MIT Art, Design and Technology University
MIT School of Computing, Pune

Department of Information Technology

Lab Manual

Practical - Web Programming

Class - S.Y. (SEM-IV), DA

Batch - DA-I

Name of the Student

Mr. Tilak Laddha

A.Y. 2024 – 2025 (SEM-IV)

File Index page given in the stationary

Web Programming

SEMESTER – IV

Course Code:	23IT2008	**Course Credits:**	02
Teaching Hours / Week (L:T:P):	0:0:4	**CA Marks:**	25
Total Number of Teaching Hours:		**END-SEM Marks:**	25

Course Pre-requisites:

Course Description:

This course provides a comprehensive introduction to web technology, designed to help students develop a strong foundation in building and managing websites and web applications. The curriculum covers key topics such as HTML, CSS, and JavaScript,PHP, MySQL, which are essential for creating interactive, well-designed web pages. Students will also explore the principles of responsive design, ensuring that web applications are optimized for different devices and screen sizes.

The course dives deeper into server-side technologies, including HTTP, web servers, and databases, allowing students to understand how websites function behind the scenes. Emphasis is placed on practical learning, and students will gain hands-on experience by working on projects that showcase their ability to design, develop, and deploy websites.

By the end of the course, students will be proficient in using modern web technologies to create web applications. They will understand how to handle client-server interactions, manage user data, and implement various web technologies to enhance the functionality of their applications.

Course Learning Objectives: This course will enable the students to:

1. Understand fundamental concepts of front-end web development.
2. Enable students to create basic web pages incorporating essential elements such as images, hyperlinks, lists, tables, and forms.
3. Teach students how to use CSS to manage fonts, lists, colors, text alignment, and background images for a cohesive and aesthetically pleasing web design.
4. Develop an understanding of JavaScript scopes to manage the visibility and lifetime of variables and functions effectively.
5. Equip students with the skills to implement and handle JavaScript events, enabling enhanced user interactions through event-driven programming.
6. Apply comprehensive knowledge of HTML, CSS, and JavaScript to develop a complete front-end application. Utilize project-based learning to showcase problem-solving skills and creativity in web development projects.
7. Configure server environments with Apache/TOMCAT.
8. Set up a PHP development environment and write basic PHP scripts.
9. Master PHP programming constructs for web development tasks.
10. Create and process HTML forms, and manage MySQL database operations.
11. Develop comprehensive back-end applications using PHP and MySQL.

Course Outcome: After taking this course, Students will be able to :

1. Apply knowledge of HTML to create the structure of the webpage and CSS to style and layout the elements, making the application visually appealing.
2. Apply comprehensive knowledge of HTML, CSS, and JavaScript to develop a complete front-end application and utilize project-based learning to showcase problem-solving skills and creativity in web development projects.
3. Set up and configure a server environment using tools like Apache or TOMCAT and set up a PHP development environment. Write & execute simple PHP scripts, understanding PHP syntax and basic features, create HTML forms to collect user data and integrate with PHP for processing.
4. Design and develop a back-end application using PHP and MySQL, implementing CRUD operations to manage data effectively.

UNIT – I	**Introduction to HTML and Cascading Style Sheet**	**09 Hours**

Module 1 - Markup Language (HTML): Introduction to HTML, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, HTML Forms

Module 2 - CSS: Need for CSS, introduction to CSS, basic syntax and structure, Levels of style sheets, Style specification formats, BOX Model, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, Background images

Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos:** **Use tools like Visual Studio Code (free).** **Videos:** **https://www.coursera.org/learn/html-css-javascript-for-web-developers**
	Self-study / Do it yourself /: **Practice creating basic HTML pages and enhancing them using CSS.**
	Experiential Learning Topics: **Design a simple webpage for coffee shop website**
	PBL - Project Based Learning: **Create a multi-page website (e.g., coffee shop website) using HTML and CSS.**

UNIT – II	**Front-End Development**	**09 Hours**

Module 3 - Overview of JavaScript, including JS in an HTML (Embedded, External), Basic JS syntax, basic interaction with HTML

Module 4 - Core features of JavaScript: Data types, Control Structures, Arrays, Functions and Scopes

Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos:**

	Use tools like Visual Studio Code (free). Videos: https://www.coursera.org/learn/javascript-basics	
	Self-study / Do it yourself /: Solve exercises on JavaScript syntax, control structures, and functions	
	Experiential Learning Topics: Build a web page with interactive elements (e.g., a simple calculator).	
	PBL - Project Based Learning: Develop an interactive webpage that uses JavaScript to validate form inputs or perform basic calculations.	
UNIT – III	Advanced Front-End Development	09 Hours
Module 5 - DOM: DOM levels, DOM Objects and their properties and methods, Manipulating DOM Module 6 - JavaScript Events: JavaScript Events, Types of JavaScript Events, Objects in JS, Event Handling		
Pedagogy	ICT Teaching / PowerPoint Presentation and Videos: https://www.coursera.org/learn/building-interactive-web-pages-using-javascript Use tools like Visual Studio Code (free).	
	Self-study / Do it yourself /: Practice exercises on DOM traversal and event handling.	
	Experiential Learning Topics: Add dynamic behavior to a webpage using DOM and events (e.g., a to-do list app).	
	PBL - Project Based Learning: Develop a web page with dynamic content (e.g., a task manager or interactive quiz) using DOM manipulation and event handling.	
UNIT – IV	Server Side Scripting	09 Hours
Module 7 - Set up and configure a server environment using tools like Apache or TOMCAT, set up a PHP development environment. Module 8 -Introduction to PHP: : Introduction to PHP, Server side scripting Vs Client side scripting, Basic		

Development Concepts (Mixing PHP with HTML), Creating, Writing & Running First PHP Script, PHP syntax, conditions & Loops, Functions, String manipulation, Arrays & Functions,

Module 9 - Form handling with HTML and PHP: Designing of Forms using HTML, Form Handling using GET and POST methods of Form

Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos:** **https://www.coursera.org/learn/web-applications-php** **Use tools like Visual Studio Code (free), XAMPP/WAMP for PHP server setup, and MySQL Workbench for database management**
	Self-study / Do it yourself /: **Practice exercises on form handling and server-side scripting with PHP.**
	Experiential Learning Topics: **Create a basic form for data submission and handle it using PHP (e.g., feedback form).**
	PBL - Project Based Learning: **Develop a small server-side application (e.g., a contact form with email validation and submission).**

UNIT – V	**Working with Databases and Web Application Development**	**09 Hours**

Module 10 - Working with databases using MySQL with PHP: MySQL database, create database, create table, primary key with AUTO_INCREMENT setting, Insert Data Into a Database Table, Select Data From a Database Table, Open or close a Connection to the MySQL Server.

Module 11 - Web Application Development (Project): Develop the web application to handle client-server interactions, manage user data, and implement various web technologies to enhance the functionality of their applications. Example: Website for a Coffee Shop

Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos:** **Use tools like Visual Studio Code (free), XAMPP/WAMP for PHP server setup, and MySQL Workbench for database management** **Videos:** **https://www.coursera.org/learn/web-app**
	Self-study / Do it yourself /:

	Exercises on creating and manipulating databases using PHP and MySQL.
	Experiential Learning Topics: Create a database and design a webpage to display its data dynamically.
	PBL - Project Based Learning: Develop a fully functional web application (e.g., a Coffee Shop website or e-commerce platform) that integrates database functionality for data management.

Experiment No.1

Problem Statement:

1. Create the basic structure of the online blogging platform website, including the home page layout with a header, main content area, and footer.

Prepare a common project website design and plan document for all assignments. Consider the following points:

1. Brief information about the project.
2. Set the goals & deliverables.
3. Finalize the modules of the project.
4. Define the audience.
5. Describe pain points & the ideal experience (On the basis of existing systems)
6. Set the visual direction
7. Map out the Project structure.
8. Plan the content for each page.
9. Add ideas for content, images & layout.
10. Determine your site structure or create content for your core website pages:

a. Home page
b. About page
c. Product/Service page
d. Testimonial/review page
e. Contact page
f. Starter blog posts

11. Create and collect design elements
12. These design elements define your brand personality and help customers feel what your brand represents through the use of:

a. Colors
b. Fonts and typography
c. Logos
d. Images and photos

Objective:

To design the basic structure of an online blogging platform website by planning its layout, content, and visual elements, ensuring it meets user needs and effectively represents the brand.

Theory:

Project Design and Plan Document for an Online Blog Platform Website

BLOGG – Online Blogging Platform

Project Website Design & Plan Document

---

1. Problem Statement

Create the basic structure of an online blogging platform website, including a clean home page layout with a **header**, **main content area**, and **footer**. Ensure consistency, responsiveness, and usability across multiple pages and components of the platform.

---

2. Brief Information about the Project

Project Title: BLOGG – A Modern Online Blogging Platform
Description:
BLOGG is a dynamic and user-friendly blogging platform designed for writers, readers, and content creators to share their ideas and stories. It features a sleek design, interactive components like "Add to Favourites," dark mode, and a structured navigation system that supports engagement and exploration.

---

3. Goals & Deliverables

Goals:

- Design a fully responsive blog website.
- Enhance user interaction with interactive UI elements.
- Provide an engaging space for blog discovery and reading.
- Enable content creators to post, manage, and share their blogs easily.

Deliverables:

- Multi-page blog platform with consistent layout
- Functional homepage, blog listings, and individual blog posts
- "Add to Favourite" feature
- Contact form integration (using JavaScript, PHP & MySQL)
- Dark mode toggle feature
- Admin backend (optional in future stages)

---

s4. Finalized Project Modules

- Header with Navigation
- Homepage (Featured + Latest Blogs)
- About Page
- Blog Post Detail Page
- Contact Form with DB Integration
- Testimonials/Reviews Page
- Starter Blog Posts Section
- Footer with Social Media and Legal Info

5. Target Audience

- Aspiring and experienced bloggers
- Readers seeking niche content
- Students and educators using blogs for knowledge sharing
- Creative professionals (writers, journalists, content creators)

6. Pain Points & Ideal Experience

Pain Points in Existing Blogging Sites:

- Overloaded interfaces with too many distractions
- Lack of personalization and interactivity
- Poor mobile optimization

Ideal Experience:

- Minimalistic and clean layout
- Smooth and intuitive navigation
- Easy-to-read typography
- Fast page loading and responsiveness
- Interactive features like bookmarking, dark mode

7. Visual Direction

- **Tone:** Clean, modern, professional
- **Color Scheme:**

Primary	`#1E88E5`	A vibrant blue for buttons/links
Secondary	`#1565C0`	Deeper blue for hover states
Background (light)	`#FFFFFF`	Clean white for readability
Background (dark)	`#121212`	Soft black for dark mode
Accent	`#FFB74D`	Warm orange for highlights/favs
Text (light)	`#212121`	Dark grey for body text

Text (dark)	`#E0E0E0`	Light grey for dark mode readability

- **Typography:** Elegant and readable fonts with strong hierarchy
- **Imagery:** Editorial-style blog banners, author avatars, and thumbnail previews

---

8. Project Folder Structure

```
BLOGG/
│
├── home.html (Home)
├── about.html
├── blog.html (Blog post listing)
├── favourite.html
├── contact.html
├── review.html
├── services.html
│
├── /css/
│   └── design.css
│
├── /js/
│   └── web.js (dark mode, form handling, etc.)
│
├── /php/
│   └── save-form.php (form submission)
│
├── /images/
│   └── logo.png
│
```

---

9. Content Plan for Pages

Home Page

- Header with site name and navigation links
- Hero section with intro to the platform
- Featured/Recent Blog Posts grid
- “Add to Favourite” functionality
- Footer with newsletter and social links

About Page

- Story of BLOGG
- Mission and values
- Platform highlights or stats
- Creator team photos and bios (optional)

Blog Listing Page

- Filterable list of blog posts
- Tags/categories for browsing
- Blog thumbnails, titles, and summaries

Individual Blog Post Page

- Full post content with author info
- Like/Bookmark/Comment section
- Related posts section

Testimonial/Review Page

- Real feedback from users
- Star ratings and quotes
- Reviewer names/photos (optional)

Contact Page

- Contact form (Name, Email, Message)
- Embedded Google map (optional)
- Contact info and support links

---

10. Content, Image & Layout Ideas

- **Images:** Stock photos, author avatars, blog feature images, quote graphics
- **Layout:**
  - Grid for blog cards
  - Sidebar for tags/recommendations
  - Single-column layout for posts
  - Responsive layout using flexbox or grid
- **Content Types:**
  - How-tos, personal stories, opinion pieces
  - Industry news, guides, tutorials

---

1. **Create and collect design elements**
2. **These design elements define your brand personality and help customers feel what your brand represents through the use of:**

1. Colors

Role	Color Code	Description
Primary	`#1E88E5`	A vibrant blue for buttons/links
Secondary	`#1565C0`	Deeper blue for hover states
Background (light)	`#FFFFFF`	Clean white for readability
Background (dark)	`#121212`	Soft black for dark mode
Accent	`#FFB74D`	Warm orange for highlights/favs
Text (light)	`#212121`	Dark grey for body text
Text (dark)	`#E0E0E0`	Light grey for dark mode readability

Consistent with a calm, trustworthy blog environment with vibrant CTAs (calls to action).

2. Fonts and Typography

Use **Google Fonts** for modern, clean typography:

```
@import
url('https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&family=Playfair+Display:wght@600&display=swap');
```

Usage	Font	Style
Headings	`Playfair Display`	Serif, stylish for elegance
Body text	`Inter`	Sans-serif, clean and readable
Accents (optional)	`Roboto Mono`	For tech-style or code snippets

```
body {
  font-family: 'Inter', sans-serif;
}
h1, h2, h3 {
  font-family: 'Playfair Display', serif;
}
code {
  font-family: 'Roboto Mono', monospace;
}
```

3. Logo Design

You'll want a simple but recognizable logo:

- **Style**: Wordmark-based (e.g., "BLOGG" in Playfair Display with slight kerning).
- **Color Versions**:
  - Blue on black (light mode)
  - White on blue or white on black (dark mode)

4. Images and Photos

Image Style Guide:

Type	Style
Blog cover images	Soft light, clean layout
Icons (e.g., heart for favorites)	Line-style or filled with hover change

- Use Unsplash or Pexels for stock images.
- Maintain a consistent filter or tone across all posts.
- Example: Warm tones with high brightness and contrast for lifestyle content.

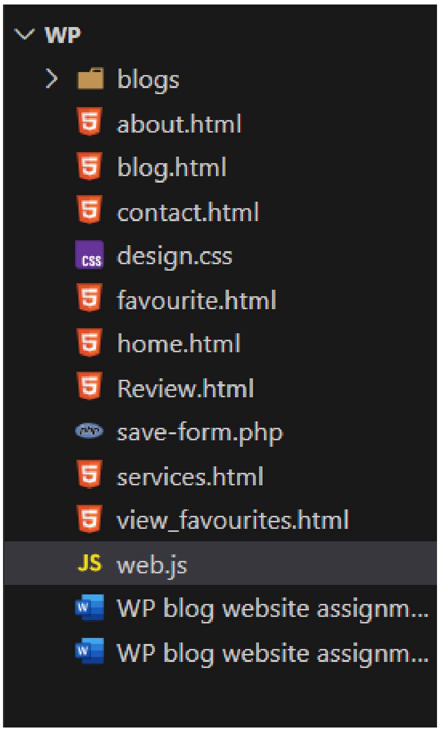
```
img {
  border-radius: 8px;
  object-fit: cover;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}
```

Implementation Suggestion:

- Add variables in CSS for easy theming:

```
:root {
  --primary-color: #1E88E5;
  --secondary-color: #1565C0;
  --accent-color: #FFB74D;
  --text-light: #212121;
  --text-dark: #E0E0E0;
  --background-light: #FFFFFF;
  --background-dark: #121212;
}
```

Output:



```
WP
  blogs
  about.html
  blog.html
  contact.html
  design.css
  favourite.html
  home.html
  Review.html
  save-form.php
  services.html
  view_favourites.html
  web.js
  WP blog website assignm...
  WP blog website assignm...
```

Conclusion :

The BLOGG platform is a clean, responsive, and user-focused blogging website designed to enhance content creation and reading experiences. With features like dark mode, interactive blog listings, and a well-structured layout, it addresses common pain points in existing platforms. The design plan outlines clear goals, target audience, visual direction, and page content, ensuring a consistent brand identity and smooth user journey. This well-documented plan provides a strong foundation for development and future scalability.

Experiment No.2

Problem Statement:

2. HTML

A. Create a detailed home page for the coffee shop website.

B. Create a detailed menu/product page for the coffee shop website, listing all available items categorized appropriately.

C. Create a cart page that allows customers to review and manage the items they wish to purchase before proceeding to checkout.

D. Create an about us page that provides detailed information about the coffee shop's history, mission, and team.

E. Create a contact page that allows customers to easily get in touch with the coffee shop through a form.

F. Design and implement admin/user registration form for the coffee shop website.

G. Design and implement admin/user login form for the coffee shop website.

Objective:

To create an online blogging platform webpage using HTML.

Theory:

HTML (HyperText Markup Language) is the standard markup language used to structure and present content on the web. It plays a crucial role in building web applications by defining the layout and structure of pages through elements like headings, paragraphs, images, links, forms, and more.In this project, we are creating an **online blogging platform** using HTML. A blogging platform is a digital space where users can publish, read, and interact with blog posts. It serves as a medium for personal expression, education, or even brand engagement.

The website includes several core pages:

- **Home Page**: Displays the latest or featured blog posts.
- **About Page**: Describes the purpose, mission, and background of the platform.
- **Product/Service Page**: (Optional) Can be used to highlight platform features or premium tools.
- **Testimonial/Review Page**: Showcases feedback from users or contributors.
- **Contact Page**: Provides a form to get in touch with the platform admins.
- **User/Admin Registration and Login Pages**: Allow users to securely create accounts and log in to post or manage content.
- **Blog Post Pages**: Allow reading and browsing of individual blog entries.
- **Starter Blog Posts**: Provide example content to demonstrate functionality.

Although this phase focuses on **HTML**, the real strength of the platform emerges when paired with **CSS** (for styling), **JavaScript** (for interactivity), and server-side scripting (like PHP or Node.js) along with databases (like MySQL) for content management.

HTML Tags and Elements

- **Headings**: <h1> to <h6> – Define title levels.
- **Paragraphs**: <p> – For text blocks.
- **Links**: <a href="..."> – Create hyperlinks.

- **Images**: <img src="..." alt="..."> – Embed images.
- **Lists**: <ul>, <ol>, <li> – Unordered/Ordered lists.
- **Tables**: <table>, <tr>, <td> – Display tabular data.
- **Forms**: <form>, <input>, <textarea>, <button> – Accept user input.

Features of HTML

1. **Platform Independent**: HTML files can run on any browser or operating system.
2. **Simple & Easy to Learn**: Readable syntax with no compilation needed.
3. **Supports Multimedia**: Can embed audio, video, and images.
4. **Semantic Markup**: HTML5 provides tags like <article>, <section>, <nav>, and <aside> to improve content structure and accessibility.
5. **Linking Documents**: Enables creation of a hyperlink-based information network (World Wide Web).

HTML5 Advancements

HTML5 is the latest version, with many new features to support modern web applications:

Feature	Description
<video> / <audio>	Embed media without plugins
<canvas>	Draw graphics using JavaScript
<section>, <article>	Semantic structuring of content
Local Storage	Save data in the browser
Geolocation API	Access user location
Responsive Forms	New input types like email, date, etc.

HTML + CSS + JavaScript

HTML forms the **structure**, CSS handles **presentation (style)**, and JavaScript enables **behavior (interactivity)**. Together, they create fully functional and dynamic websites.

Advantages of HTML

- Widely supported and standardized
- Fast loading and lightweight
- Seamless integration with CSS and JS
- Ideal for SEO and accessibility

Limitations

- Not suitable for complex logic or functionality (relies on JS for that)
- Static by nature (needs backend/server-side support for dynamic content)

Use Case: HTML in BLOGG Platform

In the **BLOGG** online blogging platform:

- HTML structures each page (home, blog posts, about, contact).
- Tags organize blog cards, navigation links, form fields, etc.
- Semantic tags enhance SEO and user accessibility.

