**THAKUR COLLEGE OF SCIENCE AND COMMERCE**

**KANDIVALI (EAST)**

**MUMBAI**

**A**

**PROJECT**

**REPORT**

**ON**

MY BLOG WEBSITE WITH ADMIN PANEL

FOR

THAKUR COLLEGE OF SCIENCE AND COMMERCE

BY

**AAYUSH SINGH**

Submitted in partial fulfillment of

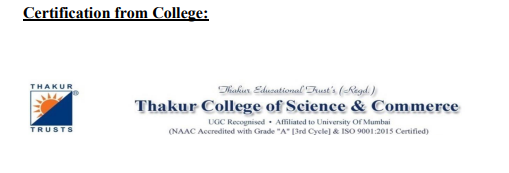
Bachelors of Science (Computer Science)

**[UNIVERSITY OF MUMBAI]**

Thakur Degree College of Science and

Commerce Kandivali (East), Mumbai-400101

**ACADEMIC YEAR: 2023-2024**



**Date**: 27/09/2023

**COMPUTER SCIENCE DEPARTMENT (2020-2021)**

Certificate of Approval This is to certify that the project work entitled "**My Blog Website with Admin Panel**” is prepared by **130 Aayush Singh** a student of “Third Year Bachelor of Science (Computer Science)” course of University of Mumbai, which is conducted by our college.

This is the original study work and important sources used have been duly acknowledged in the report. The report is submitted in partial fulfilment **of B.Sc. (Computer Science)** course as per rules of University of Mumbai.

**Ashish Trivedi** **Ashish Trivedi**

Project Guide Head of Department

External Examiner

**INDEX**

|  |  |  |
| --- | --- | --- |
| SR. No | Index Topic | Page |
|  | * Abstract * **Key Features** * Admin Panel Highlights |  |
| I. | Introduction   * Overview of the project * Significance and potential impact |  |
| II. | Project Scope   * Inclusions and exclusions * Project constraints |  |
| III. | System design and modelling   * ER Diagram * Schema Diagram |  |
| IV. | System Requirements   * Functional requirements * Non-functional requirements |  |
| V. | Deployment   * Deployment strategy * Server configuration * Release notes |  |
| VI. | Code |  |
| VII. | Snapshots |  |
| VIII. | Conclusion |  |

|  |  |  |
| --- | --- | --- |
|  | EXP. DATE | COMPLETION DATE |
| Preliminary |  |  |
| System Analysis |  |  |
| System Design |  |  |
| System Coding |  |  |
| System Implementation |  |  |
| Report Submission |  |  |

**Software Requirements**

• Operating System: Windows, Linux, Mac

• Web-Technology: PHP & JS

• Front-End: HTML, CSS, JS

• Back-End: MySQL

• Web Server: Apache SERVER

* **ABSTRACT --**

In the dynamic realm of digital expression, having a robust and user-friendly blog website is paramount. Our state-of-the-art blogging platform not only caters to the content creator's aspirations but also simplifies the administrative tasks with a powerful and intuitive admin panel.

* **Key Features:**
* Responsive Design - Our blog website boasts a responsive and visually appealing design, ensuring an optimal user experience across devices.
* **Content Management -** Easily create, edit, and organize blog posts with a user-friendly content management system. Effortlessly upload and manage multimedia content to enrich your articles.
* **Customization Options -** Tailor your blog's appearance to reflect your unique style. Choose from a variety of themes, fonts, and color schemes to create a personalized and visually appealing platform.
* **User Authentication and Permissions -** Secure user authentication ensures that only authorized individuals have access to the admin panel. Define user roles and permissions to streamline collaborative blogging efforts.
* **SEO Optimization –** Boost your blog's visibility with built-in SEO tools. Optimize meta tags, descriptions, and URLs to enhance search engine rankings and attract a wider audience.
* **Social Media Integration -** Seamlessly connect your blog to various social media platforms, allowing readers to share your content effortlessly. Expand your reach and grow your online presence.
* **Analytics Dashboard -** Gain valuable insights into your blog's performance with an intuitive analytics dashboard. Track visitor metrics, popular content, and user engagement to refine your content strategy.
* **Admin Panel Highlights:**
* **User Management -** Effortlessly manage users, roles, and permissions. Grant access to contributors, editors, and administrators as needed.
* **Content Moderation -** Maintain control over published content with moderation features. Approve, edit, or reject submissions from contributors.
* **Security Features -** Implement robust security measures to safeguard your blog against potential threats. Regular updates and encryption protocols ensure data integrity.
* **Backup and Restore –** Safeguard your content with automated backup and restore functionalities. Protect against data loss and ensure the longevity of your valuable content.
* **Intuitive Dashboard –** Navigate the admin panel with ease using a clean and intuitive dashboard. Monitor site activity, manage settings, and track performance effortlessly.

Embark on your blogging journey with confidence, knowing that our platform combines cutting-edge technology with user-friendly interfaces to empower your creativity while providing administrators with the tools they need to manage and optimize the platform efficiently.

Feel free to adjust and expand upon the points based on the specific features and functionalities you plan to incorporate into your blog website and admin panel.

* **INTRODUCTION** –

Overview of the project :

Building a blog website with an admin panel is an exciting venture that allows you to share your thoughts, insights, and expertise with the world. Whether you're a seasoned blogger or just starting out, creating a platform that is not only user-friendly but also easy for you to manage is crucial. Below is an introduction that you can use for your blog website:

**Welcome to My Blog - Where Ideas Come to Life!**

In the vast expanse of the digital realm, we are thrilled to carve out a niche for thoughtful discussions, creative expressions, and the exchange of ideas. At My Blog, we believe in the power of words to inspire, educate, and connect individuals from all walks of life.

**About Us:**

At the heart of My Blog is a community of passionate individuals dedicated to exploring the diverse tapestry of human experiences. Whether you're seeking inspiration, valuable insights, or a moment of respite from the chaos of daily life, our blog is designed to be your digital sanctuary.

**What Sets Us Apart:**

* **Diverse Perspectives -** We embrace diversity in thought, culture, and perspective. Our content spans a wide array of topics, from technology trends and lifestyle advice to deep dives into niche interests.
* **User-Friendly Interface -** Navigating through [Your Blog Name] is a breeze. Our clean and intuitive design ensures that you can seamlessly explore content that matters to you.
* **Engaging Content -** Every article is crafted with care, aiming not just to inform but to captivate. We believe in the power of storytelling to make information memorable and impactful.

**Admin Panel:**

Behind the scenes, our admin panel is the engine that drives this dynamic platform. As the architects of this digital space, our admin team works tirelessly to ensure a smooth and secure experience for both readers and contributors.

**Key Features of Our Admin Panel:**

* **Content Management –** Easily create, edit, and organize your articles with our user-friendly content management system. Tagging, categorization, and scheduling are just a few clicks away.
* **User Interaction -** Foster a sense of community by managing comments, engaging with readers, and moderating discussions. Our admin panel makes it simple to connect with your audience.
* **Analytics -** Gain insights into your audience's preferences and behaviour. Track article performance, user engagement, and more to refine your content strategy.
* **Security -** We prioritize the security of your data and the integrity of your platform. Our admin panel comes equipped with robust security features to protect your content and user information.
* **Join the Conversation:**

Become a part of the My Blog community today. Whether you're a seasoned writer or a curious reader, there's a place for you here. Let's embark on this journey of exploration, learning, and connection together.

Thank you for choosing My Blog as your digital destination. Let the adventure begin!

Feel free to customize this introduction to better align with the specific theme, tone, and goals of your blog website.

* **Significance and potential impact**:

Creating a blog website with an admin panel can have several significant benefits and potential impacts, both for the website owner/administrator and the audience. Here are some key points to consider:

1. **Content Management:**

* **Significance -** An admin panel allows easy and efficient management of blog content. Content creators can add, edit, and delete posts without needing technical expertise.
* **Impact -** Streamlined content management leads to a consistent and updated blog, enhancing user experience and engagement.

### ****User Engagement:****

### **Significance -** An admin panel often includes tools for analyzing user engagement metrics. This helps in understanding the audience, their preferences, and the popularity of various posts.

### **Impact -** Informed decisions can be made to optimize content, improve user engagement, and tailor the blog to the audience's interests.

### ****Monetization Opportunities:****

### **Significance -** An admin panel can integrate features for managing advertisements, affiliate links, or sponsored content, allowing the website owner to monetize the blog.

### **Impact -** Potential revenue streams can be explored, making the blog financially sustainable and potentially profitable.

### ****SEO Optimization:****

### **Significance -** Admin panels often include SEO tools that help in optimizing content for search engines.

### **Impact -** Improved search engine rankings can lead to increased visibility, attracting more organic traffic to the blog.

### ****Security and Access Control:****

### **Significance -** Admin panels typically offer features to manage user roles and access levels, enhancing security.

### **Impact -** Ensures that only authorized individuals have access to critical website functions, reducing the risk of unauthorized changes or data breaches.

### ****Scalability:****

### **Significance -** An admin panel designed for scalability enables the blog to grow in terms of content and user base.

### **Impact -** The website can handle increased traffic and content volume without compromising performance or user experience.

### ****Analytics and Reporting:****

### **Significance -** Admin panels often come with built-in analytics tools to track website performance.

### **Impact -** Informed decision-making is possible based on data insights, allowing for continuous improvement and optimization.

### ****Customization and Branding:****

### **Significance -** Admin panels allow easy customization of the blog's appearance and branding.

### **Impact -** The blog can be aligned with the brand identity, providing a unique and memorable experience for visitors.

### ****Time and Resource Efficiency:****

### **Significance -** An intuitive admin panel reduces the time and effort required for routine tasks.

### **Impact -** Website administrators can focus more on creating quality content and engaging with the audience, leading to overall improved productivity.

### In summary, a blog website with a well-designed admin panel can significantly enhance the efficiency of content management, user engagement, and overall website performance, providing a more enjoyable experience for both creators and consumers of content.

* **Project Scope –**
* Inclusions and exclusions :

Creating a blog website with an admin panel involves various features, inclusions, and exclusions. Below is a breakdown of key components you might want to consider:

* **Inclusions:**

1. User Authentication:

- User registration and login functionality.

- Password recovery/reset options.

- User roles (Admin, Author, Contributor, Subscriber).

2. Admin Panel:

- Dashboard for site statistics and overview.

- User management (add, edit, delete users and their roles).

