TO DEVELOP A FORUM FOR USERS TO DISCUSS VARIOUS TOPICS

A PROJECT REPORT

Submitted by

SRI RAAM V H (113222072098)

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

VELAMMAL ENGINEERING COLLEGE,
CHENNAI-66.

MARCH 2025

VELAMMAL
ENGINEERING COLLEGE
The Wheel of Knowledge rolls on!

BONAFIDE CERTIFICATE

This is to certify that this project report titled "TO DEVELOP A FORUM FOR USERS TO DISCUSS VARIOUS TOPICS" is the bonafide work of Mr. **SRI RAAM V H (113222072098),** who carried out the Project Based Assignment work under my supervision.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
Mrs.Mythili S	**Dr.Visu P**
Assistant Professor	**Professor & Head**
Department of Artificial	Department of Artificial
Intelligence and Data Science	Intelligence and Data Science
Velammal Engineering College	Velammal Engineering College
Chennai – 600066.	Chennai – 600066.

ABSTRACT

This project aims to develop a responsive discussion forum website that facilitates user engagement through topic-based discussions and interactive commenting. The platform will allow users to register securely, create new discussion threads, and participate in ongoing conversations by posting comments. Emphasizing both usability and accessibility, the website will feature a clean, intuitive interface adaptable to various devices and screen sizes. Core functionalities will include user authentication, discussion categorization, comment threading, and real-time updates to enhance the overall interaction experience. The system will be built using modern web development technologies, focusing on scalability, performance, and ease of use. This forum will serve as a collaborative space for knowledge sharing, community building, and meaningful dialogue among users.

To ensure a seamless user experience, the forum will incorporate responsive design principles using modern frontend frameworks, enabling accessibility across desktops, tablets, and mobile devices. The backend will be structured to support efficient data handling, user session management, and secure authentication protocols. Additional features such as search functionality, user profiles, upvoting or liking comments, moderation tools, and notification systems will be integrated to foster a dynamic and safe online community. By leveraging scalable architecture and clean code practices, the system will be designed to accommodate growing user activity while maintaining performance and reliability. This project not only emphasizes technical implementation but also prioritizes user engagement, community interaction, and a structured approach to digital communication.

ACKNOWLEDGEMENT

I am deeply thankful to the Lord Almighty for His abundant blessings, which enabled me to successfully complete this project.

I would like to express my sincere gratitude to our esteemed Chairman, **Dr.M.V. Muthuramalingam**, for his unwavering support and encouragement.

I also extend my heartfelt thanks to our Chief Executive Officer, Thiru. **M.V.M. Velmurugan**, for his constant support and motivation. I am equally grateful to Thiru. **V. Karthik Muthuramalingam**, the Deputy CEO, for his invaluable assistance and encouragement.

My sincere thanks go **to Dr. S. Satish Kumar**, Principal, for his continuous support and motivation throughout this project.

I extend my deepest gratitude to **Dr. P. Visu**, Professor & Head of the Department of Artificial Intelligence and Data Science, for being a guiding force and constant source of inspiration.

I am especially grateful to my Project Guide, **Mrs. S. Mythili**, Assistant Professor, Department of Artificial Intelligence and Data Science, for her invaluable guidance, support, and encouragement.

SRI RAAM V H

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Online discussion forums have become essential platforms for facilitating communication, knowledge exchange, and collaborative problem-solving across diverse communities.

In an era where digital interaction plays a critical role in both educational and professional environments, there is a growing need for accessible, well-structured, and user-friendly discussion platforms.

This project proposes the development of a responsive discussion forum website that allows users to register, initiate discussions, and engage with others through threaded comments. The goal is to create an interactive space where individuals can share ideas, seek help, and contribute to ongoing conversations in an organized manner.

The proposed forum will prioritize responsiveness, ensuring compatibility across a range of devices and screen sizes, thereby enhancing usability and inclusivity. Core functionalities will include user authentication, topic creation, comment management, and real-time content updates.

By incorporating modern web development tools and adhering to best practices in UI/UX design and backend development, the platform aims to deliver a reliable, scalable, and secure solution.

