SIMATS SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

CHENNAI-602105

Blogging Web Application

A CAPSTONE PROJECT REPORT

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

Computer Science and Engineering

Submitted by

Rohith Abishua J (192210179)

Sushil M (192210124)

Under the Supervision of

Ms.B.Jeevashri

JULY 2024

DECLARATION

We, **Sushil M, Rohith Abishua J** students of **Bachelor of Engineering in CSE**, Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Blogging Application** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

Rohith Abishua J (192210179)
Sushil M (192210124)

Date: 31/07/2024
Place: Chennai

CERTIFICATE

This is to certify that the project entitled **“Blogging Web Application”** submitted by **Sushil M, Rohith Abishua J** has been carried out under my supervision. The project has been submitted as per the requirements in the current semester of B.E. Computer Science Engineering.

Teacher-in-charge
Ms.B.Jeevashri

Table of Contents

ABSTRACT

This project involves the development of a dynamic blogging platform that emphasizes user authentication, post creation, and tag-based categorization. The platform was designed to offer an intuitive user experience with features including user-specific post editing and deletion, a responsive search bar, and a visually appealing interface. The frontend was crafted using HTML, CSS, and JavaScript, ensuring a modern and responsive design. The code was developed and managed using Visual Studio Code (VSCode), providing an efficient and streamlined development process.

The backend integrates a robust architecture, leveraging PHP and MySQL for database interactions, ensuring data integrity and security through well-defined authentication mechanisms. The goal of this project was to create a user-friendly blog where content can be easily managed and discovered, providing a seamless and engaging experience for both content creators and readers. This report details the design, implementation, and functionalities of the blog platform, alongside challenges faced and solutions devised during the development process.

1. INTRODUCTION

The Blogster web application is an endeavor to create a modern and interactive platform for users to share and manage content through blog posts. In an era where digital content creation and sharing are ubiquitous, the need for an intuitive and secure blogging platform is paramount. This web application aims to fulfill this need by developing a platform that combines ease of use, robust functionality, and aesthetic appeal.

The primary objectives of the Blogster web application are: implementing a secure and reliable system for user registration, login, and authentication to protect user data and ensure authorized access; enabling users to create, edit, and delete their own blog posts, ensuring that content management is straightforward and user-friendly; facilitating easy content discovery through tag-based categorization, allowing users to browse posts by tags and improve the overall navigability of the platform; providing a responsive search bar to help users quickly find posts based on keywords, enhancing the user experience by making content easily accessible; and ensuring that the platform is visually appealing and responsive across various devices, offering a consistent user experience on desktops, tablets, and mobile devices.

The web application utilizes a variety of technologies and tools to achieve its goals. For the frontend, HTML, CSS, and JavaScript are used to build a responsive and user-friendly interface. The design focuses on simplicity and accessibility to enhance user engagement. On the backend, PHP is employed for server-side scripting to handle user authentication, post creation, and data processing. MySQL is used as the database management system to store and manage user and post data. Visual Studio Code (VSCode) is the chosen integrated development environment (IDE) for coding, offering robust features and extensions to streamline the development process.

The Blogster web application is structured to provide a seamless and efficient blogging experience. Users can register and log in to the platform, create and manage their posts, and explore content through tags and search functionality. The application ensures that only authenticated users can modify their own content, maintaining data integrity and security. The platform's design prioritizes usability and visual appeal, making it an attractive destination for both content creators and readers. This report delves into the detailed aspects of Blogster, covering its design, implementation, and functionalities, along with the challenges encountered and the solutions implemented during the development process.

2. PROJECT DESCRIPTION

The Blogster web application is a comprehensive platform designed to provide users with a seamless and engaging experience for creating, managing, and discovering blog content. The development process focused on delivering a user-friendly interface, robust backend functionality, and efficient data management. This section provides a brief overview of the project's key features, functionality, purpose, and scope.

2.1 About my project

Purpose and Scope

The primary purpose of the Blogster web application is to provide a modern, user-friendly platform for sharing and managing blog content. By focusing on ease of use, security, and discoverability, Blogster aims to cater to both content creators and readers.

Features and Functionality

- **User Authentication and Authorization:** Blogster ensures secure user interactions through a reliable registration and login system. Users are authenticated against stored credentials, safeguarding access to personalized content and functionalities.
- **Post Creation and Management:** Users can create, edit, and delete their own blog posts. This feature is designed to be intuitive, allowing users to focus on content creation without being bogged down by technical complexities.
- **Tag-Based Categorization:** Posts can be tagged with relevant keywords, facilitating easy categorization and navigation. Users can browse posts by tags, enhancing the discoverability of content.
- **Responsive Search Bar:** A dynamic search bar enables users to quickly find posts based on keywords. The search functionality provides real-time suggestions and ensures that users can easily locate desired content.
- **User-Specific Content Editing:** Only the original creator of a post can edit or delete it, ensuring content integrity and preventing unauthorized modifications.
- **Responsive Design:** The platform is designed to be fully responsive, offering a consistent and visually appealing experience across various devices, including desktops, tablets, and mobile phones.

3. PROBLEM DESCRIPTION

In the digital age, managing and organizing blog content efficiently poses significant challenges for both individual bloggers and content management teams. Traditional blogging methods, such as manual updates and simple text editors, often fall short in handling the growing complexity and volume of content. This inadequacy is particularly evident in several key areas:

- **Content Organization and Navigation:** Users often struggle with disorganized content, leading to difficulties in locating specific posts, categories, or tags. Without a structured system, navigating through a blog can become cumbersome, detracting from the user experience.
- **User Authentication and Management: Many** blogging platforms lack robust user authentication mechanisms, which can compromise security and user management. Ensuring that only authorized users can access or edit posts is essential for maintaining content integrity and user privacy.
- **Search and Discovery:** Traditional blogs may lack advanced search functionality, making it challenging for users to find relevant posts quickly. Effective search capabilities are crucial for enhancing user engagement and ensuring that content is easily accessible.
- **Post Management:** Bloggers often face difficulties with editing, updating, or deleting posts efficiently. A lack of intuitive tools for managing content can lead to inefficiencies and errors in content handling.
- **Design and Usability:** Many blogging platforms fail to provide a user-friendly interface, resulting in a less engaging experience for both content creators and readers. Aesthetic and functional design elements are crucial for maintaining user interest and encouraging regular interaction.
- **Responsiveness and Accessibility:** Inadequate support for various devices and screen sizes can limit the reach and effectiveness of a blog. Ensuring that the blog is responsive and accessible across different platforms is vital for maximizing its audience.

The "Blogster" web application seeks to address these challenges by offering a comprehensive solution for efficient blog management. "Blogster" aims to enhance the overall blogging experience and address the deficiencies of traditional blogging methods.

4. TOOL DESCRIPTION

Hardware and Software Tools

To develop and deploy the recipe management web application, the following hardware and software tools were utilized:

Hardware Specifications

- **Laptop Model**: Dell G15 5520
- **Graphics Card**: NVIDIA GeForce RTX 3050, 4GB
- **Storage**: 500GB SSD
- **RAM**: 16GB
- **Processor**: Intel Core i5 12500H

The Dell G15 5520 laptop with its high-performance specifications provided an excellent environment for developing and testing the web application. The NVIDIA GeForce RTX 3050 graphics card ensured smooth rendering of graphics and multimedia content, enhancing the development experience, especially when dealing with high-resolution recipe images and user interface design. The 500GB SSD facilitated fast data read/write operations, significantly reducing load times for development tools and ensuring rapid access to project files. With 16GB of RAM, the laptop efficiently handled multiple development tools running concurrently, supporting a seamless multitasking environment. The Intel i5 12500H processor, known for its powerful performance and energy efficiency, enabled quick compilation and execution of code, speeding up the development cycle.

Software Tools

- **Visual Studio Code**: An integrated development environment (IDE) used for writing and debugging code. Its extensions and integrated terminal enhanced the coding experience.
- **XAMPP**: A free and open-source cross-platform web server solution stack package developed by Apache Friends. It provided the necessary Apache, MySQL, PHP, and Perl support for local development and testing.
- **phpMyAdmin**: A free software tool written in PHP, intended to handle the administration of MySQL over the web. phpMyAdmin was used for database management, allowing for easy handling of the MySQL database used in the application.
- **Google Chrome**: The primary web browser used for testing and debugging the web application. Developer tools in Chrome facilitated real-time inspection and modification of the front-end code.