- Content management (create, edit, delete blog posts).

- Comment moderation (approve, delete, or edit comments).

- Analytics integration for traffic monitoring.

3. Blog Functionality:

- Post creation with rich text formatting.

- Categorization and tagging of blog posts.

- Featured posts and recent posts sections.

- Social media sharing options for each post.

4. Content Management System (CMS):

- WYSIWYG editor for easy content creation.

- Media management (image uploads, video embedding).

- Scheduling posts for future publication.

5. Responsive Design:

- Mobile-friendly design for a seamless user experience.

- Cross-browser compatibility.

6. Search Functionality:

- Search bar for users to find specific content.

7. Comments and Interactivity:

- Commenting system with user moderation.

- Like and share buttons for posts.

8. SEO Optimization:

- Customizable meta tags for each post.

- SEO-friendly URLs.

- Sitemap and robots.txt for search engine optimization.

9. Security:

- SSL certificate for secure connections.

- Protection against common web vulnerabilities (XSS, CSRF).

- Regular security audits and updates.

10. Performance Optimization:

- Image compression and lazy loading.

- Caching mechanisms for faster page loading.

* **Exclusions:**

1. Complex Features:

- Avoid unnecessary complexity if it doesn't align with the blog's purpose.

2. Overloaded Design:

- Keep the design clean and user-friendly; avoid clutter.

3. Unnecessary Plugins:

- Only include plugins or features that add value to the user experience.

4. Resource-Intensive Elements:

- Limit large media files that can slow down the website.

5. Excessive Dependencies:

- Minimize dependencies on external services that could impact site reliability.

6. Complicated UI/UX:

- Keep the user interface intuitive and easy to navigate.

7. Unsecured Components:

- Avoid any components or plugins with known security vulnerabilities.

8. Non-Scalable Solutions:

- Choose technologies that can scale with the growth of the blog.

9. Poor Mobile Optimization:

- Ensure the website is accessible and user-friendly on mobile devices.

10. Inadequate Backup System:

- Implement a robust backup system to prevent data loss.

Remember, the specific inclusions and exclusions might vary based on your exact requirements, the target audience, and the goals of your blog. Always consider the scalability and maintainability of your solution.

* Project constraints:

When developing a blog website with an admin panel, it's important to consider various project constraints to ensure a successful and efficient implementation. Project constraints are limitations or restrictions that can affect the development process. Here are some key constraints to consider for a blog website:

1. Budget Constraints:

- Determine the budget available for the project, considering development, hosting, maintenance, and any additional costs.

- Identify cost-effective solutions for hosting, domain registration, and other necessary services.

2. Time Constraints:

- Define a realistic timeline for the project, including milestones for development, testing, and deployment.

- Identify any time-sensitive requirements, such as launching before a specific date or aligning with marketing campaigns.

3. Technical Constraints:

- Specify the technologies and programming languages to be used for development.

- Consider compatibility issues with different browsers and devices.

- Evaluate the hosting environment and ensure it meets the technical requirements of the chosen technology stack.

4. Security Constraints:

- Implement secure coding practices to protect against common web vulnerabilities.

- Use encryption protocols (SSL/TLS) to secure data transmission.

- Regularly update software dependencies to patch security vulnerabilities.

5. Scalability Constraints:

- Consider the potential growth of the blog and choose a scalable architecture.

- Plan for scalability in terms of server resources, database performance, and overall system capacity.

6. User Experience (UX) Constraints:

- Design the user interface (UI) and user experience with the target audience in mind.

- Ensure the website is accessible and usable across different devices and screen sizes.

7. Regulatory Constraints:

- Comply with relevant data protection laws and regulations (e.g., GDPR, CCPA).

- Address any legal constraints related to content publishing and copyright.

8. Content Management Constraints:

- Define content creation and management processes for both regular users and administrators.

- Implement content moderation features to ensure the quality and appropriateness of published content.

9. Performance Constraints:

- Optimize website performance to minimize load times.

- Implement caching mechanisms and optimize database queries to enhance overall speed.

10. Backup and Recovery Constraints:

- Establish a robust backup and recovery system to protect against data loss.

- Regularly test backup restoration procedures to ensure they are effective.

11. Integration Constraints:

- Identify any third-party integrations, such as social media sharing, analytics, or payment gateways.

- Ensure smooth integration with the chosen technologies and platforms.

12. Admin Panel Constraints:

- Design the admin panel to be user-friendly and intuitive for content management.

- Implement role-based access control (RBAC) to restrict access based on user roles.

13. Compliance Constraints:

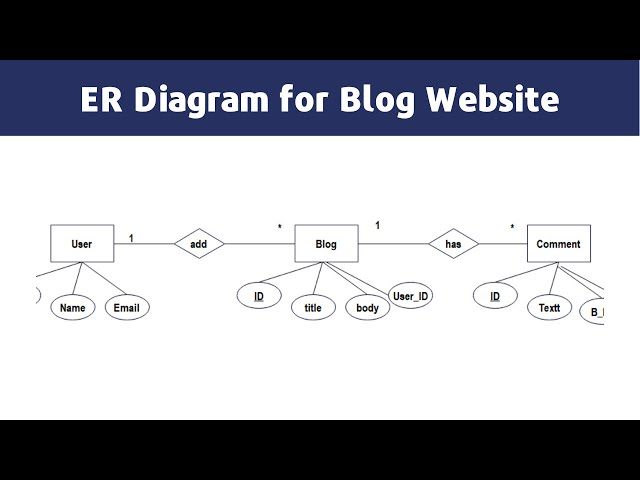
- Ensure compliance with industry standards and best practices in web development.

- Stay informed about changes in regulations that may affect the website.

By addressing these constraints, you can develop a robust and effective blog website with an admin panel that meets the needs of both users and administrators while staying within the specified budget and timeline.

* **System design and modelling** –
* ER Diagram:

Creating a detailed Entity-Relationship (ER) diagram for a blog website with an admin panel involves identifying the key entities, relationships, and attributes that are essential for the system. Below is a simplified ER diagram for a blog website with an admin panel:



Entities:

1. User:

- Attributes: UserID (Primary Key), Username, Email, Password

2. Post:

- Attributes: PostID (Primary Key), Title, Content, DatePosted, UserID (Foreign Key)

3. Comment:

- Attributes: CommentID (Primary Key), Content, DateCommented, UserID (Foreign Key), PostID (Foreign Key)

4. Category:

- Attributes: CategoryID (Primary Key), CategoryName

5. Admin:

- Attributes: AdminID (Primary Key), Email, Password

Relationships:

1. User-Post (One-to-Many):

- One user can have multiple posts, but each post is associated with only one user.

2. User-Comment (One-to-Many):

- One user can make multiple comments, but each comment is associated with only one user.

3. Post-Comment (One-to-Many):

- One post can have multiple comments, but each comment is associated with only one post.

4. Post-Category (Many-to-Many):

- Many posts can belong to many categories. This is typically implemented using an associative entity (PostCategory) with foreign keys to Post and Category.

6. User-Admin (One-to-One):

- One user can be an admin, and each admin is associated with only one user.

7. \*\*Admin-Post (One-to-Many):\*\*

- One admin can have multiple posts, but each post is associated with only one admin.

Note: This is a basic ER diagram, and the actual requirements may vary based on the specific features and functionalities you want to implement in your blog website with an admin panel. You may need to expand or modify this diagram based on additional features such as user roles, permissions, and other business rules specific to your application.

* Schema Explanation :

Creating a Explanation diagram for a blog website with an admin panel involves identifying and illustrating the key entities, relationships, and attributes that make up the database structure. Below is a simplified example of a schema diagram for a blog website with an admin panel. This example assumes a basic structure with users, blog posts, comments, and administrative features.

**Explanation:**

- Users Table: Represents information about users who interact with the blog. The "Role" attribute differentiates between regular users and administrators.

- Blog Posts Table: Contains information about individual blog posts, including the post title, content, publication date, and the user who authored the post.

- Comments Table: Stores comments made on blog posts. Each comment has a text, date, and is associated with both a user and a specific blog post.

- Categories Table: Represents different categories or topics that can be assigned to blog posts.

- Admin Panel Table: Manages administrative users. It includes a foreign key (UserID) linking to the Users table and a "Permissions" field to specify the level of administrative access.

This is a basic example, and in a real-world scenario, you might need to add more details, such as tags, images, or additional features based on your specific requirements. The diagram serves as a starting point and can be extended based on the features and functionality you want to include in your blog website.

* **System Requirements** –
* Functional Requirements :

Functional requirements for a blog website with an admin panel typically encompass various features and capabilities to ensure the effective functioning of the website. Here is a list of functional requirements that you might consider:

1. User Registration and Authentication:

- Users should be able to register for an account.

- Authentication mechanisms to ensure secure access.

2. User Roles:

- Distinct user roles, such as Admin, Author, and Regular User.

- Different access levels for each role.

3. Admin Panel:

- Admin login with secure authentication.

- Ability to manage and edit user accounts.

- Access control to manage and moderate blog posts and comments.

4. Content Management:

- Ability for authors to create, edit, and delete blog posts.

- Support for different content types (text, images, videos, etc.).

- WYSIWYG editor for easy content creation.

5. Categories and Tags:

- Ability to categorize blog posts into different topics.

- Tagging system to enhance searchability.

6. Comments:

- Users (including non-registered users) should be able to comment on blog posts.

- Moderation features to manage and delete inappropriate comments.

7. Search Functionality:

- A search bar to allow users to search for specific blog posts.

- Advanced search features, including filters by category, date, and tags.

8. User Profile:

- User profile pages displaying information about the user and their posts.

- Option to customize user profiles (avatar, bio, etc.).

9. Social Media Integration:

- Share buttons for social media platforms.

- Integration with social media logins for user registration.

10. Responsive Design:

- A responsive design that works well on various devices (desktop, tablet, mobile).

11. SEO Optimization:

- SEO-friendly URLs, meta tags, and other SEO best practices.

12. Analytics:

- Integration with analytics tools to track website traffic and user behavior.

13. Backup and Restore:

- Regular backups of the website data.

- Ability to restore the website to a previous state if needed.

15. Security:

- Secure login mechanisms (e.g., HTTPS, encryption).

16. Reporting:

- Reporting system for users to report inappropriate content or issues.

- Admin dashboard to view and manage reported issues.

17. Monetization (Optional):

- Integration with advertising platforms.