Code:

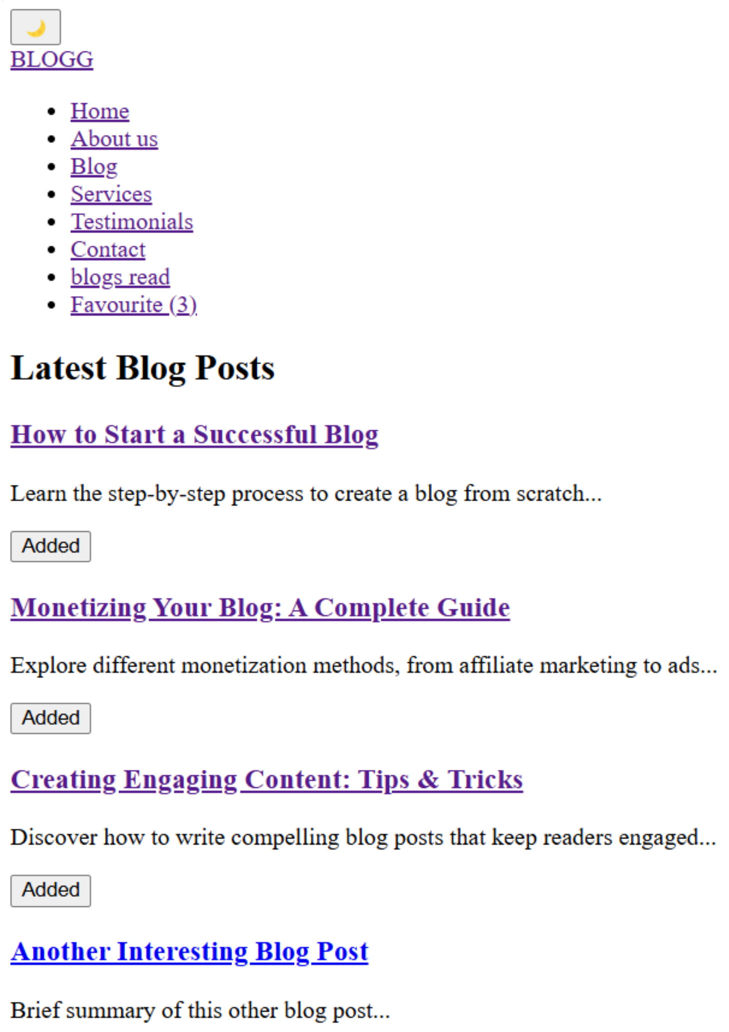
A. Home page:

code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Home</title>
  <link rel="stylesheet" href="design.css">
  <script src="web.js" defer></script>
</head>
<body class="dark-mode">
  <header>
    <div class="logo"><a href="/">BLOGG</a></div>
    <nav>
      <ul>
        <li><a href="home.html">Home</a></li>
        <li><a href="about.html">About us</a></li>
        <li><a href="blog.html">Blog</a></li>
        <li><a href="services.html">Services</a></li>
        <li><a href="Review.html">Testimonials</a></li>
        <li><a href="contact.html">Contact</a></li>
        <li><a href="favourite.html">blogs read </a></li>
        <li><a href="view_favourites.html">Favourite (<span id="favorites-count">0</span>)</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section class="latest-posts">
      <h2>Latest Blog Posts</h2>
      <article>
        <h3><a href="blogs/start-successful-blog.html">How to Start a Successful Blog</a></h3>
        <p>Learn the step-by-step process to create a blog from scratch...</p>
        <button class="fav-button" data-title="How to Start a Successful Blog" data-link="blogs/start-successful-blog.html"> Add</button>
      </article>
      <article>
        <h3><a href="blogs/monetization-guide.html">Monetizing Your Blog: A Complete Guide</a></h3>
        <p>Explore different monetization methods, from affiliate marketing to ads...</p>
        <button class="fav-button" data-title="Monetizing Your Blog: A Complete Guide" data-link="blogs/monetization-guide.html"> Add</button>
      </article>
      <article>
        <h3><a href="blogs/creating-engaging-content.html">Creating Engaging Content: Tips & Tricks</a></h3>
        <p>Discover how to write compelling blog posts that keep readers engaged...</p>
        <button class="fav-button" data-title="Creating Engaging Content: Tips & Tricks" data-link="blogs/creating-engaging-content.html"> Add</button>
      </article>
      <article>
        <h3><a href="blogs/another-blog-post.html">Another Interesting Blog Post</a></h3>
        <p>Brief summary of this other blog post...</p>
        <button class="fav-button" data-title="Another Interesting Blog Post" data-link="blogs/another-blog-post.html"> Add</button>

      </article>

    </section>

  </main>



  <footer>

    <p>&copy; 2025 BLOGG. All rights reserved.</p>

  </footer>

</body>

</html>
```

Output:

A. Index/Home page output:



🌙

BLOGG

- Home
- About us
- Blog
- Services
- Testimonials
- Contact
- blogs read
- Favourite (3)

Latest Blog Posts

How to Start a Successful Blog

Learn the step-by-step process to create a blog from scratch...

Added

Monetizing Your Blog: A Complete Guide

Explore different monetization methods, from affiliate marketing to ads...

Added

Creating Engaging Content: Tips & Tricks

Discover how to write compelling blog posts that keep readers engaged...

Added

Another Interesting Blog Post

Brief summary of this other blog post...

Code:

B. Blog page:

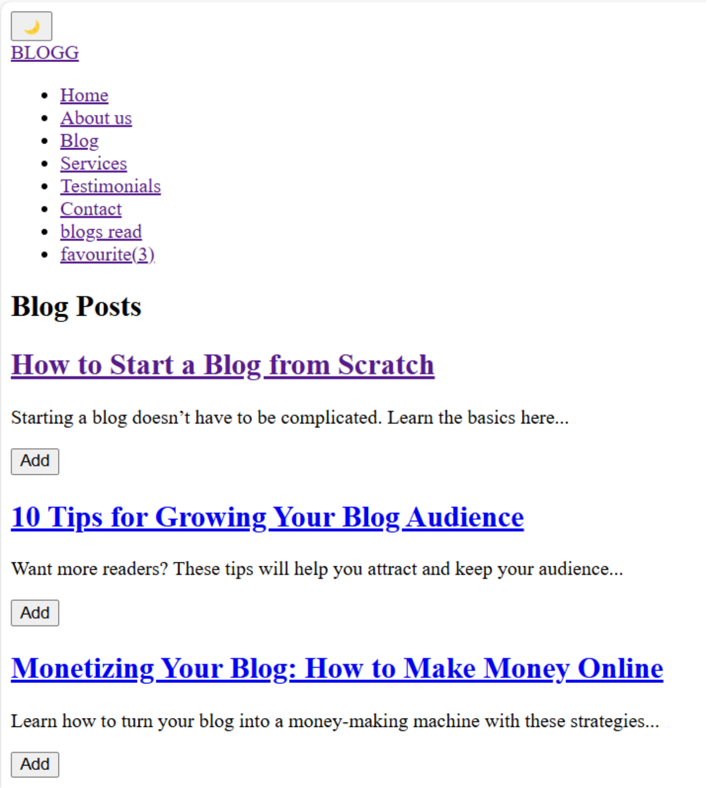
code:

```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

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

   <title>Blog | Blogging Website</title>

   <link rel="stylesheet" href="design.css">

    <script src="web.js" defer></script>

</head>

<body class="dark-mode">


   <header>

      <div class="logo"><a href="/">BLOGG</a></div>

      <nav>

         <ul>

            <li><a href="home.html">Home</a></li>

            <li><a href="about.html">About us</a></li>

            <li><a href="blog.html">Blog</a></li>

            <li><a href="services.html">Services</a></li>

            <li><a href="Review.html">Testimonials</a></li>

            <li><a href="contact.html">Contact</a></li>

             <li><a href="favourite.html">blogs read</a></li>

             <li><a href="view_favourites.html">favourite(<span id="favorites-count">0</span>)</a></li>

         </ul>

      </nav>

   </header>
```

```
    <main>
        <section class="blog-posts">
            <h1>Blog Posts</h1>
            <article>
                <h2><a href="blogs/start-successful-blog.html">How to Start a Blog from Scratch</a></h2>
                <p>Starting a blog doesn’t have to be complicated. Learn the basics here...</p>
                <button class="fav-button" data-title="How to Start a Blog from Scratch" data-link="blogs/start-successful-blog.html"> Add</button>
            </article>
            <article>
                <h2><a href="blogs/growing-blog-audience.html">10 Tips for Growing Your Blog Audience</a></h2>
                <p>Want more readers? These tips will help you attract and keep your audience...</p>
                <button class="fav-button" data-title="10 Tips for Growing Your Blog Audience" data-link="blogs/growing-blog-audience.html"> Add</button>
            </article>
            <article>
                <h2><a href="blogs/monetizing-your-blog.html">Monetizing Your Blog: How to Make Money Online</a></h2>
                <p>Learn how to turn your blog into a money-making machine with these strategies...</p>
                <button class="fav-button" data-title="Monetizing Your Blog: How to Make Money Online" data-link="blogs/monetizing-your-blog.html"> Add</button>
            </article>
            <article>
                <h2><a href="blogs/best-blogging-platforms.html">Choosing the Best Blogging Platform for You</a></h2>
                <p>Not sure which platform to use? We compare popular options to help you decide...</p>
                <button class="fav-button" data-title="Choosing the Best Blogging Platform for You" data-link="blogs/best-blogging-platforms.html"> Add</button>
            </article>
            <article>
                <h2><a href="blogs/writing-compelling-headlines.html">The Art of Writing Compelling Blog Headlines</a></h2>
        <p>Master the skill of crafting headlines that grab attention and drive clicks...</p>

        <button class="fav-button" data-title="The Art of Writing Compelling Blog Headlines" data-link="blogs/writing-compelling-headlines.html"> Add</button>

      </article>

      <article>

        <h2><a href="blogs/seo-for-bloggers.html">SEO Basics for Bloggers: Getting Your Content Found</a></h2>

        <p>Understand the fundamentals of SEO to increase your blog's visibility in search engines...</p>

        <button class="fav-button" data-title="SEO Basics for Bloggers: Getting Your Content Found" data-link="blogs/seo-for-bloggers.html"> Add</button>

      </article>

    </section>

  </main>


  <footer>

    <p>&copy; 2025 My Blog | All rights reserved.</p>

  </footer>

</body>

</html>
```

Output:

B. Blog page output:



🌙

BLOGG

- Home
- About us
- Blog
- Services
- Testimonials
- Contact
- blogs read
- favourite(3)

Blog Posts

How to Start a Blog from Scratch

Starting a blog doesn’t have to be complicated. Learn the basics here...

Add

10 Tips for Growing Your Blog Audience

Want more readers? These tips will help you attract and keep your audience...

Add

Monetizing Your Blog: How to Make Money Online

Learn how to turn your blog into a money-making machine with these strategies...

Add

Code:

C. Favourite page:

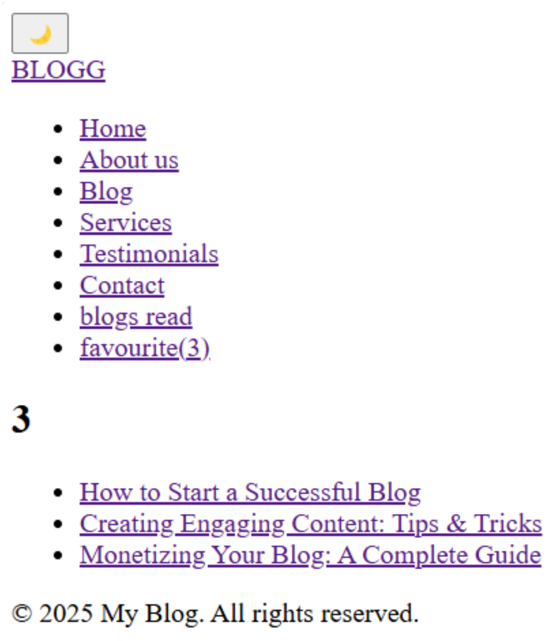
code:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Favorites</title>

    <link rel="stylesheet" href="design.css">

    <script src="web.js" defer></script>

</head>

<body class="dark-mode">

    <header>

        <div class="logo"><a href="index.html">BLOGG</a></div>

        <nav>
```

```
            <ul>
              <li><a href="home.html">Home</a></li>
              <li><a href="about.html">About us</a></li>
              <li><a href="blog.html">Blog</a></li>
              <li><a href="services.html">Services</a></li>
              <li><a href="Review.html">Testimonials</a></li>
              <li><a href="contact.html">Contact</a></li>
              <li><a href="favourite.html">blogs read</a></li>
              <li><a href="view_favourites.html">favourite(<span id="favorites-count">0</span>)</a></li>
            </ul>
          </nav>
        </header>
        <main>
          <section class="favorites">
            <h2> <span id="favorites-count">(0)</span></h2>
            <ul id="favorites-list">
              <p id="empty-message">No favorite blogs saved yet.</p>
            </ul>
          </section>
        </main>
        <footer>
          <p>&copy; 2025 My Blog. All rights reserved.</p>
        </footer>
      </body>
      </html>
```

<u>Output</u>:

C. Favourite page output:



🌙

BLOGG

- Home
- About us
- Blog
- Services
- Testimonials
- Contact
- blogs read
- favourite(3)

3

- How to Start a Successful Blog
- Creating Engaging Content: Tips & Tricks
- Monetizing Your Blog: A Complete Guide

© 2025 My Blog. All rights reserved.

Code:

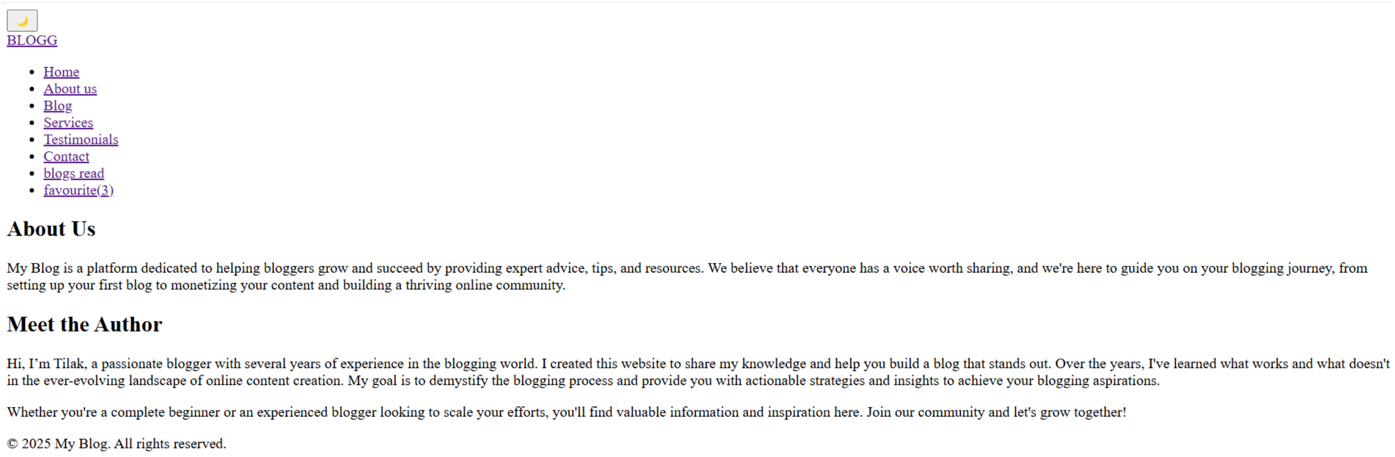
D. about us page:

code:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>About Us | My Blog</title>

  <link rel="stylesheet" href="design.css">

   <script src="web.js" defer></script>

</head>

<body class="dark-mode">

  <header>

    <div class="logo"><a href="/">BLOGG</a></div>

    <nav>

      <ul>

        <li><a href="home.html">Home</a></li>

        <li><a href="about.html">About us</a></li>
        <li><a href="blog.html">Blog</a></li>
        <li><a href="services.html">Services</a></li>
        <li><a href="Review.html">Testimonials</a></li>
        <li><a href="contact.html">Contact</a></li>
         <li><a href="favourite.html">blogs read</a></li>
         <li><a href="view_favourites.html">favourite(<span id="favorites-count">0</span>)</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section class="about">
      <h1>About Us</h1>
      <p>My Blog is a platform dedicated to helping bloggers grow and succeed by providing expert advice, tips, and resources. We believe that everyone has a voice worth sharing, and we're here to guide you on your blogging journey, from setting up your first blog to monetizing your content and building a thriving online community.</p>
      <h2>Meet the Author</h2>
      <p>Hi, I’m Tilak, a passionate blogger with several years of experience in the blogging world. I created this website to share my knowledge and help you build a blog that stands out. Over the years, I've learned what works and what doesn't in the ever-evolving landscape of online content creation. My goal is to demystify the blogging process and provide you with actionable strategies and insights to achieve your blogging aspirations.</p>
      <p>Whether you're a complete beginner or an experienced blogger looking to scale your efforts, you'll find valuable information and inspiration here. Join our community and let's grow together!</p>
    </section>
  </main>
  <footer>
    <p>&copy; 2025 My Blog. All rights reserved.</p>
  </footer>
</body>
</html>
```

Output:

D. about us page output:



🌙

BLOGG

- Home
- About us
- Blog
- Services
- Testimonials
- Contact
- blogs read
- favourite(3)

About Us

My Blog is a platform dedicated to helping bloggers grow and succeed by providing expert advice, tips, and resources. We believe that everyone has a voice worth sharing, and we're here to guide you on your blogging journey, from setting up your first blog to monetizing your content and building a thriving online community.

Meet the Author

Hi, I'm Tilak, a passionate blogger with several years of experience in the blogging world. I created this website to share my knowledge and help you build a blog that stands out. Over the years, I've learned what works and what doesn't in the ever-evolving landscape of online content creation. My goal is to demystify the blogging process and provide you with actionable strategies and insights to achieve your blogging aspirations.

Whether you're a complete beginner or an experienced blogger looking to scale your efforts, you'll find valuable information and inspiration here. Join our community and let's grow together!

© 2025 My Blog. All rights reserved.

Code:

E. contact us page:

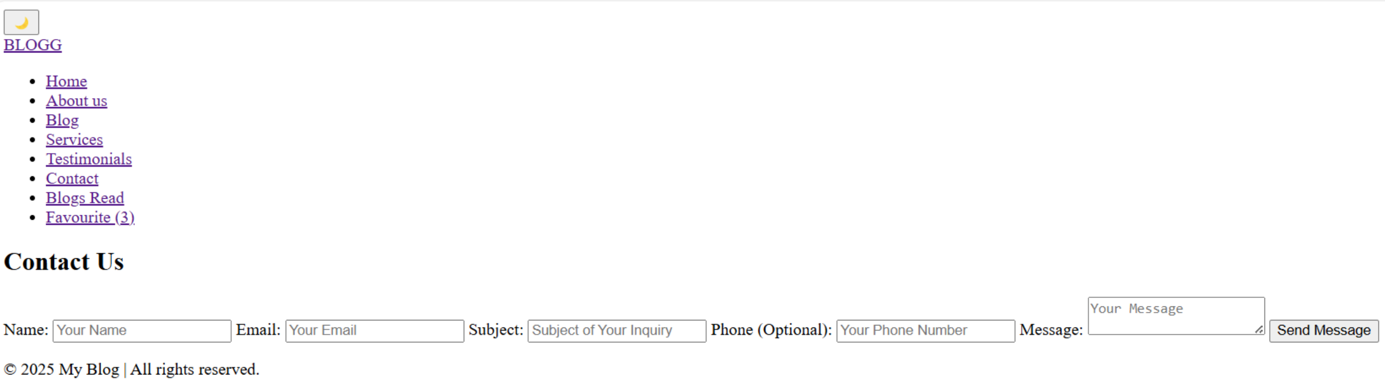
code:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Contact Us | My Blog</title>

  <link rel="stylesheet" href="design.css">

  <script src="web.js" defer></script>

</head>

<body class="dark-mode">

  <header>

    <div class="logo"><a href="/">BLOGG</a></div>

    <nav>

      <ul>

        <li><a href="home.html">Home</a></li>

        <li><a href="about.html">About us</a></li>

        <li><a href="blog.html">Blog</a></li>
```

```
            <li><a href="services.html">Services</a></li>

            <li><a href="Review.html">Testimonials</a></li>

            <li><a href="contact.html">Contact</a></li>

            <li><a href="favourite.html">Blogs Read</a></li>

            <li><a href="view_favourites.html">Favourite (<span id="favorites-count">0</span>)</a></li>

        </ul>

    </nav>

</header>


<main class="contact">

    <section class="contact-form">

        <form id="contact-form" method="post">

            <h1>Contact Us</h1>


            <label for="name">Name:</label>

            <input type="text" id="name" name="name" placeholder="Your Name" required>


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

            <input type="email" id="email" name="email" placeholder="Your Email" required>


            <label for="subject">Subject:</label>

            <input type="text" id="subject" name="subject" placeholder="Subject of Your Inquiry" required>


            <label for="phone">Phone (Optional):</label>

            <input type="tel" id="phone" name="phone" placeholder="Your Phone Number">


            <label for="message">Message:</label>

            <textarea id="message" name="message" placeholder="Your Message" required></textarea>


            <button type="submit">Send Message</button>
        <form id="contact-form" method="POST" action="save-form.php">

      </form>

      <!-- Success Message -->
      <p id="success-message" style="display: none; color: green; font-weight: bold;">
        ✅ Your message has been sent successfully!
      </p>
    </section>
  </main>

  <footer>
    <p>&copy; 2025 My Blog | All rights reserved.</p>
  </footer>
</body>
</html>
```

Output:

E. contact us page output:



🌙

BLOGG

- Home
- About us
- Blog
- Services
- Testimonials
- Contact
- Blogs Read
- Favourite (3)

Contact Us

Name: Your Name Email: Your Email Subject: Subject of Your Inquiry Phone (Optional): Your Phone Number Message: Your Message Send Message

© 2025 My Blog | All rights reserved.

