Project Report

on

**Wall Painting Visualization**

***Submitted to***

**Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur**

In partial fulfilment of the requirement of

**Master in Computer Application Part-I [Semester-II] (CBCS)**

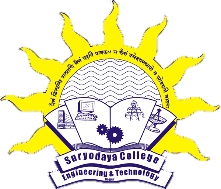
Examination Summer-2024

***Submitted By***

Chaitali S. Ghugal

***Under the Guidance of***

**Prof. Rupesh Bangre**



**Department of MCA**

**SURYODAYA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**[MCA Programme]**

Survey No. 117/1 & 117/3 A, Near Dighori Naka, Vihirgaon, Umrer Road,

Nagpur, Maharashtra 441204.

**2024-2025**

Certificate

This is to certify that the project report on **“Wall Painting Visualization”** is submitted by Chaitali S. Ghugal for partial fulfilment of the requirement of **Master in Computer Application Part-I[Semester-II] (CBCS)** examination of the Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur.

It is the original Software project carried out under the supervision and guidance of **Prof. Rupesh Bangre** and undergone requisite duration as prescribed by Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur for the project work.

## Guide

**Prof. Rupesh Bangre**

**SCET**

##### HOD

##### (Prof. Ajay D. Nanwatkar)

##### SCET

# **Internal Examiner External Examiner**

### **Place : Nagpur**

**Date :**

**DECLARATION**

To,

**The Principal,**

**Suryodaya College of Engineering & Technology,**

**Near Dighori Naka, Vihirgaon, Umrer Road Nagpur.**

Respected Sir,

We the undersigned hereby declare that the work **“Wall Painting Visualization”** developed and submitted by us is our original work. The system presented here, is developed by us is independently and has not been duplicated from any other source.

We understand that any such copying is liable to be punished in any way the University deem fit.

**Place:** Nagpur **(Chaitali S. Ghugal)**

**Date:**

**ACKNOWLEDGEMENT**

#### We wish to express our sincere thanks to the honorable **Dr. V.G.Arajpure**, Principal, **Suryodaya College of Engineering & Technology**, Nagpur and to the **Prof AjayD. Nanwatkar** H.O.D., Department of MCA for providing us varieties of opportunities, infrastructural facilities and inspiration to gather professional knowledge and material without which it would have been impossible to complete this hard task.

#### We take this opportunity to express my/our deep gratitude and whole hearted thanks to my/our project guide **Prof. Rupesh Bangre** for her guidance throughout this work. Weare very much thankful for their kindness, encouragement and valuable time, which they have devoted to me/us.

We wish to thanks all those, who have helped us in our way or the others in bringing out this project successful.

**Chaitali S. Ghugal**

***CONTENTS***

**Sr. No. CHAPTER**   **Page No**

1. **BRIEF REVIEW OF THE PROJECT**
2. TITLE
3. INTRODUCTORY / OBJECTIVES
4. PRELIMINARY INVESTIGATION
5. FLAWS IN PRESENT SYSTEM
6. NEED OF NEW SYSTEM
7. **DETAILED SYSTEM DESIGN**
8. SYSTEM FLOW CHART

2.2 DATA FLOW DIAGRAMS

2.3 DATA DICTIONARY

1. **SOFTWARE/ HARDWARE DETAILS**
2. CHOICE OF A LANGUAGE USED
3. HARDWARE/ SOFTWARE SPECIFICATION

**4. SYSTEM DESIGN**

1. PROGRAM LISTING
2. INPUT SCREEN

**5. USER DOCUMENTATION**

1. IMPLEMENTATION, PROGRAM EXECUTION & MAINTENANCE

**6.**  **CONCLUSION**

1. LIMITAIONS OF THE SYSTEM
2. SCOPE AND FUTURE MODIFICATION

**7.** **REFRENCES / BIBLIOGRAPHY**

# **1.BRIEFRE VIEW OF THE PROJECT**

## 1.1TITLE

**“WALL PAINTING VISUALIZATION”**

**1.2 INTRODUCTORY / OBJECTIVES**

**Introduction:**

Introducing a website dedicated to wall painting visualization! Have you ever struggled to envision how a certain colour or design would look on your walls before committing to it? Say goodbye to the uncertainty and hello to our innovative platform. Our website offers an interactive experience where you can digitally paint your walls in various colours and patterns, allowing you to see the results in real time. Whether you’re considering a bold accent wall or a subtle shade for the entire room, our tool lets you explore endless possibilities from the comfort of your home.

**Objectives of wall painting Applications:**

Objective for Wall Painting Visualization Website The objective of this thesis is to explore the development and impact of wall painting visualization websites, focusing on their technological foundations, user experience, and practical applications in the interior design industry.

**Specifically, the thesis aims to:**

* Analyze the Technological Framework: Investigate the underlying technologies, including computer vision, 3D rendering, and user interface design, that enable the functionality of wall painting visualization websites.
* Evaluate User Experience: Assess the usability and effectiveness of these platforms in helping users make informed decisions about paint colours and finishes, considering factors such as ease of use, accuracy of colour representation, and overall user satisfaction.
* Examine Industry Applications: Explore how interior designers, architects, and paint manufacturers utilize these websites to enhance client consultations, improve customer engagement, and increase sales.
* Identify Future Trends: Predict potential advancements in wall painting visualization technology, such as augmented reality integration and AI-driven colour recommendations, and their implications for the future of interior design.

**1.3 PRELIMINARY INVESTIGATION**

Creating a website for wall painting visualization involves several key steps, from initial research and planning to design and implementation. Here's a structured approach for conducting a primary investigation for your thesis:

* Define Objectives and Scope
* Objective: To create a user-friendly website that allows users to visualize different wall paintings in their spaces.
* Scope: Determine the types of wall paintings (e.g., murals, framed art, decals).
* Identify the target audience (e.g., homeowners, interior designers).
* Market Analysis
* Competitor Analysis: Research existing websites that offer wall painting visualizations (e.g., Art.com, Wall Pops, Wayfair).
* Identify their strengths and weaknesses.
* Trends: Look into current trends in home decor and online visualization tools.
* User Research
* Surveys and Interviews: Conduct surveys and interviews with potential users to understand their needs, preferences, and pain points.
* User Personas: Create user personas based on the collected data to represent different segments of your audience.
* Technical Requirements
* Features: Image upload for room visualization.
* Painting selection and placement. Customization options (e.g., size, frame type).
* Technologies: Web development frameworks (e.g., React, Angular).
* Image processing libraries (e.g., OpenCV).Backend services (e.g., Node.js, Django).
* Design and Prototyping
* Wireframes and Mockups: Create wireframes and mockups for the website layout and user interface.
* User Flow: Design the user flow to ensure intuitive navigation.
* Prototyping Tools: Use tools like Figma, Sketch, or Adobe XD to create interactive prototypes.
* Development Frontend
* Development: Implement the user interface using HTML, CSS, and JavaScript frameworks.
* Backend Development: Set up server-side logic, databases, and APIs to support the frontend.
* Integration: Integrate image processing capabilities to allow users to visualize paintings in their rooms.
* Testing User Testing: Conduct usability testing with real users to gather feedback and make improvements.
* Technical Testing: Perform functionality, performance, and security testing to ensure the site runs smoothly.
* Deployment Hosting: Choose a hosting service (e.g., AWS, Heroku) and deploy the website.
* Maintenance: Plan for ongoing maintenance, updates, and user support.
* Documentation and Thesis
* Writing Documentation: Document the development process, user research findings, and technical details.

**Thesis Structure:**

Introduction: Define the problem and objectives.