- Support for sponsored content.

18. Accessibility:

- Compliance with accessibility standards to ensure usability for all users.

19. Legal Compliance:

- Terms of service and privacy policy pages.

- Compliance with data protection regulations.

20. Feedback Mechanism:

- A feedback form or system for users to provide input or report issues.

Remember, these requirements can be tailored based on specific needs and preferences. Additionally, continuous testing and feedback from users can help refine and enhance the functionality of the blog website over time.

* Non-functional requirements:

Non-functional requirements are aspects of a system that describe how it performs certain functions, rather than what specific functions it performs. They are crucial for the overall performance, usability, and maintainability of a system. Here are some non-functional requirements that may be relevant for a blog website with an admin panel:

1. Performance:

- Response Time: The system should provide a response within 2 seconds for standard operations.

- Throughput: The system should support a minimum of 1000 concurrent users without degradation in performance.

2. Scalability:

- The system should be designed to easily scale horizontally to accommodate an increase in the number of users and blog posts.

3. Availability:

- The website should have 99.9% uptime, allowing for scheduled maintenance.

- The admin panel should be available 24/7 to authorized personnel.

4. Reliability:

- The system should be able to recover from a failure (e.g., server crash) within 5 minutes without data loss.

5. Security:

- All data transmissions should be encrypted using HTTPS.

- The admin panel should have secure authentication and authorization mechanisms.

- Regular security audits and penetration testing should be conducted.

6. Scalability:

- The system should be designed to handle an increasing number of users, blog posts, and comments over time.

- Database scalability should be ensured to accommodate a growing dataset.

7. Usability:

- The user interface should be intuitive and easy to use for both regular users and administrators.

- Accessibility standards should be followed to ensure the website is usable by people with disabilities.

8. Compatibility:

- The website should be compatible with major web browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).

9. Maintainability:

- Code should be well-documented to facilitate easy maintenance and updates.

- Regular backups of the database and system configuration should be performed.

10. Logging and Monitoring:

- The system should log relevant events, errors, and user activities.

- Monitoring tools should be in place to track performance metrics and alert administrators of potential issues.

11. Compliance:

- The system should comply with relevant data protection regulations (e.g., GDPR).

- Licensing and intellectual property rights should be respected.

12. Backup and Recovery:

- Regular automated backups of the entire system should be performed.

- A recovery plan should be in place to restore the system in case of data loss or system failure.

13. Caching:

- Caching mechanisms should be implemented to improve page load times and reduce server load.

These non-functional requirements will help guide the development and operation of the blog website and admin panel to ensure they meet the necessary performance, security, and usability standards. Adjustments and additional requirements may be needed based on specific project constraints and goals.

* **Code –**
* For Website :
* Index.php ---
* <?php
* require('includes/db.php');
* include('includes/function.php');
* if (isset($\_GET['page'])) {
* $page = $\_GET['page'];
* if ($page < 1) {
* $page = 1;
* }
* } else {
* $page = 1;
* }
* $post\_per\_page = 5;
* $result = ($page - 1) \* $post\_per\_page;
* ?>
* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta http-equiv="X-UA-Compatible" content="IE=edge">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl" crossorigin="anonymous">
* <title>BLOG WEBSITE</title>
* </head>
* <body>
* <?php include\_once('includes/navbar.php'); ?>
* <div>
* <div class="container m-auto mt-3 row">
* <div class="col-8">
* <?php
* if (isset($\_GET['search'])) {
* $keyword = $\_GET['search'];
* $postQuery="SELECT \* FROM posts WHERE title LIKE '%$keyword%' ORDER BY id DESC LIMIT $result,$post\_per\_page";
* }else{
* $postQuery="SELECT \* FROM posts ORDER BY id DESC LIMIT $result,$post\_per\_page";
* }
* $runPQ=mysqli\_query($db,$postQuery);
* while($post=mysqli\_fetch\_assoc($runPQ)){
* ?>
* <div class="card mb-3" style="max-width: 800px;">
* <a href="post.php?id=<?=$post['id']?>" style="text-decoration:none;color:black">
* <div class="row g-0">
* <div class="col-md-5" style="background-image: url('images/<?getPostThumb($db,$post['id'])?>');background-size: cover">
* <img src="" alt="...">
* </div>
* <div class="col-md-7">
* <div class="card-body">
* <h5 class="card-title"><?=$post['title']?></h5>
* <p class="card-text text-truncate"><?=$post['content']?></p>
* <p class="card-text"><small class="text-muted">Posted on <?=date('F jS,Y',strtotime($post['created\_at']))?></small></p>
* </div>
* </div>
* </div>
* </a>
* </div>
* <?php
* }
* ?>

* </div>
* <?php include\_once('includes/sidebar.php'); ?>
* </div>
* <?php
* if (isset($\_GET['search'])) {
* $keyword =$\_GET['search'];
* $q="SELECT \* FROM posts WHERE title LIKE '%$keyword%'";
* }
* else{
* $q="SELECT \* FROM posts";
* }
* $r = mysqli\_query($db, $q);
* $total\_posts = mysqli\_num\_rows($r);
* $total\_pages = ceil($total\_posts / $post\_per\_page);
* ?>
* <nav aria-label="Page navigation example">
* <ul class="pagination justify-content-center">
* <?php
* if($page>1){
* $switch="";
* }else{
* $switch="disabled";
* }
* if($page<$total\_pages){
* $nswitch="";
* }else{
* $nswitch="disabled";
* }
* ?>
* <li class="page-item <?=$switch?>">
* <a class="page-link" href="?<?php if(isset($\_GET['search'])){ echo"search=$keyword&";}?>page=<?=$page-1?>" tabindex="-1" aria-disabled="true">Previous</a>
* </li>
* <?php
* for ($opage = 1; $opage<=$total\_pages;$opage++) {
* ?>
* <li class="page-item "><a class="page-link" href="?<?php if (isset($\_GET['search'])) { echo "search=$keyword&"; } ?>page=<?=$opage?>"><?=$opage?></a></li>
* <?php
* }
* ?>
* <li class="page-item <?=$nswitch?>">
* <a class="page-link" href=" ?<?php  if (isset($\_GET['search'])) { echo "search=$keyword&"; } ?>page=<?=$page+1?>">Next</a>
* </li>
* </ul>
* </nav>

* <?php include\_once('includes/footer.php');?>
* <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0" crossorigin="anonymous"></script>
* </body>
* </html>
* Post.php ---
* <?php
* require('includes/db.php');
* require('includes/function.php');
* ?>
* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta http-equiv="X-UA-Compatible" content="IE=edge">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl" crossorigin="anonymous">
* <title>Blog website with admin panel</title>
* <style>
* .Feature{
* position: absolute;
* right:2px ;
* }
* </style>
* </head>
* <body>
* <?php include\_once('includes/navbar.php'); ?>
* <div class="container m-auto mt-3 row">
* <div class="col-8">
* <?php
* $post\_id=$\_GET['id'];
* $postQuery="SELECT \* FROM posts WHERE id=$post\_id";
* $runPQ=mysqli\_query($db,$postQuery);
* $post=mysqli\_fetch\_assoc($runPQ);
* ?>
* <div class="card mb-3">
* <div class="card-body">
* <h5 class="card-title"><?=$post['title']?></h5>
* <span class="badge bg-primary ">Posted on <?=date('F jS, Y',strtotime($post['created\_at']))?></small></span>
* <span class="badge bg-danger"><?=getCategory($db,$post['category\_id'])?></span>
* <div class="border-bottom mt-3"></div>
* <?php
* $post\_images=getImagesByPost($db,$post['id']);
* ?>
* <div id="carouselExampleRide" class="carousel slide" data-bs-ride="true">
* <div class="carousel-inner">
* <?php
* $c=1;
* foreach($post\_images as $image){
* if($c>1){
* $sw = "";
* }else{
* $sw = "active";
* }
* ?>
* <div class="carousel-item <?=$sw?> ">
* <img src="images/<?=$image['image']?>" class="d-block w-100" alt="...">
* </div>
* <?php
* $c++;
* }
* ?>
* </div>
* <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleRide" data-bs-slide="prev">
* <span class="carousel-control-prev-icon" aria-hidden="true"></span>
* <span class="visually-hidden">Previous</span>
* </button>
* <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleRide" data-bs-slide="next">
* <span class="carousel-control-next-icon" aria-hidden="true"></span>
* <span class="visually-hidden">Next</span>
* </button>
* </div>
* <!--    <img src="https://images.moneycontrol.com/static-mcnews/2020/04/stock-in-the-news-770x433.jpg" class="img-fluid mb-2 mt-2" alt="Responsive image"> -->
* <p class="card-text"><?=$post['content']?>
* </p>
* <div class="sharethis-inline-share-buttons"></div>
* <br>
* <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
* Comment On This
* </button>
* </div>
* </div>
* <!-- Button trigger modal -->
* <!-- Modal -->
* <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
* <div class="modal-dialog">
* <div class="modal-content">
* <div class="modal-header">
* <h1 class="modal-title fs-5" id="exampleModalLabel">Add Your Comment</h1>
* <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
* </div>
* <div class="modal-body">
* <form action="includes/add\_comment.php" method="post">
* <div class="mb-3">
* <label for="exampleInputEmail1" class="form-label">Name</label>
* <input type="name" class="form-control" name="name" id="exampleInputEmail1" aria-describedby="emailHelp">
* <div class="mb-3">
* <label for="exampleInputPassword1" class="form-label">Comment</label>
* <input type="text" class="form-control" name="comment" id="exampleInputPassword1">
* </div>
* <input type="hidden" name="post\_id" value="<?=$post\_id?>">
* <button type="submit" name="addcomment" class="btn btn-primary">Add Comment</button>
* </form>
* </div>
* </div>
* </div>
* </div>
* </div>
* <div>
* <h4>Related Posts</h4>
* <?php
* $pquery = "SELECT \* FROM posts WHERE category\_id={$post['category\_id']} ORDER BY id DESC";
* $prun = mysqli\_query($db, $pquery);
* while ($rpost = mysqli\_fetch\_assoc($prun)) {
* if ($rpost['id'] == $post\_id) {
* continue;
* }
* ?>
* <div class="card mb-3" class="related-post-link">
* <a href="post.php?id=<?=$rpost['id']?>"  style="text-decoration:none;color:black">
* <div class="row g-0">
* <div class="col-md-5" style="background-image: url();background-size: cover">
* <img src="" alt="...">
* </div>
* <div class="col-md-7">
* <div class="card-body">
* <h5 class="card-title"><?=$rpost['title']?></h5>
* <p class="card-text text-truncate"><?=$rpost['content']?></p>
* <p class="card-text"><small class="text-muted">Posted on <?=date('F jS, Y',strtotime($rpost['created\_at']))?></small></p>
* </div>
* </div>
* </div>
* </div>
* </a>
* <?php
* }
* ?>
* </div>
* </div>
* <?php include\_once('includes/sidebar.php');
* ?>
* </div>
* <nav class="navbar navbar-expand-lg navbar-light bg-light border-top">
* <div class="container m-auto">
* <a href="#" class="m-auto" style="text-decoration: none;">Developed by Aayush Singh </a>
* </div>
* </nav>
* <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0" crossorigin="anonymous"></script>
* <script type='text/javascript' src='https://platform-api.sharethis.com/js/sharethis.js#property=64bec2aa60781a00121c7f8b&product=inline-share-buttons' async='async'></script>
* </body>
* </html>
* Function –
* Navbar.php :-
* <nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark">
* <div class="container">
* <a class="navbar-brand" href="#">MyBlog</a>
* <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarScroll"
* aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle navigation">
* <span class="navbar-toggler-icon"></span>
* </button>
* <div class="collapse navbar-collapse" id="navbarScroll">
* <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll" style="--bs-scroll-height: 100px;">
* <?php
* $navQuery = "SELECT \* FROM menu";
* $runNav = mysqli\_query($db, $navQuery);
* while ($menu = mysqli\_fetch\_assoc($runNav)) {
* $no = getsmenuNo($db, $menu['id']);
* if (!$no) {
* ?>
* <li class="nav-item">
* <a class="nav-link" aria-current="page" href="<?=$menu['action']?>"><?=$menu['name']?></a>
* </li>
* <?php
* } else {
* ?>
* <li class="nav-item dropdown">
* <a class="nav-link dropdown-toggle" href="<?=$menu['action']?>" id="navbarScrollingDropdown" role="button"
* data-bs-toggle="dropdown" aria-expanded="false">
* <?=$menu['name']?>
* </a>
* <ul class="dropdown-menu" aria-labelledby="navbarScrollingDropdown">
* <?php
* $submenus = getsmenu($db, $menu['id']);
* foreach ($submenus as $sm) {
* ?>
* <li><a class="dropdown-item" href="<?=$sm['action']?>"><?=$sm['name']?></a></li>
* <?php
* }
* ?>
* </ul>
* </li>
* <?php
* }
* }
* ?>
* </ul>
* <form class="d-flex">
* <input class="form-control me-2" name="search" type="search" placeholder="Search" aria-label="Search">
* <button class="btn btn-outline-success" type="submit">Search</button>
* </form>
* </div>
* </div>
* </nav>
* Sidebar.php :-
* <div class="col-4">