Code:

F. Review page:

code:

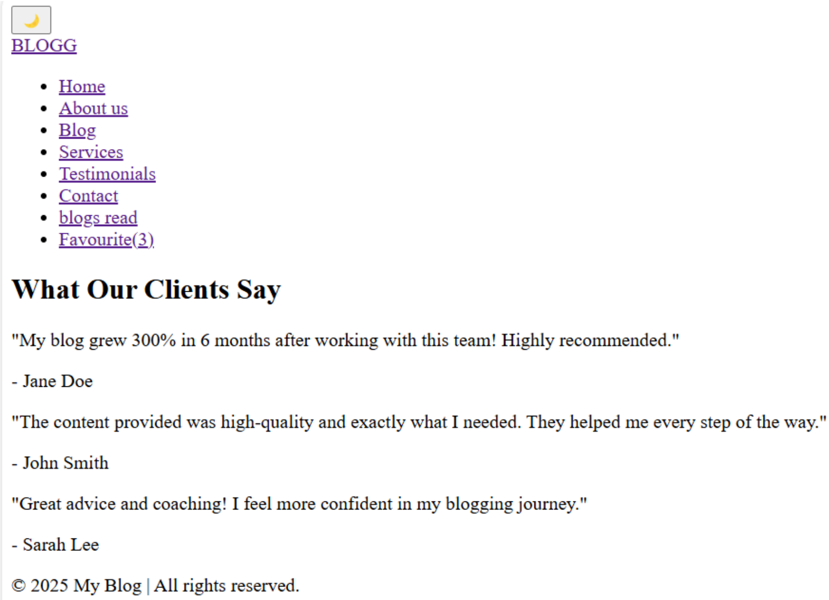
```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Testimonials | Blogging Website</title>
  <link rel="stylesheet" href="design.css">
  <script src="web.js" defer></script>
</head>
<body class="dark-mode">
  <header>
    <div class="logo"><a href="/">BLOGG</a></div>
    <nav>
      <ul>
        <li><a href="home.html">Home</a></li>
        <li><a href="about.html">About us</a></li>
        <li><a href="blog.html">Blog</a></li>
        <li><a href="services.html">Services</a></li>
        <li><a href="Review.html">Testimonials</a></li>
        <li><a href="contact.html">Contact</a></li>
         <li><a href="favourite.html">blogs read</a></li>
         <li><a href="view_favourites.html">Favourite(<span id="favorites-count">0</span>)</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section class="testimonials">
      <h1>What Our Clients Say</h1>
      <div class="testimonial-item visible">
        <p>"My blog grew 300% in 6 months after working with this team! Highly recommended."</p>
        <p>- Jane Doe</p>
      </div>
      <div class="testimonial-item">
```

```
        <p>"The content provided was high-quality and exactly what I needed. They helped me every step of the way."</p>

        <p>- John Smith</p>

      </div>

      <div class="testimonial-item">

        <p>"Great advice and coaching! I feel more confident in my blogging journey."</p>

        <p>- Sarah Lee</p>

      </div>

    </section>

  </main>

  <footer>

    <p>&copy; 2025 My Blog | All rights reserved.</p>

  </footer>

</body>

</html>
```

<u>Output</u>:

F. Review page output:



🌙

BLOGG

- Home
- About us
- Blog
- Services
- Testimonials
- Contact
- blogs read
- Favourite(3)

What Our Clients Say

"My blog grew 300% in 6 months after working with this team! Highly recommended."

- Jane Doe

"The content provided was high-quality and exactly what I needed. They helped me every step of the way."

- John Smith

"Great advice and coaching! I feel more confident in my blogging journey."

- Sarah Lee

© 2025 My Blog | All rights reserved.

<u>Code</u>:

G. Services page:

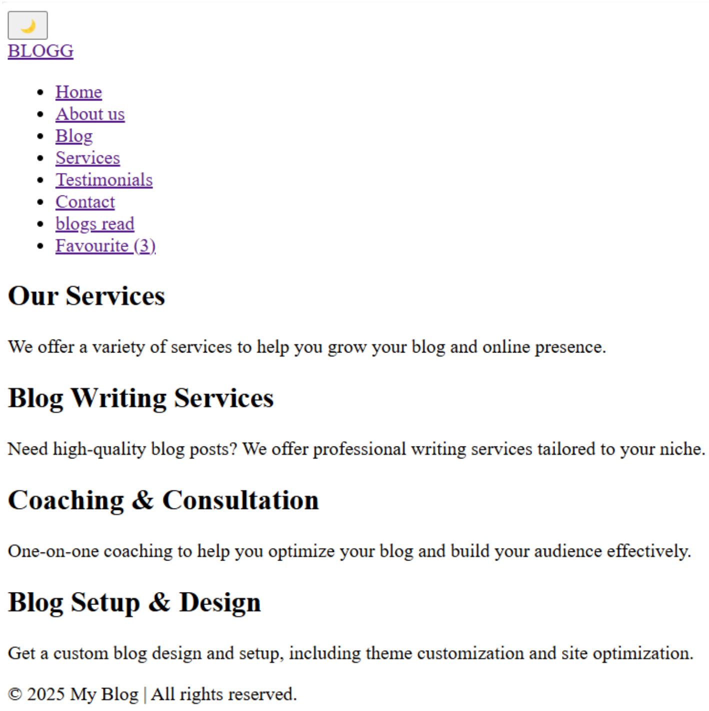
code:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Services | Blogging Website</title>
    <link rel="stylesheet" href="design.css">
     <script src="web.js" defer></script>
</head>
<body class="dark-mode">
    <header>
        <div class="logo"><a href="/">BLOGG</a></div>
        <nav>
            <ul>
                 <li><a href="home.html">Home</a></li>
                <li><a href="about.html">About us</a></li>
                <li><a href="blog.html">Blog</a></li>
                <li><a href="services.html">Services</a></li>
                <li><a href="Review.html">Testimonials</a></li>
                <li><a href="contact.html">Contact</a></li>
                 <li><a href="favourite.html">blogs read</a></li>
                 <li><a href="view_favourites.html">Favourite (<span id="favorites-count">0</span>)</a></li>
            </ul>
        </nav>
    </header>
    <main>
        <section class="services">
            <h1>Our Services</h1>
            <p>We offer a variety of services to help you grow your blog and online presence.</p>
            <div class="service-item">
                <h2>Blog Writing Services</h2>
        <p>Need high-quality blog posts? We offer professional writing services tailored to your niche.</p>
      </div>
      <div class="service-item">
        <h2>Coaching & Consultation</h2>
        <p>One-on-one coaching to help you optimize your blog and build your audience effectively.</p>
      </div>
      <div class="service-item">
        <h2>Blog Setup & Design</h2>
        <p>Get a custom blog design and setup, including theme customization and site optimization.</p>
      </div>
    </section>
  </main>
  <footer>
    <p>&copy; 2025 My Blog | All rights reserved.</p>
  </footer>
</body>
</html>
```

Output:

G. Services page output:



🌙

BLOGG

- Home
- About us
- Blog
- Services
- Testimonials
- Contact
- blogs read
- Favourite (3)

Our Services

We offer a variety of services to help you grow your blog and online presence.

Blog Writing Services

Need high-quality blog posts? We offer professional writing services tailored to your niche.

Coaching & Consultation

One-on-one coaching to help you optimize your blog and build your audience effectively.

Blog Setup & Design

Get a custom blog design and setup, including theme customization and site optimization.

© 2025 My Blog | All rights reserved.

Conclusion:

This experiment focused on creating a basic structure of an online blogging platform using HTML. We developed key pages like the home, about, contact, registration, and login pages. The project helped us understand how to build a clean, user-friendly layout and organize content effectively, forming a solid foundation for future web development.

Experiment No.3

Problem Statement

Enhance the layout of the BLOGG platform website using CSS Grid for the homepage. Use CSS Grid to structure blog listings and organize categories with headings, spacing, images, summaries, and reading links.

Theory: CSS Grid for BLOGG – Online Blogging Platform

Introduction to CSS Grid

CSS Grid Layout is a two-dimensional system that helps design responsive web layouts efficiently. Unlike Flexbox (which is one-directional), Grid works across both rows and columns—ideal for websites with structured content like BLOGG.

Why CSS Grid for BLOGG?

A blog platform like BLOGG needs structured, clean presentation for content. CSS Grid helps in:

- Creating responsive layouts for blog cards
- Arranging sections like "Featured Blogs", "Latest Posts", or "Popular Categories"
- Ensuring consistency in alignment of blog images, titles, and summaries
- Supporting clean mobile and desktop views

---

1. Homepage Layout with CSS Grid

The homepage includes:

- A full-width navigation bar
- A hero section with an intro or banner
- A blog post grid with featured/recent blogs
- A testimonial section in a row
- A footer with links and social media

Benefits:

- Better control of spacing and structure
- Easy scalability for new sections
- Clean user experience across all screen sizes

---

2. Blog Listing Layout Using CSS Grid

Each blog post is shown as a card with:

- Blog image
- Title
- Short summary
- Author/date
- "Read More" link

Example CSS:

```
css
CopyEdit
.blog-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 30px;
  padding: 20px;
}
```

Additional Styling Concepts

- **Category Headings**: Use font size and background to separate categories
- **Separators**: Use borders between sections
- **Hover Effects**: Animate card or button on hover
- **Responsive Design**: `auto-fit` and `minmax()` make content mobile-friendly

---

Mobile Responsiveness with CSS Grid

CSS Grid allows smooth transitions on small screens by adjusting column counts and row alignment. For BLOGG:

- 1–2 column layout on mobile
- Tap-friendly spacing
- Fast, clean mobile browsing

Key Features of CSS

- **Styling**: Fonts, colors, backgrounds, borders, spacing, shadows, etc.
- **Layout Control**: Positioning, display types (block, inline, flex, grid), and alignment.
- **Responsiveness**: Media queries to adapt layouts for different devices.
- **Reusability**: Styles can be reused across multiple HTML pages.
- **Maintainability**: Keeps design consistent and easier to update.

CSS Selectors

CSS uses **selectors** to target HTML elements:

- element – targets by tag (p, h1)
- #id – targets by ID (#header)
- .class – targets by class (.button)
- * – universal selector (all elements)
- div > p, ul li, a:hover – complex/nested selectors and pseudo-classes

---

CSS Box Model

Every HTML element is a box consisting of:

- **Content** – the text or image inside the element
- **Padding** – space around the content
- **Border** – surrounding line
- **Margin** – space outside the border

css

CopyEdit

```
div {

  margin: 10px;

  padding: 15px;

  border: 1px solid #000;

}
```

Advanced CSS Features

Feature	**Description**
Flexbox	Layout model for 1D layouts (row/column)
Grid	Layout model for 2D page layouts
Animations	Smooth transitions and keyframe animations
Variables	Custom properties for reusable values (--main-color)
Media Queries	Make designs responsive across screen sizes

CSS in BLOGG Platform

In the **BLOGG** project:

- CSS is used to create a **clean and modern interface**.
- A **dark mode toggle** changes the entire theme using variable overrides.
- Grid and flexbox layouts organize blog cards and navigation.
- Responsive design ensures readability on all devices.

Example:

css

CopyEdit

```
:root {
  --primary-color: #1E88E5;
  --background-light: #FFFFFF;
  --text-color: #212121;
}
body {
  background-color: var(--background-light);
  color: var(--text-color);
  font-family: 'Inter', sans-serif;
}
```

Advantages of CSS

- Separates content from design
- Enhances page loading speed
- Promotes reusability and consistency
- Enables responsive web design
- Works across all modern web browsers

Limitations

- Dependent on HTML structure
- Complex layouts may require advanced skills
- Browser compatibility issues may arise in older browsers

Code:

```
* {

   margin: 0;

   padding: 0;

   box-sizing: border-box;

}

/* General Styles */

body {

   font-family: 'serif', serif;

   background-color: #1a1a1a;

   color: #eee;

   line-height: 1.7;

}

a {

   color: #80cbc4;

   text-decoration: none;

   transition: color 0.3s ease;

}

a:hover {

   color: #a7f9ee;

}

/* Header */

header {

   background-color: #222;

   padding: 25px 0;

   text-align: center;

   position: sticky;

   top: 0;

   z-index: 100;
```

```
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
    border-bottom: 1px solid #333;
}
header .logo {
    font-size: 32px;
    color: #fff;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.5);
    letter-spacing: 2px;
}
header nav ul {
    list-style-type: none;
    display: flex;
    justify-content: center;
    gap: 40px;
    margin-top: 10px;
}
header nav a {
    color: #eee;
    font-size: 18px;
    position: relative;
    padding-bottom: 8px;
    transition: color 0.3s ease;
}
header nav a:hover {
    color: #fff;
}
header nav a:hover::after {
    content: ";
    position: absolute;
    left: 10%;
    bottom: 0;
    width: 80%;
    height: 2px;
    background-color: #80cbc4;
    transform: scaleX(0);
    transition: transform 0.3s ease-in-out;
}
header nav a:hover::after {
    transform: scaleX(1);
}
/* Mobile Navigation Toggle */
.nav-toggle {
    display: none;
    background-color: transparent;
    border: 1px solid #eee;
    color: #eee;
    font-size: 28px;
    cursor: pointer;
    padding: 8px;
    position: absolute;
    left: 10px;
    top: 10px;
    transition: border-color 0.3s ease, color 0.3s ease;
}
.nav-toggle:hover {
    border-color: #fff;
    color: #fff;
}
```

```
.nav-toggle:focus {
  outline: none;
  box-shadow: 0 0 0 2px rgba(128, 203, 196, 0.5);
}
header nav ul.show {
  display: block;
}
/* Hero Section */
.hero {
  background-color: #2a2a2a;
  color: #eee;
  padding: 80px 20px;
  text-align: center;
  border-radius: 8px;
  margin: 30px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
  animation: fadeIn 1s ease;
}
@keyframes fadeIn {
  from {
    opacity: 0;
    transform: translateY(20px);
  }

  to {
    opacity: 1;
    transform: translateY(0);
  }
}
```

```
.hero h1 {
    font-size: 2.5em;
    font-weight: bold;
    letter-spacing: 2px;
    color: #fff;
}
.hero p {
    font-size: 1.1em;
    color: #ddd;
    line-height: 1.8;
    margin-top: 20px;
}

.services {
    padding: 60px 20px;
    text-align: center;
    background-color: #1a1a1a;
}
.service-item {
    background-color: #222;
    border-radius: 8px;
    margin: 30px auto;
    padding: 30px;
    max-width: 600px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}
.service-item:hover {
    transform: translateY(-5px);
    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.4);
}
.service-item h2 {
    color: #80cbc4;
    margin-bottom: 15px;
    font-size: 24px;
    letter-spacing: 1px;
}
.service-item p {
    color: #ddd;
    line-height: 1.8;
}
.testimonials {
    padding: 60px 20px;
    background-color: #1a1a1a;
    text-align: center;
}
.testimonial-item {
    background-color: #222;
    border-radius: 8px;
    margin: 30px auto;
    padding: 30px;
    max-width: 600px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
    font-style: italic;
    color: #eee;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}
.testimonial-item:hover {
    transform: translateY(-3px);
    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.4);
}
.testimonial-item p {
    font-style: italic;
    color: #b2dfdb;
    line-height: 1.8;
}
@media (min-width: 768px) {
    .contact form {
        grid-template-columns: repeat(2, 1fr);
    }
    .contact form > h1 {
        grid-column: 1 / -1;
    }
    .contact form > button {
        grid-column: 1 / -1;
        width: fit-content;
        margin-left: auto;
        margin-right: auto;
    }
}
.contact form h1 {
    text-align: center;
    margin-bottom: 40px;
    color: #333; /* Changed heading color to dark gray for white background */
    font-size: 36px;
    letter-spacing: 2px;
    font-weight: bold;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.1); /* Adjusted shadow for light background */
}


.contact label {
    display: block;
    margin-top: 0;
    color: #555; /* Changed label color to dark gray */
    font-weight: bold;
    font-size: 1.1em;
    transition: color 0.3s ease;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.1); /* Adjusted shadow */
}
.contact label:hover {
    color: #333; /* Changed hover color */
}
.contact input,
.contact textarea {
    padding: 15px;
    background-color: #f5f5f5; /* Light gray for input fields */
    color: #333;
    border: 1px solid #ccc; /* Lighter border */
    border-radius: 10px;
    transition: background-color 0.3s ease, border-color 0.3s ease, box-shadow 0.3s ease;
    width: 100%;
    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.1); /* Subtle inset shadow */
    font-size: 1em;
}
.contact input:focus,
.contact textarea:focus {
    background-color: #fff;
    border-color: rgba(0, 0, 0, 0.3);
    outline: none;
    box-shadow: 0 4px 10px rgba(128, 203, 196, 0.3);
    color: #333;
}
.contact textarea {
    resize: vertical;
    min-height: 150px;
    color: #333;
    grid-column: 1 / -1;
}
.contact button {
    padding: 15px 30px;
    background-color: #80cbc4;
    color: #fff;
    border: none;
    border-radius: 10px;
    cursor: pointer;
    transition: background-color 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;
    font-size: 1.1em;
    width: fit-content;
    margin: 30px auto 0 auto;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1); /* Reduced shadow for light background */
}
.contact button:hover {
    background-color: #62a39e;
    transform: translateY(-3px);
    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.2); /* Reduced shadow on hover */
```

```
}

.contact button:focus {

   outline: none;

   box-shadow: 0 0 0 3px rgba(128, 203, 196, 0.5), 0 4px 10px rgba(0, 0, 0, 0.1); /* Reduced shadow on
focus */

}

.fav-button,

.remove-btn {

   padding: 12px 20px;

   border-radius: 8px;

   cursor: pointer;

   font-size: 1.1em;

   transition: background-color 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;

   box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

   border: none;

   color: #fff;

   display: flex;

   align-items: center;

   gap: 8px;

}

.fav-button:hover,

.remove-btn:hover {

   transform: translateY(-2px);

   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);

}

.fav-button:focus,

.remove-btn:focus {

   outline: none;

   box-shadow: 0 0 0 2px rgba(128, 203, 196, 0.5), 0 2px 5px rgba(0, 0, 0, 0.2);
```

```
}
.fav-button {
   background-color: #80cbc4;
}
.fav-button:hover {
   background-color: #62a39e;
}
.fav-button.active {
   background-color: #00897b;
   color: #fff;
   box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
}
.remove-btn {
   background-color: #e53e3e;
   margin-left: 15px;
}
.remove-btn:hover {
   background-color: #c53030;
}
footer {
   background-color: #222;
   color: #ddd;
   text-align: center;
   padding: 25px;
   margin-top: 40px;
   border-top: 1px solid #333;
   box-shadow: 0 -2px 5px rgba(0, 0, 0, 0.2);
}
footer p {
    margin: 0;
    font-size: 0.95em;
}
body.dark-mode {
    background-color: #1a1a1a;
    color: #eee;
}
body.dark-mode a {
    color: #80cbc4;
}
body.dark-mode a:hover {
    color: #a7f9ee;
}
body.dark-mode header {
    background-color: #222;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
    border-bottom: 1px solid #333;
}
body.dark-mode .hero {
    background-color: #2a2a2a;
    color: #eee;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
body.dark-mode .services {
    background-color: #1a1a1a;
}
body.dark-mode .service-item {
    background-color: #222;
    color: #eee;
    border-color: #333;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
body.dark-mode .testimonials {
    background-color: #1a1a1a;
}
body.dark-mode .testimonial-item {
    background-color: #222;
    color: #eee;
    border-color: #444;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
body.dark-mode .contact form {
    background-color: #333;
    color: #eee;
    border-color: #444;
    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.2);
    color: #fff;
}
body.dark-mode .contact input:focus,
body.dark-mode .contact textarea:focus {
    border-color: #80cbc4;
    background-color: #444;
    box-shadow: 0 3px 7px rgba(128, 203, 196, 0.2);
    color: #fff;
}
body.dark-mode .contact button {
    color: #fff;
}
```