The combination of powerful hardware and a robust set of development tools provided a conducive environment for the efficient development, testing, and deployment of the recipe management web application.

5. OPERATIONS

The "Blogster" web application provides a comprehensive set of functionalities designed to streamline and enhance the blogging experience for users. Below are the key operations that can be performed using this website:

a) **User Registration and Authentication**:
  - **Sign Up**: New users can create an account by providing necessary details and setting up their credentials.
  - **Login**: Registered users can securely log in to their accounts using their username and password.
  - **Password Management**: Users can update their passwords and recover forgotten passwords through a secure process.

b) **Post Creation and Management**:
  - **Create Posts**: Users can compose and publish new blog posts, including content such as text, images, and other media.
  - **Edit Posts**: Authors can edit their previously published posts to update or correct information.
  - **Delete Posts**: Users have the ability to delete posts that are no longer relevant or required.

c) **Content Organization**:
  - **Tagging and Categorization**: Posts can be tagged and categorized for better organization and easier navigation.
  - **Groups**: Users can create and join groups based on shared interests, with posts organized within these groups.

d) **Search and Discovery**:
  - **Search Bar**: A dynamic search bar allows users to quickly find posts by entering keywords, tags, or categories.
  - **Dropdown Suggestions**: The search bar provides real-time suggestions based on user input to enhance search efficiency.

e) **User Interaction**:
  - **View Posts**: Users can view detailed information about each post, including its content, tags, and associated comments.
  - **Commenting**: Users can engage with posts by leaving comments, fostering discussion and interaction.
  - **Like and Share**: Users can like posts and share them on various social media platforms.

f) **Personalized Experience**:
  - **My Posts**: Users can view a personalized list of their own posts, making it easy to manage their content.
  - **Profile Management**: Users can update their profile information and manage their account settings.

g) **Administrative Controls**:
  - **User Management**: Admin users have the ability to manage user accounts, including adding, removing, or modifying user roles.
  - **Content Moderation**: Admins can review and moderate posts and comments to ensure compliance with community guidelines.

h) **Responsive Design**:

- **Mobile and Desktop Access**: The website is designed to be responsive, ensuring a seamless experience across various devices, including smartphones, tablets, and desktops.

The "Blogster" web application aims to provide a robust and user-friendly platform for blogging, ensuring that users can easily create, manage, and discover content while engaging with the community in a meaningful way.

6. APPROACH / MODULE DESCRIPTION / FUNCTIONALITIES

The Blogster web application is built using several key modules, each playing a crucial role in the overall functionality and user experience of the platform. This section provides an overview of the main modules and their specific subsections, detailing their roles and implementation within the application.

6.1. User Authentication Module

The User Authentication module is responsible for managing user registration, login, and access control. It ensures that only authorized users can access certain features and content.

Registration and Login

- **Registration**: Users can create new accounts by providing a username and password. The registration form validates input data and stores encrypted passwords in the database for security.
- **Login**: Existing users can log in using their credentials. The login process verifies the provided credentials against stored data and establishes a user session upon successful authentication.

Session Management

- **Session Handling**: The module manages user sessions, ensuring that user data is securely stored and accessed throughout the session.
- **Logout**: Users can log out, terminating the session and ensuring that no sensitive data is left accessible.

6.2. Post Management Module

The Post Management module enables users to create, edit, delete, and view blog posts. It provides a user-friendly interface for content creation and management.

Post Creation

- **Create Post Form**: Users can create new blog posts by filling out a form that includes fields for the title, content, and tags. The form is designed to be intuitive, with validation to ensure that all required fields are filled out correctly.
- **Tagging System**: Users can add tags to their posts, facilitating easy categorization and searchability.

Post Editing and Deletion

- **Edit Post**: Users can edit their own posts, with changes being saved to the database. The edit form pre-populates with the current post data for easy modification.
- **Delete Post**: Users can delete their own posts, removing them from the database and ensuring that the content is no longer accessible.

Post Display

- **Post Listing**: Posts are displayed in a list format, with each post showing the title, author, and creation date. Users can click on a post title to view the full content.
- **Single Post View**: Individual posts can be viewed on a separate page, displaying the full content along with related tags.

6.3. Search and Tag-Based Navigation Module

This module enhances content discoverability by allowing users to search for posts and navigate through tags.

Search Functionality

- **Search Bar**: A search bar is available on the homepage, enabling users to search for posts by keywords. The search bar provides real-time suggestions and displays matching results in a dropdown menu.
- **Search Results**: Users can view a list of posts matching their search query, with links to view the full content of each post.

Tag-Based Navigation

- **Tag Links**: Tags are displayed with each post, and users can click on a tag to view all posts associated with that tag.
- **Tag Pages**: Dedicated pages for each tag display a list of posts that share the same tag, allowing users to explore related content easily.

6.4. Database Management Module

The Database Management module handles the storage and retrieval of data, ensuring efficient and secure access to user and post information.

Database Structure

- **User Table**: Stores user information, including usernames and encrypted passwords. The table is designed to ensure data integrity and security.
- **Post Table**: Stores post information, including titles, content, tags, and metadata such as the author and creation date.

Database Operations

- **CRUD Operations**: The module supports Create, Read, Update, and Delete operations for both users and posts. These operations are optimized to ensure quick access and modifications.
- **Data Validation**: Ensures that data entered into the database is valid and free of errors, preventing issues such as SQL injection attacks.

6.5. Frontend Interface Module

The Frontend Interface module focuses on the visual and interactive aspects of the web application, providing a seamless user experience.

Responsive Design

- **Layout**: The layout is designed to be fully responsive, adapting to different screen sizes and devices. This ensures that users have a consistent experience whether they are using a desktop, tablet, or mobile phone.
- **Styling**: CSS is used to style the application, with attention to detail in colors, fonts, and spacing to create an aesthetically pleasing interface.

User Interaction

- **Navigation Bar**: A navigation bar provides easy access to different sections of the application, including Home, Create Post, Groups, About Us, and Contact.
- **Interactive Elements**: Buttons, links, and forms are designed to be intuitive and responsive, enhancing the overall user experience.

By integrating these modules, Blogster offers a comprehensive and user-friendly platform for blogging. Each module plays a vital role in ensuring that the application is secure, efficient, and engaging for users.