* <?php
* if(isset($\_GET['id'])){
* ?>
* <div class="card mb-3">
* <h5 class="card-header">Comments </h5>
* <?php
* $comments = getComments($db,$post\_id);
* if(count($comments)<1){
* echo '<div class="card-body"><p class="text-center card-text">No Comments...</p></div>';
* }
* foreach($comments as $comment){
* ?>
* <div class="card-body">
* <h5 class="card-title" style="margin-bottom: 0;"> <?=$comment['name']?></h5>
* <span class="text-secondary"><small><?= date('F jS, Y', strtotime($comment['created\_at'])) ?></small></span>
* <p class="card-text"><?=$comment['comment']?></p>
* </div>
* <?php
* }
* ?>
* </div>
* <?php
* }
* ?>
* </div>
* Footer.php :-
* <nav class="navbar navbar-expand-lg navbar-light bg-light border-top">
* <div class="container m-auto">
* <a href="#" class="m-auto" style="text-decoration: none;">Developed by Aayush Singh</a>
* </div>
* </nav>
* For Admin Panel :-
* Index.php ----
* <?php
* require('../includes/db.php');
* require('../includes/function.php');
* if (!isset($\_SESSION['isUserloggedIn'])) {
* header('Location:login.php');
* }
* $admin=getAdminInfo($db, $\_SESSION['email']); // Assign the result to the $admin variable
* ?>
* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="utf-8">
* <meta content="width=device-width, initial-scale=1.0" name="viewport">
* <title>MyBlog-Admin panel</title>
* <meta content="" name="description">
* <meta content="" name="keywords">
* <!-- Favicons -->
* <link href="assets/img/favicon.png" rel="icon">
* <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">
* <!-- Google Fonts -->
* <link href="https://fonts.gstatic.com" rel="preconnect">
* <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">
* <!-- Vendor CSS Files -->
* <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
* <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
* <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
* <link href="assets/vendor/quill/quill.snow.css" rel="stylesheet">
* <link href="assets/vendor/quill/quill.bubble.css" rel="stylesheet">
* <link href="assets/vendor/remixicon/remixicon.css" rel="stylesheet">
* <link href="assets/vendor/simple-datatables/style.css" rel="stylesheet">
* <!-- Template Main CSS File -->
* <link href="assets/css/style.css" rel="stylesheet">
* </head>
* <body>
* <!-- ======= Header ======= -->
* <header id="header" class="header fixed-top d-flex align-items-center">
* <div class="d-flex align-items-center justify-content-between">
* <a href="index.php" class="logo d-flex align-items-center">
* <img src="assets/img/logo.png" alt="">
* <span class="d-none d-lg-block">CONTROL CENTRE</span>
* </a>
* </div><!-- End Logo -->
* <nav class="header-nav ms-auto">
* <ul class="d-flex align-items-center">
* </li><!-- End Search Icon-->
* <li class="dropdown">
* <a class="dropdown-toggle" href="#" data-bs-toggle="dropdown">
* <span class="username"><?=$admin['full-name']?></span>
* </a><!-- End Profile Iamge Icon -->
* <ul class="dropdown-menu dropdown-menu-end dropdown-menu-arrow profile">
* <li class="dropdown-header">
* <h6>Aayush Singh</h6>
* <span>Web Developer</span>
* </li>
* <li>
* <hr class="dropdown-divider">
* </li>
* <li>
* <a class="dropdown-item d-flex align-items-center" href="users-profile.html">
* <i class="bi bi-person"></i>
* <span>My Account</span>
* </a>
* </li>
* <li>
* <hr class="dropdown-divider">
* </li>
* <li>
* <a class="dropdown-item d-flex align-items-center" href="../includes/logout.php">
* <i class="bi bi-box-arrow-right"></i>
* <span>Log Out</span>
* </a>
* </li>
* </ul><!-- End Profile Dropdown Items -->
* </li><!-- End Profile Nav -->
* </ul>
* </nav><!-- End Icons Navigation -->
* </header><!-- End Header -->
* </body>
* <!-- ======= Sidebar ======= -->
* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="utf-8">
* <meta content="width=device-width, initial-scale=1.0" name="viewport">
* <title>MyBlog-Admin panel</title>
* <!-- Include Bootstrap CSS -->
* <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
* <!-- Include your custom CSS if needed -->
* <link href="assets/css/style.css" rel="stylesheet">
* <!-- Include CKEditor from CDN -->
* <script src="https://cdn.ckeditor.com/4.17.2/standard/ckeditor.js"></script>
* <!-- Add any additional styles or scripts here -->
* <style>
* .button-gap {
* margin-top: 10px;
* }
* </style>
* </head>
* <body>
* <!-- Sidebar -->
* <aside id="sidebar" class="sidebar">
* <ul class="sidebar-nav" id="sidebar-nav">
* <li class="active">
* <a class="nav-link active" href="index.php">
* <i class="bi bi-grid"></i>
* <span>Add Post</span>
* </a>
* </li>
* <li class="nav-item">
* <a class="nav-link" d href="index.php?managepost">
* <i class="bi bi-menu-button-wide"></i>
* <span>Manage Post</span>
* </a>
* </li>
* <li class="active">
* <a class="nav-link" href="index.php?managecategory">
* <i class="bi bi-layout-text-window-reverse"></i>
* <span>Manage Category</span>
* </a>
* </li>
* <li class="nav-item">
* <a class="nav-link" href="index.php?managemenu">
* <i class="bi bi-bar-chart"></i>
* <span>Manage Menu</span>
* </a>
* </li>
* </ul>
* </aside>
* <!-- End Sidebar -->
* <!-- Main Content -->
* <section id="main-content">
* <section class="wrapper">
* <main id="main" class="main">
* <div class="row">
* <?php
* if(isset($\_GET['managepost'])){
* ?>
* <div class="col-lg-12">
* <h5 class="card-title font-weight-bold">Posts
* <html lang="en">
* <table class="table table-bordered table-striped">
* <thead>
* <tr>
* <th>#</th>
* <th> Post Title </th>
* <th> Post Category </th>
* <th>Post Date</th>
* <th>Action</th>