```
body.dark-mode footer {
    background-color: #222;
    color: #eee;
    border-top-color: #333;
    box-shadow: 0 -2px 5px rgba(0, 0, 0, 0.2);
}
@media (max-width: 768px) {
    header nav ul {
        display: none;
        flex-direction: column;
        background-color: #000;
        position: absolute;
        top: 80px;
        left: 0;
        width: 100%;
        text-align: center;
        box-shadow: 0 6px 15px rgba(0, 0, 0, 0.5);
    }
    header nav ul.show {
        display: block;
    }
    header nav ul li {
        margin: 15px 0;
    }
    header nav a {
        padding: 12px 0;
    }
    .nav-toggle {
        display: block;
    }
    .hero {
      padding: 60px 20px;
      margin: 20px;
    }
    .services,
    .testimonials {
      padding: 40px 10px;
    }
    .service-item,
    .testimonial-item {
      margin: 20px auto;
      padding: 25px;
    }
    .contact form {
      padding: 30px;
      margin: 20px;
      gap: 20px;
      width: 95%;
    }
    .contact input,
    .contact textarea {
      font-size: 1em;
      padding: 12px;
      color: #fff;
    }
  }
  .contact form {
    display: grid; /* Changed to grid layout for better structure */
    grid-template-columns: 1fr; /* Single column by default */
    gap: 30px;
    max-width: 800px;
    margin: 60px auto;
    padding: 50px;
    background-color:rgba(0, 0, 0, 0.4) ;
    border-radius: 16px;
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
    border: 1px solid #333;
}
@media (min-width: 768px) {
    .contact form {
        grid-template-columns: repeat(2, 1fr); /* Two columns on larger screens */
    }
    .contact form > h1 { /* Ensure heading spans both columns */
        grid-column: 1 / -1;
    }
    .contact form > button { /* Ensure button spans both columns if needed, or adjust */
        grid-column: 1 / -1; /* Default to span both */
        width: fit-content; /* Keep button width to content */
        margin-left: auto;
        margin-right: auto;
    }
}
.contact form h1 {
    text-align: center;
    margin-bottom: 40px;
    color: #80cbc4;
    font-size: 36px;
    letter-spacing: 2px;
    font-weight: bold;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
}
.contact label {
    display: block;
    margin-top: 0; /* Reduced margin-top as grid provides spacing */
    color: #80cbc4;
    font-weight: bold;
    font-size: 1.1em; /* Slightly reduced label size */
    transition: color 0.3s ease;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.3);
}
.contact label:hover {
    color: #fff;
}
.contact input,
.contact textarea {
    padding: 15px; /* Slightly reduced padding */
    background-color: #333;
    color:#80cbc4 ;
    border: 1px solid #444;
    border-radius: 10px; /* Slightly less rounded */
    transition: background-color 0.3s ease, border-color 0.3s ease, box-shadow 0.3s ease;
    width: 100%;
    box-shadow: inset 0 3px 6px rgba(0, 0, 0, 0.2);
    font-size: 1em; /* Slightly reduced input size */
    color: #fff;
}
```

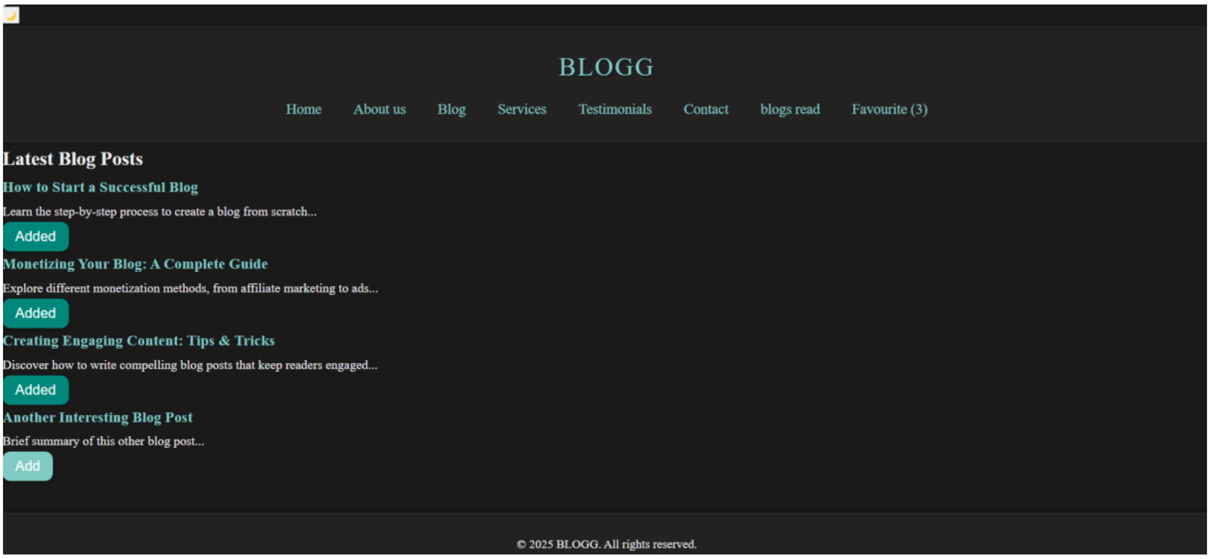
```
.contact input:focus,
.contact textarea:focus {
   background-color: #444;
   border-color: #80cbc4;
   outline: none;
   box-shadow: 0 4px 10px rgba(128, 203, 196, 0.3);
   color: #fff;
}
.contact textarea {
   resize: vertical;
   min-height: 150px; /* Adjusted textarea height */
   color: #eee;
   grid-column: 1 / -1; /* Make textarea span both columns on larger screens */
}
.contact button {
   padding: 15px 30px; /* Adjusted button padding */
   background-color: #80cbc4;
   color: #fff;
   border: none;
   border-radius: 10px; /* Slightly less rounded */
   cursor: pointer;
   transition: background-color 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;
   font-size: 1.1em; /* Adjusted button font size */
   width: fit-content;
   margin: 30px auto 0 auto;
   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
.contact button:hover {
   background-color: #62a39e;
```

```
    transform: translateY(-3px);
    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.4);
}
.contact button:focus {
    outline: none;
    box-shadow: 0 0 0 3px rgba(128, 203, 196, 0.5), 0 4px 10px rgba(0, 0, 0, 0.3);
}
/* General button */
.fav-button,
.remove-btn {
    padding: 8px 16px;
    border-radius: 10px;
    cursor: pointer;
    font-size: 1.1em;
    transition: background-color 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
    border: none;
    color: #fff;
    display: flex;
    align-items: center;
    gap: 8px;
}
.fav-button:hover,
.remove-btn:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
.fav-button:focus,
.remove-btn:focus {
    outline: none;
    box-shadow: 0 0 0 2px rgba(128, 203, 196, 0.5), 0 2px 5px rgba(0, 0, 0, 0.2);
}
.fav-button {
    background-color: #80cbc4;
}

.fav-button:hover {
    background-color: #62a39e;
}
.fav-button.active {
    background-color: #00897b;
    color: #fff;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

.remove-btn {
    background-color:#222 ;
    margin-left: 15px;
}
.remove-btn:hover {
    background-color:#62a39e;
}
/* Footer */
footer {
    background-color: #222;
    color: #ddd;
    text-align: center;
    padding: 25px;
    margin-top: 40px;
    border-top: 1px solid #333;
    box-shadow: 0 -2px 5px rgba(0, 0, 0, 0.2);
}

footer p {
    margin: 0;
    font-size: 0.95em;
}
/* Dark Mode Styles */
body.dark-mode {
    background-color: #1a1a1a;
    color: #eee;
}
body.dark-mode a {
    color: #80cbc4;
}
body.dark-mode a:hover {
    color: #a7f9ee;
}
body.dark-mode header {
    background-color: #222;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
    border-bottom: 1px solid #333;
}
body.dark-mode .hero {
    background-color: #2a2a2a;
    color: #eee;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
```

```
body.dark-mode .services {
   background-color: #1a1a1a;
}
body.dark-mode .service-item {
   background-color: #222;
   color: #eee;
   border-color: #333;
   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
body.dark-mode .testimonials {
   background-color: #1a1a1a;
}
body.dark-mode .testimonial-item {
   background-color: #222;
   color: #eee;
   border-color: #444;
   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
body.dark-mode .contact form {
   background-color: rgba(0, 0, 0, 0.3)
}
```

<u>Output</u>:

*Home page:



BLOGG

Home About us Blog Services Testimonials Contact blogs read Favourite (3)

Latest Blog Posts

How to Start a Successful Blog

Learn the step-by-step process to create a blog from scratch...

Added

Monetizing Your Blog: A Complete Guide

Explore different monetization methods, from affiliate marketing to ads...

Added

Creating Engaging Content: Tips & Tricks

Discover how to write compelling blog posts that keep readers engaged...

Added

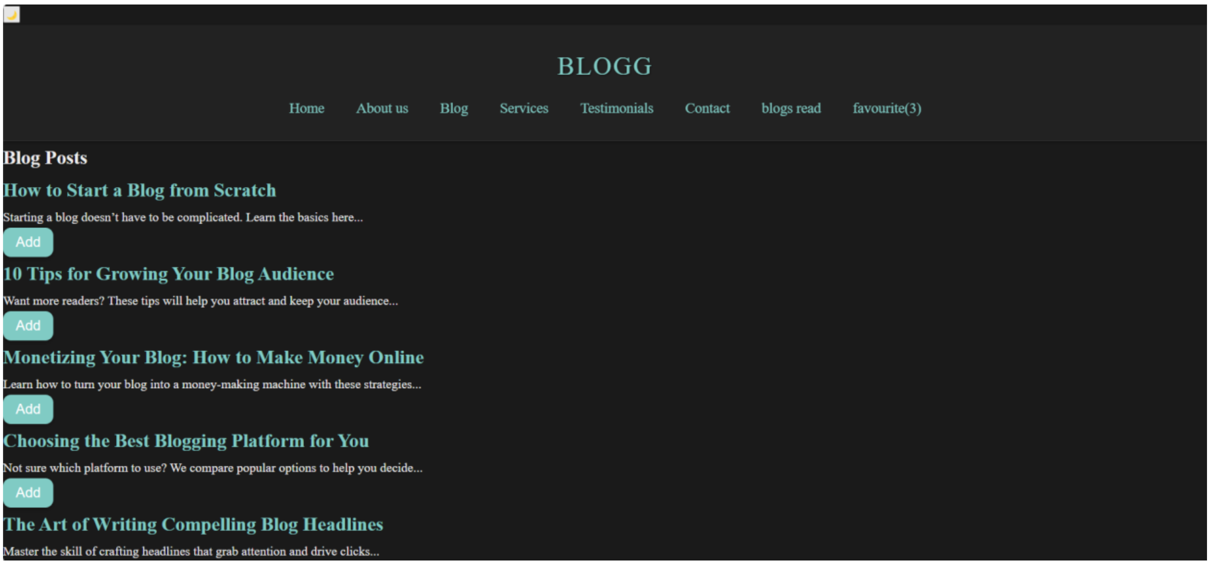
Another Interesting Blog Post

Brief summary of this other blog post...

Add

© 2025 BLOGG. All rights reserved.

*Blog / product page :



BLOGG

Home About us Blog Services Testimonials Contact blogs read favourite(3)

Blog Posts

How to Start a Blog from Scratch

Starting a blog doesn't have to be complicated. Learn the basics here...

Add

10 Tips for Growing Your Blog Audience

Want more readers? These tips will help you attract and keep your audience...

Add

Monetizing Your Blog: How to Make Money Online

Learn how to turn your blog into a money-making machine with these strategies...

Add

Choosing the Best Blogging Platform for You

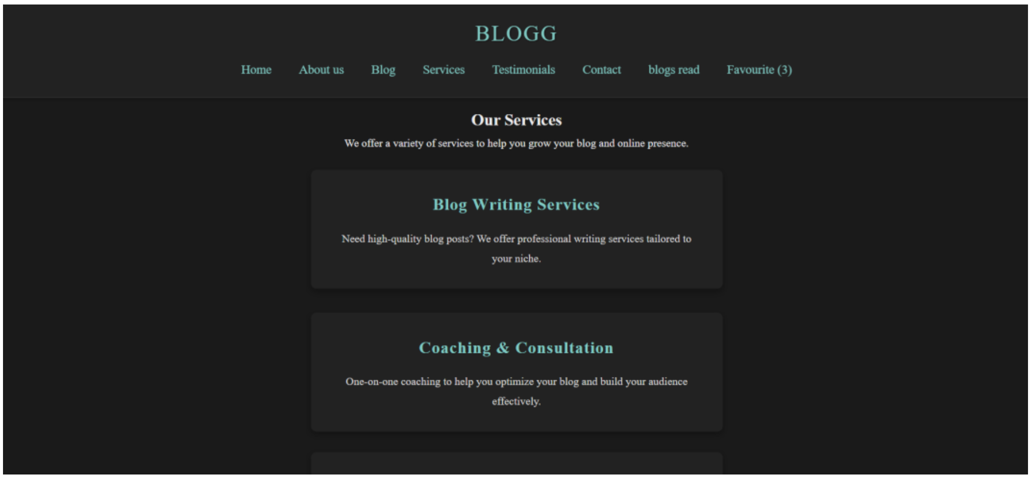
Not sure which platform to use? We compare popular options to help you decide...

Add

The Art of Writing Compelling Blog Headlines

Master the skill of crafting headlines that grab attention and drive clicks...

*Services page:



BLOGG

Home About us Blog Services Testimonials Contact blogs read Favourite (3)

Our Services

We offer a variety of services to help you grow your blog and online presence.

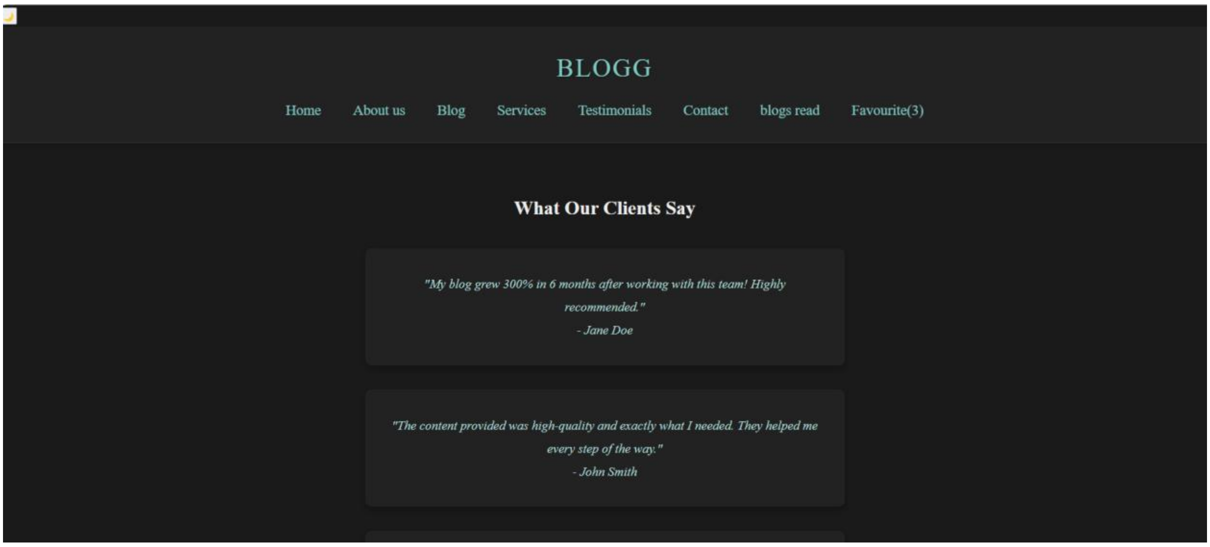
Blog Writing Services

Need high-quality blog posts? We offer professional writing services tailored to your niche.

Coaching & Consultation

One-on-one coaching to help you optimize your blog and build your audience effectively.

*Testimonials page:



BLOGG

What Our Clients Say

"My blog grew 300% in 6 months after working with this team! Highly recommended."
- Jane Doe

"The content provided was high-quality and exactly what I needed. They helped me every step of the way."
- John Smith

Conclusion

Implementing **CSS Grid** for the BLOGG platform offers a powerful and clean way to organize content visually. It enhances layout flexibility, ensures responsiveness, and provides a better **user experience** across devices. Whether it's displaying blog posts, testimonials, or categories, CSS Grid allows developers to build scalable and intuitive designs—making BLOGG a professional, easy-to-navigate blogging space for both writers and readers.

Experiment No.4

Problem statement:

CSS

A. Enhance the cart page to make it user-friendly and visually appealing. Style the cart items with appropriate margins, paddings, and input field styles to provide a seamless shopping experience.

B. Enhance and style the about us page with appropriate margins, paddings, and input field styles.

C. Enhance and style the contact page to make it user-friendly and visually appealing. Style the contact form with appropriate margins, paddings, and input field styles.

D. Enhance and style the admin/user registration form with appropriate margins, paddings, and input field styles.

E. Enhance and style the admin/user login form with appropriate margins, paddings, and input field styles.

Theory: Enhancing and Styling Key Pages in the BLOGG Platform

---

1. Why CSS Styling Matters in Blogging Platforms

First impressions are everything. For a blog website like **BLOGG**, clean layout, good spacing, and appealing visuals enhance readability and keep users engaged. Whether it's a “Favorites” page, About, Contact, or Login form, CSS styling:

- Improves reading comfort
- Creates visual hierarchy and structure
- Enhances accessibility and navigation
- Increases user trust and time-on-site

Page-wise CSS Styling Theory

---

1. Favourite / Saved Blogs Page (Cart Equivalent)

This page lets users revisit their bookmarked or liked posts.

Key Styling Techniques:

- Add spacing and margin around each saved blog card
- Include author, date, and excerpt neatly arranged with paddings

- Use “Remove” or “Unfavourite” buttons with hover effects
- Highlight blog titles using bold fonts
- Use card-style layout with subtle shadows for clarity

Result: A clean, well-spaced section that feels curated and easy to browse.

---

2. About Us Page

This page builds the platform’s story, values, and creator background.

Key Styling Techniques:

- Use increased line height and padding for paragraphs
- Space out segments like “Our Story,” “Team,” “Vision”
- Use grid or flexbox to align team photos side by side
- Round team photos and include hover text or tooltips
- Style quote blocks or values using soft boxes or colored sections

Result: A friendly, professional layout that increases connection and credibility.

---

3. Contact Page

This form allows users or collaborators to get in touch.

Key Styling Techniques:

- Equal width and height for all input fields
- Apply padding inside text areas for better typing experience
- Style form labels clearly, and highlight inputs on focus
- Use box-shadow and rounded corners for the form container
- Include button hover states and central alignment

Result: A user-friendly form that invites genuine interaction and feedback.

---

4. User/Admin Registration Form

Used for new blog creators or site administrators.

Key Styling Techniques:

- Group inputs into clear sections: Personal Info, Login Credentials
- Use descriptive placeholders and labels
- Maintain equal spacing and alignment
- Add form card background color, box shadow, and rounded borders
- Display real-time validation (e.g., password strength, email check)

Result: A trust-building form that looks secure and feels intuitive to fill.

5. User/Admin Login Form

The entry point for users to access or manage the platform.

Key Styling Techniques:

- Centrally position the login box on screen
- Provide enough padding between fields
- Style error messages clearly (red for error, green for success)
- Use minimalist fonts, slight shadow, and clean input boxes
- Add “Show Password” toggle and forgot password link

Result: A quick, effective login interface that builds user confidence.

Bonus: Styling Tips Across All Pages

- Use variables for consistent colors (`--primary-color`, `--bg-color`, etc.)
- Choose clean typography like `Inter`, `Open Sans`, or `Playfair Display`
- Responsive layouts with grid or flexbox for mobile-first design
- Use light/dark mode CSS themes to suit user preferences

Code:

```
/* Reset default margin and padding */

* {

   margin: 0;

   padding: 0;

   box-sizing: border-box;

}



/* General Styles */

body {

   font-family: 'serif', serif;

   background-color: #1a1a1a;

   color: #eee;

   line-height: 1.7;
```

```
}

a {
    color: #80cbc4;
    text-decoration: none;
    transition: color 0.3s ease;
}

a:hover {
    color: #a7f9ee;
}

/* Header */
header {
    background-color: #222;
    padding: 25px 0;
    text-align: center;
    position: sticky;
    top: 0;
    z-index: 100;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
    border-bottom: 1px solid #333;
}

header .logo {
    font-size: 32px;
    color: #fff;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.5);
    letter-spacing: 2px;
```

```
}

header nav ul {
    list-style-type: none;
    display: flex;
    justify-content: center;
    gap: 40px;
    margin-top: 10px;
}

header nav a {
    color: #eee;
    font-size: 18px;
    position: relative;
    padding-bottom: 8px;
    transition: color 0.3s ease;
}

header nav a:hover {
    color: #fff;
}

header nav a:hover::after {
    content: '';
    position: absolute;
    left: 10%;
    bottom: 0;
    width: 80%;
    height: 2px;
    background-color: #80cbc4;
    transform: scaleX(0);
    transition: transform 0.3s ease-in-out;
}

header nav a:hover::after {
    transform: scaleX(1);
}

/* Mobile Navigation Toggle */
.nav-toggle {
    display: none;
    background-color: transparent;
    border: 1px solid #eee;
    color: #eee;
    font-size: 28px;
    cursor: pointer;
    padding: 8px;
    position: absolute;
    left: 10px;
    top: 10px;
    transition: border-color 0.3s ease, color 0.3s ease;
}

.nav-toggle:hover {
    border-color: #fff;
    color: #fff;
}
```

```
.nav-toggle:focus {
    outline: none;
    box-shadow: 0 0 0 2px rgba(128, 203, 196, 0.5);
}

header nav ul.show {
    display: block;
}

/* Hero Section */
.hero {
    background-color: #2a2a2a;
    color: #eee;
    padding: 80px 20px;
    text-align: center;
    border-radius: 8px;
    margin: 30px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
    animation: fadeIn 1s ease;
}

@keyframes fadeIn {
    from {
        opacity: 0;
        transform: translateY(20px);
    }

    to {
        opacity: 1;
        transform: translateY(0);
    }
}

.hero h1 {
    font-size: 2.5em;
    font-weight: bold;
    letter-spacing: 2px;
    color: #fff;
}

.hero p {
    font-size: 1.1em;
    color: #ddd;
    line-height: 1.8;
    margin-top: 20px;
}

