BLOGIFY: A BLOG WEB APPLICATION

A project report submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

in

COMPUTER APPLICATIONS

by

PRIYAVANDNA RANAWAT (205122069)

DEPARTMENT OF COMPUTER APPLICATIONS

NATIONAL INSTITUTE OF TECHNOLOGY

TIRUCHIRAPPALLI – 620015

MAY 2024

Name: Priyavandna Ranawat
Roll No: 205122069

BONAFIDE CERTIFICATE

This is to certify that the project entitled “ **Blogify : Blog Web Application”** is a project work successfully done by **Priyavandna Ranawat (205122069)** in partial fulfilment of the requirements for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** from **NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI**, during the academic year **2024-25**.

Dr. (Mrs.) B. Janet	**Dr. Michael Arock**
Project Guide	Head of the Department

Project Viva held on ....

Name: Priyavandna Ranawat
Roll No: 205122069

ABSTRACT

In the digital era, the significance of online content creation and sharing has skyrocketed. Blogging platforms have become pivotal in facilitating individuals and businesses to express ideas, share experiences, and engage with audiences worldwide. Leveraging modern web development technologies, this project explores the creation of a Blog Web Application using the MERN (MongoDB, Express.js, React.js, Node.js)stack.

This project aims to deliver a fully functional and aesthetically pleasing blog web application with essential features such as user authentication, CRUD (Create, Read, Update, Delete) operations for blog posts, comment sections, and responsive design for various devices. Additionally, the application will incorporate features like user profile management, social sharing capabilities, and search functionality to enhance usability and engagement.

Name: Priyavandna Ranawat
Roll No: 205122069

ACKNOWLEDGEMENT

Every project, big or small, is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. I sincerely appreciate the inspiration, support, and guidance of all those people who have been instrumental in making this project successful.

I express our deep sense of gratitude to **Dr. G. Aghila**, Director, National Institute of Technology, Tiruchirappalli for giving me an opportunity to do this project.

I wish to extend my sincere gratitude to **Dr. Michael Arock**, Professor and Head, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for supporting and encouraging me to carry out this project work and duly evaluating my progress.

I express my heartfelt gratitude to my internal guide **Dr(Mrs.) B. Janet** , Professor int the Department of Computer Applications, National Institute of Technology, Tiruchirappalli for his immense support and guidance during the study.
I express my sincere and heartfelt gratitude to **Project Evaluation Committee**, Department of Computer Applications, National Institute of Technology, Tiruchirappalli. I am sincerely thankful for the constant support, care, guidance, and regular interaction throughout my project.

Finally, I would like to express my regards to all the faculty members of the Department of Computer Applications and others who have helped me develop this project directly or indirectly.

TABLE OF CONTENTS

Title **Page**

Bonafide Certificate..2
Abstract..3
Acknowledgment...4
Table of Contents...5
Title of the Project..6
1. Introduction..7
2. Objective..8
3. Platform...9
    a. Hardware Requirements
    b. Software Requirements
    c. Additional Requirements
4. Methodology..10
5. Design and Development..12
    a. Process Description
6. Challenges..16
7. Conclusion and Future Work..18
8. Appendix...21-24
9. References and Biblography...25-26

Name: Priyavandna Ranawat
Roll No: 205122069

Title of the Project:

BLOGIFY: BLOG WEB APPLICATION

1.Introduction:

In today's digital landscape, the demand for platforms enabling individuals and organizations to share content, ideas, and experiences has surged dramatically. Blogging has emerged as a prominent medium for expressing opinions, disseminating information, and building communities across diverse topics and interests. With the proliferation of web technologies, creating a robust and user-friendly blog web application has become increasingly accessible.

This introduction sets the stage for exploring the development of a Blog Web Application using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The MERN stack represents a powerful combination of technologies that streamline the development process, offering flexibility, scalability, and efficiency in building modern web applications.

