

# AU WEB DESIGNER PROJECT DOCUMENTATION

## 1. Introduction

This documentation outlines the steps for implementing a contact form on a website, including designing the form, processing submissions with PHP, and storing the data in a MySQL database. It also includes guidelines for integrating the form into a website and ensuring its functionality.

## 2. Project Overview

The contact form project consists of creating an HTML form, processing the form submissions with PHP, and storing the data in a MySQL database. This form will allow users to submit their contact details and messages, which will be captured and stored for further processing.

## 3. Contact Form Design

### 3.1 HTML Form

The contact form is designed using HTML. The form collects the user's name, email, and message. The form is placed within an HTML file named contact.php:

Html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Contact Us</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <header>

        <!-- Include header content if necessary -->

    </header>


    <main>

        <h1>Contact Us</h1>
```

```

<form action="contact_process.php" method="POST">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>

    <label for="message">Message:</label>
    <textarea id="message" name="message" rows="4" required></textarea>

    <button type="submit">Send Message</button>
</form>
</main>

<footer>
    <?php include 'footer.php'; ?>
</footer>
</body>
</html>

```

## 4. Form Processing

### 4.1 PHP Script

To handle form submissions, create a PHP file named `contact_process.php`. This script processes the form data and inserts it into a MySQL database:

```

php
<?php
// Database configuration

$servername = "localhost"; // Replace with your database server
$username = "root";        // Replace with your database username
$password = "";            // Replace with your database password

```

```
$dbname = "contact_form"; // Replace with your database name

// Create a connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check the connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Retrieve form data

$name = htmlspecialchars($_POST['name']);
$email = htmlspecialchars($_POST['email']);
$message = htmlspecialchars($_POST['message']);

// Prepare and bind

$stmt = $conn->prepare("INSERT INTO messages (name, email, message) VALUES (?, ?, ?)");
$stmt->bind_param("sss", $name, $email, $message);

// Execute the statement
if ($stmt->execute()) {
    echo "<p>Message sent successfully!</p>";
} else {
    echo "<p>Error: " . $stmt->error . "</p>";
}

// Close the statement and connection
$stmt->close();
```

```
$conn->close();
```

```
?>
```

## **5. Database Setup**

### **5.1 Create the Database**

Create a MySQL database named `contact_form`. Inside this database, create a table named `messages` to store the contact form data:

sql

```
CREATE TABLE messages (  
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    message TEXT NOT NULL,  
    reg_date TIMESTAMP  
);
```

## **6. Integration into Website**

### **6.1 Including the Contact Form**

Ensure that the `contact.php` form is properly linked within your website. For instance, include a link to the contact form from your main navigation or other relevant pages.

### **6.2 Including the Footer**

To maintain a consistent look across your website, include the footer using PHP:

php

```
<footer>
```

```
<?php include 'footer.php'; ?>
```

```
</footer>
```

## **7. Conclusion**

This documentation provides a complete guide to setting up a contact form, processing submissions with PHP, and storing data in a MySQL database. Following these steps ensures that the contact form is functional, secure, and integrated seamlessly into your website. Make sure to test the form thoroughly and verify that the data is correctly captured and stored in the database.