/* Services Section */
.services {
    padding: 60px 20px;
    text-align: center;
    background-color: #1a1a1a;
}

.service-item {
    background-color: #222;
    border-radius: 8px;
    margin: 30px auto;
    padding: 30px;
    max-width: 600px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.service-item:hover {
    transform: translateY(-5px);
    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.4);
}

.service-item h2 {
    color: #80cbc4;
    margin-bottom: 15px;
    font-size: 24px;
    letter-spacing: 1px;
}

.service-item p {
    color: #ddd;
    line-height: 1.8;
}

/* Testimonials Section */
.testimonials {
    padding: 60px 20px;
    background-color: #1a1a1a;
    text-align: center;
}
```

```
.testimonial-item {
    background-color: #222;
    border-radius: 8px;
    margin: 30px auto;
    padding: 30px;
    max-width: 600px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
    font-style: italic;
    color: #eee;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.testimonial-item:hover {
    transform: translateY(-3px);
    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.4);
}

.testimonial-item p {
    font-style: italic;
    color: #b2dfdb;
    line-height: 1.8;
}


@media (min-width: 768px) {
    .contact form {
        grid-template-columns: repeat(2, 1fr);
    }
    .contact form > h1 {
        grid-column: 1 / -1;
    }

    .contact form > button {
        grid-column: 1 / -1;
        width: fit-content;
        margin-left: auto;
        margin-right: auto;
    }
}

.contact form h1 {
    text-align: center;
    margin-bottom: 40px;
    color: #333; /* Changed heading color to dark gray for white background */
    font-size: 36px;
    letter-spacing: 2px;
    font-weight: bold;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.1); /* Adjusted shadow for light background */
}

.contact label {
    display: block;
    margin-top: 0;
    color: #555; /* Changed label color to dark gray */
    font-weight: bold;
    font-size: 1.1em;
    transition: color 0.3s ease;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.1); /* Adjusted shadow */
}

.contact label:hover {
    color: #333; /* Changed hover color */
}

.contact input,
.contact textarea {
    padding: 15px;
    background-color: #f5f5f5; /* Light gray for input fields */
    color: #333;
    border: 1px solid #ccc; /* Lighter border */
    border-radius: 10px;
    transition: background-color 0.3s ease, border-color 0.3s ease, box-shadow 0.3s ease;
    width: 100%;
    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.1); /* Subtle inset shadow */
    font-size: 1em;
}

.contact input:focus,
.contact textarea:focus {
    background-color: #fff;
    border-color: rgba(0, 0, 0, 0.3);
    outline: none;
    box-shadow: 0 4px 10px rgba(128, 203, 196, 0.3);
    color: #333;
}
```

```
.contact textarea {
    resize: vertical;
    min-height: 150px;
    color: #333;
    grid-column: 1 / -1;
}

.contact button {
    padding: 15px 30px;
    background-color: #80cbc4;
    color: #fff;
    border: none;
    border-radius: 10px;
    cursor: pointer;
    transition: background-color 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;
    font-size: 1.1em;
    width: fit-content;
    margin: 30px auto 0 auto;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1); /* Reduced shadow for light background */
}

.contact button:hover {
    background-color: #62a39e;
    transform: translateY(-3px);
    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.2); /* Reduced shadow on hover */
}

.contact button:focus {
    outline: none;
    box-shadow: 0 0 0 3px rgba(128, 203, 196, 0.5), 0 4px 10px rgba(0, 0, 0, 0.1); /* Reduced shadow on focus */
}

/* ... (rest of your CSS remains the same) ... */

/* General button */
.fav-button,
.remove-btn {
    padding: 12px 20px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 1.1em;
    transition: background-color 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
    border: none;
    color: #fff;
    display: flex;
    align-items: center;
    gap: 8px;
}

.fav-button:hover,
.remove-btn:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
```

```
.fav-button:focus,
.remove-btn:focus {
   outline: none;
   box-shadow: 0 0 0 2px rgba(128, 203, 196, 0.5), 0 2px 5px rgba(0, 0, 0, 0.2);
}

.fav-button {
   background-color: #80cbc4;
}

.fav-button:hover {
   background-color: #62a39e;
}

.fav-button.active {
   background-color: #00897b;
   color: #fff;
   box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
}

.remove-btn {
   background-color: #e53e3e;
   margin-left: 15px;
}

.remove-btn:hover {
   background-color: #c53030;
}
```

```
}
```

```
/* Footer */
footer {
    background-color: #222;
    color: #ddd;
    text-align: center;
    padding: 25px;
    margin-top: 40px;
    border-top: 1px solid #333;
    box-shadow: 0 -2px 5px rgba(0, 0, 0, 0.2);
}

footer p {
    margin: 0;
    font-size: 0.95em;
}

/* Dark Mode Styles */
body.dark-mode {
    background-color: #1a1a1a;
    color: #eee;
}

body.dark-mode a {
    color: #80cbc4;
}

body.dark-mode a:hover {
    color: #a7f9ee;
}
```

```
body.dark-mode header {

    background-color: #222;

    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

    border-bottom: 1px solid #333;

}


body.dark-mode .hero {

    background-color: #2a2a2a;

    color: #eee;

    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);

}


body.dark-mode .services {

    background-color: #1a1a1a;

}


body.dark-mode .service-item {

    background-color: #222;

    color: #eee;

    border-color: #333;

    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);

}


body.dark-mode .testimonials {

    background-color: #1a1a1a;

}


body.dark-mode .testimonial-item {
    background-color: #222;
    color: #eee;
    border-color: #444;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}

body.dark-mode .contact form {
    background-color: #333;
    color: #eee;
    border-color: #444;
    box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.2);
    color: #fff;
}

body.dark-mode .contact input:focus,
body.dark-mode .contact textarea:focus {
    border-color: #80cbc4;
    background-color: #444;
    box-shadow: 0 3px 7px rgba(128, 203, 196, 0.2);
    color: #fff;
}

body.dark-mode .contact button {
    color: #fff;
}

body.dark-mode footer {
    background-color: #222;
    color: #eee;
    border-top-color: #333;
    box-shadow: 0 -2px 5px rgba(0, 0, 0, 0.2);
}

/* Responsive Design */
@media (max-width: 768px) {
    header nav ul {
        display: none;
        flex-direction: column;
        background-color: #000;
        position: absolute;
        top: 80px;
        left: 0;
        width: 100%;
        text-align: center;
        box-shadow: 0 6px 15px rgba(0, 0, 0, 0.5);
    }

    header nav ul.show {
        display: block;
    }

    header nav ul li {
        margin: 15px 0;
    }

    header nav a {
        padding: 12px 0;
    }
```

```
.nav-toggle {
   display: block;
}

.hero {
   padding: 60px 20px;
   margin: 20px;
}

.services,
.testimonials {
   padding: 40px 10px;
}

.service-item,
.testimonial-item {
   margin: 20px auto;
   padding: 25px;
}

.contact form {
   padding: 30px;
   margin: 20px;
   gap: 20px;
   width: 95%;
}

.contact input,
    .contact textarea {
        font-size: 1em;
        padding: 12px;
        color: #fff;
    }
}

/* Improved Contact Form Styles */
/* ... (rest of your CSS remains the same) ... */

/* Improved Contact Form Styles */
.contact form {
    display: grid; /* Changed to grid layout for better structure */
    grid-template-columns: 1fr; /* Single column by default */
    gap: 30px;
    max-width: 800px;
    margin: 60px auto;
    padding: 50px;
    background-color:rgba(0, 0, 0, 0.4) ;
    border-radius: 16px;
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
    border: 1px solid #333;
}

@media (min-width: 768px) {
    .contact form {
        grid-template-columns: repeat(2, 1fr); /* Two columns on larger screens */
    }
    .contact form > h1 { /* Ensure heading spans both columns */
        grid-column: 1 / -1;
    }

    .contact form > button { /* Ensure button spans both columns if needed, or adjust */
        grid-column: 1 / -1; /* Default to span both */
        width: fit-content; /* Keep button width to content */
        margin-left: auto;
        margin-right: auto;
    }
}

.contact form h1 {
    text-align: center;
    margin-bottom: 40px;
    color: #80cbc4;
    font-size: 36px;
    letter-spacing: 2px;
    font-weight: bold;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
}

.contact label {
    display: block;
    margin-top: 0; /* Reduced margin-top as grid provides spacing */
    color: #80cbc4;
    font-weight: bold;
    font-size: 1.1em; /* Slightly reduced label size */
    transition: color 0.3s ease;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.3);
}

.contact label:hover {
    color: #fff;
}

.contact input,
.contact textarea {
    padding: 15px; /* Slightly reduced padding */
    background-color: #333;
    color:#80cbc4 ;
    border: 1px solid #444;
    border-radius: 10px; /* Slightly less rounded */
    transition: background-color 0.3s ease, border-color 0.3s ease, box-shadow 0.3s ease;
    width: 100%;
    box-shadow: inset 0 3px 6px rgba(0, 0, 0, 0.2);
    font-size: 1em; /* Slightly reduced input size */
    color: #fff;
}

.contact input:focus,
.contact textarea:focus {
    background-color: #444;
    border-color: #80cbc4;
    outline: none;
    box-shadow: 0 4px 10px rgba(128, 203, 196, 0.3);
    color: #fff;
}
```

```
.contact textarea {
   resize: vertical;
   min-height: 150px; /* Adjusted textarea height */
   color: #eee;
   grid-column: 1 / -1; /* Make textarea span both columns on larger screens */
}

.contact button {
   padding: 15px 30px; /* Adjusted button padding */
   background-color: #80cbc4;
   color: #fff;
   border: none;
   border-radius: 10px; /* Slightly less rounded */
   cursor: pointer;
   transition: background-color 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;
   font-size: 1.1em; /* Adjusted button font size */
   width: fit-content;
   margin: 30px auto 0 auto;
   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}

.contact button:hover {
   background-color: #62a39e;
   transform: translateY(-3px);
   box-shadow: 0 6px 15px rgba(0, 0, 0, 0.4);
}

.contact button:focus {
   outline: none;

   box-shadow: 0 0 0 3px rgba(128, 203, 196, 0.5), 0 4px 10px rgba(0, 0, 0, 0.3);

}


/* ... (rest of your CSS remains the same) ... */


/* General button */

.fav-button,

.remove-btn {

   padding: 8px 16px;

   border-radius: 10px;

   cursor: pointer;

   font-size: 1.1em;

   transition: background-color 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;

   box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

   border: none;

   color: #fff;

   display: flex;

   align-items: center;

   gap: 8px;

}


.fav-button:hover,

.remove-btn:hover {

   transform: translateY(-2px);

   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);

}


.fav-button:focus,
.remove-btn:focus {

   outline: none;

   box-shadow: 0 0 0 2px rgba(128, 203, 196, 0.5), 0 2px 5px rgba(0, 0, 0, 0.2);

}


.fav-button {

   background-color: #80cbc4;

}


.fav-button:hover {

   background-color: #62a39e;

}


.fav-button.active {

   background-color: #00897b;

   color: #fff;

   box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

}


.remove-btn {

   background-color:#222 ;

   margin-left: 15px;

}


.remove-btn:hover {

   background-color:#62a39e;

}


/* Footer */
```

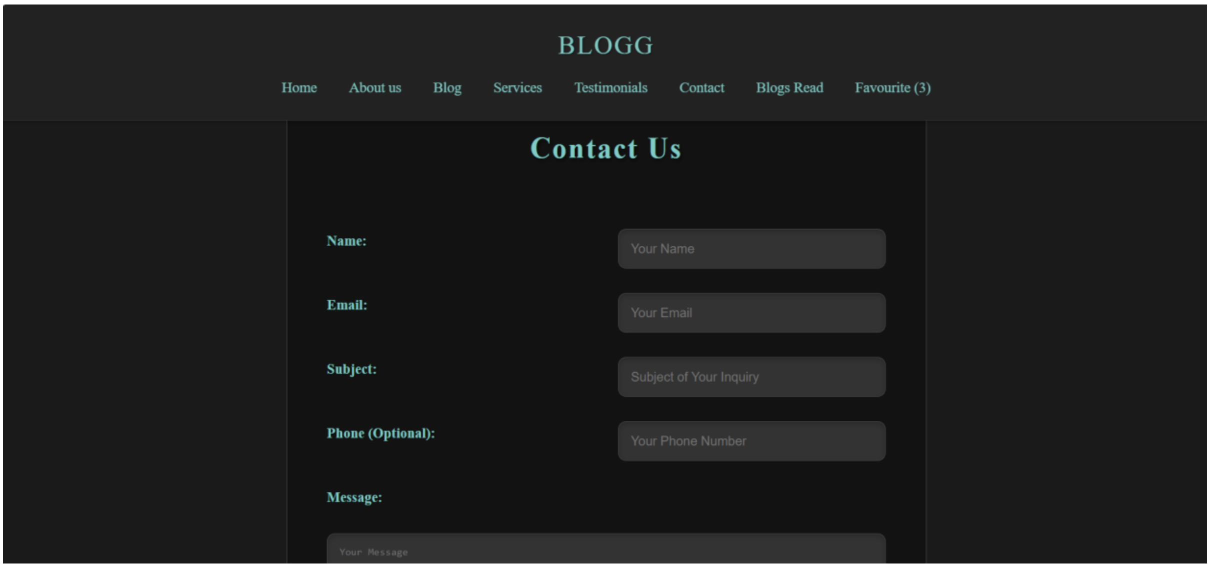
```
footer {
    background-color: #222;
    color: #ddd;
    text-align: center;
    padding: 25px;
    margin-top: 40px;
    border-top: 1px solid #333;
    box-shadow: 0 -2px 5px rgba(0, 0, 0, 0.2);
}

footer p {
    margin: 0;
    font-size: 0.95em;
}

/* Dark Mode Styles */
body.dark-mode {
    background-color: #1a1a1a;
    color: #eee;
}

body.dark-mode a {
    color: #80cbc4;
}

body.dark-mode a:hover {
    color: #a7f9ee;
}
```

```
body.dark-mode header {
   background-color: #222;
   box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
   border-bottom: 1px solid #333;
}

body.dark-mode .hero {
   background-color: #2a2a2a;
   color: #eee;
   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}

body.dark-mode .services {
   background-color: #1a1a1a;
}

body.dark-mode .service-item {
   background-color: #222;
   color: #eee;
   border-color: #333;
   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}

body.dark-mode .testimonials {
   background-color: #1a1a1a;
}

body.dark-mode .testimonial-item {
   background-color: #222;
```

```
    color: #eee;

    border-color: #444;

    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);

}


body.dark-mode .contact form {

    background-color: rgba(0, 0, 0, 0.3)

}
```

Output:

Contact us page:



Contact Us

Name: Your Name

Email: Your Email

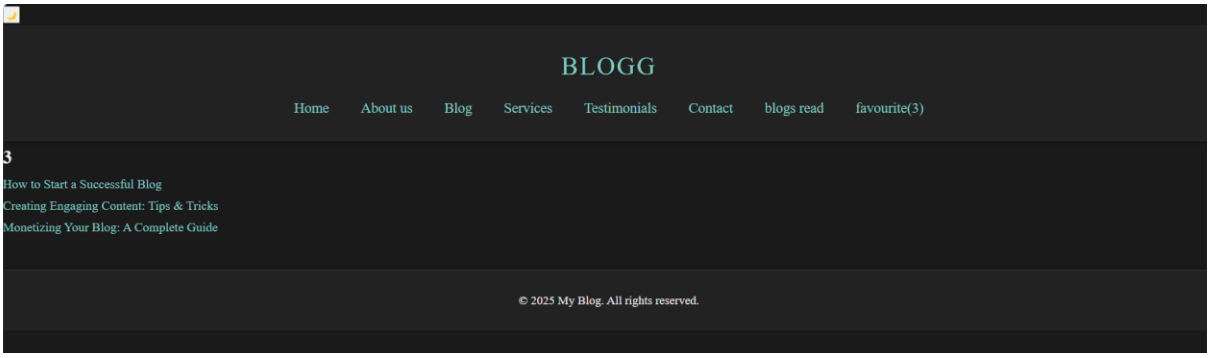
Subject: Subject of Your Inquiry

Phone (Optional): Your Phone Number

Message:

Your Message

*Blogs read / count page:



3

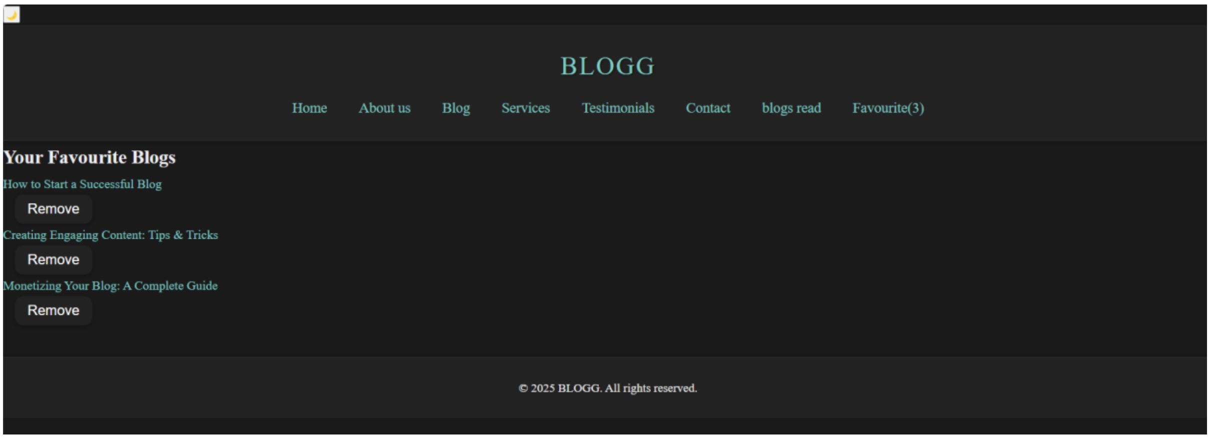
How to Start a Successful Blog

Creating Engaging Content: Tips & Tricks

Monetizing Your Blog: A Complete Guide

© 2025 My Blog. All rights reserved.

*Favourite / cart page:



Your Favourite Blogs

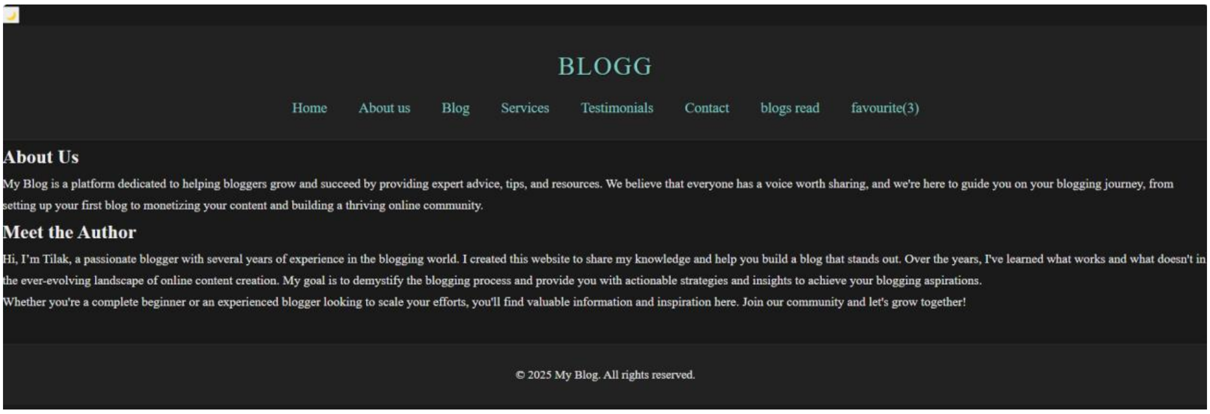
How to Start a Successful Blog

Creating Engaging Content: Tips & Tricks

Monetizing Your Blog: A Complete Guide

© 2025 BLOGG. All rights reserved.

*About us page:



About Us

My Blog is a platform dedicated to helping bloggers grow and succeed by providing expert advice, tips, and resources. We believe that everyone has a voice worth sharing, and we're here to guide you on your blogging journey, from setting up your first blog to monetizing your content and building a thriving online community.

Meet the Author

Hi, I'm Tilak, a passionate blogger with several years of experience in the blogging world. I created this website to share my knowledge and help you build a blog that stands out. Over the years, I've learned what works and what doesn't in the ever-evolving landscape of online content creation. My goal is to demystify the blogging process and provide you with actionable strategies and insights to achieve your blogging aspirations.

Whether you're a complete beginner or an experienced blogger looking to scale your efforts, you'll find valuable information and inspiration here. Join our community and let's grow together!

© 2025 My Blog. All rights reserved.

Conclusion

The success of a blogging platform like **BLOGG** doesn't rely solely on content—it also depends on **design, structure, and user experience**. With the help of **CSS Grid and modern styling techniques**, we can transform simple web pages into **visually appealing, well-structured, and highly responsive layouts**.

By styling key pages—like the Favourite Blogs, About Us, Contact, Registration, and Login forms—we improve **usability, accessibility, and trust**. From clear spacing and input feedback to responsive grids and clean typography, each CSS enhancement contributes to a **professional and engaging blogging experience**.

Ultimately, thoughtful CSS design is not just about looks—it's about making the site **welcoming, functional, and enjoyable** for every user.

Experiment No. 5

Problem statement:

JavaScript

A. Implement user registration and login forms for the blog platform website. These forms will allow users to create an account, log in, and access personalized features, such as saving favorite items or viewing order history.

User Registration Form will allow new customers to sign up and create an account on the website. The form will capture basic user details, including the name, email address, and password (not limited to these fields).

User Login Form will allow registered users to log into their accounts. The form will require an email address and a password to authenticate the user.

B. Provide validations for user registration and login forms to validate the input to ensure that all required fields are filled and that the email format is valid. (**Contents beyond Syllabus)**

B. Develop cart functionality to allow users to add items, update quantities, and remove items

Theory: User Registration, Login, Validation, and Bookmarking Functionality for BLOGG – Blogging Platform

Introduction

In the digital age, a blogging platform like BLOGG thrives on user interaction, personalized content access, and a smooth user experience. JavaScript is crucial for enhancing client-side functionality like **registration**, **login**, **form validation**, and **bookmarking blog posts**, making the platform more dynamic and user-friendly.

Key Features of JavaScript

- **Lightweight and fast**
- **Event-driven: Responds to user actions like clicks and keypresses**
- **Object-oriented: Uses objects to store and manage data**
- **First-class functions: Functions can be treated like variables**
- **Asynchronous support: Promises, async/await for non-blocking tasks**

JavaScript in the BLOGG Project

In the BLOGG blogging platform, JavaScript is used to:

- **Handle the dark mode toggle**
- **Add interactivity (e.g., Add to Favourites button)**
- **Validate and submit the contact form**

- **Dynamically update UI components**
- **Use AJAX to send form data to the backend (PHP + MySQL)**

JavaScript Examples from BLOGG

Dark Mode Toggle:

javascript

CopyEdit