This system not only serves as a communication tool but also as a community-building space where users can connect, collaborate, and grow through active participation.

the platform will integrate extended functionalities such as user profiles, topic categorization, post moderation, and notification systems to enhance user engagement and maintain a well-regulated discussion environment.

The system will also support features like post upvoting, comment sorting, and activity tracking to encourage meaningful contributions and highlight popular or relevant content. Security considerations such as data validation, password encryption, and role-based access control will be incorporated to protect user data and ensure platform integrity.

Through this project, the aim is not only to build a technically sound application but also to foster a collaborative digital community where users can freely exchange ideas and information in a structured and secure setting.

CHAPTER 2

LITERATURE SURVEY

The development of discussion forum platforms has gained increasing attention in recent years, particularly as digital collaboration and community engagement become central to online experiences. This literature survey reviews existing research and best practices in web-based forum development, user interaction design, and backend management systems.

1. Web-Based Forum Development

Discussion forums are among the earliest forms of online community platforms and have evolved significantly with the advancement of web technologies. Studies show that early bulletin board systems laid the foundation for structured asynchronous communication. Modern forums leverage full-stack web development technologies such as HTML, CSS, JavaScript (frontend), and server-side frameworks like Node.js, Django, or Laravel for backend operations. Literature emphasizes the importance of RESTful API design, database normalization, and secure data handling in building scalable and maintainable forum systems. Responsive web design is also a key consideration, enabling users to access discussions from any device. Research also suggests that integrating real-time features through WebSockets or frameworks like Socket.IO can significantly enhance user interaction and feedback.

2. User Interaction and Experience Design

User Interface (UI) and User Experience (UX) are critical in determining the success of online forums. Research indicates that clear visual hierarchy, intuitive navigation, and a clutter-free layout improve user retention and satisfaction. Studies in human-computer interaction highlight the significance of minimal cognitive load, logical content grouping, and visual cues such as icons and badges to guide users through the platform. Mobile-first design practices and accessibility standards (such as WCAG compliance) are also emphasized to ensure inclusive participation. Interactive components like upvotes, nested comments, and reaction buttons have been shown to foster greater community engagement. The literature also explores gamification elements (e.g., reputation points or badges) to incentivize quality contributions and sustained user activity.

3. Backend Architecture and Data Management

The backend structure of a forum is foundational to its performance, scalability, and data integrity. Research highlights the use of relational databases like MySQL and PostgreSQL, as well as NoSQL alternatives like MongoDB, depending on the nature and complexity of data relationships. Session management, role-based access control, and data encryption are crucial security measures discussed across multiple sources. Server-side validation and protection against SQL injection, CSRF, and XSS attacks are considered standard security practices. The literature also emphasizes the role of backend frameworks in simplifying development through modular design, middleware integration, and reusable components. Asynchronous processing and job queues are discussed as solutions for handling tasks like email notifications or data updates efficiently.

4. Real-Time Interaction and Notification Systems

Recent studies emphasize the importance of real-time interaction in discussion forums to emulate the responsiveness of modern chat platforms. Technologies such as WebSockets, Firebase, and SignalR are often employed to push new content updates instantly, improving engagement and reducing perceived latency. Literature on notification systems suggests that personalized, timely alerts (such as replies, mentions, and thread subscriptions) significantly improve user retention. These systems often rely on event-driven architecture and publish-subscribe models to scale efficiently. Best practices also highlight the importance of allowing users to control notification preferences, ensuring relevance without overwhelming them with information.

5. Security and Community Moderation

Security and content moderation are recurring themes in forum development literature. Studies discuss the need for robust user authentication systems, including support for OAuth and multi-factor authentication. Anti-spam measures such as CAPTCHA, rate limiting, and content filtering are explored in depth. Moreover, literature on community moderation highlights the importance of reporting tools, admin dashboards, and automated content analysis using machine learning to detect offensive language or spam. Transparent moderation policies and community guidelines are also identified as key factors in sustaining healthy online interactions.

CHAPTER 3
SYSTEM REQUIREMENTS

1.1 SOFTWARE REQUIREMENTS

Client-Side:

- **Operating System:** Any modern OS such as Windows 10/11, macOS, or Linux (Ubuntu, Fedora) capable of running a web browser.
- **Web Browser:** Latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari to ensure compatibility with modern web standards.
- **Languages & Technologies:** • HTML5 – for structuring the game interface
  • CSS3 – for styling the user interface and layout
  • JavaScript – for implementing game logic and interactivity.
- **Code Editor:** Any code editor like Visual Studio Code, Sublime Text, or Atom for editing HTML, CSS, and JavaScript files.

Server-Side:

- **Web Server (for hosting):** Apache, Nginx, or Node.js server (if deploying online).
- **Hosting Platform:** GitHub Pages, Netlify, or any basic static site hosting service for deploying the application.
- **Backend (for advanced features like AI, multiplayer, or persistent scores):** Node.js, Flask, or Django (optional depending on feature extensions).

1.2 HARDWARE REQUIREMENTS

Client-Side:

- **Processor:** Dual-Core or better (Intel i3/i5/i7, AMD Ryzen 3 or better).
- **Memory (RAM):** Minimum 4 GB RAM (8 GB recommended for smooth multitasking).
- **Storage:** At least 200 MB of free space (for source code, browser cache, and local development environment).
- **Display:** 1024x768 resolution or higher for proper UI rendering.
- **Network:** Internet connection for downloading files, assets, or online hosting (optional for local use)

Server-Side:

- **Processor:** Quad-Core or higher if hosting the application on a server.
- **Memory (RAM):** Minimum 8 GB RAM for hosting multiple sessions or extended features.
- **Storage:** 500 GB SSD or higher for application hosting, logs, and media storage (if extended)
- **Network:** Reliable internet with minimum 100 Mbps (1 Gbps preferred) for faster access and deployment.

CHAPTER 4
SYSTEM DESIGN

Absolutely! Here's a detailed **System Design** section for your **Responsive Discussion Forum Website**, written in the same structured and in-depth style as the previous sections:

System Design

The system design for the responsive discussion forum website is structured to ensure scalability, modularity, user interactivity, and seamless communication across devices. The design adopts a layered architecture encompassing frontend, backend, and database components, with attention to user authentication, content management, and real-time interaction. This section outlines the architectural layout, component interactions, and technologies used in building a functional and robust forum system.

4.1 System Architecture Overview

The system follows a **client-server architecture**, where the frontend communicates with a backend server via RESTful APIs. A three-tier model is adopted:

- **Presentation Layer (Frontend)** – Handles UI rendering, user input, and interaction.
- **Application Layer (Backend)** – Contains business logic, API endpoints, and handles data processing.
- **Data Layer (Database)** – Manages persistent storage, including user accounts, discussions, and comments.

To enhance performance and scalability, asynchronous processing and optional caching mechanisms (e.g., Redis) can be implemented.

4.2 Frontend Design

The frontend is built using modern JavaScript frameworks such as **React.js** or **Vue.js**, enabling the creation of dynamic, component-based UIs. Key design principles include:

- **Responsive Layouts:** Utilization of CSS Grid, Flexbox, and media queries to ensure compatibility across mobile, tablet, and desktop screens.
- **Single-Page Application (SPA):** Efficient routing and state management for smooth navigation without page reloads.

- **Component Breakdown:** Modular components for login/register forms, discussion threads, comment sections, profile pages, etc.

For improved UX, loading indicators, modals, and client-side validation are incorporated using libraries like **Bootstrap**, **Tailwind CSS**, or **Material UI**.

4.3. Backend Development

The backend is developed using technologies such as **Node.js with Express**, **Django**, or **Laravel**, depending on the preferred stack. It is responsible for:

- **User Authentication:** Secure login/registration using JWT (JSON Web Tokens), OAuth2, or session-based authentication.
- **CRUD Operations:** API endpoints for creating, retrieving, updating, and deleting discussions and comments.
- **Middleware Functions:** Logging, input sanitization, access control, and error handling.

Role-based permissions (e.g., admin, moderator, user) are enforced to manage user access and administrative functionalities effectively.