* Literature Review: Discuss related work and market analysis.
* Methodology: Explain the design, development, and testing process.
* Results: Present findings from user research and testing.
* Discussion: Analyze the results and their implications.
* Conclusion: Summarize the project and suggest future work.
* Tools and Resources
* Project Management: Trello, Asana
* Design Tools: Figma, Adobe XD, Sketch
* Development Tools: VS Code, GitHub
* Testing Tools: Selenium, Jest

By following these steps, you can conduct a thorough primary investigation for your thesis on a wall painting visualization website. This structured approach will help ensure that your project is well-researched, user-centric, and technically sound.

**1.4 FLAWS IN PRESENT SYSTEM**

Despite the significant advancements and benefits of wall painting visualization websites, several flaws and limitations persist in the current systems. These shortcomings can impact user experience, accuracy, and overall effectiveness. Identifying and addressing these flaws is crucial for the continued improvement and adoption of these platforms. The primary flaws in present wall painting visualization systems are outlined below:

**1. Limited Realism and Accuracy**

* **Colour Representation**: Many visualization systems struggle with accurately representing colours due to variations in display screens, image processing algorithms, and lighting conditions. This can lead to discrepancies between the virtual preview and the actual paint colour when applied on walls.
* **Texture and Finish Simulation**: Simulating different paint finishes (e.g., matte, gloss, satin) and wall textures accurately is challenging. Current systems often provide only a basic representation, which may not fully convey the visual and tactile differences.

**2. User Interface and Experience Issues**

* **Complexity**: Some platforms have complex interfaces that can be overwhelming for novice users. The need for multiple steps to upload images, select colours, and adjust settings can be cumbersome and deter user engagement.
* **Limited Customization Options**: Many systems offer limited customization, such as the inability to adjust for different types of lighting or to see how colours change throughout the day. This can restrict users' ability to fully explore their options.

**3. Technical Limitations**

* **Image Processing**: Accurate detection and segmentation of walls in uploaded images remain a significant challenge. Poor segmentation can result in unrealistic previews, with paint colours bleeding into areas they shouldn't, such as furniture or floors.
* **Performance and Speed**: High-quality rendering and processing of images can be resource-intensive, leading to slow performance, especially on devices with limited computational power.

**4. Lack of Integration with Other Design Tools**

* **Interoperability**: Many visualization websites operate as standalone tools and lack integration with other interior design software and smart home technologies. This fragmentation can hinder a holistic design approach, where paint choices are considered alongside furniture, decor, and lighting.

**5. Accessibility and Inclusivity**

* **Device Compatibility**: Some platforms are not optimized for all devices, particularly mobile phones and tablets. This limits accessibility for users who prefer or need to use these devices.
* **Language and Regional Limitations**: Limited language options and regional settings can make these tools less accessible to a global audience, reducing their usability for non-English speakers or those in different geographic regions.

**6. Cost and Availability**

* **Subscription Models**: Some advanced features are often locked behind paywalls or subscription models, limiting access for users who are unwilling or unable to pay. This creates a barrier for casual users who may benefit from the full range of features.
* **Limited Paint Brand Options**: Many visualization tools are tied to specific paint brands, restricting the variety of colors and finishes available to users. This can be a significant limitation for those looking to explore a broader range of products.

While wall painting visualization websites offer valuable tools for interior design, several flaws hinder their effectiveness and user satisfaction. Addressing these issues through technological advancements, improved user interface design, better integration with other tools, and greater accessibility can significantly enhance the utility and adoption of these platforms. By overcoming these challenges, wall painting visualization websites can provide more accurate, realistic, and comprehensive solutions for users seeking to transform their living spaces.

**1.5 NEED OF NEW SYSTEM**

The development and widespread use of wall painting visualization websites have revolutionized the process of selecting paint colors and finishes. However, the existing systems have several limitations that affect their accuracy, usability, and overall effectiveness. To address these shortcomings and meet the evolving needs of users, there is a pressing need for a new, improved wall painting visualization system. This section outlines the key reasons why a new system is necessary and the potential benefits it can bring.

**1. Enhanced Accuracy and Realism**

* **Colour Fidelity**: Current systems often struggle with accurate color representation due to variations in display screens and image processing limitations. A new system should employ advanced algorithms and calibration techniques to ensure colors are represented as accurately as possible across different devices.
* **Texture and Finish Simulation**: To provide a more realistic preview, the new system should incorporate sophisticated rendering techniques that accurately simulate different paint finishes and wall textures. This would help users better understand the visual and tactile impact of their choices.

**2. Improved User Experience**

* **Intuitive Interface**: The new system should prioritize a user-friendly interface that simplifies the process of uploading images, selecting colours, and customizing settings. An intuitive design will make the tool accessible to both novice and experienced users.
* **Advanced Customization**: Users should be able to adjust for different lighting conditions, view colour changes throughout the day, and apply multiple colours and finishes to different sections of a room. Enhanced customization options will provide a more comprehensive visualization experience.

**3. Technical Advancements**

* **Robust Image Processing**: Implementing advanced computer vision techniques can improve the accuracy of wall detection and segmentation in uploaded images. This will result in more precise and realistic previews.
* **Performance Optimization**: The new system should leverage cloud computing and optimized algorithms to ensure fast and responsive performance, even on devices with limited computational power.

**4. Integration with Other Design Tools**

* **Interoperability**: To offer a holistic design solution, the new system should integrate seamlessly with other interior design software and smart home technologies. This will enable users to coordinate paint choices with furniture, decor, and lighting for a cohesive design.
* **Collaboration Features**: Adding features for real-time collaboration and sharing will allow users to work with designers, family members, or friends, enhancing the overall design process.