```
const toggle = document.getElementById("darkModeToggle");


toggle.addEventListener("click", () => {
  document.body.classList.toggle("dark-mode");
});
```

Form Submission with Validation:

javascript

CopyEdit

```
document.getElementById("contactForm").addEventListener("submit", function(e) {
  e.preventDefault(); // Prevent default submission


  const name = document.getElementById("name").value;
  if (name.trim() === "") {
    alert("Please enter your name.");
    return;
  }


  // AJAX logic to send data to PHP
});
```

JavaScript Core Concepts

Concept	Description
Variables	Use let, const, or var to store data
Functions	Block of code to perform a task
DOM Manipulation	Access and change HTML elements
Events	Handle user actions (click, submit, hover)
Loops	Repeat tasks (for, while)
Conditions	if, else, switch for decision-making
Objects/Arrays	Store and organize data efficiently
ES6 Features	Modern JS syntax like arrow functions, classes, template literals

---

Advantages of JavaScript

- **Enhances user experience with real-time interactions**
- **Enables rich web applications without reloading**
- **Supported by all modern browsers**
- **Fast execution due to just-in-time (JIT) compilation**
- **Versatile: Runs on both client-side and server-side (Node.js)**

Limitations of JavaScript

- **Runs on the client-side, so it can be disabled by the user**
- **May have browser compatibility issues if not coded properly**
- **Security concerns (e.g., XSS) if user input is not sanitized**

🔧 Common JavaScript Tools/Libraries

- **jQuery – Simplifies DOM operations and AJAX**
- **React – Build dynamic UIs with reusable components**
- **Node.js – Server-side JavaScript execution**

- **Vanilla JS – Using pure JavaScript without frameworks**

JavaScript & Modern Web Development

JavaScript is essential for modern web applications. With frameworks like React, Vue, and Angular, and tools like Webpack, Babel, and ESLint, JavaScript powers highly interactive and complex web platforms used daily across the globe.

1. User Registration and Login Forms

These forms help establish user identity and enable personal dashboards for readers or bloggers.

Registration Form

The registration form collects data such as name, email, and password to create a new user account.

JavaScript Functions in Registration:

- Ensures no input fields are left blank
- Validates email format using RegEx
- Verifies password strength and matches with confirm password
- Displays real-time error messages and hints

Login Form

Enables users to access their accounts securely.

JavaScript Functions in Login:

- Checks for empty fields
- Validates input credentials
- Provides error messages for failed attempts
- Redirects to dashboard or blog list on success

---

2. JavaScript Form Validations

Form validation is essential to prevent incorrect or incomplete entries.

Common Validations:

- Required fields check
- Email format validation
- Password criteria (length, characters)
- Confirm password match

- Displaying in-line feedback messages

Client-side validation enhances UX by offering immediate feedback, reducing server load, and maintaining data integrity.

---

3. Bookmark / Favourite Blog Post Feature

Instead of a shopping cart, BLOGG includes a **“Add to Favourite”** or bookmark system to save posts for later reading.

JavaScript Responsibilities:

- Clicking the favourite icon toggles saved state
- Adds blog ID or title to a local storage array
- Displays saved posts in a “My Favourites” page
- Allows removal from favourites dynamically

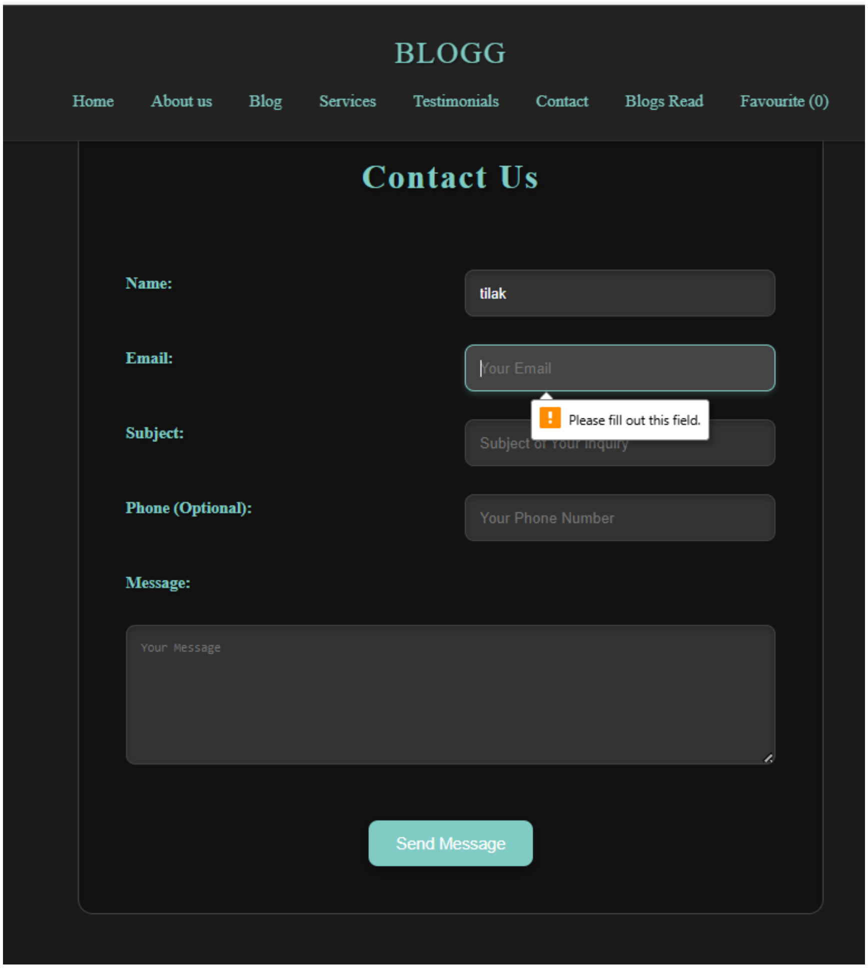
Code:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Contact Us | My Blog</title>

  <link rel="stylesheet" href="design.css">

  <script src="web.js" defer></script>

</head>

<body class="dark-mode">

  <header>

    <div class="logo"><a href="/">BLOGG</a></div>

    <nav>

      <ul>

        <li><a href="home.html">Home</a></li>

        <li><a href="about.html">About us</a></li>
```

```
                <li><a href="blog.html">Blog</a></li>
                <li><a href="services.html">Services</a></li>
                <li><a href="Review.html">Testimonials</a></li>
                <li><a href="contact.html">Contact</a></li>
                <li><a href="favourite.html">Blogs Read</a></li>
                <li><a href="view_favourites.html">Favourite (<span id="favorites-count">0</span>)</a></li>
            </ul>
        </nav>
    </header>

    <main class="contact">
        <section class="contact-form">
            <form id="contact-form" method="post">
                <h1>Contact Us</h1>

                <label for="name">Name:</label>
                <input type="text" id="name" name="name" placeholder="Your Name" required>

                <label for="email">Email:</label>
                <input type="email" id="email" name="email" placeholder="Your Email" required>

                <label for="subject">Subject:</label>
                <input type="text" id="subject" name="subject" placeholder="Subject of Your Inquiry"
required>

                <label for="phone">Phone (Optional):</label>
                <input type="tel" id="phone" name="phone" placeholder="Your Phone Number">

                <label for="message">Message:</label>
                <textarea id="message" name="message" placeholder="Your Message" required></textarea>


                <button type="submit">Send Message</button>
                <form id="contact-form" method="POST" action="save-form.php">


            </form>


            <!-- Success Message -->
            <p id="success-message" style="display: none; color: green; font-weight: bold;">
                ✅ Your message has been sent successfully!
            </p>
        </section>
    </main>


    <footer>
        <p>&copy; 2025 My Blog | All rights reserved.</p>
    </footer>
</body>
</html>


// Dark Mode Toggle
const darkModeToggle = document.createElement("button");
darkModeToggle.textContent = "🌙";
darkModeToggle.classList.add("dark-mode-toggle");
document.body.prepend(darkModeToggle);


darkModeToggle.addEventListener("click", function () {
    document.body.classList.toggle("dark-mode");
    darkModeToggle.textContent = document.body.classList.contains("dark-mode") ? "☀" : "🌙";
    localStorage.setItem("dark-mode", document.body.classList.contains("dark-mode"));
});

// Load dark mode preference
if (localStorage.getItem("dark-mode") === "true") {
    document.body.classList.add("dark-mode");
    darkModeToggle.textContent = "☀";
}

// Favorite Blogs - Add & Display
function getFavorites() {
    const storedFavorites = localStorage.getItem("favorites");
    return storedFavorites ? JSON.parse(storedFavorites) : [];
}

function saveFavorites(favorites) {
    localStorage.setItem("favorites", JSON.stringify(favorites));
    updateFavoritesCount();
}

function updateFavoritesCount() {
    const count = getFavorites().length;
    document.querySelectorAll("#favorites-count").forEach(el => {
        el.textContent = count;
    });
}
```

```
function updateFavButtonUI(button, isFavorite) {

    if (button) {

        button.classList.toggle('active', isFavorite);

        button.textContent = isFavorite ? 'Added' : 'Add';

    }

}


function addToFavorites(title, link, buttonElement) {

    let favorites = getFavorites();

    const existingIndex = favorites.findIndex(item => item.title === title);

    if (existingIndex === -1) {

        favorites.push({ title, link });

        saveFavorites(favorites);

        updateFavButtonUI(buttonElement, true);

        alert("Added to favorites!");

    } else {

        favorites.splice(existingIndex, 1);

        saveFavorites(favorites);

        updateFavButtonUI(buttonElement, false);

        alert("Removed from favorites!");

    }

    displayFavorites();

}


// Event listener for favorite buttons

document.addEventListener('click', function (event) {

    if (event.target.classList.contains('fav-button')) {

        const button = event.target;

        const title = button.dataset.title;
        const link = button.dataset.link;
        addToFavorites(title, link, button);
    }
});


// Initialize favorite buttons on page load
document.querySelectorAll(".fav-button").forEach(button => {
    const title = button.dataset.title;
    updateFavButtonUI(button, getFavorites().some(item => item.title === title));
});
function displayFavorites() {
    const favoritesList = document.getElementById("favorites-list");
    const emptyMessage = document.getElementById("empty-message");
    const isFavouritePage = window.location.pathname.includes("favourite.html"); // Check if it's the 'blogs
read' page


    if (!favoritesList) return console.log("favorites-list element not found!");


    favoritesList.innerHTML = "";
    const favorites = getFavorites();


    console.log("Favorites retrieved:", favorites);


    if (favorites.length === 0) {
        emptyMessage.style.display = 'block';
        favoritesList.style.display = 'none';
        return;
    } else {
        emptyMessage.style.display = 'none';
        favoritesList.style.display = 'block';
    }

    favorites.forEach((item) => {
        const li = document.createElement("li");
        li.innerHTML = `
            <a href="${item.link || "#"}" target="_blank">${item.title || "Untitled"}</a>
            ${!isFavouritePage ? `<button class="remove-btn" data-title="${item.title}">Remove</button>` :
""}
        `;
        favoritesList.appendChild(li);
    });
}

// Event listener for removing from favorites
document.addEventListener('click', function (event) {
    if (event.target.classList.contains('remove-btn')) {
        removeFromFavorites(event.target.dataset.title);
    }
});
function removeFromFavorites(title) {
    saveFavorites(getFavorites().filter(item => item.title !== title));
    displayFavorites();
}
// Ensure favorites count updates when the page loads
updateFavoritesCount();

// Display favorites if on the view_favourites.html page
if (window.location.pathname.includes('view_favourites.html')) {
    document.addEventListener("DOMContentLoaded", displayFavorites);
}
// Display favorites if on the view_favourites.html page
if (window.location.pathname.includes('favourite.html')) {
    document.addEventListener("DOMContentLoaded", displayFavorites);
}
document.getElementById("contact-form").addEventListener("submit", async function (e) {
    e.preventDefault();

    const formData = new FormData(this);
    const response = await fetch("save-form.php", {
        method: "POST",
        body: formData
    });

    const text = await response.text(); // get raw text
    console.log("Server returned:", text); // log raw response

    try {
        const json = JSON.parse(text); // try to convert to JSON
        if (json.success) {
            document.getElementById("success-message").style.display = "block";
            this.reset();
        } else {
            alert("Error from server: " + json.error);
        }
    } catch (e) {
        alert("Could not parse JSON from PHP file. See console for raw output.");
        console.error("JSON parsing failed. Full response:", text);
    }
});
```

Output:



Contact Us

Name: tilak

Email: Your Email

Please fill out this field.

Subject: Subject of Your Inquiry

Phone (Optional): Your Phone Number

Message:

Your Message

Conclusion

Implementing **registration**, **login**, **validation**, and **bookmark features** using **JavaScript** is fundamental for an interactive and personalized blogging platform like BLOGG.

These JavaScript functionalities:

- Enhance user engagement through smoother interactions
- Improve usability with real-time validations
- Enable readers to save and revisit their favourite posts easily

As BLOGG evolves, these features form the backbone for adding more dynamic capabilities such as personalized recommendations, comment systems, or admin controls. JavaScript thus transforms BLOGG from a static website into an interactive, user-centric content platform.

Experiment No.6

Problem Statement:

JavaScript

A. The user login form will allow registered users to log into their accounts. The form will require an email address and a password to authenticate the user.

B. If the login is successful, the user should be redirected to the homepage or their user dashboard. (**Contents beyond Syllabus)**

C. Use localStorage or sessionStorage to store authentication data, such as the user's email and login status. This ensures that once a user is logged in, they remain authenticated even after the page reloads or when they visit the site again. (**Contents beyond Syllabus)**

Save the cart data to local storage when items are added, updated, or removed. Retrieve and load the cart data from local storage when the page loads. (**Contents beyond Syllabus)**

Theory: Persistent Login and Bookmarking using Web Storage API in BLOGG Platform

---

Introduction

A smooth and engaging user experience is critical for a blogging platform like **BLOGG**. JavaScript's **Web Storage API**, which includes localStorage and sessionStorage, plays a vital role in maintaining user session state and preferences such as bookmarks/favourites without needing a backend database or server session.

---

1. Persistent Login using localStorage / sessionStorage

The login system allows users to authenticate and continue accessing personalized content without needing to log in repeatedly.

JavaScript Implementation:

- On successful login:
  - Store userEmail and isLoggedIn flags in localStorage
- On every page load:
  - JavaScript checks if the login data exists in localStorage
  - If yes, the user is auto-logged in and redirected to their dashboard or content page
- On logout:
  - Clear localStorage values to end the session

Benefits:

- Enhances usability with auto-login features
- Supports session continuity across multiple blog visits
- Useful for simple prototypes or static platforms without backend integration

---

2. Bookmark/Favourite Blog Posts using localStorage

Instead of a shopping cart, BLOGG uses bookmarking to let users save articles for later reading.

Implementation Highlights:

- When a user clicks "Add to Favourites":
  - Blog ID or slug is stored in a favourites array in localStorage
- On the "Favourites" page:
  - JavaScript retrieves and displays all saved blog entries
- Bookmarks persist across sessions unless cleared by the user

Benefits:

- Enhances engagement by letting users curate content
- Offers a personalized reading experience
- Stores data even after browser closure or refresh

Bonus Use Cases for BLOGG:

- Save theme preference (e.g., dark mode) using localStorage
- Maintain reading progress in long articles
- Store comment drafts for unsaved posts

Code:

*favourites:

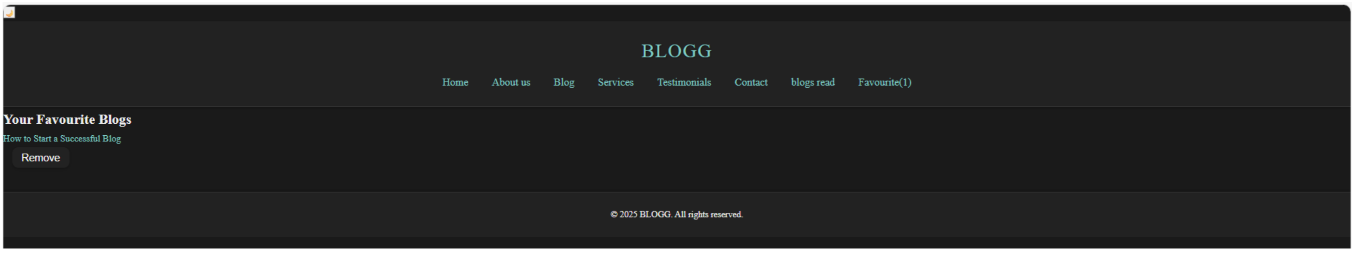
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Favorites</title>

    <link rel="stylesheet" href="design.css">

    <script src="web.js" defer></script>

</head>

<body class="dark-mode">

    <header>

        <div class="logo"><a href="index.html">BLOGG</a></div>

        <nav>

            <ul>

                <li><a href="home.html">Home</a></li>

                <li><a href="about.html">About us</a></li>

                <li><a href="blog.html">Blog</a></li>

                <li><a href="services.html">Services</a></li>

                <li><a href="Review.html">Testimonials</a></li>

                <li><a href="contact.html">Contact</a></li>

                <li><a href="favourite.html">blogs read</a></li>

                <li><a href="view_favourites.html">favourite(<span id="favorites-count">0</span>)</a></li>

            </ul>

        </nav>

    </header>

    <main>

        <section class="favorites">

            <h2> <span id="favorites-count">(0)</span></h2>

            <ul id="favorites-list">

                <p id="empty-message">No favorite blogs saved yet.</p>

            </ul>

        </section>

    </main>

    <footer>
```

```
        <p>&copy; 2025 My Blog. All rights reserved.</p>
    </footer>
</body>
</html>
```

*view_favourites:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Your Favourite Blogs</title>
    <link rel="stylesheet" href="design.css">
    <script src="web.js" defer></script>
</head>
<body class="dark-mode">
    <header>
        <div class="logo"><a href="/">BLOGG</a></div>
        <nav>
            <ul>
                <li><a href="home.html">Home</a></li>
                <li><a href="about.html">About us</a></li>
                <li><a href="blog.html">Blog</a></li>
                <li><a href="services.html">Services</a></li>
                <li><a href="Review.html">Testimonials</a></li>
                <li><a href="contact.html">Contact</a></li>
                <li><a href="favourite.html">blogs read</a></li>
                <li><a href="favourite.html">Favourite(<span id="favorites-count">0</span>)</a></li>
            </ul>
        </nav>
    </header>
    <main>
        <section class="favourite-blogs-list">
            <h1>Your Favourite Blogs</h1>
            <ul id="favorites-list">
                </ul>
            <p id="empty-message" style="display:none;">No favorite blogs added yet.</p>
        </section>
    </main>
    <footer>
        <p>&copy; 2025 BLOGG. All rights reserved.</p>
    </footer>
</body>
</html>
```

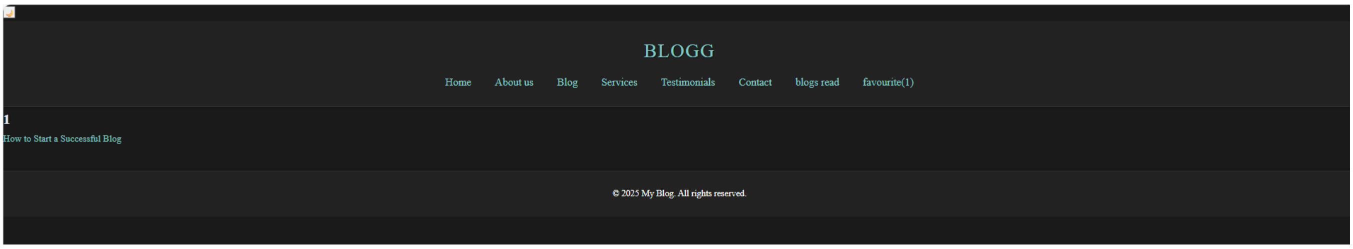
Output:



Your Favourite Blogs

How to Start a Successful Blog

© 2025 BLOGG. All rights reserved.



