai besoin de l'aide, pouriez vous me co...	2025-06-20 17:49:09