Home Project “Interior Design Studio Website”

**1. Introduction**

This document provides a comprehensive overview of the Interior Design Studio website, a platform designed to offer information about the studio's services, showcase its portfolio, and facilitate client engagement for interior renovation, refurbishment, and rehabilitation projects. The website aims to provide a user-friendly interface for potential customers seeking professional interior design services for their apartments, offices, factories, and other facilities.

**2. Target Audience**

The primary target audience for this website includes:

* Individuals and businesses planning interior renovation or refurbishment projects.
* Property owners and managers seeking professional interior design expertise.
* Anyone interested in viewing the studio's portfolio and understanding their design process.

**3. Website Features and Functionality**

The website consists of the following key sections and functionalities:

**3.1. Homepage (index.html)**

* **Navigation:** A clear and consistent top navigation bar provides access to all main sections of the website (HOME, PORTFOLIO, SERVICES, PROCESS, EDUCATION).
* **Hero Section:** Likely a visually engaging section at the top, potentially showcasing key services or featured projects to immediately capture visitor attention.
* **About Section:** Introduces the Interior Design Studio, highlighting their expertise, philosophy, and unique selling points.
* **Services Overview:** Briefly outlines the main service categories offered by the studio.
* **Portfolio Teaser:** A glimpse into the studio's past projects, encouraging users to explore the full portfolio.
* **Contact Information/Call to Action:** Provides clear ways for potential clients to get in touch with the studio.
* **Footer:** Contains copyright information and potentially contact details.

**3.2. Portfolio (Gallery.html)**

* **Visual Showcase:** This page displays a collection of the studio's completed interior design projects through images.
* **CSS Grid Layout:** Utilizes CSS Grid (Gridstyle.css) to present the portfolio in a visually appealing and organized manner.
* **Project Variety:** Showcases the studio's experience across different types of facilities (residential, commercial, etc.).
* **Potentially Clickable Images:** Users might be able to click on the images to view more details about specific projects (though this functionality isn't explicitly in the provided HTML).
* **Navigation:** Consistent top navigation for easy access to other sections.
* **Footer:** Contains copyright information.

**3.3. Services (Services.html)**

* **Detailed Service Descriptions:** This page provides in-depth information about the various services offered by the studio, such as design, art studio, turnkey projects, and technical specifications.
* **Service Packages:** Presents different service packages (Express, PRO, Business) tailored to varying client needs and project scopes.
* **"SELECT" Button Functionality:** Each service package has a "SELECT" button that triggers a **pop-up window** for payment details.
  + **Payment Pop-up:** A modal (paymentPopupOverlay) appears, displaying payment method options (Credit Card, PayPal, Other).
  + **AJAX Payment Processing:** Upon selecting a payment method and clicking "Proceed with Payment," an AJAX request (/process\_payment\_ajax) is sent to the server (app.py) to *simulate payment processing* for the selected service.
* **Artwork Purchase Carousel:** Features a carousel showcasing *individual artworks available for one-time purchase*.
  + **"BUY NOW" Button Functionality:** Each artwork has a "BUY NOW" button that triggers a separate pop-up window for artwork purchase.
  + **Artwork Purchase Pop-up:** A modal (artworkPurchasePopupOverlay) appears, displaying payment method options specifically for artwork purchase.
  + **AJAX Artwork Purchase Processing:** Upon selecting a payment method and clicking "Proceed with Purchase," an AJAX request (/process\_artwork\_purchase\_ajax) is sent to the server (app.py) to simulate processing the artwork purchase.
* **Video Integration:** Includes an embedded YouTube iframe, likely showcasing the studio's work or testimonials.
* **Navigation:** Consistent top navigation.
* **Footer:** Contains copyright information.

**3.4. Process (Link in Navigation)**

* **(Content Not Provided):** This page would likely detail the studio's **interior design process**, outlining the typical stages involved in a project from initial consultation to final execution.

**3.5. Education (Link in Navigation)**

* **(Content Not Provided):** This page might offer resources, articles, or information related to interior design, potentially positioning the studio as knowledgeable in the field. For corporate clients having IT departments, there are options for consultations and if requested the client’s employees can be offered courses on frameworks and techniques applied for web applications.