7. IMPLEMENTATION/CODING

Home Page

```
<?php
session_start();

// Database connection
$servername = "localhost";
$dbusername = "root";
$dbpassword = "";
$dbname = "blog_db";

$conn = new mysqli($servername, $dbusername, $dbpassword, $dbname);

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

// Retrieve the latest posts
$sql_latest = "SELECT id, title, author FROM posts ORDER BY created_at DESC LIMIT 5";
$result_latest = $conn->query($sql_latest);

// Retrieve posts created by the logged-in user
$user = isset($_SESSION['username']) ? $_SESSION['username'] : '';
$sql_user_posts = "SELECT id, title, author FROM posts WHERE author = ?";
$stmt_user_posts = $conn->prepare($sql_user_posts);
$stmt_user_posts->bind_param("s", $user);
$stmt_user_posts->execute();
$result_user_posts = $stmt_user_posts->get_result();

// Merge results
$posts = [];
while ($row = $result_latest->fetch_assoc()) {
    $posts[$row['id']] = $row;
```

```
}
while ($row = $result_user_posts->fetch_assoc()) {
    if (!isset($posts[$row['id']])) {
        $posts[$row['id']] = $row;
    }
}
$posts = array_values($posts); // Re-index the array

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <style>
        /* Base Styles */
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }

        /* Header Styles */
        header {
            background: #007bff; /* Blue color */
            color: white;
            padding: 20px;
            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
            position: relative; /* Ensure the header is positioned relative for absolute positioning inside it */
            display: flex;
            flex-direction: column;
            align-items: center;
        }

        header h1 {
            margin: 0;
            font-size: 2.5rem;
            text-align: center;
        }

        nav ul {
            list-style-type: none;
            padding: 0;
            margin: 10px 0;
            text-align: center;
        }

        nav ul li {
            display: inline;
            margin: 0 15px;
        }

        nav ul li a {
            color: white;
            text-decoration: none;
            font-size: 1.1rem;
        }

        nav ul li a:hover {
            text-decoration: none; /* Removed underline from links on hover */
        }

        /* Search Bar Styles */
        .search-bar-container {
            width: 100%;
            display: flex;
            justify-content: center; /* Center horizontally */
            margin-top: 10px; /* Space between navigation and search bar */
            position: relative;
        }

        .search-bar {
            padding: 10px;
            border-radius: 5px;
            border: none;
            font-size: 1rem;
            width: 300px;
        }

        /* Dropdown Menu Styles */
        .dropdown {
            position: absolute;
            top: calc(100% + 10px); /* Positioned just below the search bar with a gap */
            left: 50%; /* Center horizontally */
            transform: translateX(-50%); /* Center horizontally */
            width: 300px; /* Matches the width of the search bar */
            background-color: white;
            border: 1px solid #ddd;
            border-radius: 5px;
            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
            display: none;
            z-index: 1000;
        }

        .dropdown-item {
            padding: 10px;
            text-decoration: none;
            display: block;
            color: #333;
        }

        .dropdown-item:hover {
            background-color: #f4f4f4;
        }

        .dropdown.active {
            display: block;
        }

        /* Logout Button Styles */
        .logout-btn {
            background-color: #ff4500; /* OrangeRed color */
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 5px;
            font-size: 1rem;
            cursor: pointer;
            text-decoration: none;
            transition: background-color 0.3s ease;
            position: absolute; /* Position logout button absolutely */
            bottom: 20px; /* Position from the bottom */
            right: 20px; /* Position from the right */
        }

        .logout-btn:hover {
            background-color: #ff6347; /* Tomato color */
        }

        /* My Posts Button Styles */
        .my-posts-btn {
            position: absolute;
            top: 20px;
            left: 20px;
            background-color: #4CAF50; /* Green color */
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 5px;
            font-size: 1rem;
            cursor: pointer;
            text-decoration: none;
            transition: background-color 0.3s ease;
        }

        .my-posts-btn:hover {
            background-color: #45a049; /* Darker green on hover */
        }

        /* Hero Section Styles */
        .hero {
            width: 100%;
            height: 400px; /* Adjust height as needed */
            background-image: url('imghome.jpg'); /* Update with your image path */
            background-size: cover;
            background-position: center;
            display: flex;
            align-items: center;
            justify-content: center;
            text-align: center;
            border-radius: 20px; /* Curved border */
            overflow: hidden; /* Ensures content is clipped within the rounded corners */
        }

        .hero .btn {
            background-color: rgba(0, 0, 0, 0.7); /* Black with 70% opacity */
            color: white;
            padding: 15px 30px;
            border: none;
            border-radius: 5px;
            font-size: 1.2rem;
            cursor: pointer;
            text-decoration: none;
            transition: background-color 0.3s ease;
        }

        .hero .btn:hover {
            background-color: rgba(0, 0, 0, 0.9); /* Black with 90% opacity on hover */
        }

        /* Main Content Styles */
        main {
            padding: 20px;
            background-color: #000000; /* Background color */
        }

        section#latest-posts {
            padding: 40px 20px;
            background: white;
            border-radius: 8px;
            box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
            margin-top: 20px;
        }

        section#latest-posts h2 {
            margin-top: 0;
            font-size: 1.8rem;
            color: #a52a2a; /* Reddish-brown color */
        }
```

```
/* Floating Boxes for Posts */
#post-list {
   display: flex;
   flex-wrap: wrap;
   gap: 20px;
   padding: 20px;
}

#post-list .post {
   flex: 1 1 calc(25% - 20px);
   box-sizing: border-box;
   padding: 20px;
   background: #ffffff;
   border-radius: 8px;
   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
   transition: box-shadow 0.3s ease, transform 0.3s ease;
   text-align: center;
}

#post-list .post:hover {
   box-shadow: 0 6px 12px rgba(0, 0, 0, 0.2);
   transform: translateY(-5px);
}

#post-list .post h3 {
   margin: 0;
   font-size: 1.2rem;
   color: #a52a2a; /* Reddish-brown color */
}

#post-list .post h3 a {
   text-decoration: none; /* Remove underline */
   color: #a52a2a; /* Reddish-brown color for links */
}
```

```
        #post-list .post h3 a:hover {
            text-decoration: underline; /* Add underline on hover, if desired */
        }

        /* User Button and Dropdown Styles */
        .user-btn {
            position: absolute;
            top: 20px;
            right: 20px;
            width: 40px;
            height: 40px;
            background-color: #ff0000; /* Red color */
            border-radius: 50%;
            color: white;
            display: flex;
            align-items: center;
            justify-content: center;
            cursor: pointer;
            font-size: 1.2rem;
            text-decoration: none;
            text-align: center; /* Center text horizontally */
            line-height: 1; /* Adjust line height to vertically center */
}


        .user-btn:hover {
            background-color: #cc0000; /* Darker red on hover */
        }

        .user-dropdown {
            position: absolute;
            top: 70px; /* Adjust according to the button's position */
            right: 20px;
            background-color: white;
            border: 1px solid #ddd;
            border-radius: 5px;
            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
            display: none;
            z-index: 1000;
            width: 150px; /* Adjust width as needed */
        }

        .user-dropdown.active {
            display: block;
        }

        .user-dropdown a {
            display: block;
            padding: 10px;
            color: #333;
            text-decoration: none;
        }

        .user-dropdown a:hover {
            background-color: #f4f4f4;
        }
        .username {
            padding: 10px;
            font-weight: bold;
            color: #333;
            border-bottom: 1px solid #ddd;
}
    </style>
</head>
<body>
    <header>
        <h1>My Blog</h1>
        <nav>
            <ul>
                <li><a href="home.php">Home</a></li>
                <li><a href="create.php">Create Blog</a></li>
                <li><a href="categories.php">Categories</a></li>
                <li><a href="aboutus.html">About Us</a></li>
                <li><a href="contact.html">Contact</a></li>

            </ul>
        </nav>
        <div class="search-bar-container">
            <input type="text" id="search-bar" class="search-bar" placeholder="Search posts...">
            <div id="search-results" class="dropdown"></div>
        </div>
        <a href="myposts.php" class="my-posts-btn">My Posts</a>
        <a href="#" class="user-btn"><?php echo strtoupper(substr($user, 0, 1)); ?></a> <!-- User round button -->
        <div class="user-dropdown">
        <div class="username"><?php echo htmlspecialchars($user); ?></div>
            <a href="logout.php">Logout</a>
        </div>
    </header>

    <main>
        <section class="hero">
            <a href="create.php" class="btn">Create Blog</a>
        </section>

        <section id="latest-posts">
            <h2>Latest Posts</h2>
            <div id="post-list">
                <?php
                foreach ($posts as $post) {
                    echo '<div class="post">';
                    echo '<h3><a href="post.php?id=' . $post['id'] . '">' . htmlspecialchars($post['title']) . '</a></h3>';
                    echo '<p class="author">By ' . htmlspecialchars($post['author']) . '</p>';
                    echo '</div>';
                }
                ?>
            </div>
        </section>
    </main>

    <script>
        // JavaScript for handling dropdown visibility and search functionality
        document.getElementById('search-bar').addEventListener('focus', function() {
            document.getElementById('search-results').classList.add('active');
        });

        document.getElementById('search-bar').addEventListener('blur', function() {
            setTimeout(function() {
                document.getElementById('search-results').classList.remove('active');
            }, 200);
        });

        document.getElementById('search-bar').addEventListener('input', function() {
            let query = this.value;
            if (query.length > 2) {
                // Fetch search results via AJAX
                fetch('search.php?q=' + encodeURIComponent(query))
                    .then(response => response.json())
                    .then(data => {
                        let resultsDiv = document.getElementById('search-results');
                        resultsDiv.innerHTML = '';
                        data.forEach(item => {
                            let link = document.createElement('a');
                            link.href = 'post.php?id=' + item.id;
                            link.textContent = item.title;
                            link.className = 'dropdown-item';
                            resultsDiv.appendChild(link);
                        });
                    });
            } else {
                document.getElementById('search-results').innerHTML = '';
            }
        });

        // JavaScript for user dropdown menu
        document.querySelector('.user-btn').addEventListener('click', function() {
            document.querySelector('.user-dropdown').classList.toggle('active');
        });
    </script>
</body>
</html>

<?php
$stmt_user_posts->close();
$conn->close();
?>
```