**5. Greater Accessibility and Inclusivity**

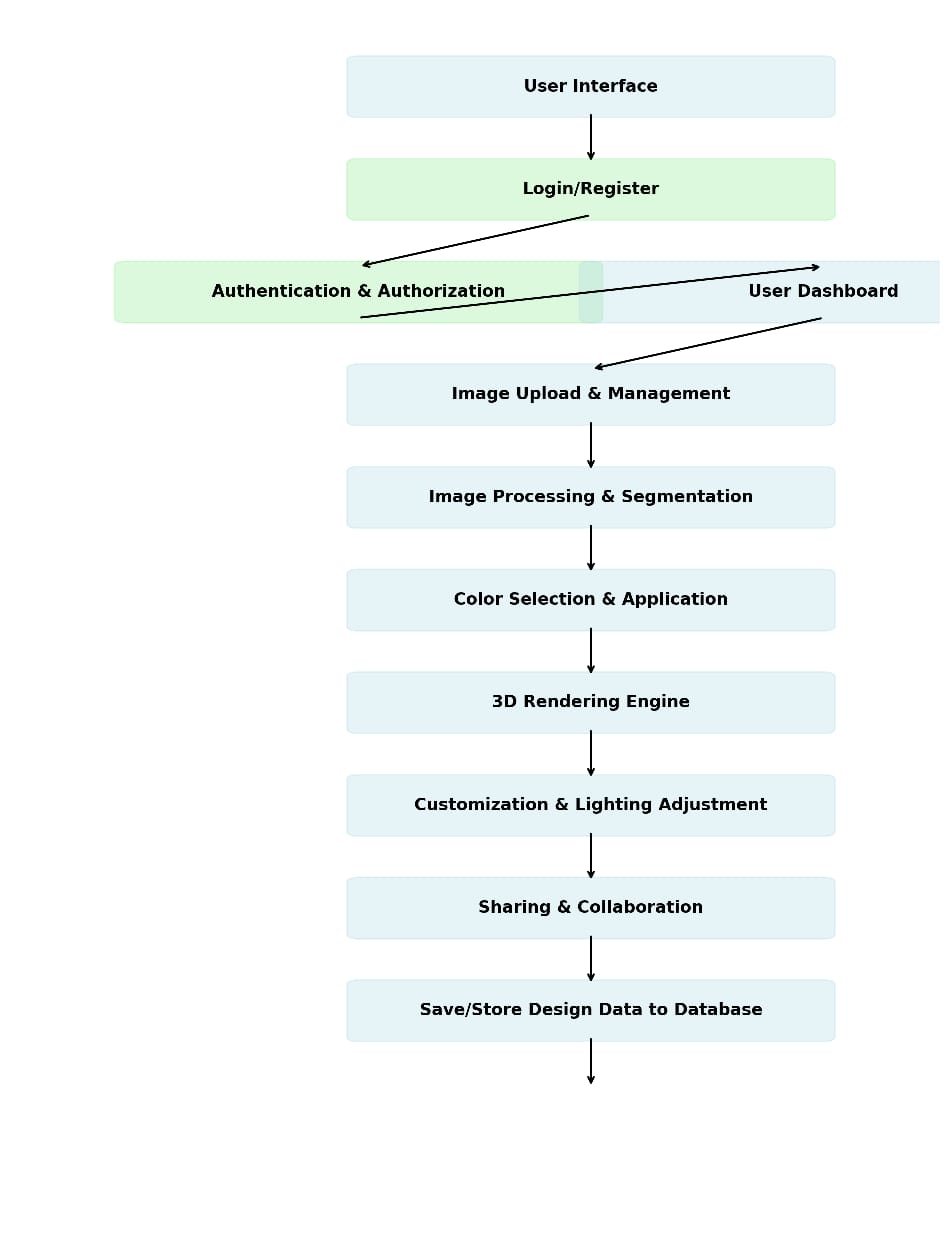
* **Device Compatibility**: The new system should be optimized for use on a wide range of devices, including smartphones, tablets, and desktops, ensuring accessibility for all users.
* **Multilingual Support**: Incorporating multiple languages and regional settings will make the tool more inclusive, catering to a global audience.

**6. Expanded Paint Options and Cost Efficiency**

* **Broad Paint Brand Integration**: The new system should include a wide variety of paint brands and products, providing users with more options to choose from. This can be achieved through partnerships with multiple paint manufacturers.
* **Cost-Effective Solutions**: Offering a range of pricing models, including free access to basic features and affordable subscriptions for advanced tools, will make the system accessible to a broader audience.

**2. DETAILED SYSTEM DESIGN**

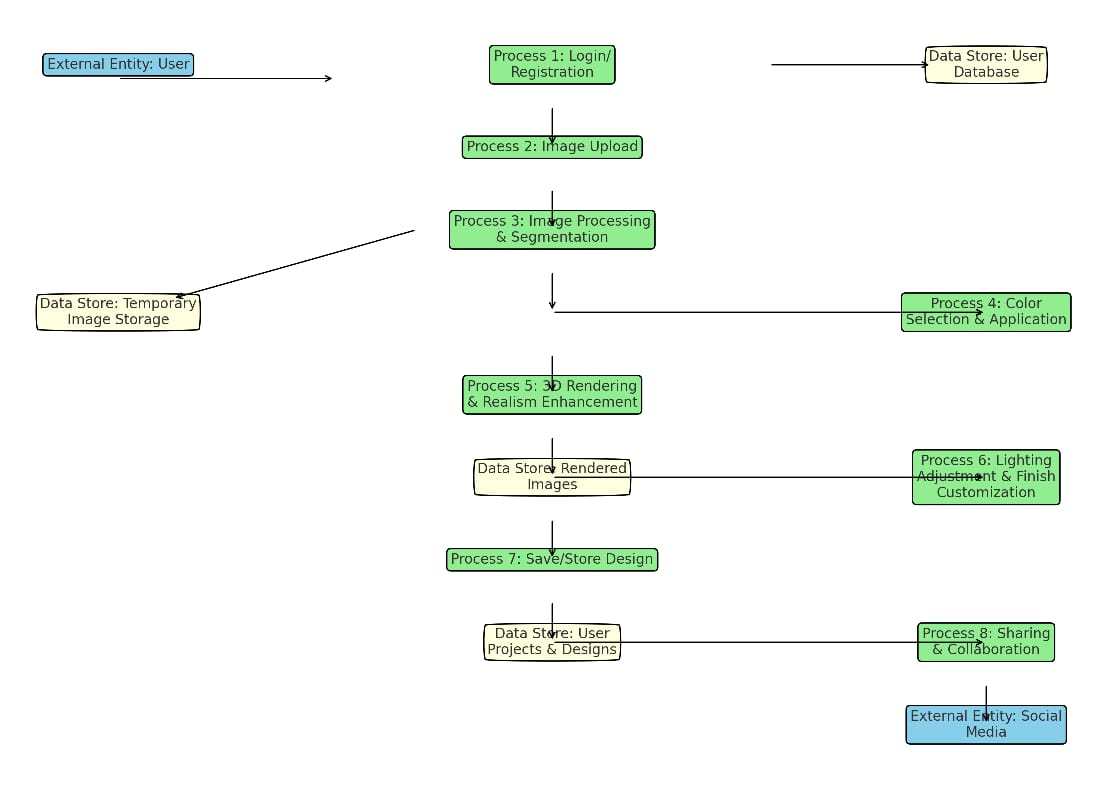
**2.1 SYSTEM FLOW CHART**



Flow chart of the wall painting visualization website, detailing the system design and data flow:

* **External Entity: User**:
* Actions: Upload Image, Select Color, View Design, Save Project, Share Design.
* **Process 1: Login/Registration**:
* Handles user authentication and account creation.
* **Data Store: User Database**:
* Stores user credentials and profile information.
* **Process 2: Image Upload**:
* Allows users to upload room images for visualization.
* **Process 3: Image Processing & Segmentation**:
* Processes the uploaded image to segment the walls from other elements.
* **Data Store: Temporary Image Storage**:
* Temporarily holds the processed image for further modifications.
* **Process 4: Color Selection & Application**:
* Users select paint colors which are applied to the segmented image.
* **Process 5: 3D Rendering & Realism Enhancement**:
* Enhances the image to provide a realistic preview of the paint color on the walls.
* **Data Store: Rendered Images**:
* Stores the rendered images for user previews.
* **Process 6: Lighting Adjustment & Finish Customization**:
* Allows users to adjust lighting and finish to see how colors look under different conditions.
* **Process 7: Save/Store Design**:
* Saves the finalized design for future reference or modifications.
* **Data Store: User Projects & Designs**:
* Stores user projects and designs securely.
* **Process 8: Sharing & Collaboration**:
* Facilitates sharing of designs via social media and collaboration with others.
* **External Entity: Social Media**:
* Platform for sharing the visualized designs.

This flow chart provides a clear and detailed visual representation of the data flow and system design for your wall painting visualization website project.



**2.4 DATA DICTIONARY**

**User Table:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| user\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each user |
| username | VARCHAR(50) | UNIQUE, NOT NULL | User's chosen username |
| password\_hash | VARCHAR(255) | NOT NULL | Hashed password for security |
| email | VARCHAR(100) | UNIQUE, NOT NULL | User's email address |
| first\_name | VARCHAR(50) | NOT NULL | User's first name |
| last\_name | VARCHAR(50) | NOT NULL | User's last name |
| registration\_date | DATETIME | NOT NULL | Date and time when the user registered |
| last\_login | DATETIME | NULL | Date and time of the user's last login |

**Image Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| image\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each uploaded image |
| user\_id | INT | FOREIGN KEY | Identifier of the user who uploaded the image |
| image\_path | VARCHAR(255) | NOT NULL | File path to the stored image |
| upload\_date | DATETIME | NOT NULL | Date and time when the image was uploaded |
| segmented | BOOLEAN | DEFAULT 0 | Indicates if the image has been processed for wall segmentation |

**Project Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| project\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each project |
| user\_id | INT | FOREIGN KEY | Identifier of the user who created the project |
| project\_path | VARCHAR(100) | NOT NULL | Name of the project |
| created\_date | DATETIME | NOT NULL | Date and time when the project was created |
| segmented | DATETIME | NULL | Date and time of the last modification |

**Colour Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** | **Description** |
| color\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each colour |
| Color\_name | VARCHAR(50) | NOT NULL | Name of the color |
| hex\_value | CHAR(7) | NOT NULL | Hexadecimal color code (e.g., #FFFFFF) |
| brand | VARCHAR(50) | NOT NULL | Paint brand associated with the color |

**o DATABASE FILE**

* **Create Users Table**

**CREATE TABLE Users (**

**user\_id SERIAL PRIMARY KEY,**

**username VARCHAR(50) UNIQUE NOT NULL,**

**email VARCHAR(100) UNIQUE NOT NULL,**

**password\_hash VARCHAR(255) NOT NULL,**

**created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP**

**);**

* **Create Projects Table**

**CREATE TABLE Projects (**

**project\_id SERIAL PRIMARY KEY,**

**user\_id INTEGER REFERENCES Users(user\_id),**

**project\_name VARCHAR(100),**

**created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP**

**);**

* **Create Images Table**

**CREATE TABLE Images (**

**image\_id SERIAL PRIMARY KEY,**

**project\_id INTEGER REFERENCES Projects(project\_id),**

**image\_url VARCHAR(255),**

**created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP**

**);**

* **Create Colors Table**

**CREATE TABLE Colors (**

**color\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**color\_name VARCHAR(50),**

**hex\_value VARCHAR(7),**

**manufacturer VARCHAR(100)**

**);**

* **Create ProjectColors Table**

**CREATE TABLE ProjectColors (**

**project\_color\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**project\_id INT,**

**color\_id INT,**

**applied\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**FOREIGN KEY (project\_id) REFERENCES Projects(project\_id),**

**FOREIGN KEY (color\_id) REFERENCES Colors(color\_id)**

**);**

* **Create Collaborations Table**

**CREATE TABLE Collaborations (**

**collab\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**project\_id INT,**

**user\_id INT,**

**role VARCHAR(50),**

**invited\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**FOREIGN KEY (project\_id) REFERENCES Projects(project\_id),**

**FOREIGN KEY (user\_id) REFERENCES Users(user\_id)**

**);**

* **JavaScript Code**

**const express = require('express');**

**constbodyParser = require('body-parser');**

**const { Pool } = require('pg');**

**const app = express();**

**const port = process.env.PORT || 3000;**

**const pool = new Pool({**

**user: 'your\_db\_user',**

**host: 'your\_db\_host',**

**database: 'your\_db\_name',**

**password: 'your\_db\_password',**

**port: 5432,**

**});**

**app.use(bodyParser.json());**

**app.post('/login', async (req, res) => {**

**const { email, password } = req.body;**

**const result = await pool.query('SELECT \* FROM Users WHERE email = $1', [email]);**

**if (result.rows.length> 0 &&result.rows[0].password\_hash === hashPassword(password)) {**

**res.status(200).send({ message: 'Login successful', userId: result.rows[0].user\_id });**

**} else {**

**res.status(401).send({ message: 'Invalid credentials' });**

**}**

**});**

**app.post('/upload-image', (req, res) => {**

**// Handle image upload and store the URL in the database**

**});**

**app.listen(port, () => {**

**console.log(`Server running on port ${port}`);**

**});**

**function hashPassword(password) {**

**// Implement password hashing**

**}**

* **React.js Code**

**import React, { useState } from 'react';**

**import axios from 'axios';**

**constImageUpload = ({ projectId }) => {**

**const [image, setImage] = useState(null);**

**consthandleImageUpload = (event) => {**

**setImage(event.target.files[0]);**

**};**

**consthandleSubmit = async () => {**

**constformData = new FormData();**

**formData.append('image', image);**

**formData.append('project\_id', projectId);**

**try {**

**const response = await axios.post('/upload-image', formData, {**

**headers: {**

**'Content-Type': 'multipart/form-data',**

**},**

**});**

**console.log('Image uploaded successfully:', response.data);**

**} catch (error) {**

**console.error('Error uploading image:', error);**

**}**

**};**

**return (**

**<div>**

**<input type="file" onChange={handleImageUpload} />**

**<button onClick={handleSubmit}>Upload Image</button>**

**</div>**

**);**

**};**

**export default ImageUpload;**

1. **SOFTWARE AND HARDWARE DETAILS**

**3.1 CHOICE OF LANGUAGE USED**

* HTML. CSS
* JavaScript, Rect.js

**3.2 HARDWARE / SOFTWARE SPECIFICATION**

* **DEVICES USED**
* DELL LAPTOP
* ANDROID SMARTPHONE
* **SOFTWARE USED**
* WORDPRESS
* BOOTSTRAP
* APACHE
* MYSQL

1. **SYSTEM DESIGN**

**4.1 PROGRAM LISTING **

* **LOGIN PORTAL**
* **Login.js Code**

**import React, { useState } from 'react';**

**import axios from 'axios';**

**import './Login.css';**

**function Login() {**

**const [email, setEmail] = useState('');**

**const [password, setPassword] = useState('');**

**const [error, setError] = useState('');**

**consthandleSubmit = async (e) => {**

**e.preventDefault();**

**try {**

**const response = await axios.post('/api/login', { email, password });**

**localStorage.setItem('token', response.data.token);**

**window.location.href = '/dashboard';**

**} catch (err) {**

**setError('Invalid email or password');**

**}**

**};**

**return (**

**<div className="login-container">**

**<form onSubmit={handleSubmit}>**

**<h2>Login</h2>**

**<div className="form-group">**

**<label>Email</label>**

**<input**

**type="email"**

**value={email}**

**onChange={(e) =>setEmail(e.target.value)}**

**required**

**/>**

**</div>**

**<div className="form-group">**

**<label>Password</label>**

**<input**

**type="password"**

**value={password}**

**onChange={(e) =>setPassword(e.target.value)}**

**required**

**/>**

**</div>**

**{error &&<p className="error">{error}</p>}**

**<button type="submit">Login</button>**

**</form>**

**</div>**

**);**

**}**

**export default Login;**

* **Login.css**

**.login-container {**

**width: 300px;**

**margin: 100px auto;**

**padding: 20px;**

**border: 1px solid #ccc;**

**border-radius: 5px;**

**background-color: #f9f9f9;**

**}**

**.form-group {**

**margin-bottom: 15px;**

**}**

**.form-group label {**

**display: block;**

**margin-bottom: 5px;**

**}**

**.form-group input {**

**width: 100%;**

**padding: 8px;**

**box-sizing: border-box;**

**}**

**.error {**

**color: red;**

**margin-top: 10px;**

**}**

**button {**

**width: 100%;**

**padding: 10px;**

**background-color: #007bff;**

**border: none;**

**color: white;**

**border-radius: 5px;**

**cursor: pointer;**

**}**

**button:hover {**

**background-color: #0056b3;**

**}**

* **Server.js Backend Code(Login Page)**

**const express = require('express');**

**constbodyParser = require('body-parser');**

**const mongoose = require('mongoose');**

**constbcrypt = require('bcrypt');**

**constjwt = require('jsonwebtoken');**

**const User = require('./models/User');**

**const app = express();**

**const PORT = process.env.PORT || 5000;**

**app.use(bodyParser.json());**

**mongoose.connect('mongodb://localhost:27017/wall-painting-visualization', {**

**useNewUrlParser: true,**

**useUnifiedTopology: true,**

**});**

**app.post('/api/login', async (req, res) => {**

**const { email, password } = req.body;**

**const user = await User.findOne({ email });**

**if (!user) {**

**return res.status(400).json({ message: 'Invalid email or password' });**

**}**

**constisMatch = await bcrypt.compare(password, user.password);**

**if (!isMatch) {**

**return res.status(400).json({ message: 'Invalid email or password' });**

**}**

**const token = jwt.sign({ userId: user.\_id }, 'secret\_key', { expiresIn: '1h' });**

**res.json({ token });**

**});**

**app.listen(PORT, () => {**

**console.log(`Server is running on port ${PORT}`);**

**});**

* **Models/User.js**

**const mongoose = require('mongoose');**

**constbcrypt = require('bcrypt');**

**constuserSchema = new mongoose.Schema({**

**email: {**

**type: String,**

**required: true,**

**unique: true,**

**},**

**password: {**

**type: String,**

**required: true,**

**},**

**});**

**userSchema.pre('save', async function (next) {**

**const user = this;**

**if (!user.isModified('password')) {**

**return next();**

**}**

**user.password = await bcrypt.hash(user.password, 8);**

**next();**

**});**

**const User = mongoose.model('User', userSchema);**

**module.exports = User;**

**REGISTER**

* **HTML Code**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>Register</title>**

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

**</head>**

**<body>**

**<div class="container">**

**<h2>Register</h2>**

**<form id="registerForm">**

**<div class="form-group">**

**<label for="username">Username</label>**

**<input type="text" id="username" name="username" required>**

**</div>**

**<div class="form-group">**

**<label for="email">Email</label>**

**<input type="email" id="email" name="email" required>**

**</div>**

**<div class="form-group">**

**<label for="password">Password</label>**

**<input type="password" id="password" name="password" required>**

**</div>**

**<button type="submit">Register</button>**

**</form>**

**</div>**

**<script src="scripts.js"></script>**

**</body>**

**</html>**

**JavaScript Code**

**document.getElementById('registerForm').addEventListener('submit', async function (e) {**

**e.preventDefault();**

**const username = document.getElementById('username').value;**

**const email = document.getElementById('email').value;**

**const password = document.getElementById('password').value;**

**const response = await fetch('/register', {**

**method: 'POST',**

**headers: {**

**'Content-Type': 'application/json'**

**},**

**body: JSON.stringify({ username, email, password })**

**});**

**const result = await response.json();**

**if (result.success) {**

**alert('Registration successful!');**

**} else {**

**alert('Registration failed: ' + result.message);**

**}**

**});**

* **Backend Code**

**const express = require('express');**

**constbodyParser = require('body-parser');**

**constbcrypt = require('bcrypt');**

**const mongoose = require('mongoose');**

**const User = require('./models/User'); // Assuming a User model is defined in models/User.js**

**const app = express();**

**app.use(bodyParser.json());**

**// MongoDB connection**

**mongoose.connect('mongodb://localhost:27017/wall\_painting\_website', {**

**useNewUrlParser: true,**

**useUnifiedTopology: true**

**});**

**// Registration endpoint**

**app.post('/register', async (req, res) => {**

**const { username, email, password } = req.body;**

**try {**

**constexistingUser = await User.findOne({ email });**

**if (existingUser) {**

**return res.status(400).json({ success: false, message: 'Email already in use' });**

**}**

**consthashedPassword = await bcrypt.hash(password, 10);**

**constnewUser = new User({ username, email, password: hashedPassword });**

**await newUser.save();**

**res.status(201).json({ success: true, message: 'User registered successfully' });**

**} catch (error) {**

**res.status(500).json({ success: false, message: 'Server error' });**

**}**

**});**

**app.listen(3000, () => {**

**console.log('Server is running on port 3000');**

**});**

* **Models/User.js Code**

**const mongoose = require('mongoose');**

**constUserSchema = new mongoose.Schema({**

**username: {**

**type: String,**

**required: true,**

**unique: true**

**},**

**email: {**

**type: String,**

**required: true,**

**unique: true**

**},**

**password: {**

**type: String,**

**required: true**

**}**

**});**

**module.exports = mongoose.model('User', UserSchema);**

* **GET START PAGE**
* **HTML Code**

**<!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>**

**<div class="container">**

**<h1>Get in Touch</h1>**

**<p>Contact Us for Consultation</p>**

**<div class="contact-info">**

**<div class="contact-item">**

**<h2>Phone</h2>**

**<p>8177894099</p>**

**</div>**

**<div class="contact-item">**

**<h2>Email</h2>**

**<p><a href="mailto:chaitalighugal@gmail.com">chaitalighugal@gmail.com</a></p>**

**</div>**

**</div>**

**<div class="follow-us">**

**<h2>Follow Us</h2>**

**<!-- Add social media links here -->**

**<!-- Example: <a href="https://www.facebook.com/example">Facebook</a> -->**

**</div>**

**<button id="colorful-consultation">Get Started for a Colorful Consultation</button>**

**</div>**

**<script src="script.js"></script>**

**</body>**

**</html>**

* **CSS Code**

**body {**

**font-family: Arial, sans-serif;**

**margin: 0;**

**padding: 0;**

**}**

**.container {**

**max-width: 800px;**

**margin: 0 auto;**

**padding: 20px;**

**}**

**h1 {**

**color: #333;**

**}**

**.contact-info {**

**display: flex;**

**justify-content: space-between;**

**}**

**.contact-item {**

**flex: 0 0 48%;**

**}**

**.follow-us {**

**margin-top: 20px;**

**}**

**.follow-us h2 {**

**color: #333;**

**}**

**.follow-us a {**

**display: block;**

**margin-top: 5px;**

**text-decoration: none;**

**color: #007bff;**

**}**

**.follow-us a:hover {**

**text-decoration: underline;**

**}**

**button {**

**display: block;**

**margin-top: 20px;**

**padding: 10px 20px;**

**background-color: #007bff;**

**color: #fff;**

**border: none;**

**border-radius: 5px;**

**cursor: pointer;**

**}**

**button:hover {**

**background-color: #0056b3;**

**}**

* **JavaScript.js Code**

**document.getElementById("colorful-consultation").addEventListener("click", function() {**

**// Change background color of the page**

**document.body.style.backgroundColor = getRandomColor();**

**});**

**function getRandomColor() {**

**var letters = "0123456789ABCDEF";**

**varcolor = "#";**

**for (vari = 0; i< 6; i++) {**

**color += letters[Math.floor(Math.random() \* 16)];**

**}**

**return color;**

**}**

**HOME PAGE**

* **HTML Code**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>Wall Painting Visualization</title>**