4.4 Database Design

A **relational database** such as **MySQL** or **PostgreSQL** is used for structured data management. The database schema includes tables such as:

- **Users:** Stores user credentials, profile info, roles, and status.
- **Discussions:** Contains thread titles, content, timestamps, author references.
- **Comments:** Stores comments linked to discussion threads and users.
- **Categories/Tags:** Optional feature to organize discussions for better navigation.

Foreign keys and indexing are used to maintain referential integrity and optimize query performance.

4.5 Real-Time and Notification Systems

For enhanced user experience, the system integrates **real-time features** using **WebSockets** (e.g., Socket.IO with Node.js) or **Firebase Realtime Database**. These allow:

- Live updates on new comments or replies
- Real-time notifications for mentions, likes, or replies
- Instant content refresh without page reloads

Push notifications and email alerts can also be integrated to notify users of new interactions outside the platform.

4.6 Security and Data Protection

Security is a key design consideration. The system implements:

- **Password Hashing:** Using bcrypt or Argon2
- **Input Validation & Sanitization:** To prevent SQL injection and XSS attacks
- **Rate Limiting & CAPTCHA:** To mitigate spam and brute force attacks
- **Role-Based Access Control (RBAC):** Ensures only authorized users can access certain features

Secure HTTPS communication, database encryption, and regular backups are also recommended for production deployment.

4.7 Scalability & Future Enhancements

The modular design ensures that the forum can scale horizontally by deploying backend services across multiple servers and vertically by optimizing individual components. Future enhancements may include:

- AI-powered moderation tools (for filtering offensive content)
- Rich media support (image, video, file uploads)
- Integration with third-party services (e.g., GitHub login, Disqus-style embedding)
- Analytics dashboard for admin insights

Containerization tools like **Docker** and orchestration platforms like **Kubernetes** may be adopted for deployment in cloud environments.

CHAPTER 5
MODULES DESCRIPTION

5.1: User Registration and Authentication Module

1. Secure registration and login forms using HTML, CSS, and JavaScript for collecting user credentials.
2. Backend authentication logic using secure password hashing (e.g., bcrypt) and optional JWT/session-based login management.
3. Validation for unique usernames, strong passwords, and properly formatted emails.
4. Role-based access control (e.g., user, moderator, admin) implemented to restrict or allow specific functionalities.

5.2: Discussion Thread Management Module

1. Users can create new discussion threads with titles, content, and optional tags or categories.
2. Threads are displayed in a paginated list with summaries, timestamps, and author details.
3. Each discussion page loads the full content along with associated comments and metadata.
4. Edit/delete options are available for the original author or moderators, following access control rules.

5.3: Commenting and Interaction Module

1. Logged-in users can post comments under discussion threads in a nested or flat format.
2. Real-time validation ensures empty or inappropriate comments are flagged or restricted.
3. Features like "Reply", "Edit", "Delete", and "Like/Upvote" enhance interaction within threads.
4. Comments auto-refresh or update in real time (using AJAX or WebSockets) to avoid page reloads.

5.4: User Profile and Dashboard Module

1. Each user has a personal profile page displaying their discussions, comments, and reputation.
2. Users can edit personal info like bio, profile picture, and account settings.
3. Dashboard offers quick access to recent activities, favorite threads, and notification settings.
4. Admin or moderator dashboards include tools for user management and content moderation.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION:

- The **Responsive Discussion Forum Website** serves as a dynamic platform designed to foster digital communication, collaboration, and knowledge exchange among users.

- With its clean user interface, responsive design, and interactive features, the platform enables users to easily register, initiate discussions, and engage in meaningful conversations through threaded comments.

- Developed using modern web technologies such as HTML5, CSS3, JavaScript, and a robust backend framework, the forum system ensures secure authentication, seamless content management, and real-time interaction. The incorporation of user roles, discussion categorization, and intuitive navigation improves the overall user experience and facilitates organized community engagement.

- This project not only showcases full-stack web development capabilities but also emphasizes modular system design, user-focused functionality, and web security practices. It lays a solid foundation for expanding into more complex social or educational platforms and can be easily customized for academic, professional, or hobbyist communities.