The aim of this project is to guide developers through the process of creating a feature-rich and dynamic blog platform, leveraging the capabilities of each component within the MERN stack. By harnessing MongoDB's versatile NoSQL database for data storage, Express.js for building robust backend APIs, React.js for crafting interactive user interfaces, and Node.js for server-side logic execution, developers can construct a comprehensive solution tailored to the needs of bloggers and their audiences.

Throughout this exploration, key aspects such as user authentication, CRUD operations for managing blog posts, comment functionalities, responsive design for optimal viewing across devices, and additional features like user profiles, social sharing, and search capabilities will be addressed.

By delving into the development of a blog web application using the MERN stack, developers can gain valuable insights into modern web development practices, modular architecture, and best practices for creating scalable and engaging user experiences. Ultimately, this project aims to equip developers with the knowledge and tools necessary to embark on their journey of building sophisticated web applications tailored to the blogging ecosystem.

2.Objective:

Intuitive User Interface: Design a user-friendly interface prioritizing ease of navigation, enabling individuals of varying technical expertise to efficiently navigate and utilize the blog application.

Reliable Local Storage: Employ local storage technology to bolster the application's speed and responsiveness, providing users with a dependable platform for storing and accessing their blog posts.

Versatile Editing Capabilities: Empower users with a diverse array of editing features, enabling them to customize their posts to align with their preferences, whether through text formatting, bullet points, or other editing options.

Secure Data Management: Uphold the security of user data through the implementation of robust data management practices, safeguarding sensitive information stored within the blog application.

Seamless Post Deletion: Simplify the post deletion process, allowing users to effortlessly declutter their blog space and efficiently manage their content.
Customization Options: Offer users a spectrum of customization choices, enabling them to tailor their blog experience to suit their unique preferences and workflow.

CRUD Functionality: Implement Create, Read, Update, and Delete (CRUD) operations to empower users to manage their blog content effectively. This functionality allows users to create new posts, view existing posts, edit or update posts as needed, and delete posts when they are no longer required, providing a comprehensive set of tools for content management within the blog application

3. Platform:

- **Hardware Requirements:**
  Processor: Intel Core i5 or an equivalent AMD processor
  Memory: Minimum of 8GB RAM
  Storage: At least 128GB SSD storage
  Networking: Ethernet or Wi-Fi connectivity
  Web Browser: Compatible with Chrome, Firefox, or Safari

- **Software Requirements:**
  Operating System: Compatible with Windows 10, macOS, or Linux
  Development Environment: MERN stack, npm, Solidity compiler (if applicable)
  Front-End Technologies: Utilizing HTML, CSS, Bootstrap, and React

These outlined hardware and software prerequisites lay the groundwork for establishing and operating a blog application using the MERN stack. Adequate processor capability, memory allocation, and storage capacity, alongside a stable internet connection, are essential for seamless development and deployment. The specified operating systems ensure broad compatibility, while the incorporation of front-end technologies and development environment tools facilitates the creation of a dynamic and user-friendly blogging interface. Additionally, the mention of the Solidity compiler hints at the potential integration of blockchain technology for enhanced features, if desired.

4. Methodology:

The methodology for developing Blogify, the blog web application, involves a systematic and iterative approach to ensure a robust and user-friendly product. The development process encompasses several key stages:

Project Planning:

- Define the project scope, objectives, and key features.
- Conduct a market analysis to identify user needs and preferences.
- Establish a development timeline and allocate resources accordingly.

Requirement Analysis:

- Gather detailed requirements through stakeholder meetings, user surveys, and competitor analysis.
- Define the essential features, user interface elements, and technical specifications for SnapNotes.

Design Phase:

- Create wireframes and prototypes to visualize the user interface and overall user experience.
- Collaborate with UX/UI designers to ensure a seamless and intuitive design.
- Obtain feedback from potential users to refine the design.

Technology Stack Selection:

- Choose the appropriate technologies and frameworks based on the project requirements, such as front-end and back-end technologies, databases, and APIs.

Front-end Development:

- Develop the user interface using HTML, CSS, and JavaScript.
- Implement responsive design to ensure compatibility with various devices and screen sizes.
- Incorporate local storage technology for efficient note management.