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

**</head>**

**<body>**

**<header>**

**<nav>**

**<ul>**

**<li><a href="#">Home</a></li>**

**<li><a href="#">Features</a></li>**

**<li><a href="#">About</a></li>**

**<li><a href="#">Contact</a></li>**

**</ul>**

**</nav>**

**</header>**

**<main>**

**<section class="upload-section">**

**<h1>Upload Your Room Image</h1>**

**<input type="file" id="imageUpload" accept="image/\*">**

**</section>**

**<section class="color-selection">**

**<h1>Select a Color</h1>**

**<input type="color" id="colorPicker">**

**</section>**

**<section class="visualization">**

**<h1>Visualization</h1>**

**<canvas id="canvas"></canvas>**

**</section>**

**</main>**

**<footer>**

**<p>&copy; 2024 Wall Painting Visualization</p>**

**</footer>**

**<script src="scripts.js"></script>**

**</body>**

**</html>**

* **CSS Code**

**body {**

**font-family: Arial, sans-serif;**

**margin: 0;**

**padding: 0;**

**box-sizing: border-box;**

**}**

**header {**

**background-color: #333;**

**color: white;**

**padding: 10px 0;**

**}**

**navul {**

**list-style: none;**

**display: flex;**

**justify-content: center;**

**padding: 0;**

**}**

**navul li {**

**margin: 0 15px;**

**}**

**navul li a {**

**color: white;**

**text-decoration: none;**

**}**

**main {**

**padding: 20px;**

**display: flex;**

**flex-direction: column;**

**align-items: center;**

**}**

**.upload-section, .color-selection, .visualization {**

**margin: 20px 0;**

**text-align: center;**

**}**

**footer {**

**background-color: #333;**

**color: white;**

**text-align: center;**

**padding: 10px 0;**

**position: fixed;**

**bottom: 0;**

**width: 100%;**

**}**

* **JavaScript Code**

**document.getElementById('imageUpload').addEventListener('change', handleImageUpload);**

**document.getElementById('colorPicker').addEventListener('input', handleColorChange);**

**let canvas = document.getElementById('canvas');**

**let ctx = canvas.getContext('2d');**

**let image = new Image();**

**let imageLoaded = false;**

**function handleImageUpload(event) {**

**let reader = new FileReader();**

**reader.onload = function(e) {**

**image.src = e.target.result;**

**image.onload = function() {**

**canvas.width = image.width;**

**canvas.height = image.height;**

**ctx.drawImage(image, 0, 0);**

**imageLoaded = true;**

**}**

**}**

**reader.readAsDataURL(event.target.files[0]);**

**}**

**function handleColorChange(event) {**

**if (!imageLoaded) return;**

**let color = event.target.value;**

**ctx.drawImage(image, 0, 0);**

**ctx.globalCompositeOperation = 'source-in';**

**ctx.fillStyle = color;**

**ctx.fillRect(0, 0, canvas.width, canvas.height);**

**ctx.globalCompositeOperation = 'source-over';**

**}**

**ABOUT PAGE**

* **HTML Code**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>About Us</title>**

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

**</head>**

**<body>**

**<div class="container">**

**<h1>Innovative Website Solutions</h1>**

**<h2>About Chaitali Creation Website AI</h2>**

**<p>At Chaitali Creation, we specialize in creating AI-powered websites that enable users to visualize paint combinations on their walls through advanced technology.</p>**

**<h2>Our Expertise</h2>**

**<p>Our expertise lies in web development, image processing, and integrating cutting-edge color matching algorithms for a seamless user experience.</p>**

**<h2>Our Unique Value Proposition</h2>**

**<p>Discover the reasons why Chaitali Creation's AI website stands out from the rest:</p>**

**<ul>**

**<li><strong>Personalized Experience:</strong> Tailored to individual preferences, our website offers a personalized experience for each user, enhancing satisfaction and engagement.</li>**

**<li><strong>Real-time Visualization:</strong> Experience paint combinations in real time, allowing users to make informed decisions instantly for their home decor.</li>**

**<li><strong>Seamless Integration:</strong> Our AI website seamlessly integrates web development, image processing, and color matching algorithms for a cohesive and efficient system.</li>**

**</ul>**

**</div>**

**</body>**

**</html>**

* **CSS Code**

**body {**

**font-family: Arial, sans-serif;**

**margin: 0;**

**padding: 0;**

**}**

**.container {**

**max-width: 800px;**

**margin: 0 auto;**

**padding: 20px;**

**}**

**h1, h2 {**

**color: #333;**

**}**

**ul {**

**list-style-type: none;**

**padding: 0;**

**}**

**ul li {**

**margin-bottom: 10px;**

**}**

**strong {**

**font-weight: bold;**

**}**

**Services**

* **Main services**
* **Web Development**
* **Image Processing**
* **Color Matching Algorithm Integration**
* **Mobile campatibility**

**HTML Code (Service)**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>Wall Painting Visualization Services</title>**

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

**</head>**

**<body>**

**<header>**

**<h1>Our Services</h1>**

**</header>**

**<main>**

**<section class="service">**

**<h2>Web Development</h2>**

**<p>Our expert team specializes in creating user-friendly websites that are tailored to your needs. From design to implementation, we ensure a seamless experience.</p>**

**</section>**

**<section class="service">**

**<h2>Image Processing</h2>**

**<p>Utilizing cutting-edge technology, we enhance images to bring out the best in your home walls. Our careful processing ensures accurate depiction of paint combinations.</p>**

**</section>**

**<section class="service">**

**<h2>Color Matching Algorithm Integration</h2>**

**<p>Integration of advanced algorithms enables us to match colors accurately, providing you with a comprehensive visualization of various paint combinations.</p>**

**</section>**

**<section class="service">**

**<h2>Mobile Compatibility</h2>**

**<p>Our websites are designed to be fully mobile-responsive, ensuring that you can access our services smoothly on any device.</p>**