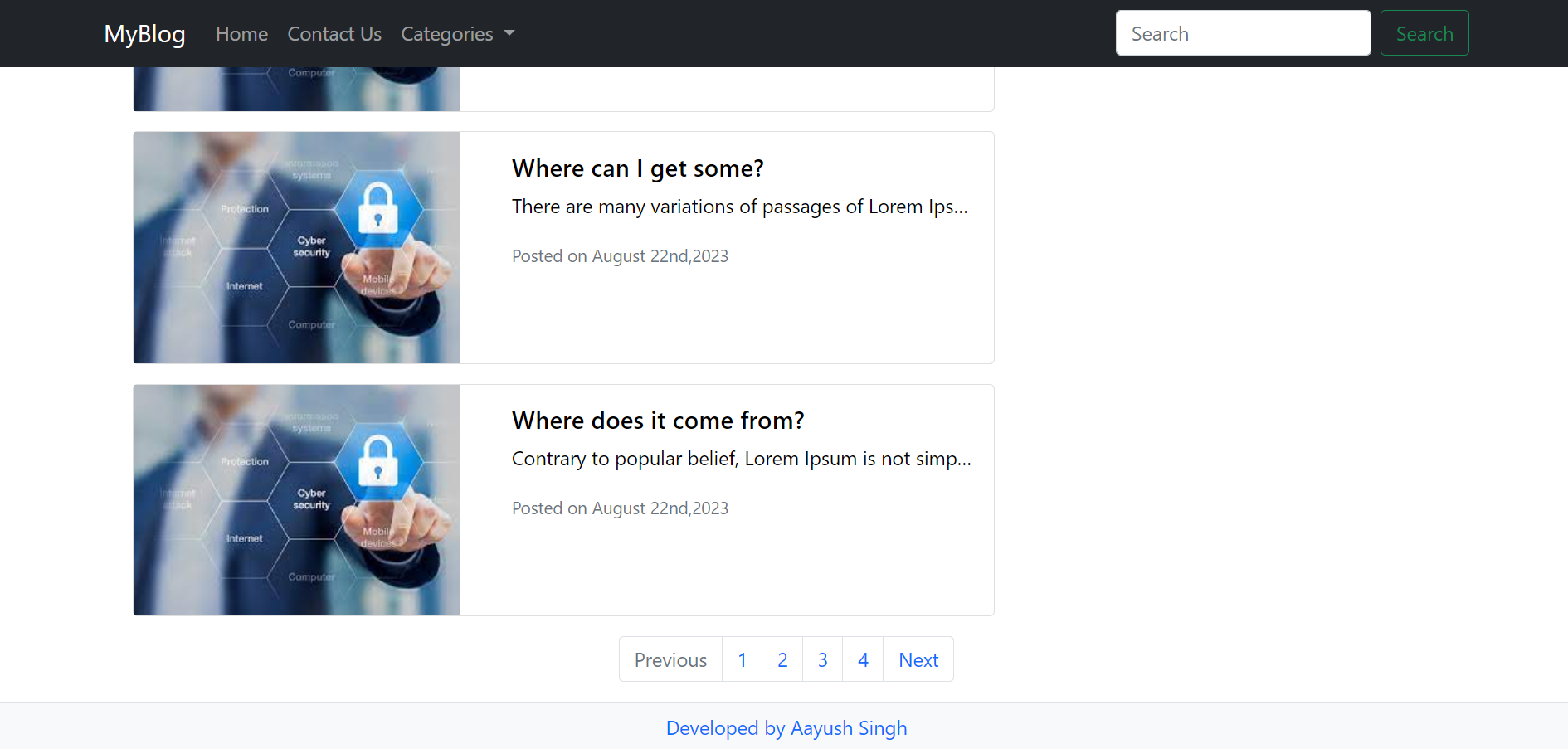
* </tr>
* </thead>
* <tbody>
* <?php
* $posts = getAllPosts($db);
* $count=1;
* foreach($posts as $post){
* ?>
* <tr>
* <td><?=$count?></td>
* <td><?=$post['title']?></td>
* <td><?=getCategory($db,$post['category\_id'])?></td>
* <td><?=date('F jS,Y',strtotime($post['created\_at']))?></td>
* <td >
* <div class="btn-group">
* <a href="../includes/removepost.php?id=<?=$post['id']?>" class="btn btn-danger">Remove</a>
* </div>
* </td>
* </tr>
* <?php
* $count++;
* }
* ?>
* <!-- Add more rows with user data as needed -->
* </tbody>
* </table>
* <?php
* }else if(isset($\_GET['managemenu'])){
* ?>
* <div class="col-lg-12">
* <h5 class="card-title font-weight-bold">Menu --
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>    Add New Menu
* </title>
* <!--
* <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
* </head>
* <body>
* <!-- Button to trigger the modal -->
* <button type="button" class="text-primary" data-toggle="modal" href="#myModal">
* Add New Menu
* </button>
* <a>
* <!-- Modal -->
* <div class="modal fade" id="myModal">
* <div class="modal-dialog">
* <div class="modal-content">
* <!-- Modal Header -->
* <div class="modal-header">
* <label class="modal-title">Add New Menu </label>
* <button type="button" class="close" data-dismiss="modal">&times;</button>
* </div>
* <!-- Modal body -->
* <div class="modal-body">
* <form role="form" method="post" action="../includes/addmenu.php">
* <div class="form-group">
* <label for="email">Menu Title:</label>
* <input type="text" name="menu\_name" class="form-control" id="email" placeholder="Enter Menu Name...">
* </div>
* <div class="form-group">
* <label for="email">Menu Link:</label>
* <input type="text" name="menu\_link" class="form-control" id="email" value="#" placeholder="Enter Menu Link...">
* </div>

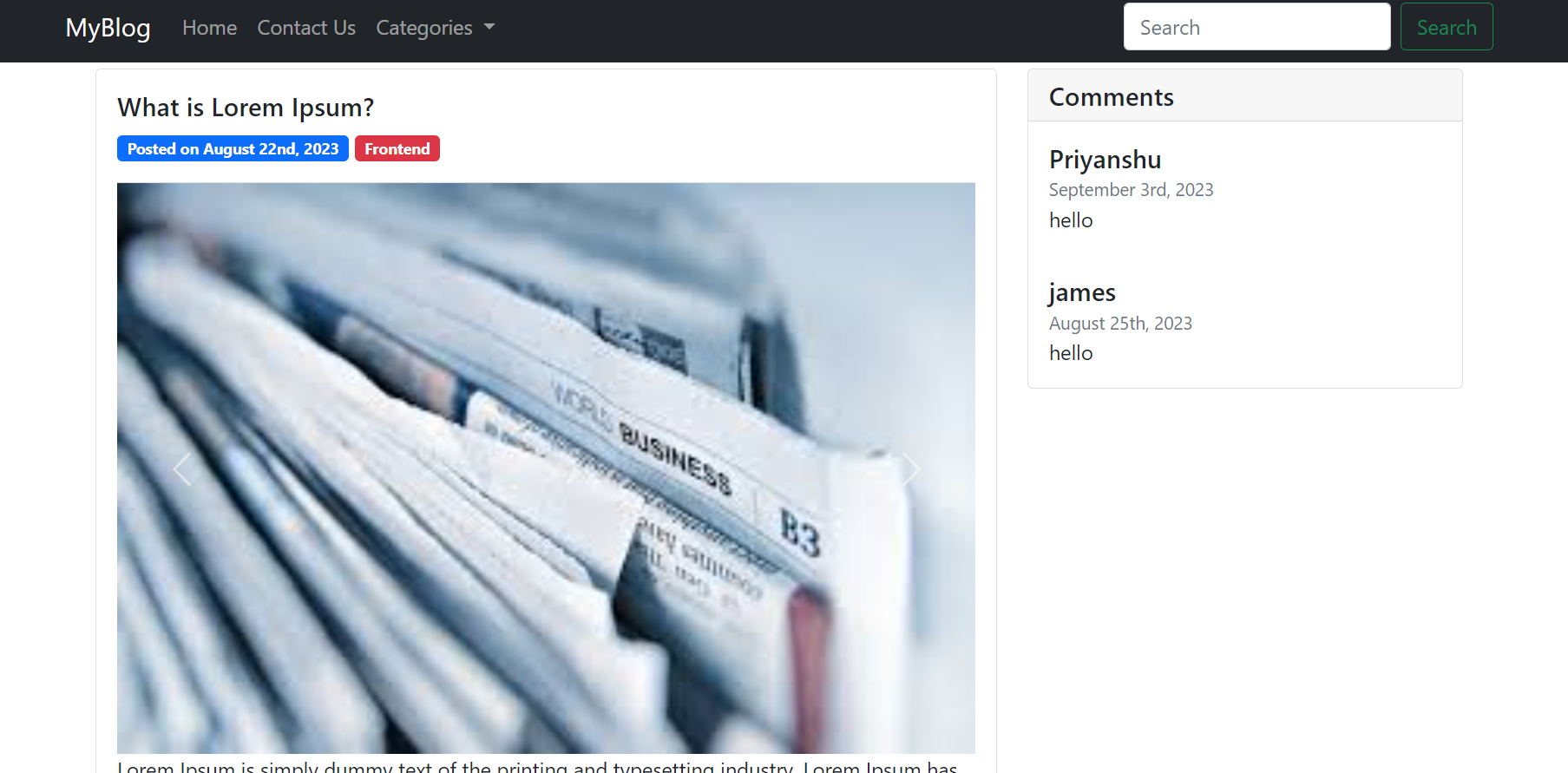

* <!-- Modal footer -->
* <div class="modal-footer">
* <!-- Add JavaScript to trigger the form submission -->
* <button type="submit" name="addmenu" class="btn btn-primary" id="addCategoryBtn">Add</button>
* </form>
* </div>
* </div>
* </div>
* </div>
* <script>
* // Add an event listener to the "Add" button
* document.getElementById('addCategoryBtn').addEventListener('click', function () {
* // Submit the form when the button is clicked
* document.querySelector('form').submit();
* });
* </script>
* <!-- Add Bootstrap JS (make sure to include Bootstrap and jQuery libraries) -->
* <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
* <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
* <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
* </body>
* </html>
* </h5>
* </div>
* <table class="table table-bordered table-striped">
* <thead>
* <tr>
* <th>#</th>
* <th> Menu </th>
* <th>Link</th>
* <th>Action</th>