**4. Technology Stack**

The website is built using the following technologies:

* **Frontend:**
  + **HTML:** Provides the structural foundation of the web pages.
  + **CSS:** (style.css, Gridstyle.css) Styles the visual presentation of the website, including layout (likely using CSS Grid for the portfolio), colors, typography, and responsiveness.
  + **JavaScript:** (services.js) Implements interactive elements such as the payment and artwork purchase pop-up windows, handles user interactions with buttons, and makes asynchronous requests (AJAX) to the backend.
  + **Bootstrap:** (Included via CDN links in Services.html) A CSS framework used for styling and providing components like the navigation bar, cards, and the carousel.
  + **Bootstrap Icons:** (Included via CDN link in Services.html) A library of icons used for visual enhancements.
* **Backend:**
  + **Python:** The server-side logic is implemented using the Python programming language.
  + **Bottle:** A lightweight Python web framework used to handle routing (app.py) and serve static files.

**5. Backend Functionality (app.py)**

The app.py file defines the server-side routes and logic:

* **/ (Homepage):** Serves the index.html file.
* **/Gallery.html:** Serves the Gallery.html file.
* **/Services.html:** Serves the Services.html file.
* **/select\_payment (POST):** Handles the submission of the service selection from a form (though a form isn't directly present in the provided Services.html - likely used in an earlier iteration). It renders a payment\_options.html template (not provided).
* **/process\_payment\_ajax (POST):** An AJAX endpoint that receives the selected service and payment method. It currently returns a placeholder message indicating that payment processing is not fully implemented.
* **/process\_artwork\_purchase\_ajax (POST):** An AJAX endpoint that receives the selected artwork and payment method. It currently returns a placeholder message indicating that artwork purchase processing is not fully implemented.
* **/<filename:path>:** A generic route to serve static files (CSS, JavaScript, images, etc.).
* **/style.css:** Specifically serves the style.css file.
* **/Gridstyle.css:** Specifically serves the Gridstyle.css file.
* **/services.js:** Specifically serves the services.js file.

**6. Frontend JavaScript (services.js)**

This file contains the JavaScript code for the Services.html page:

* **Handles Service Payment Pop-up:**
  + Selects all elements with the class select-button (for service packages).
  + Attaches a click event listener to each "SELECT" button to:
    - Prevent default link behavior.
    - Retrieve the selected service from the data-service attribute.
    - Update the text of the payment pop-up title.
    - Set the value of a hidden input field with the selected service.
    - Display the paymentPopupOverlay.
  + Provides functions (closePopup, submitPayment) to close the payment pop-up and handle the simulated payment submission via AJAX.
* **Handles Artwork Purchase Pop-up:**
  + Selects all elements with the class buy-now-button (for artworks).
  + Attaches a click event listener to each "BUY NOW" button to:
    - Prevent default link behavior.
    - Stop the click event from propagating to parent elements (to prevent carousel navigation).
    - Retrieve the selected artwork name from the data-artwork attribute.
    - Update the text of the artwork purchase pop-up title and display the artwork name.
    - Set the value of a hidden input field with the selected artwork.
    - Display the artworkPurchasePopupOverlay.
  + Provides functions (closeArtworkPopup, submitArtworkPurchase) to close the artwork purchase pop-up and handle the simulated artwork purchase submission via AJAX.

**7. Installation and Setup (Based on Prerequisites)**

1. **Install Bottle:**

Bash

pip3 install bottle

or

Bash

wget https://github.com/bottlepy/bottle/raw/master/bottle.py

(If using wget, ensure the bottle.py file is in the same directory as your HTML, CSS, and JavaScript files).

1. **Save Website Files:** Ensure all the provided files (index.html, Gallery.html, Services.html, style.css, Gridstyle.css, services.js, app.py) are saved in the same directory on your server or local machine.
2. **Run the Bottle Application:** Open your terminal or command prompt, navigate to the directory where you saved the files, and run the app.py script: