

MERN STACK BLOG APP

Project report submitted to



by

Somaansh Virmani

Registration No: 215805032

Department of Computer Science and Engineering

Manipal Institute of Technology

Bengaluru

Under the supervision of

Dr. Kumar Shekhar Roy

Associate Professor

Department of Computer Science and Engineering

Manipal Institute of Technology

Bengaluru

Manipal Institute of Technology

Bengaluru Campus-560064, Karnataka, India.

Acknowledgement

Apart from the efforts of team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

The completion of any inter-disciplinary project depends upon cooperation, co-ordination, and combined efforts of several sources of knowledge.

We are eternally grateful to our teacher **Kumar Sekhar Roy** for his even willingness to give us valuable advice and direction under which we executed this project. His constant guidance and willingness to share his vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

Certificate

This is to certify that the **Project report** entitled “**MERN Blog App**” submitted by **Somaansh Virmani (Registration No:215805032)** as the record of the project work carried out by them, is accepted as the Project report submission in partial fulfilment of the requirements for the award of degree of Bachelor of Technology (BTech).

Examiners Name

Signature with the Date

.....

.....

.....

.....

.....

.....

Table of Contents

	Page No.
ACKNOWLEDGEMENT	ii
ABSTRACT	1
Chapter 1: INTRODUCTION	
1.1 Introduction	2
1.2 Scope	2
1.3 Problem statement	3
Chapter 2: BACKGROUND MATERIAL	
2.1 Background overview	7
2.2 Technologies Involved	8
Chapter 3: METHODOLOGY	
3.1 Developing methodology	9
3.2 Overall Project Timeline	11
Chapter 4: IMPLEMENTATION	
4.1 Modules	14
4.2 Prototype	18
Chapter 5: CONCLUSIONS & FUTURE SCOPE	
5.1 Conclusion	20
5.2 Future Scope	20
REFERENCES	21

ABSTRACT

A **MERN** stack blog app is a comprehensive web application that leverages the MERN stack—MongoDB, Express.js, React.js, and Node.js—to provide a full-featured blogging platform. This abstract outlines the key components and functionalities of such an app.

MongoDB serves as the database layer, offering a schema-less NoSQL database that stores data in JSON-like documents. This flexibility allows for the storage of diverse content types found in blog posts, such as text, images, and videos, as well as user data and comments.

Express.js operates as the backend framework running on Node.js, the JavaScript runtime environment. Together, they manage server-side logic, API endpoints, and interaction with the MongoDB database. Express.js simplifies the creation of robust RESTful APIs that serve as the communication bridge between the frontend and the database.

React.js is employed for the frontend, providing a dynamic and responsive user interface. It enables the creation of a single-page application (SPA) where users can seamlessly navigate through different parts of the blog without the need for page reloads. React's component-based architecture allows for reusable UI components, making the app scalable and maintainable.

The app features user authentication, allowing individuals to sign up and login. Users can create, edit, and publish blog posts, with the option to categorize them by tags or topics. A commenting system enables interaction, while search functionality allows users to find content easily.

Security is a paramount concern, and the app incorporates best practices to protect user data and prevent unauthorized access. Performance optimization ensures that the app can handle high traffic and large volumes of data efficiently.

In summary, a MERN stack blog app is a modern solution for online content sharing and community building, offering a blend of powerful technologies that work in harmony to deliver a seamless blogging experience.

INTRODUCTION

Introduction

A blog serves as a platform for sharing information, opinions, and experiences with a wider audience. It fosters communication and knowledge exchange. This app is built on the MERN tech stack which comprises MongoDB, Express.js, React.js, and Node.js.

React is a powerful JavaScript library for building dynamic and interactive user interfaces (UIs). Developed by Facebook, React is known for its component-based architecture, which allows developers to create reusable UI elements

Node.js is a cross-platform, open-source JavaScript runtime environment that can run on Windows, Linux, Unix, macOS, and more. Node.js runs on the V8 JavaScript engine, and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting.

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs

MongoDB is a source-available, cross-platform, document-oriented database program. Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with optional schemas.

Purpose

The purpose of this app is to provide a platform for individuals or organizations to publish, manage, and distribute content online. It serves as a digital space for expression, sharing knowledge, and engaging with a community.

Scope

The scope of a MERN stack blog app encompasses creating a dynamic blogging platform with user authentication, content management, interactive features, and responsive design, leveraging MongoDB, Express.js, React.js, and Node.js for a seamless full-stack experience

Problem Statement

Creating a comprehensive platform that allows users to express their thoughts, share knowledge, and engage with a community through written content. The app should provide a seamless experience for both content creators and readers, ensuring ease of use, security, and interactivity:

- **User Experience:** The app must have an intuitive interface that is easy to navigate for posting and reading content. It should be responsive, ensuring compatibility with various devices and screen sizes.

- **Content Management:** Users should be able to create, edit, publish, and delete blog posts. The app should support text formatting, media uploads, and categorization of posts by tags or topics.
- **User Authentication and Profiles:** Secure user authentication is essential for managing user accounts. Profiles should display user information, their posts, and activity on the platform.
- **Interactivity:** The app should allow readers to interact with content through comments, likes, and shares. A moderation system should be in place to manage user-generated content.
- **Performance and Scalability:** The app must be optimized for performance, handling large volumes of data and traffic without compromising speed or availability.
- **Security:** Implement robust security measures to protect user data and prevent unauthorized access or data breaches.
- **Legal and Compliance:** Ensure compliance with data protection regulations and provide clear terms of use and privacy policies.