**</section>**

**</main>**

**<footer>**

**<p>&copy; 2024 Wall Painting Visualization. All rights reserved.</p>**

**</footer>**

**</body>**

**</html>**

* **CSS Code**

**body {**

**font-family: Arial, sans-serif;**

**margin: 0;**

**padding: 0;**

**background-color: #f4f4f4;**

**}**

**header {**

**background-color: #4CAF50;**

**color: white;**

**text-align: center;**

**padding: 1em 0;**

**}**

**main {**

**padding: 2em;**

**}**

**.service {**

**background-color: white;**

**margin-bottom: 1em;**

**padding: 1em;**

**border-radius: 8px;**

**box-shadow: 0 0 10px rgba(0,0,0,0.1);**

**}**

**.service h2 {**

**color: #333;**

**}**

**footer {**

**background-color: #4CAF50;**

**color: white;**

**text-align: center;**

**padding: 1em 0;**

**position: fixed;**

**width: 100%;**

**bottom: 0;**

**}**

* **JavaScript Code**

**document.addEventListener('DOMContentLoaded', (event) => {**

**const services = document.querySelectorAll('.service');**

**services.forEach(service => {**

**service.addEventListener('click', () => {**

**alert(`You clicked on ${service.querySelector('h2').innerText}`);**

**});**

**});**

**});**

* **BLOG Page**
* **JavaScript.js Code**

**import React, { useState } from 'react';**

**import './App.css';**

**import ImageUploader from './components/ImageUploader';**

**import ColorPicker from './components/ColorPicker';**

**import Preview from './components/Preview';**

**function App() {**

**const [image, setImage] = useState(null);**

**const [color, setColor] = useState('#ffffff');**

**return (**

**<div className="App">**

**<h1>Wall Painting Visualization</h1>**

**<ImageUploadersetImage={setImage} />**

**<ColorPickersetColor={setColor} />**

**<Preview image={image} color={color} />**

**</div>**

**);**

**}**

**export default App;**

* **Image Uploader.js Code**

**import React from 'react';**

**function ImageUploader({ setImage }) {**

**consthandleUpload = (event) => {**

**const file = event.target.files[0];**

**const reader = new FileReader();**

**reader.onloadend = () => {**

**setImage(reader.result);**

**};**

**reader.readAsDataURL(file);**

**};**

**return (**

**<div>**

**<input type="file" onChange={handleUpload} />**

**</div>**

**);**

**}**

**export default ImageUploader;**

* **ColorPicker.js Code**

**import React from 'react';**

**function ColorPicker({ setColor }) {**

**return (**

**<div>**

**<input type="color" onChange={(e) =>setColor(e.target.value)} />**

**</div>**

**);**

**}**

**export default ColorPicker;**

* **Preview.js Code**

**import React from 'react';**

**function Preview({ image, color }) {**

**constpreviewStyle = {**

**backgroundImage: `url(${image})`,**

**backgroundColor: color,**

**width: '500px',**

**height: '500px',**

**backgroundBlendMode: 'multiply',**

**};**

**return (**

**<div style={previewStyle}></div>**

**);**

**}**

**export default Preview;**

* **Server.js Code**

**const express = require('express');**

**constmulter = require('multer');**

**const path = require('path');**

**const { applyColor } = require('./imageProcessing');**

**const app = express();**

**const upload = multer({ dest: 'uploads/' });**

**app.post('/upload', upload.single('image'), (req, res) => {**

**constfilePath = req.file.path;**

**constcolor = req.body.color;**

**applyColor(filePath, color)**

**.then((outputPath) =>res.sendFile(outputPath))**

**.catch((err) =>res.status(500).send(err));**

**});**

**app.listen(3000, () => {**

**console.log('Server started on http://localhost:3000');**

**});**

* **ImageProcessing.js Code**

**const { createCanvas, loadImage } = require('canvas');**

**function applyColor(imagePath, color) {**

**return loadImage(imagePath).then((image) => {**

**const canvas = createCanvas(image.width, image.height);**

**constctx = canvas.getContext('2d');**

**ctx.drawImage(image, 0, 0);**

**ctx.globalCompositeOperation = 'multiply';**

**ctx.fillStyle = color;**

**ctx.fillRect(0, 0, image.width, image.height);**

**constoutputPath = `output/${Date.now()}.png`;**

**return new Promise((resolve, reject) => {**

**const out = fs.createWriteStream(outputPath);**

**const stream = canvas.createPNGStream();**

**stream.pipe(out);**

**out.on('finish', () => resolve(outputPath));**

**out.on('error', reject);**

**});**

**});**

**}**

**module.exports = { applyColor };**

**Contact page**

* **Get in touch**
* **Contact us for consultation**
* **Phone 8177894099**
* **Email: chaitalighugal@gmail.Com**
* **Follow us**

**HTML Code**

**<!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>**

**<div class="container">**

**<h1>Get in Touch</h1>**

**<p>Contact Us for Consultation</p>**

**<div class="contact-info">**

**<div class="contact-item">**

**<h2>Phone</h2>**

**<p>8177894099</p>**

**</div>**

**<div class="contact-item">**

**<h2>Email</h2>**

**<p><a href="mailto:chaitalighugal@gmail.com">chaitalighugal@gmail.com</a></p>**

**</div>**

**</div>**

**<div class="follow-us">**

**<h2>Follow Us</h2>**

**<!-- Add social media links here -->**

**<!-- Example: <a href="https://www.facebook.com/example">Facebook</a> -->**

**</div>**

**</div>**

**</body>**

**</html>**

* **CSS Code**

**body {**

**font-family: Arial, sans-serif;**

**margin: 0;**

**padding: 0;**

**}**

**.container {**

**max-width: 800px;**

**margin: 0 auto;**

**padding: 20px;**

**}**

**h1 {**

**color: #333;**

**}**

**.contact-info {**

**display: flex;**

**justify-content: space-between;**

**}**

**.contact-item {**

**flex: 0 0 48%;**

**}**

**.follow-us {**

**margin-top: 20px;**

**}**

**.follow-us h2 {**

**color: #333;**

**}**

**.follow-us a {**

**display: block;**

**margin-top: 5px;**

**text-decoration: none;**

**color: #007bff;**

**}**

**.follow-us a:hover {**

**text-decoration: underline;**

**}**

**Working Status**

* **Working Hours**
* **Lorem Ipsum**
* **Lorem Ipsum**
* **Sunday: Closed**

**HTML Code**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>Wall Painting Website</title>**

**<!-- Add your CSS file link here if you have one -->**

**</head>**

**<body>**

**<div class="working-hours">**

**<h2>Working Hours</h2>**

**<p>We're available to serve you during the following hours:</p>**

**<ul>**

**<li><strong>Monday - Saturday:</strong> 9:00 AM - 6:00 PM</li>**

**<li><strong>Sunday:</strong> Closed</li>**

**</ul>**

**</div>**

**</body>**

**</html>**

* **Copyright © 2024 AI website by chaitali creation**

**HTML Code**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>Wall Painting Website</title>**

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

**</head>**

**<body>**

**<header>**

**<!-- Header content goes here -->**

**</header>**

**<main>**

**<!-- Main content goes here -->**

**</main>**

**<footer>**

**<p>Copyright © 2024 AI website by Chaitali Creation</p>**

**</footer>**

**</body>**

**</html>**

* **CSS Code**

**footer {**

**background-color: #f8f8f8;**

**padding: 20px;**

**text-align: center;**

**}**

**footer p {**

**color: #555;**

**font-size: 14px;**

**}**

**4.2 INPUT SCREENS**

**➢ USER PORTAL PAGE**

The homepage of the Chaitali Creation wall painting visualization website is effectively designed to introduce users to its AI-powered capabilities. It combines clear branding, an engaging vist layout, and user-friendly navigation to provide a seamless and inviting user experience. The strategic use of images, color schemes, and CTAs ensures that users are both informed and encouraged to interact with the website’s features right from the start.

**USER HOME PAGE**

The page from the wall painting visualization website, "Visualize Your Home," highlights the main services offered:

**1. Web Development**: Crafting beautiful and functional websites.

**2. Image Processing**: Enhancing visual elements for stunning results.

**3. Colour Matching Algorithm Integration**: Personalized paint color suggestions for home decor.

The page aims to transform users' walls with personalized paint colour suggestions using their innovative AI technology. The page layout includes a header with navigation links (Home, About, Services, Blog, Contact) and a prominent "Get Started" button. Some image links appear to be broken, indicated by placeholder text.

The "Get in Touch" page on the wall painting visualization website, "Visualize Your Home," provides contact information for consultation. Key details include:

* **Phone Number**: 8177894099
* **Email**: [chaitalighugal@gmail.com](mailto:chaitalighugal@gmail.com)
* **Social Media**: Follow Us (specific platforms not listed)

The page also includes navigation links (Home, About, Services, Blog, Contact) and a prominent "Get Started" button for easy access to further services. The layout is simple and focused on providing clear contact options.

**5. USER DOCUMENTATION**

**5.1 IMPLEMENTATION, PROGRAM EXECUTION & MAINTENANCE**

1. **Requirement Gathering** 
   * **Objective:** Identify user needs and project goals.
   * **Activities:** Conduct surveys, interviews, and market research to understand user requirements and expectations.
   * **Deliverables:** Requirement specification document detailing functional and non-functional requirements.
2. **System Design**

* **Objective**: Plan the architecture and design of the system.
* **Activities:** Create system architecture diagrams, design database schema, and define UI/UX workflows.
* **Deliverables:** System design document including data flow diagrams, ER diagrams, and wireframes.

1. **Technology Selection**

* **Objective:** Choose appropriate technologies for development.
* **Activities:** Evaluate different technologies for frontend, backend, database, and hosting.
* **Technologies:**
* **Frontend:** HTML, CSS, JavaScript, React.js
* **Backend:** Node.js, Express.js
* **Database:** MongoDB Hosting: AWS, Google Cloud.