6.2 FUTURE ENHANCEMENTS:

While the current system provides a solid core for online discussions, several future enhancements can improve its scalability, usability, and community value:

1. Real-Time Chat and Live Commenting:

- Implement WebSocket-based live commenting so users can view new replies instantly without refreshing the page.
- Add a direct messaging or group chat feature for real-time user interactions.

2. Advanced User Profiles and Gamification:

- Enable detailed user profiles with avatars, bios, badges, and activity timelines.
- Introduce gamification through a reputation system, upvotes, and achievements to increase user engagement.

3. Moderation Tools and Reporting System:

- Equip admins and moderators with tools to review, flag, or remove inappropriate content.
- Add automated content filtering using AI for spam or offensive language detection.

4. Mobile App and PWA Support:

- Convert the forum into a **Progressive Web App (PWA)** to enable installation on mobile devices.
- Support offline mode with data synchronization for use in low-connectivity environments.

5. Thread Sorting and Smart Search:

- Implement advanced search functionality with filters for keywords, authors, tags, and dates.
- Allow users to sort threads by latest activity, popularity, or relevance.

6. Email Notifications and Subscriptions:

- Notify users via email or push notifications when they are mentioned, replied to, or followed.
- Allow users to subscribe to categories or threads for regular updates.

7. Multilingual and Accessibility Features:

- Provide localization support to cater to users from different linguistic backgrounds.
- Ensure full compliance with accessibility standards (e.g., screen reader support, keyboard navigation).

8. Analytics Dashboard:

- Integrate an admin analytics panel to track forum activity, popular discussions, and user engagement trends.
- Display insights such as most active users, discussion heatmaps, and time-based usage statistics.

CHAPTER 7

REFERENCES

1. Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. Wiley.
2. Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley.
3. Mozilla Developer Network (MDN). *HTML Documentation*. Available at: https://developer.mozilla.org/en-US/docs/Web/HTML. Accessed March 28, 2025.
4. Mozilla Developer Network (MDN). *CSS Documentation*. Available at: https://developer.mozilla.org/en-US/docs/Web/CSS. Accessed March 28, 2025.
5. Mozilla Developer Network (MDN). *JavaScript Guide*. Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript. Accessed March 28, 2025.
6. W3Schools. *HTML, CSS, and JavaScript Tutorials*. Available at: https://www.w3schools.com/. Accessed March 28, 2025.
7. Stack Overflow. *Community Help for JavaScript Game Logic*. Available at: https://stackoverflow.com/. Accessed March 28, 2025.

APPENDIX

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Discussion Forum</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <!-- Username Modal -->
    <div class="modal fade" id="usernameModal" data-bs-backdrop="static" tabindex="-1">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title">Welcome to Discussion Forum</h5>
                </div>
                <div class="modal-body">
                    <p>Please enter a username to participate in discussions:</p>
                    <input type="text" id="usernameInput" class="form-control" placeholder="Enter username" required>
                </div>
                <div class="modal-footer">
```

```
                    <button type="button" class="btn btn-primary"
id="setUsernameBtn">Continue</button>
                </div>
            </div>
        </div>
    </div>

    <!-- Header -->
    <nav class="navbar navbar-expand-lg bg-white shadow-sm">
        <div class="container">
            <a class="navbar-brand d-flex align-items-center" href="#">
                <i class="bi bi-chat-dots me-2"></i>
                Discussion Forum
            </a>
            <div class="d-flex align-items-center">
                <button class="btn btn-link me-2" id="darkModeToggle">
                    <i class="bi bi-moon-fill"></i>
                </button>
                <button class="btn btn-primary" data-bs-toggle="modal" data-bs-
target="#newPostModal">
                    <i class="bi bi-plus-circle me-2"></i>
                    New Discussion
                </button>
            </div>
        </div>
    </nav>
```

```
<!-- New Post Modal -->
<div class="modal fade" id="newPostModal" tabindex="-1">