Back-end Development:

- Build the server-side logic and functionality using a Node Js and Express Js.
- Integrate local storage mechanisms for efficient data handling.

Testing:

- Conduct thorough testing, including unit testing, integration testing, and user acceptance testing.
- Identify and fix bugs or issues promptly.
- Ensure cross-browser compatibility and optimal performance.

Throughout the development process, collaboration and communication among the development team, stakeholders, and potential users are critical to ensure the alignment of the final product with user expectations and market needs. The methodology is flexible, allowing for adjustments based on feedback and changing requirements**.**

6. Design and Development:

1. **User Experience (UX) Design:**

- Conduct research to understand the target audience and their preferences in blog consumption.
- Develop user personas and journey maps to visualize how users interact with the blog application.
- Create wireframes outlining the application's structure and layout, focusing on user-centric design principles.
- Refine wireframes iteratively based on feedback from user testing and stakeholder input.

2. **User Interface (UI) Design:**

- Translate wireframes into high-fidelity UI designs, incorporating the brand identity and visually appealing elements.
- Design intuitive navigation elements to ensure seamless user interaction.
- Develop interactive prototypes for user testing and stakeholder feedback.

3. **Feedback and Iteration:**

- Collect feedback on wireframes and prototypes through usability testing and stakeholder reviews.
- Iterate on the design based on user input, addressing any usability concerns.
- Collaborate closely with front-end developers to ensure design feasibility within the MERN stack.

4. **Front-end Development:**

- **Implementation of UI Design:**
- Develop the user interface based on finalized UI designs, ensuring consistency with approved prototypes.
- Implement animations and interactive elements to enhance user engagement.

5. **Local Storage Integration:**

- Integrate local storage mechanisms into the front-end logic for efficient post creation, editing, and deletion.
- Synchronize local storage with the user's device to maintain data consistency.

6. **Responsive Design:**

- Implement responsive design practices to provide a consistent user experience across various devices and screen sizes.

7. Challenges:

Limited Storage Capacity:

- Challenge: The blog application's reliance on local storage may lead to capacity constraints determined by browser settings.
- Solution: Employ data management strategies such as periodic archiving or empower users with tools for effective data management.

Data Security and Privacy:

- Challenge: Storing sensitive data locally raises concerns about security and privacy, exposing it to local threats.
- Solution: Encrypt sensitive data before local storage and educate users on securing their local storage, including robust password practices.

Cross-Device Synchronization:

- Challenge: Achieving seamless synchronization across devices without a dedicated backend poses challenges.
- Solution: Explore cloud-based synchronization or user-initiated data transfer methods, accompanied by clear instructions for users on syncing data.

Data Loss Risks:

- Challenge: Users face potential data loss risks due to device loss, damage, or intentional/unintentional clearing of local storage.
- Solution: Implement backup options or encourage regular data exports, and provide warning prompts before critical actions like note deletion**.**

Offline Access:

- Challenge: Maintaining functionality offline without a dedicated backend is challenging.
- Solution: Implement offline access features for creating, editing, and viewing notes, while clearly communicating offline limitations to users.

User Authentication:

- Challenge: Implementing secure authentication without a backend presents challenges.

- Solution: Utilize third-party authentication services or client-side authentication methods, and transparently communicate security measures.

Limited Collaboration Features:

- Challenge: Enabling collaboration features like note sharing is complex without backend support for user access management.
- Solution: Simplify collaboration features or opt for lightweight solutions such as shareable links with read-only access.

Browser Compatibility:

- Challenge: Varying browser behaviors towards local storage can lead to inconsistencies.
- Solution: Thoroughly test across browsers, implement fallback mechanisms, and offer recommendations for optimal browser usage.

Scalability Concerns:

- Challenge: Local storage solutions may encounter scalability issues as data volume increases.
- Solution: Optimize storage processes, prompt users to archive or delete old data, and provide guidance for improved performance.