Create Blog

```
<?php
session_start();

// Debugging session info
echo "<pre>";
print_r($_SESSION);
echo "</pre>";

// Redirect to login page if not logged in
if (!isset($_SESSION['username'])) {
    header("Location: login.html");
    exit();
}

$servername = "localhost";
$dbusername = "root";
$dbpassword = "";
$dbname = "blog_db";

// Create a new connection
$conn = new mysqli($servername, $dbusername, $dbpassword, $dbname);

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

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve and sanitize form data
    $title = $conn->real_escape_string($_POST['title']);
    $content = $conn->real_escape_string($_POST['content']);
    $tags = $conn->real_escape_string($_POST['tags']);
    $author = $_SESSION['username']; // Get the logged-in username

    // Handle file upload
    $image_path = null;
    if (isset($_FILES['image']) && $_FILES['image']['error'] == UPLOAD_ERR_OK) {
        $tmp_name = $_FILES['image']['tmp_name'];
        $image_name = basename($_FILES['image']['name']);
        $upload_dir = 'uploads/'; // Directory where you want to save uploaded images

        // Check if upload directory exists and is writable
        if (!is_dir($upload_dir)) {
            mkdir($upload_dir, 0755, true);
        }

        // Move the uploaded file to the target directory
        $image_path = $upload_dir . $image_name;
        if (move_uploaded_file($tmp_name, $image_path)) {
            $image_path = $conn->real_escape_string($image_path); // Escape the file path
        } else {
            echo "Failed to upload image.";
            $image_path = null;
        }
    }

    // Prepare SQL query
    $sql = "INSERT INTO posts (title, author, content, tags, image, created_at) VALUES (?, ?, ?, ?, ?, NOW())";
    $stmt = $conn->prepare($sql);
    if ($stmt === false) {
        die("Prepare failed: " . $conn->error);
    }

    // Bind parameters
    $stmt->bind_param("sssss", $title, $author, $content, $tags, $image_path);

    // Execute the query
    if ($stmt->execute()) {
        // Redirect to myposts.php after successful post creation
        header("Location: myposts.php");
        exit();
    } else {
        echo "Error: " . $stmt->error;
    }

    // Close statement
    $stmt->close();
}

// Close connection
$conn->close();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create Post</title>
    <style>
        /* Base Styles */
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            margin: 0;
            padding: 0;
            background-image: url('img111.png'); /* Background image */
            background-size: cover;
            background-position: center;
            background-attachment: fixed;
            background-repeat: no-repeat;
            color: #333;
        }

        .home-button {
            background-color: #007bff;
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 5px;
            font-size: 1rem;
            cursor: pointer;
            text-decoration: none;
            display: inline-block;
            margin: 20px;
            transition: background-color 0.3s ease;
        }

        .home-button:hover {
            background-color: #0056b3;
        }

        main {
            padding: 40px 20px;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: calc(100vh - 80px); /* Full height minus header height */
        }

        section {
            padding: 40px;
            background: rgba(255, 255, 255, 0.9); /* Semi-transparent white background */
            border-radius: 12px;
            box-shadow: 0 6px 12px rgba(0, 0, 0, 0.2);
            max-width: 700px;
            width: 100%;
        }

        section h2 {
            margin-top: 0;
            color: #333;
            font-size: 2rem;
            border-bottom: 2px solid #8B4513; /* Reddish brown underline */
            padding-bottom: 10px;
            margin-bottom: 20px;
```

```
}

form {
   display: flex;
   flex-direction: column;
}

label {
   font-weight: bold;
   margin-bottom: 8px;
   color: #333;
   font-size: 1rem;
}

input[type="text"],
textarea,
input[type="file"] {
   width: 100%;
   padding: 12px;
   margin-bottom: 20px; /* Increased space between elements */
   border: 1px solid #ccc;
   border-radius: 8px;
   background-color: #f9f9f9;
   color: #333;
   box-sizing: border-box;
}

textarea {
   resize: vertical; /* Allow vertical resizing */
}

button {
   background-color: #8B4513; /* Reddish brown color */
   color: #ffffff;
```

```
            padding: 14px;
            border: none;
            border-radius: 8px;
            cursor: pointer;
            font-size: 1.1rem;
            transition: background-color 0.3s ease, transform 0.2s ease;
            margin-top: 10px;
        }

        button:hover {
            background-color: #6f3f1c; /* Darker reddish-brown color for hover effect */
            transform: translateY(-2px); /* Subtle lift effect */
        }
    </style>
</head>
<body>
    <main>
        <section id="create-post">
            <a href="home.php" class="home-button">Home</a> <!-- Home button to return to home.php -->
            <h2>Create a New Post</h2>
            <form id="post-form" action="create.php" method="post" enctype="multipart/form-data">
                <label for="title">Title:</label>
                <input type="text" id="title" name="title" required>

                <label for="content">Content:</label>
                <textarea id="content" name="content" rows="6" required></textarea>

                <label for="tags">Tags (comma-separated):</label>
                <input type="text" id="tags" name="tags" required>

                <label for="image">Image:</label>
                <input type="file" id="image" name="image">
        <!-- Display the author's username -->
        <label for="author">Author:</label>
        <input type="text" id="author" name="author" value="<?php echo htmlspecialchars($_SESSION['username']); ?>" readonly>

        <button type="submit">Post</button>
      </form>
    </section>
  </main>
</body>
</html>
```

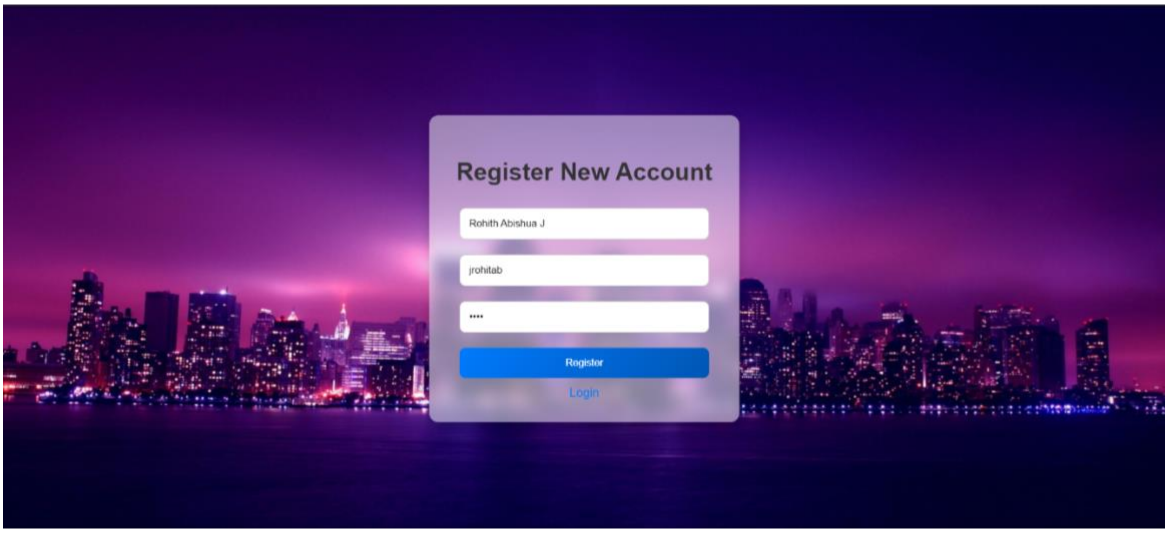
Submit Post

```
<?php
session_start(); // Start the session

// Check if user is logged in
if (!isset($_SESSION['username'])) {
  echo "User not logged in.";
  header("Location: login.html");
  exit();
} else {
  echo "User logged in as: " . htmlspecialchars($_SESSION['username']);
}

// Database connection
$servername = "localhost";
$dbusername = "root";
$dbpassword = "";
$dbname = "blog_db";

// Create a new connection
$conn = new mysqli($servername, $dbusername, $dbpassword, $dbname);
```