1

How to Start a Successful Blog

© 2025 My Blog. All rights reserved.

Conclusion

Using JavaScript’s Web Storage API significantly elevates the interactivity of BLOGG. **Persistent login** allows users to seamlessly access their reading or writing dashboards, while **bookmarking features** make the content experience more personalized and user-friendly.

These functionalities:

- Increase return visits and engagement
- Provide a smooth experience without relying on a server or database
- Enable powerful features in a static or front-end-heavy environment

In summary, JavaScript not only powers the interactivity of BLOGG but also ensures user preferences and sessions persist effectively—making the platform smarter, faster, and more user-focused.

Experiment No.7

Problem statement:

PHP

A. Develop a PHP script to handle user registration for the Blog platform website. The script should accept input from users for their name, email address, password, etc. (all required fields for registration).

B. Implement error handling to notify users of any issues during registration, such as validation errors.

C. Provide feedback to the user upon successful registration, either through a confirmation message or a redirect to a login page.

Theory: User Registration Script for BLOGG – Online Blogging Platform

---

Problem Statement

Develop a PHP script to handle user registration for the BLOGG platform website.
The script should accept input from users for their name, email address, password, etc., validate the data, and store it securely. It should also provide appropriate feedback or error handling for user experience.

Introduction

User registration is a fundamental component of any interactive website, especially on a content-driven platform like **BLOGG**. PHP is a widely-used server-side language that facilitates **form processing**, **database storage**, and **secure input validation**.

On BLOGG, user registration enables personalized features like:

- Creating and managing blog posts
- Bookmarking favorite content
- Managing user profiles

Core Elements of the PHP Registration Script

1. **Form Handling**
   - The form captures details like name, email, password, and confirms the password.
   - Data is submitted using the POST method to the PHP script.
2. **Validation**
   - Fields must not be empty.
   - Email is validated with regex to ensure proper formatting.
   - Password is checked for a minimum length and complexity.

  - Password and Confirm Password must match.

3. **Password Hashing**
    - Before storing, passwords are hashed using password_hash() to ensure secure storage and prevent plaintext password exposure.

4. **Database Interaction**
    - The script uses MySQLi or PDO to connect with the MySQL database and store user data into the users table.
5. **Error Handling**
    - Displays messages for missing/invalid input, duplicate accounts, or connection failures.
6. **User Feedback**
    - If registration is successful, the user is redirected to a login page or shown a success message.

Key Features of PHP:

- **Server-side execution**: Code runs on the server, not in the browser.
- **Open-source**: Freely available and widely supported.
- **Easy to integrate**: Works well with HTML, CSS, JavaScript, and MySQL.
- **Platform-independent**: Works on Windows, Linux, and macOS.
- **Supports databases**: Especially MySQL, for building dynamic websites.

◆ Basic Syntax

php

CopyEdit

```
<?php
  echo "Hello, BLOGG!";
?>
```

- PHP code is enclosed within <?php ... ?> tags.
- Statements end with a semicolon ;.
- Output is commonly displayed using echo or print.

◆ Variables in PHP

Variables store data and start with a $ symbol.

php

CopyEdit

```
$name = "Alice";
$age = 25;
```

❖ Common PHP Data Types

- String: "Hello"
- Integer: 42
- Float: 3.14
- Boolean: true or false
- Array: ["blog", "post", "title"]
- Object: Custom data types

❖ Conditional Statements

php

CopyEdit

```
if ($age > 18) {
  echo "Adult";
} else {
  echo "Minor";
}
```

❖ Loops

php

CopyEdit

```
for ($i = 0; $i < 5; $i++) {
  echo $i;
}
```

Other loops: while, do...while, foreach

❖ Functions

Reusable blocks of code:

php

CopyEdit

```
function greet($name) {
  return "Hello, $name!";
}
```

❖ PHP with HTML

php

CopyEdit

```
<html>
  <body>
    <h1><?php echo "Welcome to BLOGG!"; ?></h1>
  </body>
</html>
```

❖ Superglobals

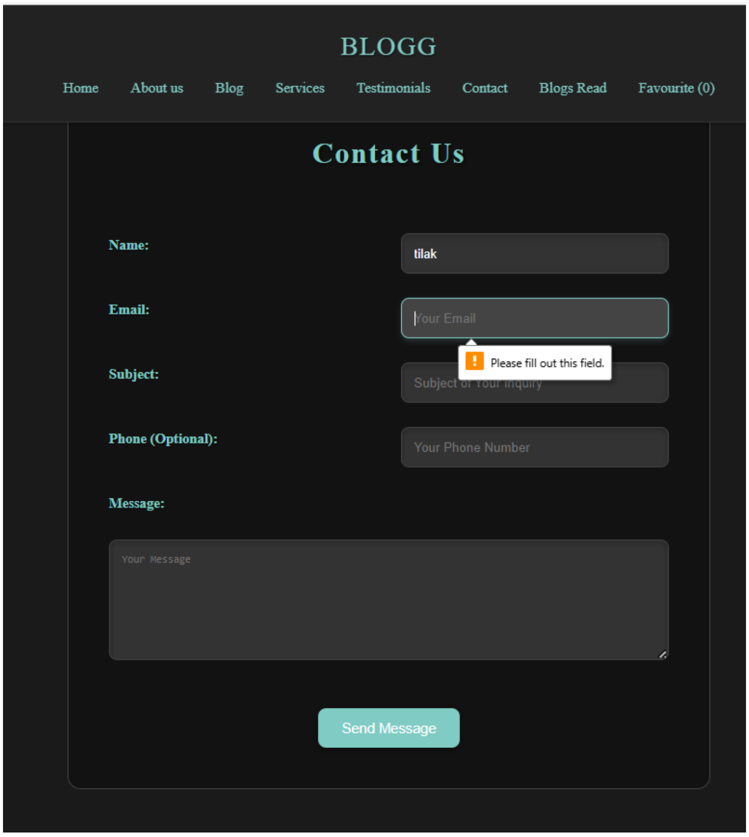
- $_GET, $_POST: Get user input from forms
- $_SESSION, $_COOKIE: Store user data
- $_SERVER: Server/environment info
- $_FILES: File uploads

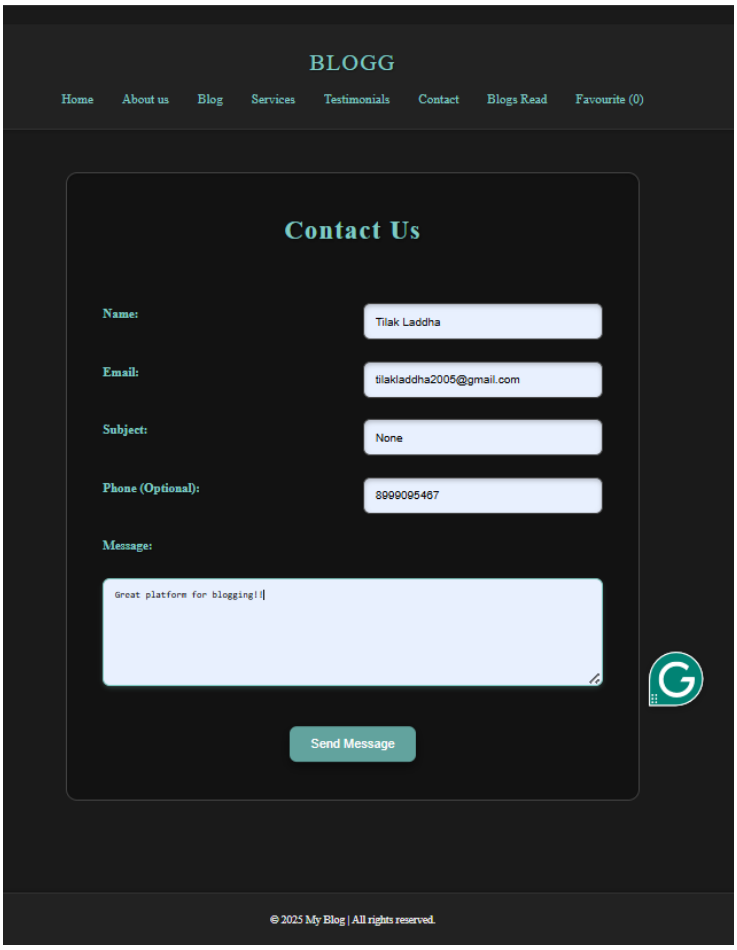
Code:

```
<?php

error_reporting(E_ALL);

ini_set('display_errors', 1);

header('Content-Type: application/json');


// Debug: Check if request is POST

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {

    echo json_encode(['success' => false, 'error' => 'Invalid request method.']);

    exit;
}

$host = 'localhost';

$user = 'root';

$pass = 'tilak2005';  // use your real password

$db = 'blog_db';


$conn = new mysqli($host, $user, $pass, $db);


// Debug: Check connection

if ($conn->connect_error) {

    echo json_encode(['success' => false, 'error' => 'DB connection failed: ' . $conn->connect_error]);

    exit;

}

// Get form data safely

$name = $_POST['name'] ?? '';

$email = $_POST['email'] ?? '';

$subject = $_POST['subject'] ?? '';

$phone = $_POST['phone'] ?? '';

$message = $_POST['message'] ?? '';


// Debug: Check missing fields

if (!$name || !$email || !$message) {

    echo json_encode(['success' => false, 'error' => 'Missing required fields.']);

    exit;

}


$stmt = $conn->prepare("INSERT INTO contact_form (name, email, subject, phone, message) VALUES (?, ?, ?, ?, ?)");

$stmt->bind_param("sssss", $name, $email, $subject, $phone, $message);
```

```
if ($stmt->execute()) {
    echo json_encode(['success' => true]);
} else {
    echo json_encode(['success' => false, 'error' => 'Insert failed: ' . $stmt->error]);
}

$stmt->close();
$conn->close();
?>
```

Output:





Contact Us

Name: Tilak Laddha

Email: tilakladdha2005@gmail.com

Subject: None

Phone (Optional): 8999095467

Message:

Great platform for blogging!!

© 2025 My Blog | All rights reserved.

Conclusion:

Implementing a **PHP-based registration system** is essential for managing user identity in the **BLOGG platform**. A well-structured registration script ensures **secure account creation**, **input validation**, and **database integration**, enabling a personalized experience for every blogger or reader.

Through this PHP implementation:

- Users can **register with valid credentials** (name, email, password).
- **Server-side validation** guarantees that no incomplete or incorrect data is stored.
- **Password hashing** protects user data from breaches.
- **Database integration (MySQL)** ensures persistent storage of user details.
- Clear **error messages** and **success feedback** enhance user interaction and trust.

For BLOGG, this registration system lays the foundation for:

- **User-authored blog posts**
- **Commenting systems**
- **Personalized dashboards**
- **Bookmarking/favourites**, and more

Experiment No. 8

Problem statement:

PHP

A. Develop a PHP script to handle user login for the Blog platform website. The script should accept input from users for their login credentials. (all required fields for login).

B. Provide feedback to the user upon successful login, either through a confirmation message or a redirect to a welcome page.

C. Implement error handling to notify users of login failures due to incorrect credentials or other errors.

D. Provide feedback to the user upon successful login, either through a welcome user name message or a redirect to a home page.

Theory: User Login Script for BLOGG – Online Blogging Platform

---

Problem Statement

Develop a PHP script to handle **user login** for the BLOGG website.
The script should accept input for login credentials (email and password), validate the input, compare it with database records, and start a user session upon success. It must also provide feedback for success or failure.

Introduction

The **login system** is a vital feature of the BLOGG platform, enabling registered users to access **personalized content**, **create or edit blogs**, and **bookmark their favourite articles**. Using PHP for login management allows secure backend communication, user session tracking, and robust error handling.

PHP login implementation involves:

- Capturing credentials through a form
- Validating and sanitizing input
- Verifying the credentials against a MySQL database
- Starting a session upon successful login
- Displaying errors or redirecting to the homepage if successful

---

Key Components of the Login System

1. Form Handling

- Accepts email and password from the login form using the POST method.

2. Input Validation

- Checks if fields are filled
- Verifies if email format is correct
- Avoids SQL injection using prepared statements

3. Password Verification

- Passwords are hashed during registration using password_hash()
- At login, password_verify() checks if the entered password matches the hash

4. Database Authentication

- Connects to MySQL
- Searches for a user with the provided email
- Verifies the password against the stored hash

5. Session Management

- Upon successful login, session_start() stores the user’s session data
- Enables content access and dashboard features

6. User Feedback

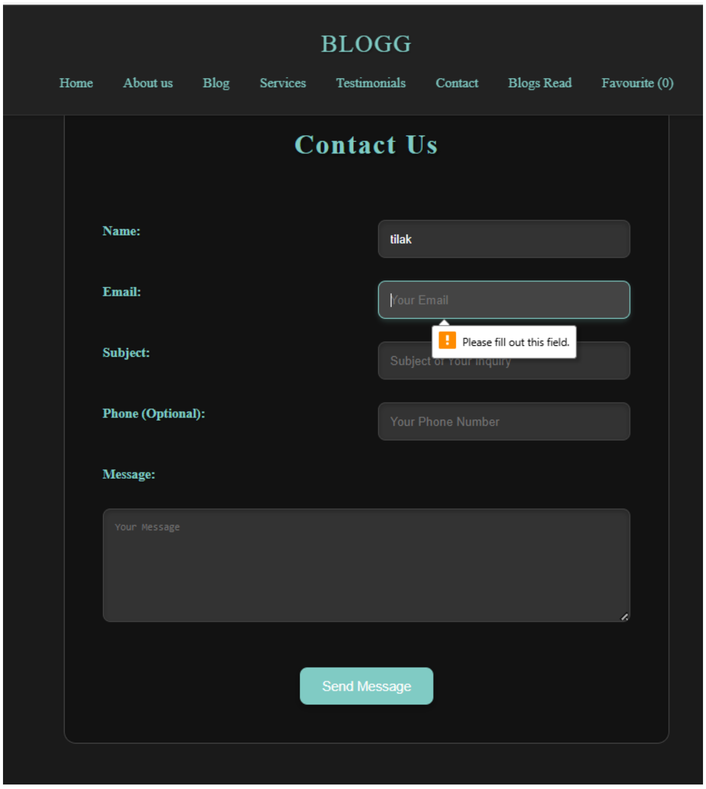
- If successful: Redirects to home/dashboard or shows a welcome message
- If failed: Displays relevant error messages (e.g., "Invalid credentials")

Code:

```
<?php

error_reporting(E_ALL);

ini_set('display_errors', 1);

header('Content-Type: application/json');



// Debug: Check if request is POST

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {

    echo json_encode(['success' => false, 'error' => 'Invalid request method.']);

    exit;

}



$host = 'localhost';

$user = 'root';
```

```
$pass = 'tilak2005';  // use your real password

$db = 'blog_db';


$conn = new mysqli($host, $user, $pass, $db);

// Debug: Check connection

if ($conn->connect_error) {

    echo json_encode(['success' => false, 'error' => 'DB connection failed: ' . $conn->connect_error]);

    exit;

}

// Get form data safely

$name = $_POST['name'] ?? '';

$email = $_POST['email'] ?? '';

$subject = $_POST['subject'] ?? '';

$phone = $_POST['phone'] ?? '';

$message = $_POST['message'] ?? '';


// Debug: Check missing fields

if (!$name || !$email || !$message) {

    echo json_encode(['success' => false, 'error' => 'Missing required fields.']);

    exit;

}

$stmt = $conn->prepare("INSERT INTO contact_form (name, email, subject, phone, message) VALUES (?, ?, ?, ?, ?)");

$stmt->bind_param("sssss", $name, $email, $subject, $phone, $message);


if ($stmt->execute()) {

    echo json_encode(['success' => true]);

} else {

    echo json_encode(['success' => false, 'error' => 'Insert failed: ' . $stmt->error]);
}
$stmt->close();
$conn->close();
?>
```

Output:



Contact Us

Name: tilak

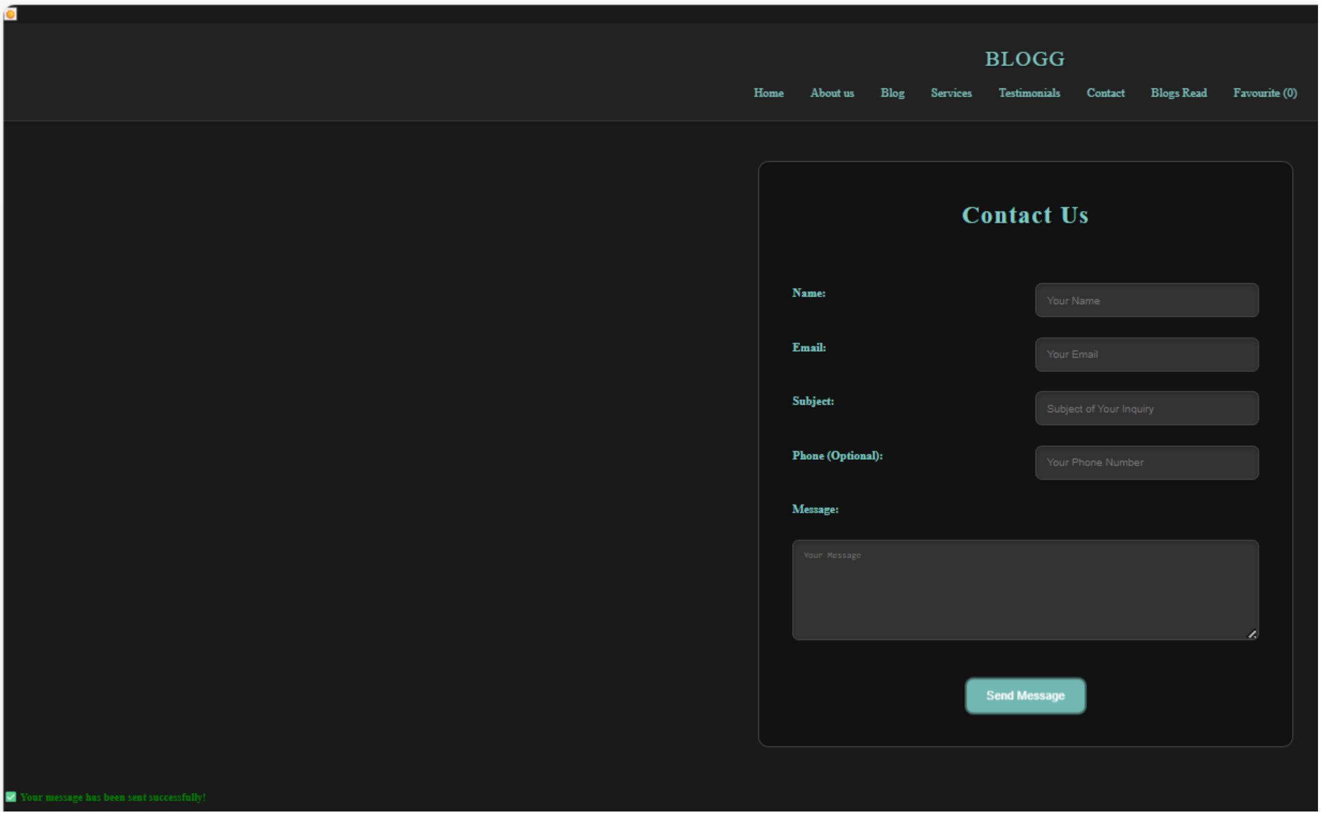
Email: Your Email

Please fill out this field.

Subject: Subject of Your Inquiry

Phone (Optional): Your Phone Number

Message: Your Message



Contact Us

Name: Your Name

Email: Your Email

Subject: Subject of Your Inquiry

Phone (Optional): Your Phone Number

Message: Your Message

✅ Your message has been sent successfully!

Conclusion:

Implementing a **PHP-based login system** on the BLOGG platform is critical for managing user access and enabling personalized functionality like **blog creation**, **commenting**, and **bookmarking**.

With this PHP login system:

- Users securely log in using their credentials
- Sessions maintain their authenticated state across pages
- Passwords are verified using **hashed encryption** for security
- Users are provided with **immediate feedback**, improving usability

For BLOGG, this login functionality ensures that content access is **secure**, **reliable**, and **scalable**. It enables features like:

- Personal dashboards
- Saving and editing blogs
- Bookmarking favourites

Experiment No. 9

Problem statement:

PHP

A. Develop a PHP script that allows users to manage their shopping cart for an e-commerce website (e.g., a online blog platform). The script should allow users to add items to their cart, view their cart contents, and remove items if needed.