8. Conclusion and Future Work:

Conclusion:

In conclusion, the development of a blog web application using the MERN stack presents a multifaceted endeavor with numerous challenges and opportunities. Throughout the project, we have encountered and addressed various technical and user-centric challenges, including scalability, data security, cross-device synchronization, and offline access.

By leveraging the capabilities of MongoDB, Express.js, React.js, and Node.js, we have created a robust and feature-rich platform for users to create, edit, and share blog posts effortlessly. The intuitive user interface, coupled with versatile editing capabilities, ensures a seamless and engaging blogging experience for users of all technical backgrounds.

However, challenges such as limited storage capacity, data security concerns, and cross-device synchronization complexities required careful consideration and innovative solutions. Implementing encryption for sensitive data, providing backup options, and optimizing storage processes were essential steps in mitigating these challenges and ensuring a reliable and secure blogging platform.

Furthermore, the project has underscored the importance of user experience design and thorough testing across different browsers and devices. By prioritizing user feedback and iteratively refining the design and functionality, we have created a blog application that meets the needs and expectations of our target audience.

In summary, the development of the blog web application using the MERN stack has been a rewarding journey, highlighting the significance of robust architecture, meticulous planning, and continuous iteration. As we conclude this project, we are proud to deliver a scalable, secure, and user-friendly platform that empowers users to share their thoughts, ideas, and experiences with the world

Future Work:

Advanced Collaboration Features: Expand collaboration capabilities by implementing features such as real-time co-editing, user mentions, and collaborative commenting to enhance user engagement and interaction.

Enhanced Search Functionality: Integrate advanced search functionality, including keyword search, filters, and sorting options, to help users easily discover relevant content within the blog application

Integration with External Services: Explore integrations with external services such as social media platforms, content syndication networks, and analytics tools to expand the reach of the blog application and provide valuable insights to users.

Mobile Application Development: Develop native mobile applications for iOS and Android platforms to offer users a seamless and optimized experience on mobile devices, leveraging technologies like React Native for cross-platform development.

Monetization Strategies: Implement monetization strategies such as ad placements, sponsored content, subscription models, or premium features to generate revenue and sustain the growth and development of the blog application

Content Moderation and User Management: Introduce content moderation tools and user management features to maintain quality standards, enforce community guidelines, and manage user permissions effectively as the user base grows

Internationalization and Localization: Support multiple languages and regions by implementing internationalization (i18n) and localization (l10n) features to cater to a diverse audience and enhance accessibility for users worldwide.

Blockchain Integration: Explore opportunities for integrating blockchain technology to enhance data security, transparency, and integrity, such as blockchain-based authentication, content ownership verification, or decentralized storage solutions

Performance Optimization: Continuously optimize performance by implementing techniques such as code splitting, lazy loading, server-side rendering (SSR), and caching mechanisms to improve page load times and overall user experience.

Community Engagement and Feedback: Foster a thriving community around the blog application by encouraging user feedback, hosting discussions, and soliciting feature requests to prioritize future development efforts based on user needs and preferences