BACKGROUND MATERIAL

Background Overview

Your MERN stack blog app is well-structured, with clear separation of concerns between the client and server modules. Here's a detailed abstract based on your description:

The project is composed of two primary modules: the client application and the server. The server module is built to handle HTTP requests, including GET, PUT, and POST operations, which are essential for retrieving, updating, and creating data, respectively. It plays a pivotal role in user authentication, managing session states through cookies, and ensuring secure access to the system. Additionally, the server acts as the communication hub, interfacing with the MongoDB database to perform data transactions and persistence.

On the frontend, the client application is developed using React, a powerful library for building user interfaces. The client is tasked with presenting data from the server in an accessible and aesthetically pleasing manner. It handles user input for critical functions such as user registration and authentication, as well as the creation and deletion of blog posts. This module is designed with a focus on user experience, featuring a collection of React components that are seamlessly connected via React-router. This setup not only facilitates smooth site navigation but also enhances the overall user experience by providing a fluid and intuitive interaction flow.

The architecture of this MERN stack blog app reflects a modern approach to web application development, emphasizing modularity, scalability, and a user-centric design. By leveraging the strengths of the MERN stack, the app is positioned to offer a robust platform for blogging, content management, and community engagement.

Technologies Involved

- | | |
|---------------------|-------------------|
| 1. Node.js | 9. Date-fns |
| 2. Express.js | 10. React-quill |
| 3. React.js | 11. Cors |
| 4. MongoDB | 12. Brypt |
| 5. Mongoose | 13. Cookie parser |
| 6. React-router-dom | 14. Jsonwebtoken |
| 7. Npm | 15. Multer |
| 8. Nodemon | 16. Fs |

METHODOLOGY

Modules

1) Front-End

- a) App: Manages routing between various pages
- b) Index: Initializes the react DOM
- c) React Components
 - i) Layout: Contains the blueprint of a page, including header and content.
 - ii) Post: Manages how posts are shown at the index page.
 - iii) User Context: Used for creating the user's context in the web page for the website.
 - iv) Header: Contains the information that the header of the page should show.
 - v) Editor: Contains the react-quill component and subsequent dependencies, for creating and editing posts.
- d) React Pages
 - i) Create-Post: Contains the React component that is used for creating pages
 - ii) Edit-post: Used for taking user input data which would be required for editing a post
 - iii) Index-page: Is the home page of the blog site, contains a brief of all the blogs
 - iv) Login-page: Contains a web form for user authentication
 - v) Post-page: Contains all the information of a blog post
 - vi) Register-page: Contains a web form for user registration

2) Back-End

- a) Index.js: Handles all the GET, POST and PUT calls to the server and performs the required operation. It is also responsible for user registration, along with password encryption into the database along with user authentication. It also handles cookies and the user session.
- b) Database Models
 - i) Post: contains the schema for the individual posts that a user would make. It has the following information:-

```
_id: ObjectId('6620085782b9a63dfbc48757')
title: "MyPost 23"
summary: "new Post fallout 3"
content: "<p>cool game</p>"
cover: "uploads\4f9845d7e5b8b26111e84ba161c65580.webp"
author: ObjectId('661fb8e15d1545b81f2d1b6e')
createdAt: 2024-04-17T17:35:19.335+00:00
updatedAt: 2024-04-17T21:15:06.453+00:00
__v: 0
```

- ii) User: Contains information of all the various users on the platform. Following image contains an example of an object inside the User schema.

```
_id: ObjectId('661fb5638b2a50bf89d0726c')
username: "hello"
password: "$2b$10$zx/B9Se1Dr4lMd9Xe13NE.P51UuKqGkUbh.I4ITEfx4sBYL0.LVZu"
__v: 0
```

Prototype

Index Page:

MyBlog

[Login](#) [Register](#)



Title 2

my_name Apr 18, 2024 13:29

hi this is a summary nice to mete you



my title post of this post is a title

somaansh Apr 17, 2024 23:19

this is a short summary of a poost regarding posting posts



MyPost 23

somaansh Apr 17, 2024 23:05

new Post fallout 3

Login Page:

MyBlog

[Login](#) [Register](#)

Login

username
password
Login

Registraion Page:

MyBlog

[Login](#) [Register](#)

Register

username
password
Register

Create Post Page:

MyBlog

[Create new post](#) [Logout](#)

Choose File

No file chosen

Normal

⌵

B

I

U

S

”

☰

☷

☰

☷

🔗

🖼️

↶

Create Post

Post Page:

MyBlog

[Create new post](#) [Logout](#)

Title 2

2024-04-18 13:29:10
by my_name



wow this a cool

this is a less cool

- there's this also
- and this
- wow

1. damn


Edit Post Page:

MyBlog

[Create new post](#) [Logout](#)

my title post of this post is a title

2024-04-17 23:09:15
by somaansh

 Edit the post



this post has an image of ps5 in it

CONCLUSION & FUTURE SCOPE

Conclusion

In conclusion, the MERN stack blog app represents a powerful fusion of modern technologies, enabling content creators to share their insights, stories, and expertise with a global audience. By leveraging MongoDB for data storage, Express.js for server-side logic, React.js for dynamic user interfaces, and Node.js for seamless communication, this app achieves a harmonious balance between functionality, scalability, and user experience.

Future Scope

- 1) Delete Post: There is still no functionality to delete a post, this can be added to the project for allowing users to remove unwanted posts.
- 2) Optimization and performance: The app can further be optimized by using hooks and a better database solution. Performance can also be improved on by switching to a different front end framework.
- 3) Deployment: The app is only running on localhost, it can be hosted on a server so it is accessible by a wider audience. It's scope is further enhanced by the opportunity to perform server-side optimizations using Nginx