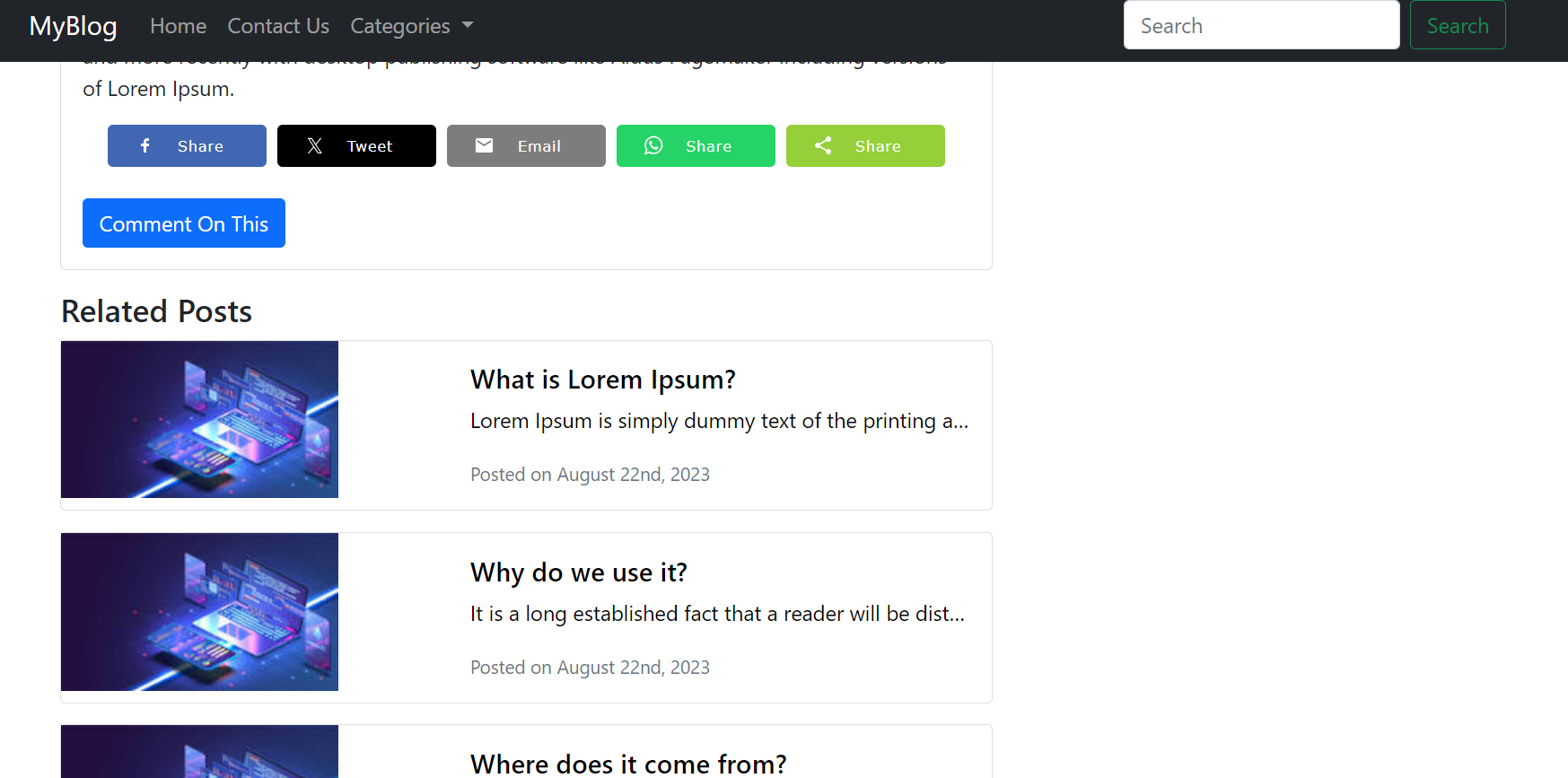
* </tr>
* </thead>
* <tbody>
* <?php
* $menus = getmenu($db);
* $count=1;
* foreach($menus as $menu){
* ?>
* <tr>
* <td><?=$count?></td>
* <td><?=$menu['name']?></td>
* <td><?=$menu['action']?></td>
* <td >
* <div class="btn-group">
* <a href="../includes/removemenu.php?id=<?=$menu['id']?>" class="btn btn-danger">Remove</a>
* </div>
* </td>
* </tr>
* <?php
* $count++;
* }
* ?>
* <!-- Add more rows with user data as needed -->
* </tbody>
* </table>
* <div class="col-lg-12">
* <h5 class="card-title font-weight-bold">SubMenu -- <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>    Add New SubMenu
* </title>
* <!-- Add Bootstrap CSS (make sure to include Bootstrap and jQuery libraries) -->
* <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
* </head>
* <body>
* <button type="button" class="text-primary" data-toggle="modal" href="#myModal1">
* Add New Submenu
* </button></a>
* <!-- Modal -->
* <div class="modal fade" id="myModal1">
* <div class="modal-dialog">
* <div class="modal-content">
* <!-- Modal Header -->
* <div class="modal-header">
* <label class="modal-title">Add New SubMenu</label>
* <button type="button" class="close" data-dismiss="modal">&times;</button>
* </div>
* <!-- Modal body -->
* <div class="modal-body">
* <form role="form" method="post" action="../includes/addmenu.php">
* <div class="form-group">
* <label for="email">SubMenu Title:</label>
* <input type="text" name="submenu-name" class="form-control" id="email" placeholder="Enter Menu Name...">
* </div>
* <div class="form-group">
* <label for="email">Select Parent menu:</label>
* <select name="parent-id" class="form-control" id="email" >
* <?php
* $mlist = getAllmenu($db);
* foreach($mlist as $m){
* ?>
* <option value="<?=$m['id']?>"><?=$m['name']?></option>
* <?php
* }
* ?>
* </select>
* </div>
* <div class="form-group">
* <label for="email">SubMenu Link:</label>
* <input type="text" name="submenu-link" class="form-control" id="email" value="#" placeholder="Enter Menu Link...">
* </div>
* <!-- Add JavaScript to trigger the form submission -->
* <button type="submit" name="addsmenu" class="btn btn-primary" id="addCategoryBtn">Add</button>
* </form>
* </div>
* </div>
* </div>
* </div>
* <script>
* // Wrap your code in a document ready function
* // Add an event listener to the "Add" button
* document.getElementById('addCategoryBtn').addEventListener('click', function () {
* // Submit the form when the button is clicked
* document.querySelector('form').submit();
* });
* </script>
* <!-- Add Bootstrap JS (make sure to include Bootstrap and jQuery libraries) -->
* <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
* <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
* <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
* </body>
* </html>
* </h5>
* </div>
* <table class="table table-bordered table-striped">
* <thead>
* <tr>
* <th>#</th>
* <th> Sub Menu </th>
* <th> Parent Menu </th>
* <th>Link</th>
* <th>Action</th>
* </tr>
* </thead>
* <tbody>
* <?php
* $smenus = getAllsmenu($db);
* $count=1;
* foreach($smenus as $menu){
* ?>
* <tr>
* <td><?=$count?></td>
* <td><?=$menu['name']?></td>
* <td><?=getMenuName($db,$menu['parent\_menu\_id'])?></td>
* <td><?=$menu['action']?></td>
* <td >
* <div class="btn-group">
* <a href="../includes/removesubmenu.php?id=<?=$menu['id']?>" class="btn btn-danger">Remove</a>
* </div>
* </td>
* </tr>
* <?php
* $count++;
* }
* ?>
* <!-- Add more rows with user data as needed -->
* </tbody>
* </table>
* <?Php
* }
* else if(isset($\_GET['managecategory'])){
* ?><div class="col-lg-12">
* <h5 class="card-title font-weight-bold">Category -- <!DOCTYPE html>
* <html lang="en">
* <!-- Add Bootstrap CSS (make sure to include Bootstrap and jQuery libraries) -->
* <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
* </head>
* <body>
* <!-- Button to trigger the modal -->
* <button type="button" class="text-primary" data-toggle="modal" href="#myModal">
* Add New Category
* </button>
* <!-- Modal -->
* <div class="modal fade" id="myModal">
* <div class="modal-dialog">
* <div class="modal-content">
* <!-- Modal Header -->
* <div class="modal-header">
* <h4 class="modal-title">Add New Category</h4>
* <button type="button" class="close" data-dismiss="modal">&times;</button>
* </div>
* <!-- Modal body -->
* <!-- Modal body -->
* <div class="modal-body">
* <form role="form" method="post" action="../includes/addct.php">
* <div class="form-group">
* <label for="category-name">Category Name:</label>
* <input type="text" name="category\_name" class="form-control" id="category-name" placeholder="Enter Category...">
* </div>
* <!-- Add a hidden input for the 'addct' key to be sent -->
* <button type="submit" name="addct" class="btn btn-primary" id="addCategoryBtn">Add</button>
* </form>
* </div>
* </div>
* </div>
* </div>
* <script>
* // Add an event listener to the "Add" button
* document.getElementById('addCategoryBtn').addEventListener('click', function () {
* // Submit the form when the button is clicked
* document.querySelector('form').submit();
* });
* </script>
* <!-- Add Bootstrap JS (make sure to include Bootstrap and jQuery libraries) -->
* <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
* <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
* <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
* </body>
* </html>
* </h5>
* </div>
* <table class="table table-bordered table-striped">
* <thead>
* <tr>
* <th>#</th>
* <th>Category NAme </th>
* <th>Action</th>
* </tr>
* </thead>
* <tbody>
* <?php
* $categories = getAllCategory($db);
* $count=1;
* foreach($categories as $ct){
* ?>
* <tr>
* <td><?=$count?></td>
* <td><?=$ct['name']?></td>
* <td >
* <div class="btn-group">
* <a class="btn btn-danger" href="../includes/removect.php?id=<?=$ct['id']?>">Remove</a>
* </div>
* </td>
* </tr>
* <?php
* $count++;
* }
* ?>
* <!-- Add more rows with user data as needed -->
* </tbody>
* </table>
* <?php
* }else{
* ?>
* <div class="col-lg-12">
* <h5 class="card-title">Add Post</h5>
* <form action="../includes/addpost.php" method="post" enctype="multipart/form-data" class="form-horizontal">
* <div class="form-group">
* <div class="col-sm-12">
* <label for="post-title">Post Title</label>
* <input type="text" class="form-control" id="post-title" name="post\_title">
* </div>
* </div>
* <div class="form-group">
* <div class="col-sm-12">
* <label for="post-content text-truncate">Post Content</label>
* <textarea id="editor" name="post\_content"></textarea>
* </div>
* </div>
* <div class="row">
* <div class="form-group col-md-6">
* <label for="post-category">Select Post Category</label>
* <select name="post\_category" class="form-control" id="post-category">
* <?php
* // Replace this with actual category data retrieval
* $categories = getAllCategory($db);
* foreach ($categories as $ct) {
* ?>
* <option value="<?=$ct['id']?>"><?=$ct['name']?></option>
* <?php
* }
* ?>
* </select>
* </div>
* <div class="form-group col-md-6">
* <label for="post-category">Upload photo (max 5)</label>
* <input type="file" class="form-control" name="post\_image[]" accept="image/\*" multiple/>
* </div>
* </div>
* <div class="col-md-12 button-gap">
* <input type="submit" name="addpost" class="btn btn-primary" value="Add Post">
* </div>
* </form>
* </div>
* <?php
* }
* ?>
* </div>