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

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve and sanitize form data
    $title = htmlspecialchars($conn->real_escape_string($_POST['title']));
    $content = htmlspecialchars($conn->real_escape_string($_POST['content']));
    $tags = htmlspecialchars($conn->real_escape_string($_POST['tags'])); // This should be a comma-separated string
    $author = htmlspecialchars($_SESSION['username']); // Get the logged-in username

    // Handle file upload
    $image_path = null;
    if (isset($_FILES['image']) && $_FILES['image']['error'] == UPLOAD_ERR_OK) {
        $tmp_name = $_FILES['image']['tmp_name'];
        $image_name = basename($_FILES['image']['name']);
        $upload_dir = 'uploads/'; // Directory where you want to save uploaded images
        $image_path = $upload_dir . $image_name;

        // Check if upload directory exists and is writable
        if (!is_dir($upload_dir)) {
            mkdir($upload_dir, 0755, true);
        }

        // Move the uploaded file to the target directory
        if (move_uploaded_file($tmp_name, $image_path)) {
            $image_path = $conn->real_escape_string($image_path); // Escape the file path
        } else {
            echo "Failed to upload image.";
            $image_path = null;
        }
    }

    // Prepare SQL query
    $sql = "INSERT INTO posts (title, author, content, tags, image, created_at) VALUES (?, ?,
?, ?, ?, NOW())";
    $stmt = $conn->prepare($sql);

    if ($stmt === false) {
        die("Prepare failed: " . $conn->error);
    }

    // Bind parameters
    $stmt->bind_param("sssss", $title, $author, $content, $tags, $image_path);

    // Execute the query
    if ($stmt->execute()) {
        echo "Post created successfully. <a href='home.php'>Back to home</a>";
    } else {
        echo "Error: " . $stmt->error;
    }

