**SOCIAL MEDIA WEB APPLICATION**

#### **A PROJECT REPORT**

***Submitted by***

**Samarth Lawania - Q - 2115000902**

**Aditya Singh - B - 2115000383**

**Vedant Singh -- W - 211570007**

**Piyush Anand Jangid –W - 211570005**

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

**BACHELOR OF ENGINEERING**

**IN**

##### Computer Engineering and Application

**GLA University, Mathura**

#### **BONAFIDE CERTIFICATE**

Certified that this project report **Social Media Wb Application** is the bonafide work of **Vedant Singh, Samarth Lawania, Piyush Anand Jangid, Aditya Singh** who carried out the project work under my supervision.

|  |  |
| --- | --- |
|  |  |
| Signature of the HoD  **SIGNATURE**  Rohit Agrawal  **HEAD OF THE DEPARTMENT**  CEA Department | Signature of the Supervisor  **SIGNATURE**  Er. Sanjay Madan  **SUPERVISOR**  Technical Trainer  CEA Department |

**Table Contents:-**

1. ABSTRACT
2. INTRODUCTION
3. MOTIVATION
   1. Technical evaluation
   2. User Experience analysis
   3. Business Assessment
   4. Comparative Analysis
4. PROBLEM STATEMENT
5. WHAT IS MERN STACK?
   1. MongoDB
   2. Node JS
   3. Express JS
   4. ReactJS
6. PROJECT DESCRIPTION
   1. Responsive User Interface
   2. Efficient Data Handling
   3. Interactive Components
   4. Dynamic Content Updates
   5. Efficient Backend Processing
   6. Content Sharing, Not Real-Time Chatting
   7. Responsive Design for Cross-Device Accessibility
   8. Technologies Used
   9. Benefits
   10. Future Enhancements
   11. Conclusion

7.Result

8. Refrences

**1.ABSTRACT**

In this era of today’s world , there are lots of social media platforms like LinkedIn, Instagram, Twitter, Pinterest, Facebook for connecting with family, friends for the people of different communities.

This web application aims for the sharing of content like photos, videos, text about social issues, healthcare and welfare. Others who are on this platform can react or give their opinion about the content shared. This platform is built using MERN stack technology where MERN stands for MongoDB, Express JS, React, Nodejs.

It is designed to work well on computers as well as on mobile phones and what makes our application different is that it’s all about sharing content, not so much about chatting. It’s perfect for the people who wants to share and see stuff related to the community, social issues, healthcare and welfare. The MERN stack ensures that the platform is strong and easy to use, and it's set up as a single-page application, meaning everything happens on one page for a straightforward experience. The project includes both the behind-the-scenes work (backend) and what users see (frontend), using important tools, Social Media ideas, APIs.

**2.INTRODUCTION**

The Internet has swiftly transformed from a simple social platform to a dynamic social network used for sharing content, ideas, and information. This shift in communication styles has impacted various aspects of our lives, influencing education, communication, employment, politics, healthcare, public relations, and personal productivity. A Social Networking Service (SNS) is an online platform facilitating the building and development of relationships between people. It offers users the means to interact online with individuals who share similar interests and social purposes.

Within these online spaces, users can engage through emails, instant messages, online comments, wikis, digital photographs, videos, and blog posts. Social networking platforms provide a platform for people, including those with disabilities, to express their thoughts and ideas publicly. Social networks play a dual role as both content providers and consumers. Users have the ability to decide who can access their information, creating profiles that include details such as age, location, hobbies, and more. Some platforms even allow users to upload photos, share multimedia content, and customize their profiles.

Privacy is a significant aspect of social media platforms, as users can control who can view, edit, and add to their friend list. This control over personal information aims to maintain privacy in the online social sphere. Social networking has fundamentally altered how people communicate, share information, and engage with others. It facilitates communication and interaction over the internet, and as the popularity of social media continues to grow, new technologies become increasingly prevalent.

**3.MOTIVATION**

Creating a report on the MERN stack social media app can stem from various motivations, depending on the goals of the individual or organization behind it. Some potential motivations include:

**Technical Evaluation**:

1. Purpose: To assess the technical aspects of the MERN stack social media app, such as its architecture, design, functionality, performance, and scalability.
2. Audience: Developers who seek insights into the codebase quality and areas for improvement.

**User Experience Analysis:**

1. Purpose: To analyze the user experience of the MERN stack social media app, evaluating ease of use, engagement features, and alignment with user needs.
2. Audience: Designers aiming to enhance usability and improve overall user experience.