* <!-- End Main Content -->
* <!-- Vendor JS Files (Bootstrap) -->
* <script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
* <!-- Initialize CKEditor after the textarea -->
* <script>
* // Replace 'editor' with the ID of your textarea element
* CKEDITOR.replace('editor');
* </script>
* </body>
* </html>
* **login.php---**
* <?php
* require('../includes/db.php'); // Include your database connection script
* if (isset($\_SESSION['isUserloggedIn']) && $\_SESSION['isUserloggedIn']) {
* header('Location:index.php');
* exit;
* }
* if (isset($\_POST['login'])) {
* $email = mysqli\_real\_escape\_string($db, $\_POST['email']);
* $password = mysqli\_real\_escape\_string($db, $\_POST['password']);
* // Use prepared statement to prevent SQL injection
* $query = "SELECT \* FROM admin WHERE email='$email' AND password='$password' ";
* $runQuery = mysqli\_query($db, $query);
* if (mysqli\_num\_rows($runQuery)) {
* session\_start(); // Start the session
* $\_SESSION['isUserloggedIn'] = true;
* $\_SESSION['email'] = $email;
* header('Location: index.php');
* exit(); // Make sure to exit after redirecting
* } else {
* echo "<script>alert('Incorrect email or password!');</script>";
* }
* }
* ?>
* <!  DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="utf-8">
* <meta content="width=device-width, initial-scale=1.0" name="viewport">
* <title> MyBlog-Admin Panel</title>
* <meta content="" name="description">
* <meta content="" name="keywords">
* <!-- Favicons -->
* <link href="assets/img/favicon.png" rel="icon">
* <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">
* <!-- Google Fonts -->
* <link href="https://fonts.gstatic.com" rel="preconnect">
* <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">
* <!-- Vendor CSS Files -->
* <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
* <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
* <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
* <link href="assets/vendor/quill/quill.snow.css" rel="stylesheet">
* <link href="assets/vendor/quill/quill.bubble.css" rel="stylesheet">
* <link href="assets/vendor/remixicon/remixicon.css" rel="stylesheet">
* <link href="assets/vendor/simple-datatables/style.css" rel="stylesheet">
* <!-- Template Main CSS File -->
* <link href="assets/css/style.css" rel="stylesheet">
* </head>
* <body class="login-img3-body">
* <div class="container">
* <div class="login-form" method="post" >
* <section class="section register min-vh-100 d-flex flex-column align-items-center justify-content-center py-4">
* <div class="container">
* <div class="row justify-content-center">
* <div class="col-lg-4 col-md-6 d-flex flex-column align-items-center justify-content-center">
* <div class="d-flex justify-content-center py-4">
* <a href="index.html" class="logo d-flex align-items-center w-auto">
* <img src="assets/img/logo.png" alt="">
* <span class="d-none d-lg-block">ADMIN LOGIN</span>
* </a>
* </div><!-- End Logo -->
* <div class="card mb-3">
* <div class="card-body">
* <div class="pt-4 pb-2">
* <h5 class="card-title text-center pb-0 fs-4">Login to Your Account</h5>
* </div>
* <form method = "POST" class="row g-3 needs-validation" novalidate>
* <div class="col-12">
* <label for="youremail" class="form-label">Email</label>
* <div class="input-group has-validation">
* <span class="input-group-text" id="inputGroupPrepend">@</span>
* <input type="email" name="email" class="form-control" id="email"  required>
* <div class="invalid-feedback">Please enter your Email.</div>
* </div>
* </div>
* <div class="col-12">
* <label for="yourPassword" class="form-label">Password</label>
* <div class="input-group has-validation">
* <span class="input-group-text" id="inputGroupPrepend">\*</span>
* <input type="password" name="password" class="form-control" id="yourPassword" required>
* <div class="invalid-feedback">Please enter your password!</div>
* </div>
* </div>
* </div>
* <div class="col-12">
* <button class="btn btn-primary w-100" name="login" type="submit">Login</button>
* </div>
* </form>
* </div>
* </div>
* </div>
* </div>
* </div>
* </div>

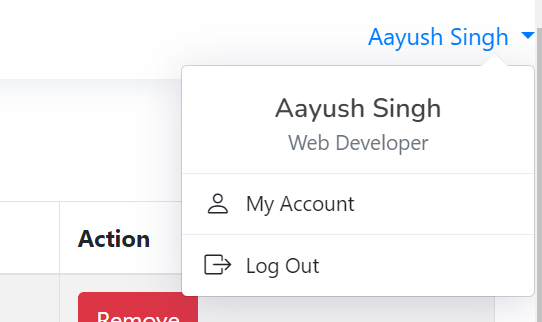
* **Database ---**
* **DB.php :-**
* <?php
* session\_start();
* $host = 'localhost';  // Database host
* $username = 'root';  // Database username
* $password = '';  // Database password
* $database = 'myblog';  // Database name
* // Create a database connection
* $db = mysqli\_connect($host, $username, $password, $database);
* // Check if the connection was successful
* if (!$db) {
* die('Database connection error: ' . mysqli\_connect\_error());
* }
* ?>
* **Snapshots –**
* **Websites**

