PROJECT DOCUMENTATION

The theme I choose for the project is call Wedding Photography.

I stated building my code by first doing the **html pages (Home).**

In the HTML page, I created five different kinds of pages.

1. **Home**

The home page is all about what the wedding Photography will look like. There are images and logo for you to have an idea of how the page is going to look like.

2. **Contact**

The contact page is to give you the information about how every individual can reach out to us.

3. **Gallery**

The gallery is to showcase some of the picture we have from most of the work done by us.

4. **Service**

It talks about the services we give to all our customers and how we do those services without any complains.

5. **About:** That is if anyone needs more information.

This page show a more detailed on everything the page is about and the images presented give the website a clear view of how it is. Regarding the images it make the page looks more nice and colorful.

At the end of the page I use a < strong > < strong /> to make it look more bold, so people can see what I am talking about.

php pages (Service).

The **Services Page** class is a specialized subclass of the Page class, used to generate a dynamic web page displaying various services related to a wedding photography business. The class primarily extends the functionalities of the Page class by adding additional buttons and a content section for a list of services.

The page.php file is included, which contains the base class Page. This base class likely provides methods for rendering common elements like titles, keywords, styles, headers, menus, and footers.

The page.php file is included, which contains the base class Page. This base class likely provides methods for rendering common elements like titles, keywords, styles, headers, menus, and footers.

The row2buttons array holds a collection of service names (keys) mapped to corresponding service pages (values). These are additional buttons that will be rendered in the page's menu.

The content property of the $services object is set to a paragraph describing the available services.

The Display() method is called on the $services object, rendering the complete HTML page to the browser.

The Services Page class extends the functionality of the base Page class to generate a specific services page with additional buttons and customizable content. This modular approach ensures reusability and a clear separation of concerns, making it easier to maintain and update.

php pages (processeorder).

This PHP script processes an order form for a wedding photography service, calculates the total cost, applies tax, and writes the order to a file. It also generates a basic HTML page displaying the order summary and the shipping address.

Form Input: Retrieves and sanitizes form data from an HTML POST request, converting the values of Customersqty, picturesqty, and colorsqty into integers.

Address: The address field is sanitized by removing any tab or newline characters.

Document Root: Stores the root directory of the server in $document_root, useful for file operations.

Date: Captures the current date and time in the format Hour:Minute, Day Month Year.

This section provides the basic HTML structure, with headers for the order results page.

Order Date: Displays the date and time when the order was processed.

Order Details: Outputs the message "Your order is as follows:".

Total Quantity: Calculates the total number of items ordered by summing up the quantities of customers, pictures, and colors.

Pricing Constants: Defines the prices of each item:

- Customer price: $100
- Picture price: $10
- Color price: $4

Total Amount: Calculates the total amount for the order based on the item quantities and their respective prices.

Subtotal: Displays the subtotal (before tax) of the order.

Tax Calculation: A 10% tax is applied to the subtotal.

Total with Tax: Displays the final total including the tax.

File Handling: The script attempts to open the orders.txt file for appending ('ab' mode) within the parent directory (../orders/).

File Locking: Uses flock() to ensure exclusive access to the file while writing.

Error Handling: If the file cannot be opened, it displays an error message and exits the script.

Writing to File: The order string is written to the file, and the file is then closed.

Confirmation: Once the order is successfully written to the file, a message confirming the order is displayed.

Order Validation: Validates if any items were ordered and provides feedback if the order is empty.

Total Price Calculation: Computes the total cost, including item pricing and tax.

Order Logging: Logs the order details to a server file for record-keeping.

User-Friendly Messages: Provides clear feedback on the success or failure of the order processing.

This script processes a customer's wedding photography order by calculating totals, applying taxes, and storing the order in a log file. It also provides feedback to the user in a simple HTML page.

php pages (orderform).

This is an HTML form designed to collect order details for a wedding photography service. The form allows users to input the quantity of customers, pictures, background colors, and their shipping address. When the form is submitted, it sends the data to a PHP script (processorder.php) for processing.

DOCTYPE: Specifies that this is an HTML5 document.

Title: The page title is set to "WEDDING PHOTOGRAPHY - Order Form" and appears in the browser tab.

Form Action: When the form is submitted, the data is sent to processorder.php using the POST method.

Method: The post method is used to send form data securely to the server.

Table Layout: The form is structured using a table to align the input fields for better readability.

Header Row: The table has two columns labeled "Item" and "Quantity." The background of the header row is light gray (#cccccc), and the text is center-aligned.

Form Action: The form data is sent via POST to processorder.php.

Input Validation: The size and maxlength attributes on the input fields help to restrict the length of user input, aiding in data validation.

User-Friendly Layout: The table structure provides an organized, easy-to-read layout for entering order details.

Submit Button: Users can submit their order by clicking the "Submit Order" button, which triggers the data submission.

This form provides a simple, structured way for users to place an order for wedding photography services. It collects data on the number of customers, pictures, background colors, and the shipping address, and sends this data for processing when submitted.