**Business Assessment:**

1. Purpose: To assess the business potential of the MERN stack social media app, examining market demand, revenue potential, and growth opportunities.
2. Audience: Entrepreneurs considering the launch of a social media app and investors looking to evaluate its viability.

**Comparative Analysis:**

1. Purpose: To compare the MERN stack social media app with other apps in the market, considering features, performance, user base, and revenue.
2. Audience: Marketers seeking to identify competitive advantages and weaknesses, aiding in the development of a strategic marketing approach to attract users.

These motivations reflect a range of perspectives, from technical intricacies and user-centric considerations to business viability and market positioning. The report, shaped by these motivations, serves as a valuable tool for decision-makers in their respective domains, offering insights tailored to their specific needs and objectives.

**4.PROBLEM STATEMENT**

In the rapidly evolving landscape of social media, it's crucial to delve into how these platforms shape user behavior and influence our social connections. Specifically, we're honing in on our social media app developed with the MERN stack. The goal is to comprehensively assess its performance, user experience, and business potential. This evaluation is not only about understanding how well our app is doing but also about finding ways to make it even better.

Moreover, we're keen on comparing our social media app with others out there. This comparative analysis helps us see where our app excels and where there might be room for improvement. We want to know what makes our app unique and how it stacks up against the competition.

What sets our app apart is its distinct focus—it's not an chatting platform. Instead, our app is designed specifically for sharing posts and fostering interactions around those posts. This approach caters to users who prefer meaningful engagement through shared content rather than mere conversations.

The challenge we're tackling head-on is optimizing our platform to meet users' content-sharing needs seamlessly. We want our app to be a standout, user-friendly space where people can easily share and interact with posts. In essence, our objective is to make our social media app the go-to place for a simple, enjoyable, and distinctive content-sharing experience.

**5.What is MERN Stack :-**

The MERN stack is a framework specifically designed for web application development, with MongoDB, Express JS, ReactJS, and NodeJS forming its integral components. Each element plays a distinct role in the creation of a web application:

**MongoDB:**

Role: MongoDB serves as the document-oriented, No-SQL database where the application data is stored. It provides a flexible and scalable solution for handling data.

**NodeJS:**

Role: NodeJS acts as the JavaScript runtime environment, enabling the execution of JavaScript code directly on the machine rather than in a browser. It plays a fundamental role in server-side scripting.

**Express JS:**

Role: Express JS is a framework that sits on top of NodeJS, serving as a powerful tool to build the backend of a website. It utilizes NodeJS functions and structures to create the server-side logic. Express JS was developed to address the need for building websites with NodeJS, which was initially designed for running JavaScript on computers.

**ReactJS:**

Role: Developed by Facebook, ReactJS is a library used to construct the user interface (UI) elements of a single-page web application. Users interact with the UI components in the front-end, situated within the browser. The backend of the application, located on a server, is managed by Express JS, which is built upon NodeJS.

In the application flow, when a user interacts with the ReactJS UI components, a request to modify data is sent to the Express server. This server, built on NodeJS, handles the request. If data from the MongoDB database is required, Express fetches the information and sends it back to the application's front end, where it is displayed to the user. This seamless interaction between MongoDB, Express JS, ReactJS, and NodeJS enables the creation of dynamic and responsive web applications.

**6. PROJECT DESCRIPTION**

**Responsive User Interface:**

* Utilizing ReactJS's component-based architecture to craft a responsive and visually appealing user interface.
* Implementing a flexible layout that adapts seamlessly to various screen sizes and devices.

**Efficient Data Handling:**

* Leveraging React's virtual DOM to efficiently manage and update the user interface, ensuring smooth interactions without full page reloads.
* Implementing optimized data fetching strategies to enhance performance and minimize server requests.

**Interactive Components:**

* Building interactive components that respond dynamically to user actions, providing a fluid and engaging experience.
* Incorporating state management to maintain the application's state on the client side.

**Dynamic Content Updates:**

* Enabling real-time updates and changes without requiring constant communication with the server.
* Implementing React's capabilities for efficient data binding to ensure immediate reflection of changes in the user interface.

**Efficient Backend Processing:**

* Leveraging the MERN stack for server-side logic and data queries to ensure seamless content sharing.
* Optimizing backend processes for swift response times, particularly tailored for content-oriented interactions.