    // Close statement
    $stmt->close();
}

// Close connection
$conn->close();
?>
```

8. RESULT

a) Resister page



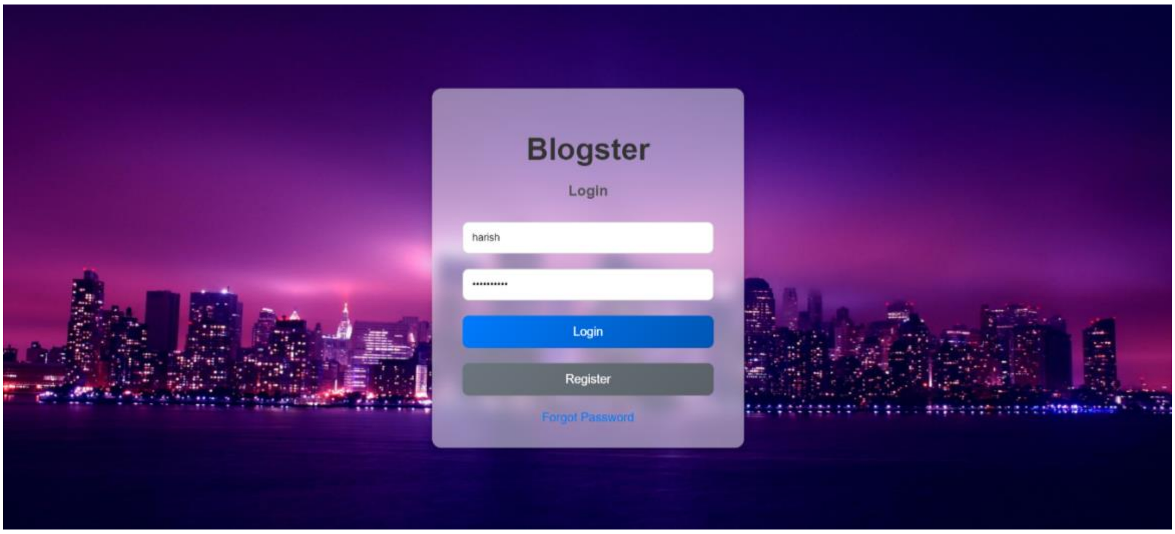
Register New Account

Rohith Abishua J

jrohitab

••••

b) Login page



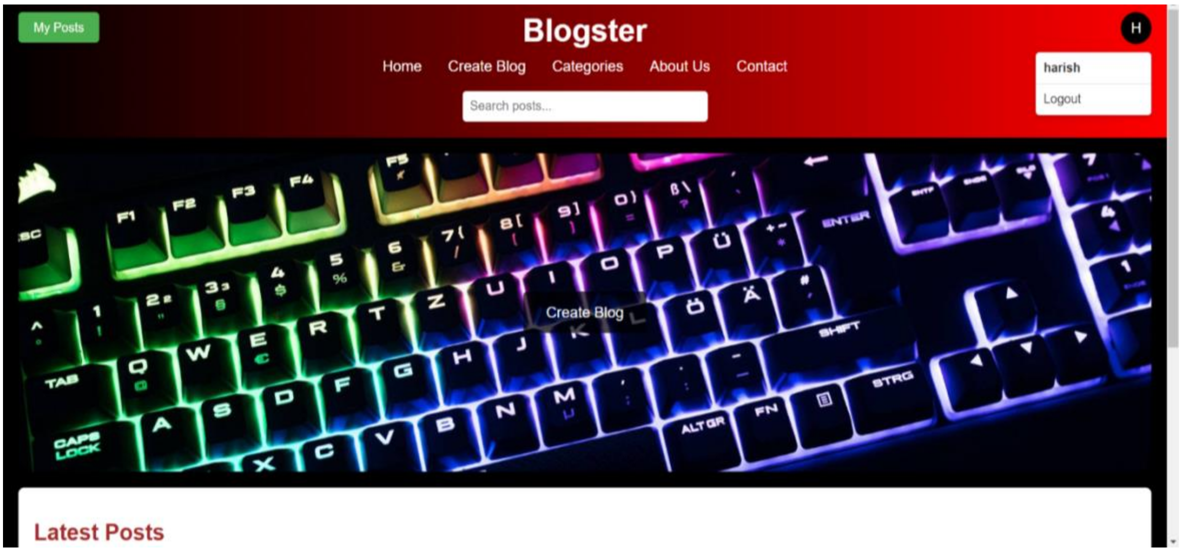
Blogster

Login

harish

••••••••••

c) Home page



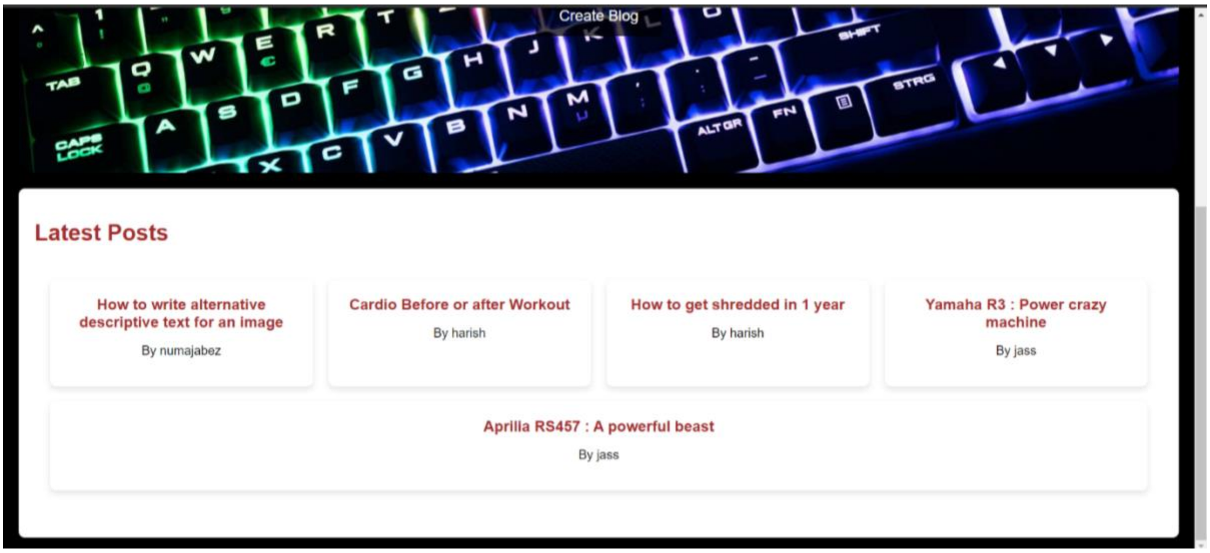
Blogster

Search posts...

harish

Latest Posts

d) Home page-latest post



Create Blog

Latest Posts

How to write alternative descriptive text for an image
By numajabez

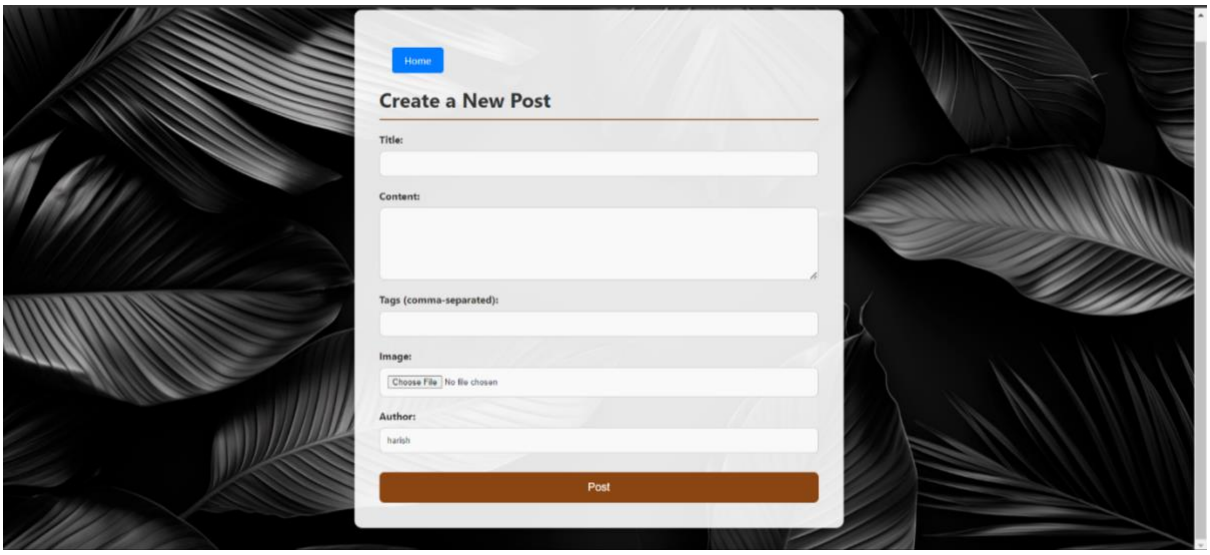
Cardio Before or after Workout
By harish

How to get shredded in 1 year
By harish

Yamaha R3 : Power crazy machine
By jass

Aprilia RS457 : A powerful beast
By jass

e) Create blog page



Create a New Post

Title:

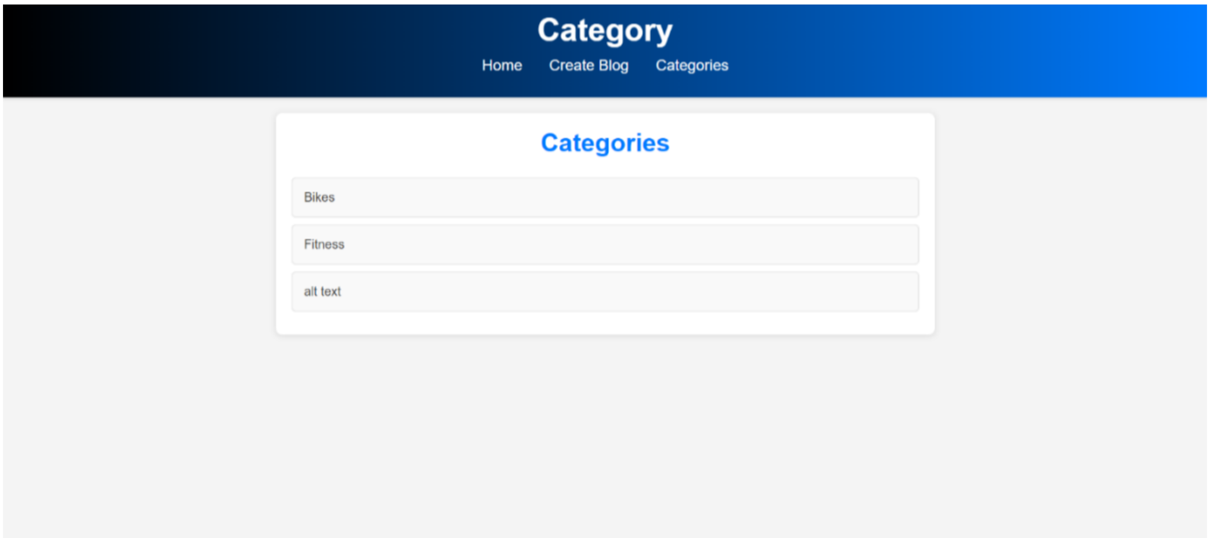
Content:

Tags (comma-separated):

Image: Choose File No file chosen

Author: harish

f) Category page



Category

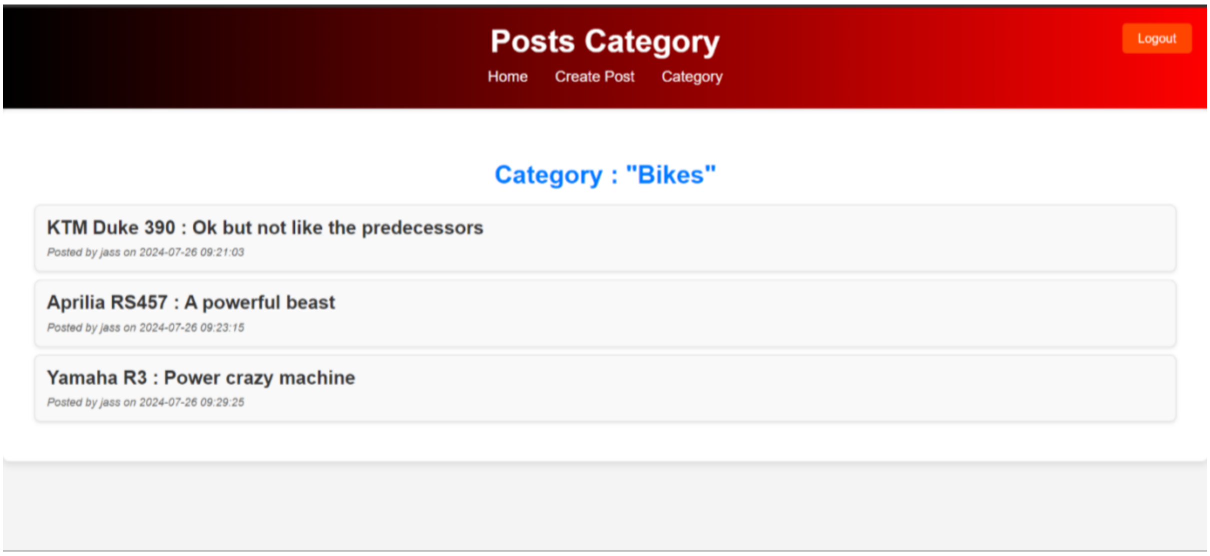
Categories

Bikes

Fitness

alt text

g) Bike Category posts



Posts Category

Category : "Bikes"

KTM Duke 390 : Ok but not like the predecessors

Posted by jass on 2024-07-26 09:21:03

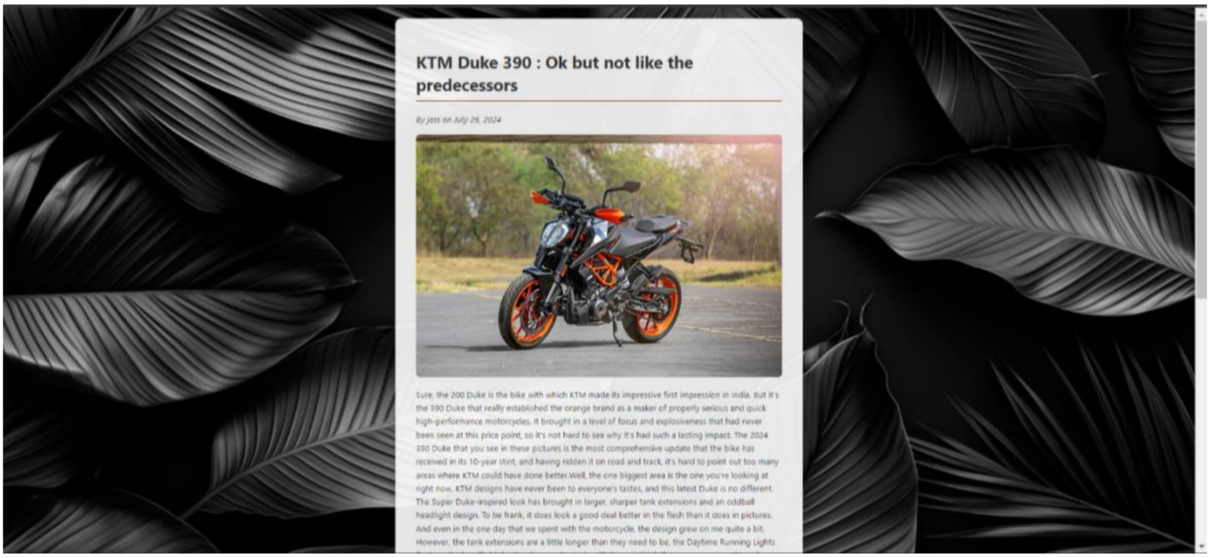
Aprilia RS457 : A powerful beast

Posted by jass on 2024-07-26 09:23:15

Yamaha R3 : Power crazy machine

Posted by jass on 2024-07-26 09:29:25

h) Post view page

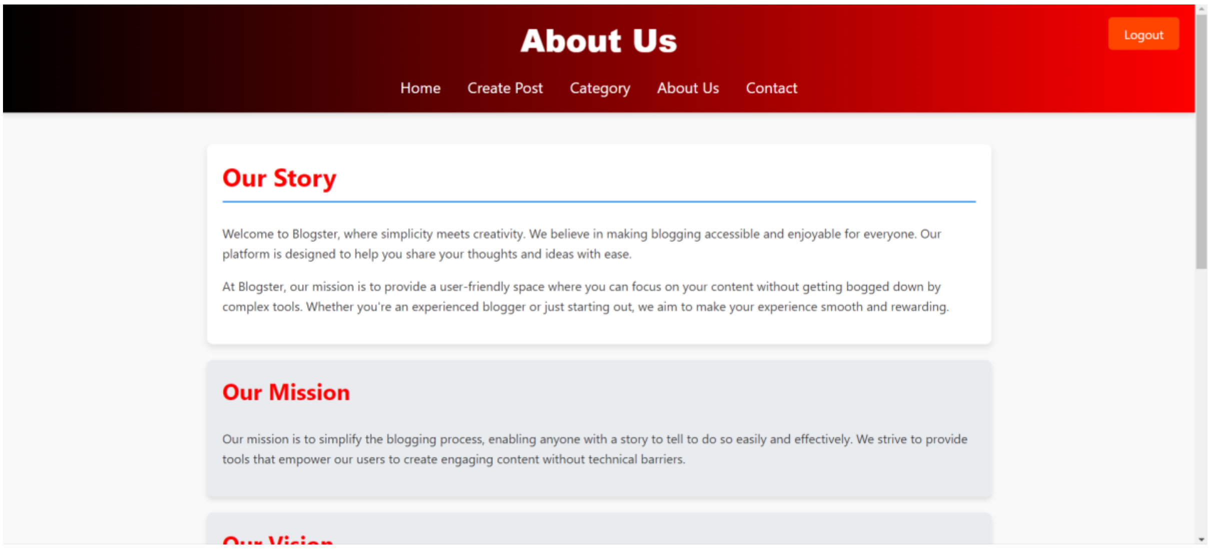


KTM Duke 390 : Ok but not like the predecessors

By jass on July 26, 2024

Sure, the 200 Duke is the bike with which KTM made its impressive first impression in india. But it's the 390 Duke that really established the orange brand as a maker of properly serious and quick high-performance motorcycles. It brought in a level of focus and explosiveness that had never been seen at this price point, so it's not hard to see why it's had such a lasting impact. The 2024 390 Duke that you see in these pictures is the most comprehensive update that the bike has received in its 10-year stint, and having ridden it on road and track, it's hard to point out too many areas where KTM could have done better.Well, the one biggest area is the one you're looking at right now. KTM designs have never been to everyone's tastes, and this latest Duke is no different. The Super Duke-inspired look has brought in larger, sharper tank extensions and an oddball headlight design. To be frank, it does look a good deal better in the flesh than it does in pictures. And even in the one day that we spent with the motorcycle, the design grew on me quite a bit. However, the tank extensions are a little longer than they need to be, the Daytime Running Lights

i) About Us page



About Us

Our Story