    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Create New Discussion</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
            </div>
            <div class="modal-body">
                <form id="newPostForm">
                    <div class="mb-3">
                        <label class="form-label">Title</label>
                        <input type="text" class="form-control" id="postTitle" required>
                    </div>
                    <div class="mb-3">
                        <label class="form-label">Category</label>
                        <input type="text" class="form-control" id="postCategory" required>
                    </div>
                    <div class="mb-3">
                        <label class="form-label">Content</label>
                        <textarea class="form-control" id="postContent" rows="4"
required></textarea>
                    </div>
                </form>
            </div>
            <div class="modal-footer">
```

```
                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancel</button>
                    <button type="button" class="btn btn-primary" id="submitPost">Post Discussion</button>
                </div>
            </div>
        </div>
    </div>

    <!-- Main Content -->
    <main class="container py-4">
        <!-- Search Bar -->
        <div class="mb-4">
            <div class="input-group">
                <span class="input-group-text">
                    <i class="bi bi-search"></i>
                </span>
                <input type="text" class="form-control" id="searchInput" placeholder="Search discussions...">
            </div>
        </div>

        <!-- Discussion List -->
        <div id="discussionList" class="discussion-list"></div>
    </main>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
    <script src="script.js"></script>
```

```
</body>

</html>
```

JS

```
// State management
let currentUser = '';
let darkMode = localStorage.getItem('darkMode') === 'true';
let posts = [
    {
        id: 1,
        title: "Getting Started with Web Development",
        author: "Sarah Johnson",
        content: "What's the best way to start learning web development? I'm completely new to this field.",
        category: "Web Development",
        likes: 15,
        replies: 1,
        timestamp: "2 hours ago",
        likedBy: [],
        repliesList: [
            {
                id: 1,
                author: "John Doe",
                content: "I recommend starting with HTML, CSS, and then JavaScript. Take it step by step!",
                timestamp: "1 hour ago"
            }
        ]
    },
    {
        id: 2,
        title: "JavaScript vs Python",
        author: "Mike Chen",
        content: "Which language would you recommend for a beginner? JavaScript or Python?",
        category: "Programming",
        likes: 8,
        replies: 0,
        timestamp: "4 hours ago",
        likedBy: [],
        repliesList: []
    }
];

// DOM Elements
const usernameModal = new bootstrap.Modal(document.getElementById('usernameModal'));
const darkModeToggle = document.getElementById('darkModeToggle');
const discussionList = document.getElementById('discussionList');
const searchInput = document.getElementById('searchInput');
```

```
// Initialize
document.addEventListener('DOMContentLoaded', () => {
    if (!currentUser) {
        usernameModal.show();
    }
    updateDarkMode();
    renderPosts();
    setupEventListeners();
});

// Event Listeners Setup
function setupEventListeners() {
    // Username submission
    document.getElementById('setUsernameBtn').addEventListener('click', () => {
        const username = document.getElementById('usernameInput').value.trim();
        if (username) {
            currentUser = username;
            usernameModal.hide();
        }
    });

    // Dark mode toggle
    darkModeToggle.addEventListener('click', () => {
        darkMode = !darkMode;
        localStorage.setItem('darkMode', darkMode);
        updateDarkMode();
    });

    // New post submission
    document.getElementById('submitPost').addEventListener('click', handleNewPost);

    // Search functionality
    searchInput.addEventListener('input', handleSearch);
}

// Dark Mode
function updateDarkMode() {
    document.body.classList.toggle('dark-mode', darkMode);
    darkModeToggle.innerHTML = darkMode ? '<i class="bi bi-sun-fill"></i>' : '<i class="bi bi-moon-fill"></i>';
}

// Post Management
function handleNewPost() {
    if (!currentUser) {
        usernameModal.show();
        return;
    }

    const title = document.getElementById('postTitle').value.trim();
    const category = document.getElementById('postCategory').value.trim();
    const content = document.getElementById('postContent').value.trim();

    if (title && category && content) {
        const newPost = {
            id: posts.length + 1,
            title,
            author: currentUser,
            content,
            category,
            likes: 0,
            replies: 0,
            timestamp: "Just now",
            likedBy: [],
            repliesList:
        posts.unshift(newPost);
        renderPosts();
        bootstrap.Modal.getInstance(document.getElementById('newPostModal')).hide();
        document.getElementById('newPostForm').reset();
    }