1. **Deployment**

* **Objective:** Launch the website for users.
* **Activities:** Set up server infrastructure, deploy code to production servers, configure domain and SSL.
* **Tools:** Docker for containerization, Jenkins for CI/CD, AWS/GCP for hosting.

1. **Testing**

* Objective: Ensure the system works correctly and meets requirements.
* Activities: Perform unit testing, integration testing, system testing, and user acceptance testing.
* **Tools:** Jest for unit testing, Selenium for automated UI testing.

1. **Maintenance & Updates**

* Objective: Keep the system up-to-date and operational.
* Activities: Monitor system performance, fix bugs, implement new features, update security patches.
* Tools: Monitoring tools like New Relic, logging tools like ELK stack.

**Implementation**

**Steps:**

1. Set up development environment with chosen technologies.
2. Implement frontend and backend modules as per design.
3. Integrate modules and perform iterative testing. Deploy the website on a staging environment for final testing.
4. Move the tested code to the production environment.

**Program Execution**

**Steps:**

1. Run the server using Node.js.
2. Ensure the database server is active and connected.
3. Access the website through the configured domain.
4. Use the website features as per user roles (admin, user).

**Maintenance & Update**

**Steps:**

1. Regularly back up the database and server configurations.
2. Monitor user feedback and system logs for issues.
3. Schedule updates for off-peak hours to minimize disruption.
4. Apply security patches and new feature updates as needed.
5. Conduct periodic reviews and refactor code to maintain performance and security standards.

This documentation provides an overview of the essential stages involved in the development, deployment, and maintenance of the wall painting visualization website. Each section outlines the objectives, activities, and tools used to ensure a smooth process from conception to execution and beyond.

**6. CONCLUSION**

**6.1 LIMITATIONS OF THE SYSTEM**

While the wall painting visualization website presents a significant advancement in the interior design process, it does have several limitations that must be acknowledged. Firstly, the accuracy of colour representation can vary due to differences in device screens and lighting conditions, potentially leading to discrepancies between the visualization and actual paint application. Additionally, the complexity of accurately segmenting walls and applying textures in real-time can strain the system, particularly on devices with limited processing power. Moreover, the reliance on high-quality image uploads for accurate visualizations means users with low-resolution or poorly lit photos may experience suboptimal results.

#### **Detailed Explanation**

**1. Colour Representation Variability**:

**Issue**: Different devices have varying screen calibrations, affecting how colors are displayed.

**Impact**: Users might see different shades on their devices compared to the actual paint colour.

**Solution**: Implement colour calibration tools and guidelines for users to adjust their device settings.

1. **Image Processing Strain:**

**Issue**: Advanced image processing and real-time rendering require significant computational power.

**Impact**: Devices with limited resources may experience lag or reduced functionality.

**Solution**: Optimize algorithms for performance and offer a simplified mode for low-end devices.

1. **Dependence on High-Quality Images:**

**Issue**: Accurate visualizations rely on high-quality, well-lit images of the rooms.

**Impact**: Poor image quality can lead to inaccurate wall segmentation and colour application.

**Solution**: Provide users with tips for taking better photos and incorporate basic image enhancement tools.

1. **Lighting and Finish Simulation:**

**Issue**: Simulating different lighting conditions and paint finishes can be complex.

**Impact**: The visualized effect might not perfectly match real-world conditions.

**Solution**: Continuously improve the rendering engine and offer a variety of lighting presets to enhance realism.

By addressing these limitations through ongoing development and user education, the wall painting visualization website can further enhance its utility and accuracy, providing an even more valuable tool for interior design enthusiasts and professionals.

**6.2 SCOPE AND FUTURE MODIFICATION**

The wall painting visualization website offers users an interactive platform to visualize different paint colours and finishes on their walls. Key features include:

* **Image Upload and Management**: Users can upload images of their rooms.
* **Colour Selection and Application**: Users can choose from a variety of colours and finishes.
* **Realistic Rendering**: Advanced algorithms ensure accurate colour representation and lighting effects.
* **User Authentication**: Secure login and registration processes.
* **Sharing and Collaboration**: Users can share designs and collaborate with others.

#### **Future Modifications**

To enhance the functionality and user experience, several future modifications can be implemented:

**1. Augmented Reality (AR) Integration**:

**Feature**: Enable real-time visualization of paint colours using AR through smartphone cameras.

**Benefit**: Provides a more immersive and accurate way for users to see how colours will look in their actual space.

**2. AI-Driven Colour Suggestions**:

**Feature**: Use AI to suggest colour combinations based on room type, existing furniture, and user preferences.

**Benefit**: Helps users make informed decisions with personalized recommendations.

**3. Enhanced Customization Options**:

**Feature**: Offer more customization tools such as pattern overlays, wall art, and virtual furniture arrangement.

**Benefit**: Allows users to create a more comprehensive design plan for their interiors.

**4. Integration with E-Commerce Platforms**:

**Feature**: Integrate with online paint and decor stores for direct purchasing of selected colours and products.

**Benefit**: Streamlines the user journey from visualization to purchase, enhancing convenience.

**5. Collaborative Design Features**:

**Feature**: Introduce real-time collaborative tools for users to work on designs with family, friends, or interior designers.

**Benefit**: Facilitates shared decision-making and collective creativity.

**6. Enhanced Performance and Scalability**:

**Feature**: Optimize backend infrastructure and use scalable cloud solutions to handle increased user traffic.

**Benefit**: Ensures smooth and responsive performance, even during peak usage times.

**7. User Feedback and Analytics**:

**Feature**: Implement tools to collect user feedback and analyse usage patterns.

**Benefit**: Provides insights for continuous improvement and feature updates based on actual user needs and preferences.

By implementing these future modifications, the wall painting visualization website can significantly enhance its utility, user satisfaction, and market competitiveness.

**7. REFRENCES / BIBLIOGRAPHY**

The following books were used extensively for the project development and implementation.

1. Krug, Steve. *Don't Make Me Think: A Common Sense Approach to Web Usability*. New Riders, 2014.
2. Duckett, Jon. *HTML and CSS: Design and Build Websites*. Wiley, 2011
3. Gonzalez, Rafael C., and Richard E. Woods. *Digital Image Processing*. Pearson, 2018
4. Szeliski, Richard. "Computer Vision: Algorithms and Applications". *Springer*, 2010.
5. Reddy, Kalilur. *Node.js Web Development*. Packt Publishing, 2020
6. Redmond, Elizabeth, and Michael Redmond. *Learning SQL*. O'Reilly Media, 2020.
7. Parisi, Tony. *Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages*. O'Reilly Media, 2014.
8. Hwang, Kai, Geoffrey C. Fox, and Jack J. Dongarra. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. Morgan Kaufmann, 2012.
9. Grigorik, Ilya. *High Performance Browser Networking: What Every Web Developer Should Know About Networking and Web Performance*. O'Reilly Media, 2013.
10. "Collaborative Design Tools: How They Are Changing the Design Process". *UX Design Blog*, 2021.

These references collectively provide a comprehensive foundation for the development and enhancement of a wall painting visualization website, covering technical, design, and user experience aspects.

**Notice**

Students of **MCA-I [Sem-II]** should follow following instruction regarding the **Project Report**:

1. Paper Size: **A4 (Executive Bond)**

2. Font Size:

**Normal Text Size : 12**

**Heading** **: 16**

**Sub-Heading** **: 14**

**Font** **: Times New Roman**

5. Header : **Name of Project (Right)**

6. Footer : **College Name (Left),** **Page Number (Right)**

7. Line Spacing : **1.5**

8. Margin : **Left: 1.5, Right: 1, Top: 1, Bottom: 1**

9. BindingColour : **Brown Colour.**