Welcome to Blogster, where simplicity meets creativity. We believe in making blogging accessible and enjoyable for everyone. Our platform is designed to help you share your thoughts and ideas with ease.

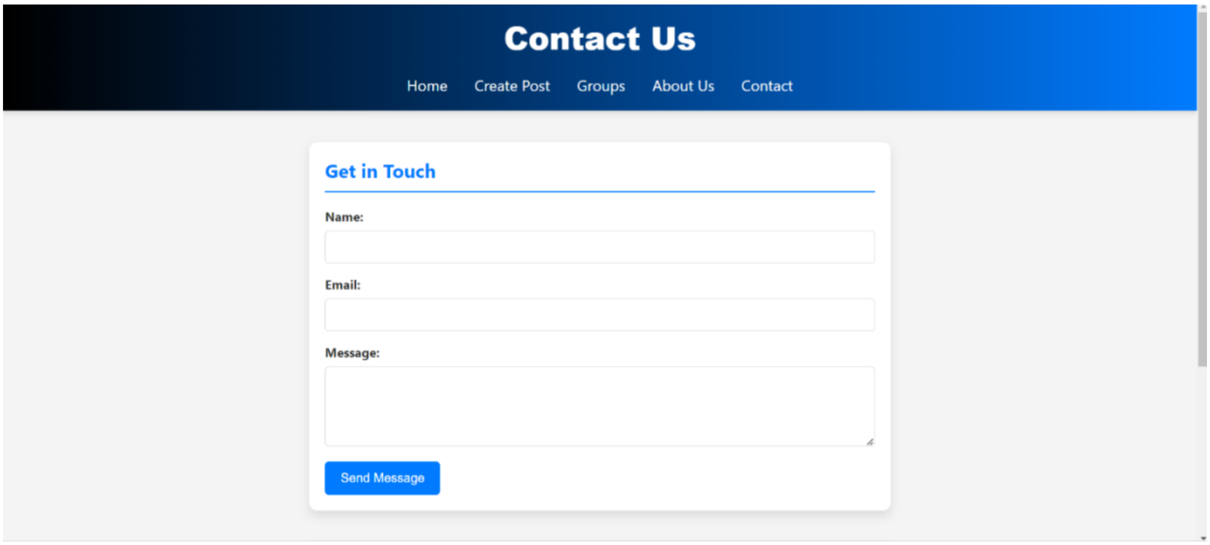
At Blogster, our mission is to provide a user-friendly space where you can focus on your content without getting bogged down by complex tools. Whether you're an experienced blogger or just starting out, we aim to make your experience smooth and rewarding.

Our Mission

Our mission is to simplify the blogging process, enabling anyone with a story to tell to do so easily and effectively. We strive to provide tools that empower our users to create engaging content without technical barriers.

Our Vision

j) Contact Us page



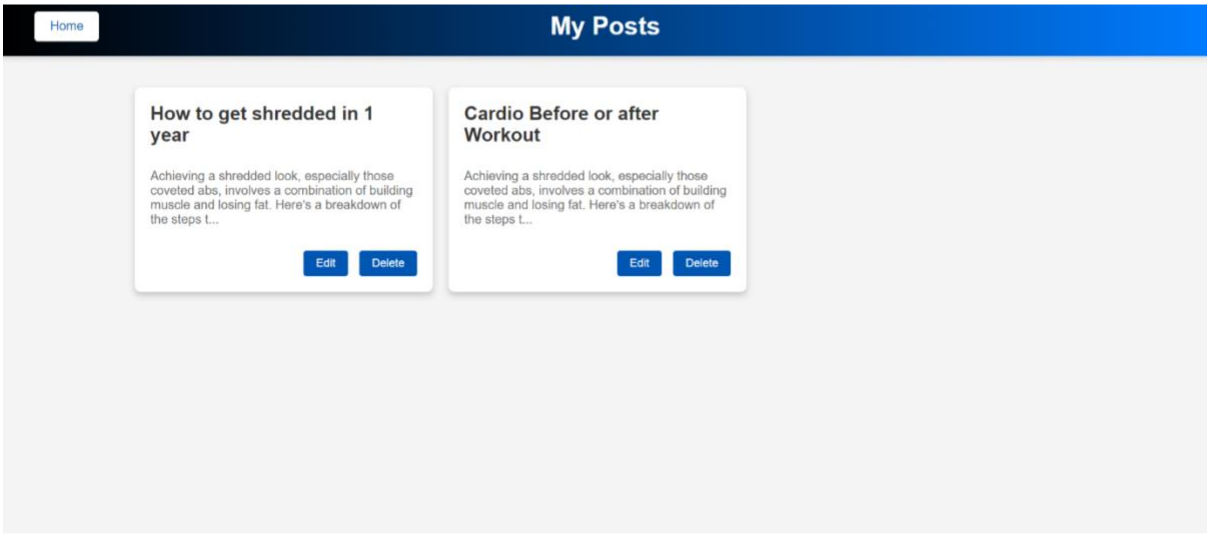
Get in Touch

Name:

Email:

Message:

k) My Posts page



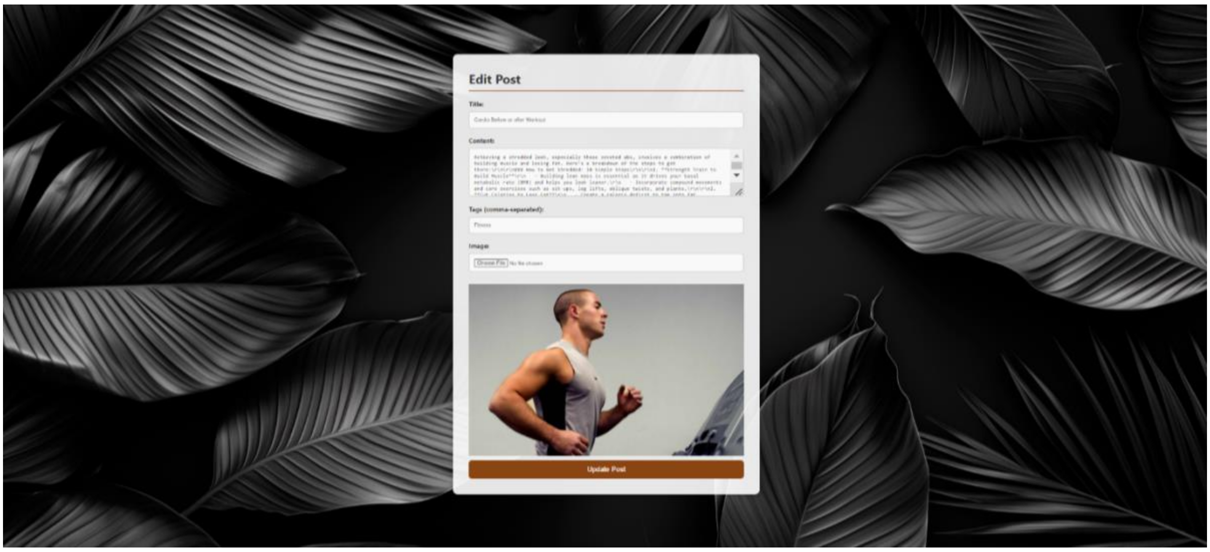
How to get shredded in 1 year

Achieving a shredded look, especially those coveted abs, involves a combination of building muscle and losing fat. Here's a breakdown of the steps t...

Cardio Before or after Workout

Achieving a shredded look, especially those coveted abs, involves a combination of building muscle and losing fat. Here's a breakdown of the steps t...

l) Edit Post page



Edit Post

Title:

Cardio Before or after Workout

Content:

Achieving a shredded look, especially those coveted abs, involves a combination of building muscle and losing fat. Here's a breakdown of the steps to get there:\r\n\r\n### How to Get Shredded: 10 Simple Steps\r\n\r\n1. **Strength Train to Build Muscle**\r\n   - Building lean mass is essential as it drives your basal metabolic rate (BMR) and helps you look leaner.\r\n   - Incorporate compound movements and core exercises such as sit-ups, leg lifts, oblique twists, and planks.\r\n\r\n2. **Cut Calories to Lose Fat**\r\n   - Create a caloric deficit to tap into fat

Tags (comma-separated):

Fitness

Image:

9. CONCLUSION

The Blogster web application has successfully achieved its primary objective of providing a user-friendly, secure, and efficient platform for blogging. Through the integration of various modules, including user authentication, post management, search functionality, and a responsive frontend interface, the application offers a seamless experience for both content creators and readers. The focus on security and data integrity ensures that user information is protected, while the intuitive design and navigation enhance user engagement. Overall, Blogster stands as a robust and reliable blogging platform, catering to the needs of its users effectively.

9.1. Future Enhancements