}

function handleLike(postId) {
    if (!currentUser) {
        usernameModal.show();
        return;
    }

    const post = posts.find(p => p.id === postId);
    if (post) {
        const hasLiked = post.likedBy.includes(currentUser);
        if (hasLiked) {
            post.likedBy = post.likedBy.filter(user => user !== currentUser);
            post.likes--;
        } else {
            post.likedBy.push(currentUser);
            post.likes++;
        }
        renderPosts();
    }
}

function handleReply(postId) {
    if (!currentUser) {
        usernameModal.show();
        return;
    }

    const replyContent = document.getElementById(`replyContent-${postId}`).value.trim();
    if (replyContent) {
        const post = posts.find(p => p.id === postId);
        if (post) {
            const reply = {
                id: post.repliesList.length + 1,
                author: currentUser,
                content: replyContent,
                timestamp: "Just now"
            };
            post.repliesList.push(reply);
            post.replies++;
            renderPosts();
        }
    }
}

function handleSearch(e) {
    const searchTerm = e.target.value.toLowerCase();
    const filteredPosts = posts.filter(post =>
        post.title.toLowerCase().includes(searchTe


rm) ||
        post.content.toLowerCase().includes(searchTerm) ||
        post.category.toLowerCase().includes(searchTerm)
    );

    renderPosts(filteredPosts);
}

// Rendering
function renderPosts(postsToRender = posts) {
    discussionList.innerHTML = postsToRender.map(post => `
        <div class="card mb-4">
            <div class="card-body">
                <h5 class="card-title">${post.title}</h5>
                <div class="d-flex text-muted mb-3">
                    <span>${post.author}</span>
                    <span class="mx-2">•</span>
                    <span>${post.timestamp}</span>
                    <span class="mx-2">•</span>
                    <span><i class="bi bi-tag me-1"></i>${post.category}</span>
                </div>
                <p class="card-text">${post.content}</p>

                <div class="d-flex gap-3">
                    <button onclick="handleLike(${post.id})" class="interaction-btn
${post.likedBy.includes(currentUser) ? 'liked' : ''}">
                        <i class="bi bi-hand-thumbs-up me-1"></i>
                        ${post.likes} Likes
                    </button>
                    <button onclick="document.getElementById('replyForm-
${post.id}').classList.toggle('d-none')" class="interaction-btn">
                        <i class="bi bi-chat me-1"></i>
                        ${post.replies} Replies
```

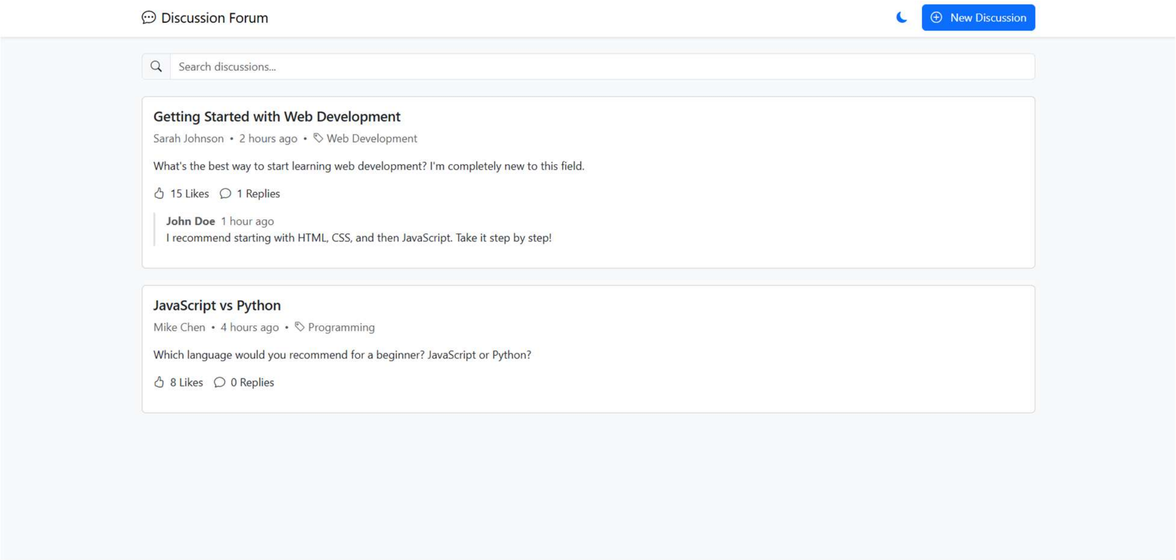
```
          </button>
        </div>