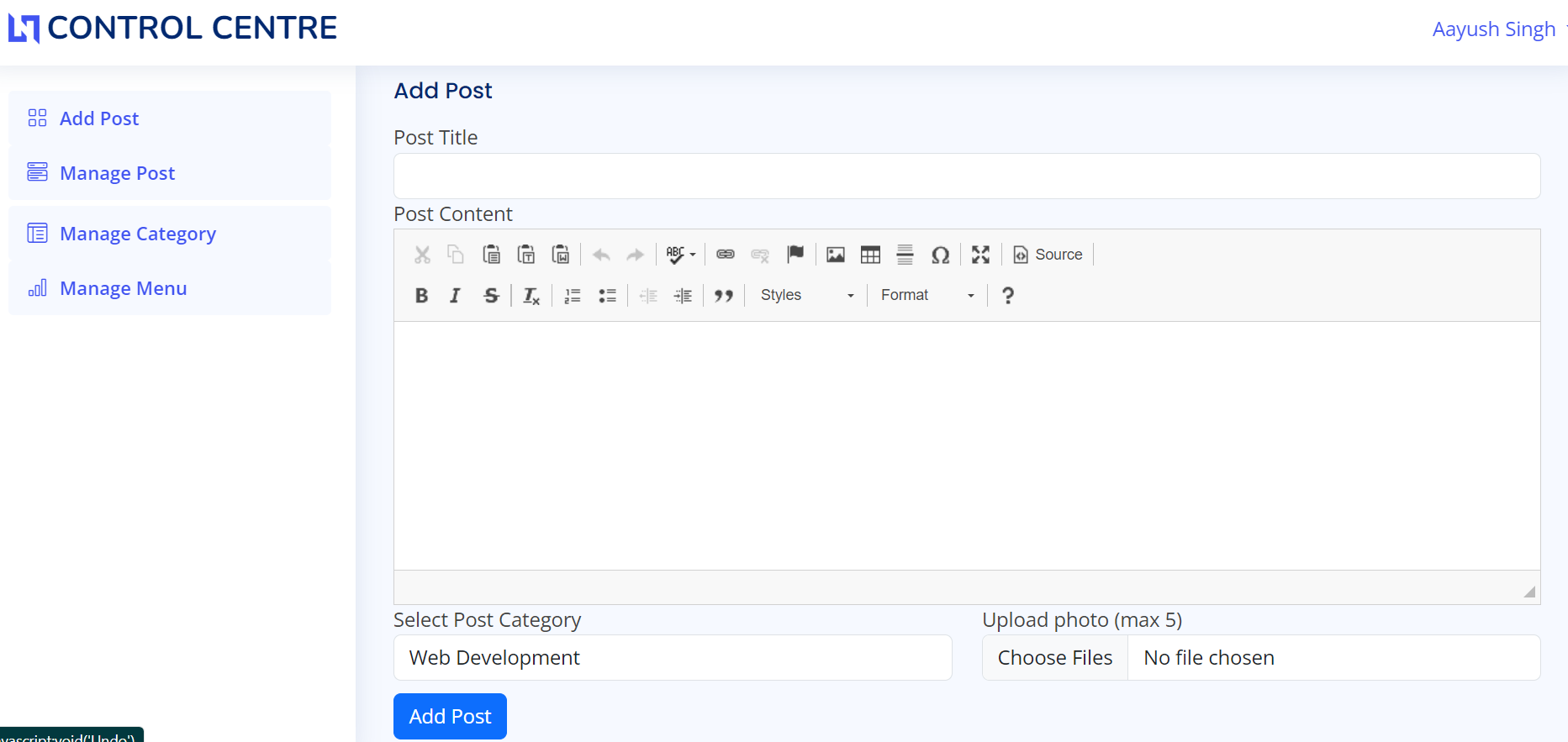
****

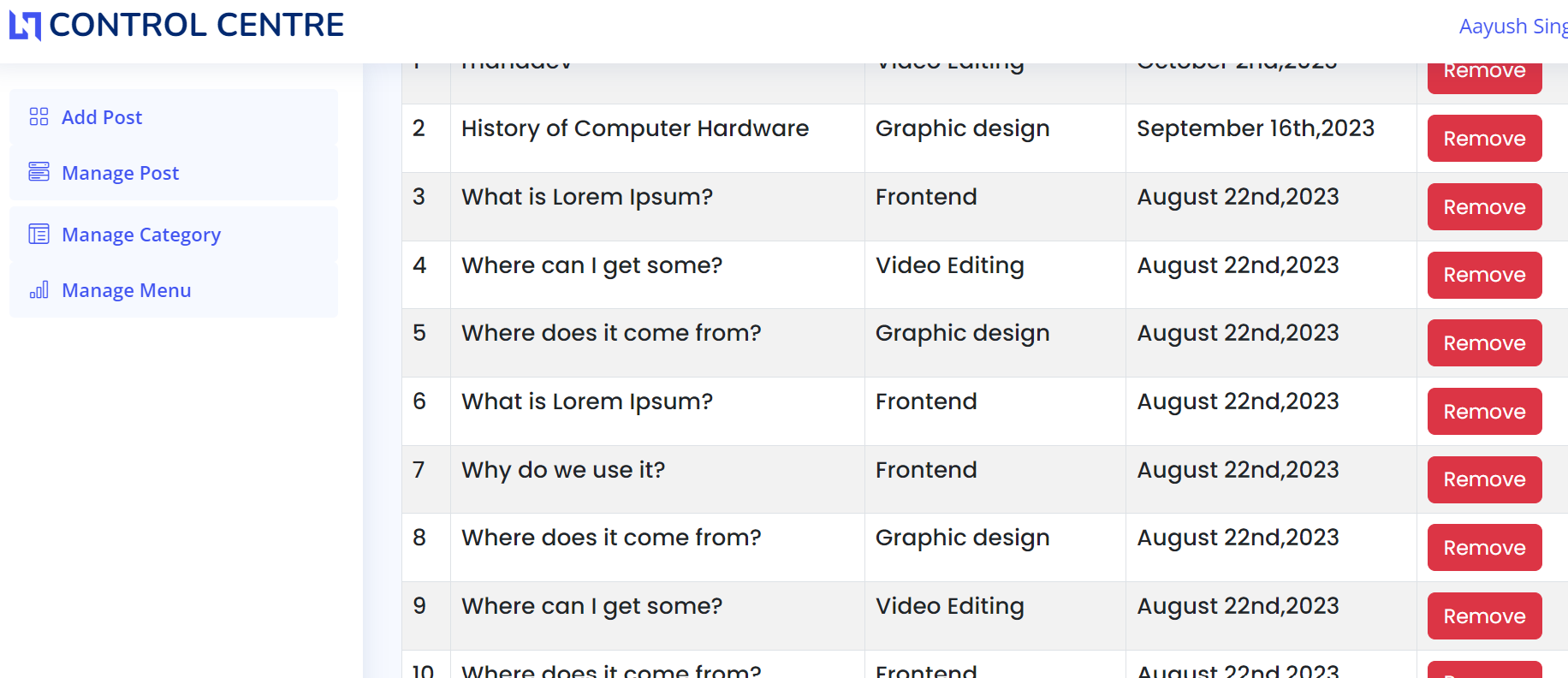


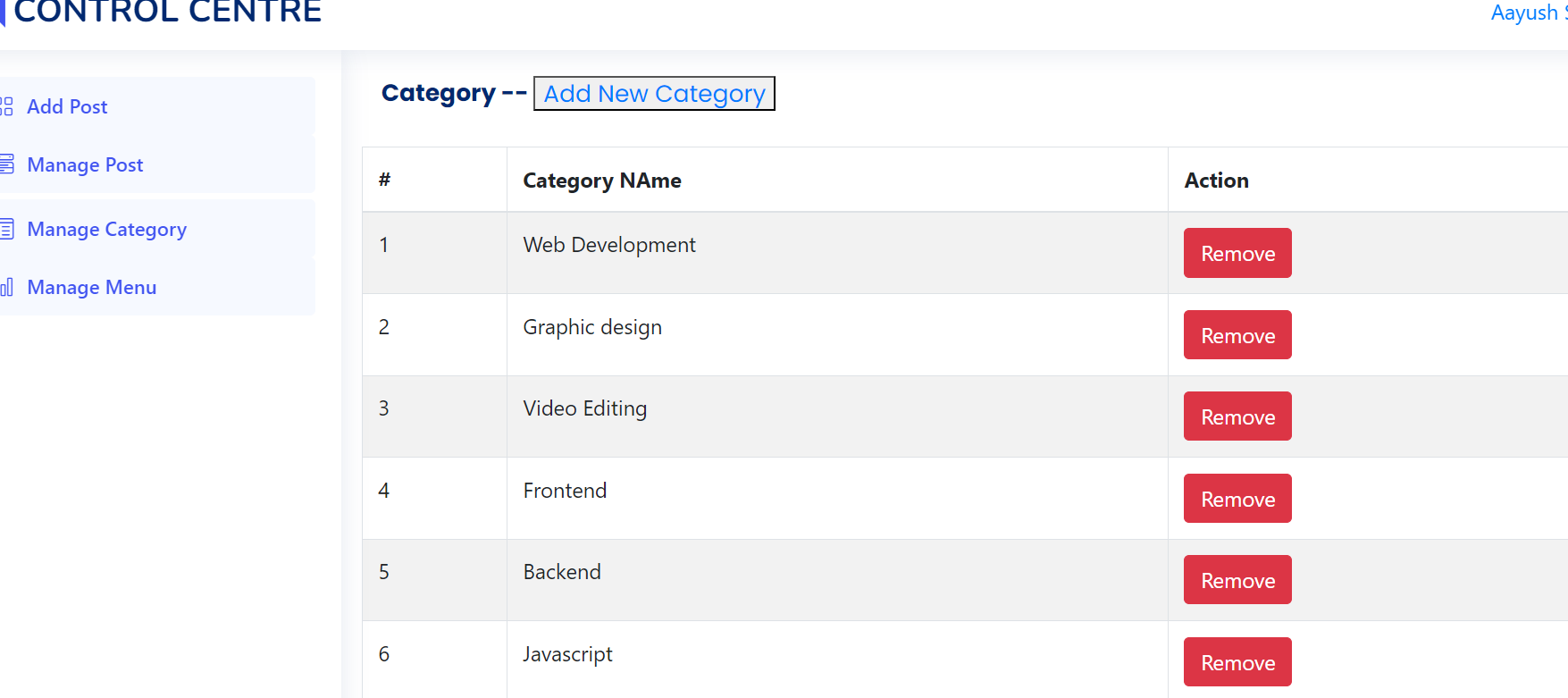


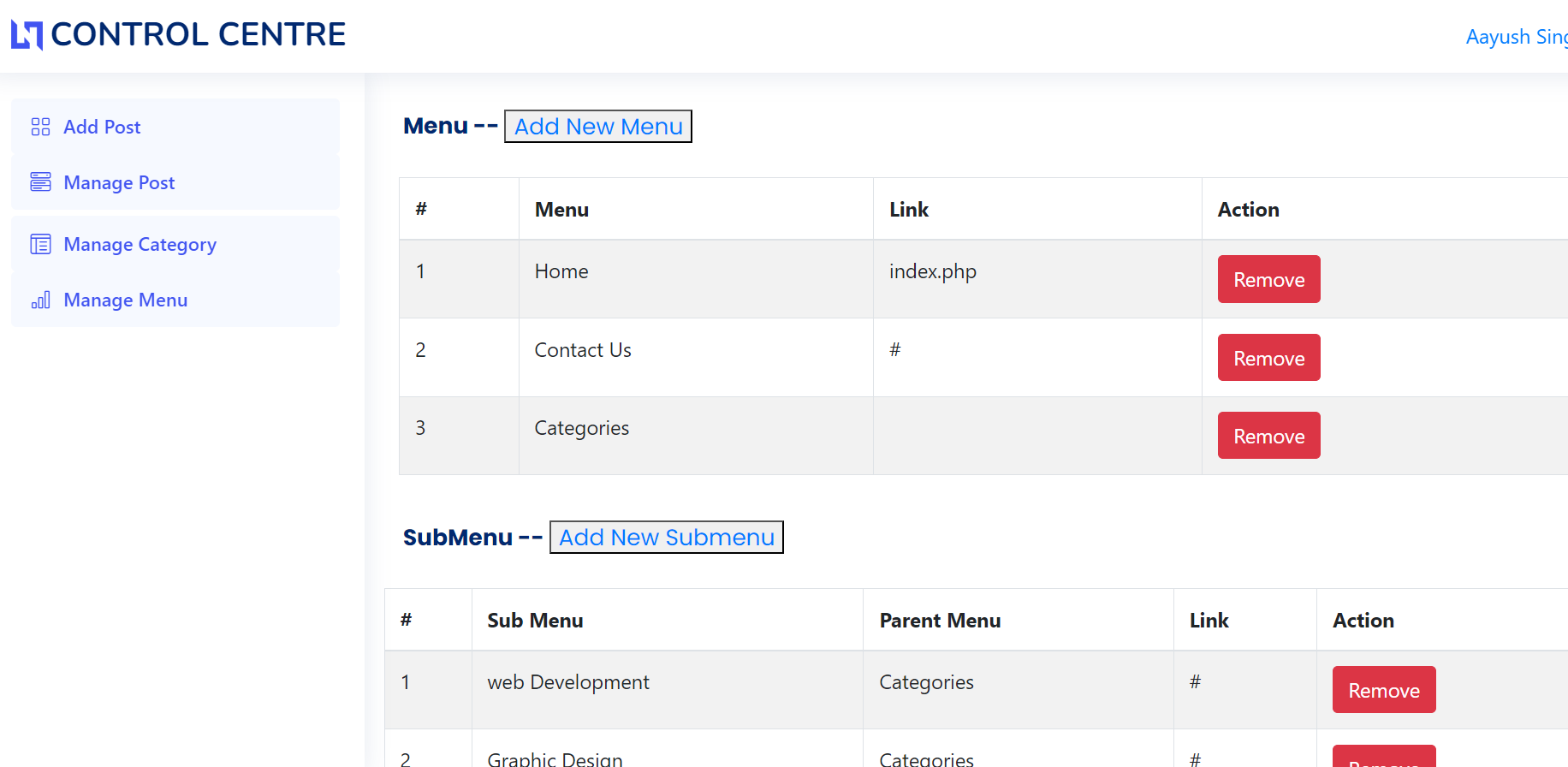
* **Admin Panel**











* **Conclusion –**

In conclusion, the integration of an admin panel into a blog website is a significant enhancement that empowers website owners and administrators with greater control, efficiency, and flexibility in managing their content and user base. The admin panel serves as a centralized hub for various tasks, ranging from content creation and publication to user management and site customization.

With the admin panel, website administrators can streamline the content creation process, ensuring a consistent flow of high-quality posts. The ability to easily manage and organize articles, images, and multimedia content enhances the overall user experience and keeps the website dynamic and engaging.

User management features offered by the admin panel enable administrators to monitor and control user activities, ensuring a secure and respectful online environment. From user registration and authentication to role assignments and content moderation, the admin panel provides robust tools to maintain a well-managed and thriving online community.

Additionally, the customization options available through the admin panel allow website owners to adapt and evolve their platform according to changing trends and user preferences. Whether it's adjusting the website's layout, modifying the theme, or implementing additional features, the admin panel serves as a versatile tool for tailoring the website to meet specific needs.

In conclusion, the inclusion of an admin panel in a blog website is not just a technical feature; it's a strategic asset that empowers administrators to efficiently manage and grow their online presence. It enhances the user experience, ensures content integrity, and provides the flexibility needed to stay relevant in the ever-evolving digital landscape. As the internet continues to evolve, having a robust admin panel becomes increasingly crucial for website owners looking to stay ahead in the dynamic world of online content creation and community management**.**