Blogster can be further enhanced through several key improvements to expand its capabilities and elevate the user experience. Enhancing user profiles with customization options, activity logs, and social media integration will personalize the user experience and foster a sense of community. Advanced search filters based on categories, tags, dates, and author names, along with search engine optimization (SEO) practices, will make content discovery more efficient and drive organic traffic to the platform. Adding social features like a commenting system and the ability to like and share posts will encourage interaction and broaden the platform's reach. Implementing two-factor authentication (2FA) and conducting regular security audits will enhance security, ensuring user data protection.

Performance optimization through caching mechanisms and scalability improvements will maintain the platform's responsiveness and reliability as it grows. Additionally, developing a dedicated mobile application with push notifications and offline access will provide a tailored experience for mobile users. These enhancements will not only improve functionality and user experience but also position Blogster as a leading platform in the blogging space, continuously adapting to meet the evolving needs of its users.

References

1. Herring, S. C., Scheidt, L. A., Bonus, S., & Wright, E. (2005). "Weblogs as a bridging genre." *Information, Technology & People, 18*(2), 142-171.
2. Nardi, B. A., Schiano, D. J., Gumbrecht, M., & Swartz, L. (2004). "Why we blog." *Communications of the ACM, 47*(12), 41-46.
3. Blood, R. (2002). *The weblog handbook: Practical advice on creating and maintaining your blog*. Perseus Publishing.
4. Rettberg, J. W. (2014). *Blogging*. Polity Press.
5. Rosenberg, S. (2009). *Say everything: How blogging began, what it's becoming, and why it matters*. Crown Publishing Group.
6. Schmidt, J. (2007). "Blogging practices: An analytical framework." *Journal of Computer-Mediated Communication, 12*(4), 1409-1427.
7. Marlow, C. (2004). "Audience, structure and authority in the weblog community." Presented at the International Communication Association Conference, New Orleans, LA.
8. Efimova, L., & de Moor, A. (2005). "Beyond personal webpublishing: An exploratory study of conversational blogging practices." *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*.
9. Trammell, K. D., & Keshelashvili, A. (2005). "Examining the new influencers: A self-presentation study of A-list blogs." *Journalism & Mass Communication Quarterly, 82*(4), 968-982.
10. boyd, d. m., & Ellison, N. B. (2007). "Social network sites: Definition, history, and scholarship." *Journal of Computer-Mediated Communication, 13*(1), 210-230.