        <!-- Replies Section -->
        <div class="reply-section">
          ${post.repliesList.map(reply => `
            <div class="mb-3">
              <div class="d-flex text-muted">
                <span class="fw-bold">${reply.author}</span>
                <span class="ms-2">${reply.timestamp}</span>
              </div>
              <p class="mb-0">${reply.content}</p>
            </div>
          `).join('')}
        </div>

        <!-- Reply Form -->
        <div id="replyForm-${post.id}" class="mt-3 d-none">
          <textarea id="replyContent-${post.id}" class="form-control mb-2" rows="2"
placeholder="Write a reply..."></textarea>
          <button onclick="handleReply(${post.id})" class="btn btn-primary btn-sm">Post
Reply</button>
        </div>
      </div>

    </div>
  `).
```

OUTPUT SCREENSHOT:



Getting Started with Web Development

Sarah Johnson • 2 hours ago • Web Development

What's the best way to start learning web development? I'm completely new to this field.

15 Likes 1 Replies

John Doe 1 hour ago

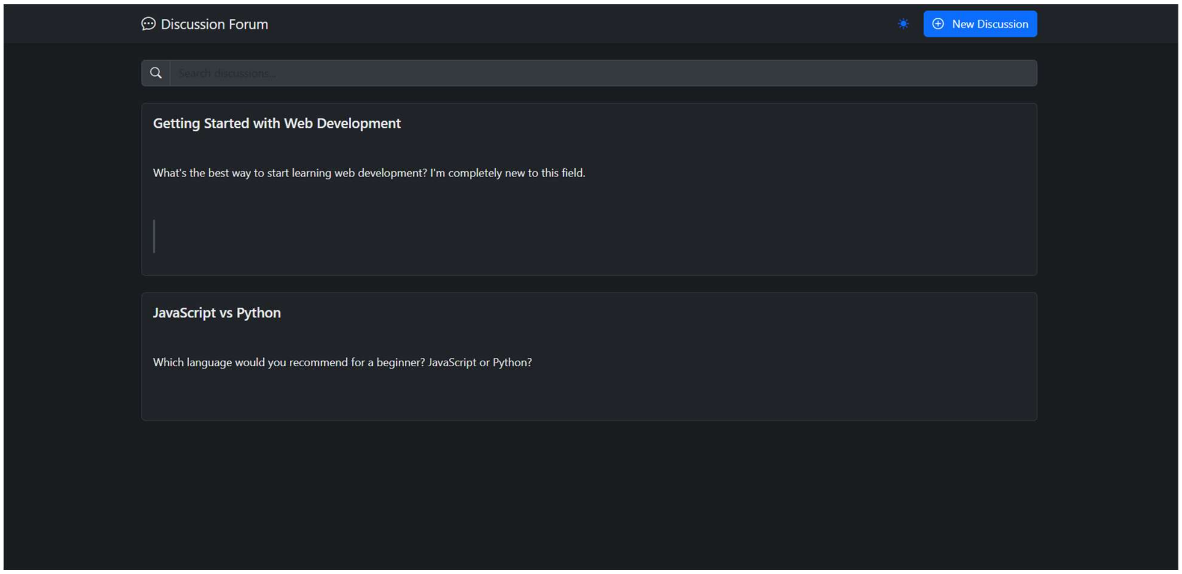
I recommend starting with HTML, CSS, and then JavaScript. Take it step by step!

JavaScript vs Python

Mike Chen • 4 hours ago • Programming

Which language would you recommend for a beginner? JavaScript or Python?

8 Likes 0 Replies



Getting Started with Web Development

What's the best way to start learning web development? I'm completely new to this field.

JavaScript vs Python

Which language would you recommend for a beginner? JavaScript or Python?