B. Develop a PHP script to manage the cart for an website (e.g. online blog platform,) using MySQL. This script should allow users to add items to their cart, view their cart contents, and remove items from the cart. The cart data should be stored in the MySQL database to allow persistence across sessions.

Theory: Bookmark (Read Later) Management System for BLOGG Platform

---

A. Problem Statement

Develop a PHP system for BLOGG that allows users to:

- **Add blog posts** to their bookmark or "Read Later" list
- **View saved blog posts**
- **Remove bookmarked posts** if desired
  The bookmark data can be stored temporarily via session or permanently in a MySQL database for logged-in users.

Theory: Bookmark (Read Later) Feature in BLOGG

Bookmarks are an essential component of content platforms. They allow readers to save articles for later reading, revisit them conveniently, and manage their favorite content.

Two Bookmark Management Methods

A. Session-Based Bookmarking (Without MySQL)

This method uses PHP $_SESSION to temporarily store bookmarked blog post IDs.

Key Characteristics:

- Bookmarks are stored in session memory
- Quick and doesn't require user login
- Data is lost once session expires or browser closes

Operations Supported:

- Add: Save blog ID to session

- View: Display blog details using stored blog IDs
- Remove: Unset blog ID from session array

Pros:

- Easy to implement
- Good for guest users

Cons:

- Not persistent
- Cannot track across devices or sessions

B. Database-Based Bookmarking (With MySQL)

A persistent solution using a bookmarks table in MySQL.

Key Characteristics:

- Requires login (user ID linkage)
- Saves blog ID and user ID in database
- Persistent across sessions and devices

Operations Supported:

- Add: Insert blog ID for logged-in user
- View: Fetch blog details using SQL JOIN
- Remove: Delete entry by user and blog ID

Pros:

- Persistent and scalable
- Supports personalized blog management

Cons:

- Requires login system
- Needs proper error and session handling

Code:

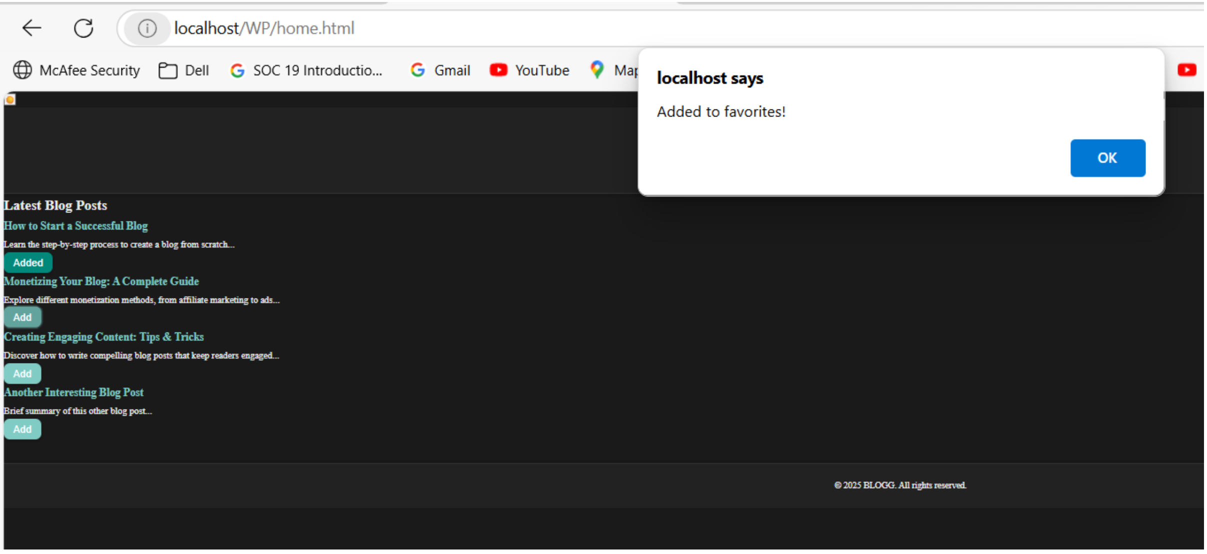
```
// Dark Mode Toggle

const darkModeToggle = document.createElement("button");
```

```
darkModeToggle.textContent = "🌙";
darkModeToggle.classList.add("dark-mode-toggle");
document.body.prepend(darkModeToggle);

darkModeToggle.addEventListener("click", function () {
    document.body.classList.toggle("dark-mode");
    darkModeToggle.textContent = document.body.classList.contains("dark-mode") ? "☀️" : "🌙";
    localStorage.setItem("dark-mode", document.body.classList.contains("dark-mode"));
});

// Load dark mode preference
if (localStorage.getItem("dark-mode") === "true") {
    document.body.classList.add("dark-mode");
    darkModeToggle.textContent = "☀️";
}

// Favorite Blogs - Add & Display
function getFavorites() {
    const storedFavorites = localStorage.getItem("favorites");
    return storedFavorites ? JSON.parse(storedFavorites) : [];
}

function saveFavorites(favorites) {
    localStorage.setItem("favorites", JSON.stringify(favorites));
    updateFavoritesCount();
}

function updateFavoritesCount() {
    const count = getFavorites().length;
    document.querySelectorAll("#favorites-count").forEach(el => {
        el.textContent = count;
    });
}

function updateFavButtonUI(button, isFavorite) {
    if (button) {
        button.classList.toggle('active', isFavorite);
        button.textContent = isFavorite ? 'Added' : 'Add';
    }
}

function addToFavorites(title, link, buttonElement) {
    let favorites = getFavorites();
    const existingIndex = favorites.findIndex(item => item.title === title);
    if (existingIndex === -1) {
        favorites.push({ title, link });
        saveFavorites(favorites);
        updateFavButtonUI(buttonElement, true);
        alert("Added to favorites!");
    } else {
        favorites.splice(existingIndex, 1);
        saveFavorites(favorites);
        updateFavButtonUI(buttonElement, false);
        alert("Removed from favorites!");
    }
    displayFavorites();
}
```

```
// Event listener for favorite buttons
document.addEventListener('click', function (event) {
    if (event.target.classList.contains('fav-button')) {
        const button = event.target;
        const title = button.dataset.title;
        const link = button.dataset.link;
        addToFavorites(title, link, button);
    }
});

// Initialize favorite buttons on page load
document.querySelectorAll(".fav-button").forEach(button => {
    const title = button.dataset.title;
    updateFavButtonUI(button, getFavorites().some(item => item.title === title));
});
function displayFavorites() {
    const favoritesList = document.getElementById("favorites-list");
    const emptyMessage = document.getElementById("empty-message");
    const isFavouritePage = window.location.pathname.includes("favourite.html"); // Check if it's the 'blogs read' page

    if (!favoritesList) return console.log("favorites-list element not found!");

    favoritesList.innerHTML = "";
    const favorites = getFavorites();

    console.log("Favorites retrieved:", favorites);
    if (favorites.length === 0) {
        emptyMessage.style.display = 'block';
        favoritesList.style.display = 'none';
        return;
    } else {
        emptyMessage.style.display = 'none';
        favoritesList.style.display = 'block';
    }

    favorites.forEach((item) => {
        const li = document.createElement("li");
        li.innerHTML = `
            <a href="${item.link || "#"}" target="_blank">${item.title || "Untitled"}</a>
            ${!isFavouritePage ? `<button class="remove-btn" data-title="${item.title}">Remove</button>` :
""}
        `;
        favoritesList.appendChild(li);
    });
}

// Event listener for removing from favorites
document.addEventListener('click', function (event) {
    if (event.target.classList.contains('remove-btn')) {
        removeFromFavorites(event.target.dataset.title);
    }
});
function removeFromFavorites(title) {
    saveFavorites(getFavorites().filter(item => item.title !== title));
    displayFavorites();
```

```
}
// Ensure favorites count updates when the page loads
updateFavoritesCount();


// Display favorites if on the view_favourites.html page
if (window.location.pathname.includes('view_favourites.html')) {
    document.addEventListener("DOMContentLoaded", displayFavorites);
}
// Display favorites if on the view_favourites.html page
if (window.location.pathname.includes('favourite.html')) {
    document.addEventListener("DOMContentLoaded", displayFavorites);
}
document.getElementById("contact-form").addEventListener("submit", async function (e) {
    e.preventDefault();


    const formData = new FormData(this);
    const response = await fetch("save-form.php", {
        method: "POST",
        body: formData
    });


    const text = await response.text(); // get raw text
    console.log("Server returned:", text); // log raw response


    try {
        const json = JSON.parse(text); // try to convert to JSON
        if (json.success) {
            document.getElementById("success-message").style.display = "block";
            this.reset();
        } else {
          alert("Error from server: " + json.error);
        }
    } catch (e) {
        alert("Could not parse JSON from PHP file. See console for raw output.");
        console.error("JSON parsing failed. Full response:", text);
    }
});
```

Output:



localhost says

Added to favorites!

Latest Blog Posts

How to Start a Successful Blog

Learn the step-by-step process to create a blog from scratch...

Added

Monetizing Your Blog: A Complete Guide

Explore different monetization methods, from affiliate marketing to ads...

Add

Creating Engaging Content: Tips & Tricks

Discover how to write compelling blog posts that keep readers engaged...

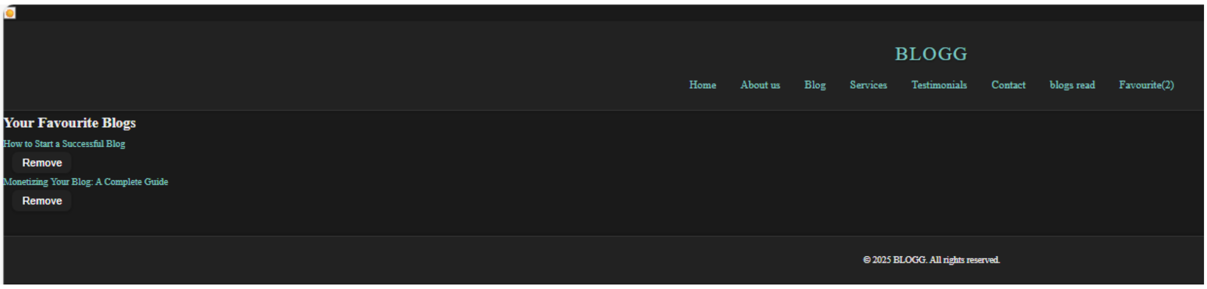
Add

Another Interesting Blog Post

Brief summary of this other blog post...

Add

© 2025 BLOGG. All rights reserved.



BLOGG

Your Favourite Blogs

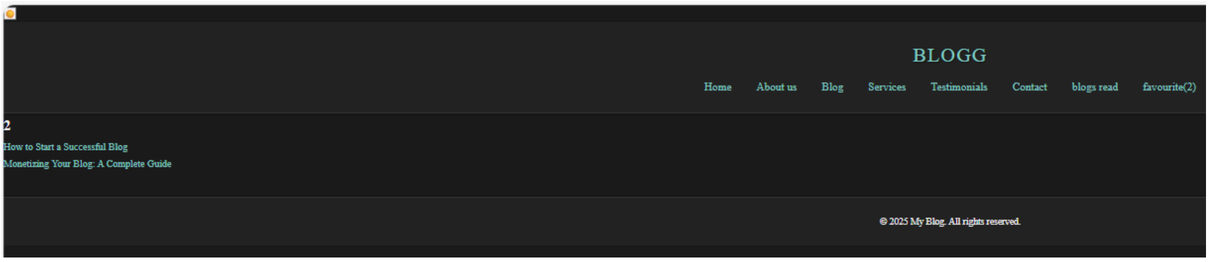
How to Start a Successful Blog

Remove

Monetizing Your Blog: A Complete Guide

Remove

© 2025 BLOGG. All rights reserved.



BLOGG

2

How to Start a Successful Blog

Monetizing Your Blog: A Complete Guide

© 2025 My Blog. All rights reserved.

Conclusion – BLOGG Bookmark System

The “Read Later” feature adds significant value to the BLOGG platform by:

- Letting users curate content based on interest
- Encouraging them to return and engage more
- Supporting content discovery and personalization

Session-based bookmarks are simple for temporary readers, while **MySQL bookmarks** provide persistent, user-specific functionality.

For real-world usage, the **database-driven approach** is highly recommended.

Experiment No. 10

Problem statement:

PHP

A. Develop a PHP script to handle the checkout process for users who are ready to complete their purchase. The script should process the cart data and provide feedback to the user upon successful or failed checkout.

B. Develop a PHP script that processes the checkout process for users who are ready to complete their purchase, integrating the MySQL database for handling user and order information. The script should validate user input, process the cart data, and provide feedback upon successful or failed checkout.

Theory: Blog Submission (Checkout Analogy) for BLOGG Platform

---

A. Problem Statement

Develop a PHP script that allows users to submit blog posts (analogous to a checkout in e-commerce), validating and storing the submission in a MySQL database. Feedback should be shown for both successful and failed submissions.

Theory: Blog Post Submission System

In BLOGG, the blog post submission is the equivalent of a checkout process. It turns a draft or composed blog into published content visible on the platform.

Two Types of Blog Submission

A. Session-Based Draft Blog Submission

- User writes a blog post
- Content is saved in session variables (optional step)
- On submit, it may be displayed as a preview and saved later to DB manually

Pros:

- Simple, good for preview/demos
- No database dependency

Cons:

- Non-persistent
- Not production-ready

B. Database-Based Blog Submission

Blog posts are inserted into a blogs table on submission. This supports versioning, publishing, and later editing.

Workflow:

1. Validate that the user is logged in
2. Validate blog inputs (title, content, category)
3. Sanitize input to prevent XSS/SQL injection
4. Insert into blogs table with user ID
5. Return success/failure response

Code:

Sql:

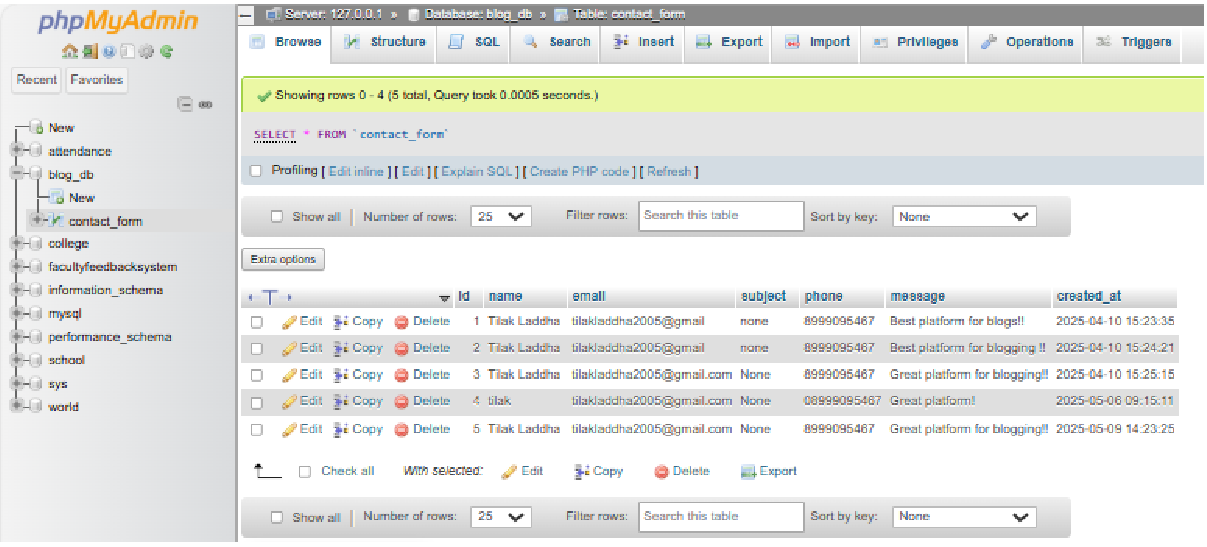
```
CREATE TABLE contact_form (

    id INT AUTO_INCREMENT PRIMARY KEY,

    name VARCHAR(100) NOT NULL,

    email VARCHAR(100) NOT NULL,

    subject VARCHAR(255),

    phone VARCHAR(20),

    message TEXT,

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);
```

Php:

```
<?php

error_reporting(E_ALL);

ini_set('display_errors', 1);

header('Content-Type: application/json');

// Debug: Check if request is POST

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {

    echo json_encode(['success' => false, 'error' => 'Invalid request method.']);

    exit;
```

```
}
$host = 'localhost';
$user = 'root';
$pass = 'tilak2005';  // use your real password
$db = 'blog_db';
$conn = new mysqli($host, $user, $pass, $db);
// Debug: Check connection
if ($conn->connect_error) {
    echo json_encode(['success' => false, 'error' => 'DB connection failed: ' . $conn->connect_error]);
    exit;
}
// Get form data safely
$name = $_POST['name'] ?? '';
$email = $_POST['email'] ?? '';
$subject = $_POST['subject'] ?? '';
$phone = $_POST['phone'] ?? '';
$message = $_POST['message'] ?? '';
// Debug: Check missing fields
if (!$name || !$email || !$message) {
    echo json_encode(['success' => false, 'error' => 'Missing required fields.']);
    exit;
}
$stmt = $conn->prepare("INSERT INTO contact_form (name, email, subject, phone, message) VALUES (?, ?, ?, ?, ?)");
$stmt->bind_param("sssss", $name, $email, $subject, $phone, $message);

if ($stmt->execute()) {
    echo json_encode(['success' => true]);
} else {
    echo json_encode(['success' => false, 'error' => 'Insert failed: ' . $stmt->error]);
}
$stmt->close();
$conn->close();
?>
```

Output:



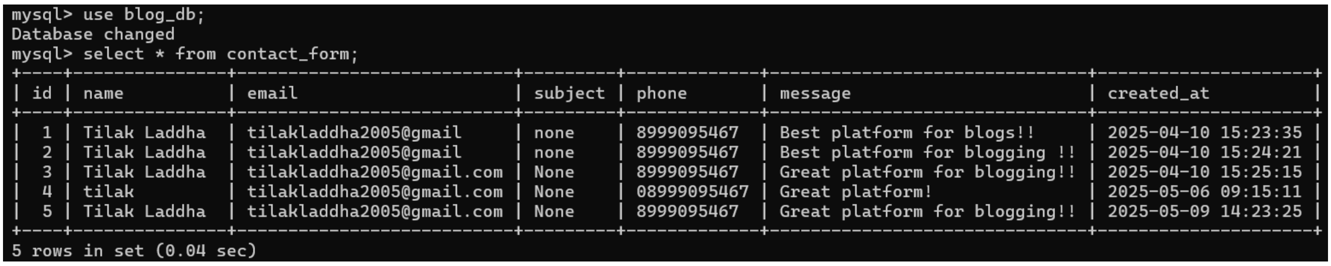
✓ Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

```
SELECT * FROM `contact_form`
```

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	name	email	subject	phone	message	created_at
□ Edit Copy Delete	1	Tilak Laddha	tilakladdha2005@gmail	none	8999095467	Best platform for blogs!!	2025-04-10 15:23:35
□ Edit Copy Delete	2	Tilak Laddha	tilakladdha2005@gmail	none	8999095467	Best platform for blogging !!	2025-04-10 15:24:21
□ Edit Copy Delete	3	Tilak Laddha	tilakladdha2005@gmail.com	None	8999095467	Great platform for blogging!!	2025-04-10 15:25:15
□ Edit Copy Delete	4	tilak	tilakladdha2005@gmail.com	None	08999095467	Great platform!	2025-05-06 09:15:11
□ Edit Copy Delete	5	Tilak Laddha	tilakladdha2005@gmail.com	None	8999095467	Great platform for blogging!!	2025-05-09 14:23:25

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None



```
mysql> use blog_db;
Database changed
mysql> select * from contact_form;
+----+--------------+---------------------------+---------+-------------+-------------------------------+---------------------+
| id | name         | email                     | subject | phone       | message                       | created_at          |
+----+--------------+---------------------------+---------+-------------+-------------------------------+---------------------+
|  1 | Tilak Laddha | tilakladdha2005@gmail     | none    | 8999095467  | Best platform for blogs!!     | 2025-04-10 15:23:35 |
|  2 | Tilak Laddha | tilakladdha2005@gmail     | none    | 8999095467  | Best platform for blogging !! | 2025-04-10 15:24:21 |
|  3 | Tilak Laddha | tilakladdha2005@gmail.com | None    | 8999095467  | Great platform for blogging!! | 2025-04-10 15:25:15 |
|  4 | tilak        | tilakladdha2005@gmail.com | None    | 08999095467 | Great platform!               | 2025-05-06 09:15:11 |
|  5 | Tilak Laddha | tilakladdha2005@gmail.com | None    | 8999095467  | Great platform for blogging!! | 2025-05-09 14:23:25 |
+----+--------------+---------------------------+---------+-------------+-------------------------------+---------------------+
5 rows in set (0.04 sec)
```

Conclusion – BLOGG Blog Submission System

The blog post submission system is the heart of the BLOGG platform. It allows content creators to:

- Compose and share their ideas securely
- Store drafts or publish immediately
- Receive confirmation of successful submission

The **database-driven approach** ensures all blog data is:

- Securely stored
- Linked to users
- Ready for display on blog listing pages

This process transforms BLOGG from a static site into a **dynamic content platform** where users actively contribute and engage.