**Content Sharing, Not Real-Time Chatting:**

* Focusing on content sharing functionalities rather than real-time chatting.
* Enabling users to post, view, and interact with diverse content types seamlessly.

**Responsive Design for Cross-Device Accessibility:**

* Implementing a responsive design to ensure accessibility across various devices and screen sizes.
* Enhancing user engagement through a visually appealing and user-friendly interface.

**Technologies Used:**

* Frontend: ReactJS
* Backend: MERN Stack (MongoDB, Express JS, ReactJS, NodeJS)
* JavaScript and ES6+ for coding functionalities.

**Benefits:**

* Faster loading times and improved performance due to client-side rendering.
* Enhanced user interaction and engagement through dynamic content updates.
* Scalable architecture allowing for easy integration of additional features.

**Future Enhancements:**

* Integration of additional React libraries or frameworks for extended functionality.
* Implementation of advanced state management solutions for larger-scale applications.
* Continuous optimization for performance enhancements and an ever-improving user experience.

.

**Scope:**

This project will encompass the development of ChatFusion, including user registration, profile management, content sharing, and real-time chat. It will provide features for user generated content and messaging capabilities as well .

**Methodology:**

The project will employ the following methodologies, tools, and technologies:

• Programming Languages: Node JS (for Backend Development), React (for FrontEnd Development)

• Software: Vs Code, Firebase for backend services, WebSocket for real-time chat, web development tools.

• Hardware: System for testing, Web server for hosting web components.

**Proposed System:**

ChatFusion will be an intuitive and user-friendly social media app with a well-designed user interface. Users can register, create profiles, share content, and engage in real-time chat conversations. The chat system will include one-on-one chat and message status indicators.

**Features:**

• User Registration and Profile Management

• Content Sharing

• Liking and Commenting the Posts

• Real-time One-on-One Chat

• Message Delivery Confirmation

• User Authentication and Data Security

**Working**

**User Interaction:**

* Users access the web application through their browsers, interacting with the frontend developed using ReactJS.
* The user interface provides options for content creation, exploration, and interaction.

**Content Creation:**

* Users can create and share various types of content, such as text, images, or multimedia, through dedicated forms or interfaces.
* The content creation triggers requests to the backend server, which is powered by the MERN stack (MongoDB, Express JS, ReactJS, and NodeJS).

**Backend Processing:**

* The backend, running on NodeJS and Express, receives the content creation requests.
* It processes the requests, which may involve validating user input, storing content data in MongoDB, and handling any required server-side logic.

**Database Interaction:**

* MongoDB, as the document-oriented database, stores and manages the content data efficiently.
* User-generated content, along with relevant metadata, is stored in collections, allowing for easy retrieval and manipulation.

**Content Retrieval and Display:**

When users access the application to view content, ReactJS on the frontend sends requests to the backend.

The backend fetches relevant content data from MongoDB and sends it back to the frontend for display.

**Client-Side Rendering:**

ReactJS, on the client side, dynamically renders the received content, providing a seamless and responsive user experience.

Users can navigate through the content, interact with it (e.g., liking, commenting), and explore various features offered by the application.

**Responsive Design:**

The application's responsive design ensures that users can access and interact with content across different devices and screen sizes.

**No Real-Time Chatting:**

Unlike real-time chatting applications, the web app focuses on asynchronous content sharing. Users can post and interact with content at their convenience.

**Scalability and Performance:**

The MERN stack's scalability allows the application to handle a growing number of users and content.

Client-side rendering reduces the load on the server, contributing to improved performance and faster content retrieval.

**Conclusion**

The goal of this project is to create a multifaceted social media platform with integrated chat functionality to foster user engagement and seamless communication. Our web application aims to address the growing need for a unified platform that combines social networking and messaging features using mern stack. The web application is fully functional and responsive and provides great user experience along side serving a purpose. Social Media today owns the market, casting influence and shaping behavioral and purchase patterns. As a close friend casts much more influence, through the connectivity they offer, social media posts became the most influential source across all generations.

Our web application represents an exciting endeavor to create an social media and. With a well-structured plan and a dedicated team, we are confident in our ability to deliver a user-friendly and feature-rich application that enhances online social interactions.

**9. Result : -**

* Photo Posting: Users posted a total of any number of photos, utilizing intuitive features such as filters and captions.
* User Interaction: The app recorded any number of likes per photo on average, with active engagement in comments..

**10.References:**

• MDN Documentation

• Node Documentation

• Express Documentation

• Mongoose Documentation