8. Appendix:

App.js:

```
import { useState } from 'react';

import { Box } from '@mui/material';
import { BrowserRouter, Routes, Route, Navigate, Outlet } from 'react-router-
dom';

//components
import DataProvider from './context/DataProvider';
import Header from './components/header/Header';
import Home from './components/home/Home';
import CreatePost from './components/create/CreatePost';
import DetailView from './components/details/DetailView';
import Update from './components/create/Update';
import About from './components/about/About';
import Contact from './components/contact/Contact';
import Login from './components/account/Login';

const PrivateRoute = ({ isAuthenticated, ...props }) => {
  const token = sessionStorage.getItem('accessToken');
  return isAuthenticated && token ?
    <>
      <Header />
      <Outlet />
    </> : <Navigate replace to='/account' />
};

function App() {

  const [isAuthenticated, isUserAuthenticated] = useState(false);

  return (
    <DataProvider>
      <BrowserRouter>
        <Box style={{ marginTop: 64 }}>
          <Routes>
            <Route path='/account' element={<Login
isUserAuthenticated={isUserAuthenticated} />} />

            <Route path='/' element={<PrivateRoute
isAuthenticated={isAuthenticated} />} >
              <Route path='/' element={<Home />} />
            </Route>

            <Route path='/create' element={<PrivateRoute
isAuthenticated={isAuthenticated} />} >
              <Route path='/create' element={<CreatePost />} />
            </Route>

            <Route path='/details/:id' element={<PrivateRoute
isAuthenticated={isAuthenticated} />} >
              <Route path='/details/:id' element={<DetailView />} />
            </Route>

            <Route path='/update/:id' element={<PrivateRoute
isAuthenticated={isAuthenticated} />} >
              <Route path='/update/:id' element={<Update />} />
            </Route>

            <Route path='/about' element={<PrivateRoute
isAuthenticated={isAuthenticated} />} >
              <Route path='/about' element={<About />} />
            </Route>

            <Route path='/contact' element={<PrivateRoute
isAuthenticated={isAuthenticated} />} >
              <Route path='/contact' element={<Contact />} />
            </Route>
          </Routes>
        </Box>
      </BrowserRouter>
    </DataProvider>
  );
}

export default App;
```

API.js-

```
import axios from 'axios';
import { API_NOTIFICATION_MESSAGES ,SERVICE_URLS} from '../constants/config';
import { getAccessToken ,getType} from '../utils/common-utils';

const API_URL = 'http://localhost:8000';


const axiosInstance = axios.create({
    baseURL: API_URL,
    timeout: 10000,
    headers: {
        "content-type": "application/json"
    }
})


axiosInstance.interceptors.request.use(
    function(config) {
        if (config.TYPE.params) {
            config.params = config.TYPE.params
        } else if (config.TYPE.query) {
            config.url = config.url + '/' + config.TYPE.query;
        }
        return config;
    },
    function(error) {
        return Promise.reject(error);
    }
);

axiosInstance.interceptors.response.use(
    function(response) {
        // Stop global loader here
        return processResponse(response);
    },
    function(error) {
        // Stop global loader here
        return Promise.reject(ProcessError(error));
    }
)

// ///////////////
const processResponse = (response) => {
    if (response?.status === 200) {
        return { isSuccess: true, data: response.data }
    } else {
        return {
            isFailure: true,
            status: response?.status,
            msg: response?.msg,
            code: response?.code
        }
    }
}


const ProcessError = async (error) => {
    if (error.response) {
        console.log('ERROR IN RESPONSE:', error.response);
        return {
            isError: true,
            msg: API_NOTIFICATION_MESSAGES.responseFailure,
            code: error.response.status
        };
    } else if (error.request) {
        // The request was made but no response was received
        console.log("ERROR IN REQUEST:", error.request);
        return {
            isError: true,
            msg: API_NOTIFICATION_MESSAGES.requestFailure,
            code: ""
        };
    } else {
        // Something happened in setting up the request that triggered an Error
        console.log("ERROR IN NETWORK:", error.message);
        return {
            isError: true,
            msg: API_NOTIFICATION_MESSAGES.networkError,
            code: ""
        };
    }
};

const API={};


for (const [key, value] of Object.entries(SERVICE_URLS)) {
    API[key] = (body, showUploadProgress, showDownloadProgress) =>
        axiosInstance  ({
            method: value.method,
            url: value.url,
            data: value.method==='DELETE' ? null: body,
            responseType: value.responseType,
           headers: {
               authorization: getAccessToken(),
           },
            TYPE: getType(value, body),
            onUploadProgress: function(progressEvent) {
                if (showUploadProgress) {
                    let percentCompleted = Math.round((progressEvent.loaded *
100) / progressEvent.total);
                    showUploadProgress(percentCompleted);
                }
            },
            onDownloadProgress: function(progressEvent) {
                if (showDownloadProgress) {
                    let percentCompleted = Math.round((progressEvent.loaded *
100) / progressEvent.total);
                    showDownloadProgress(percentCompleted);
                }
            }
        })
}
export { API };
```

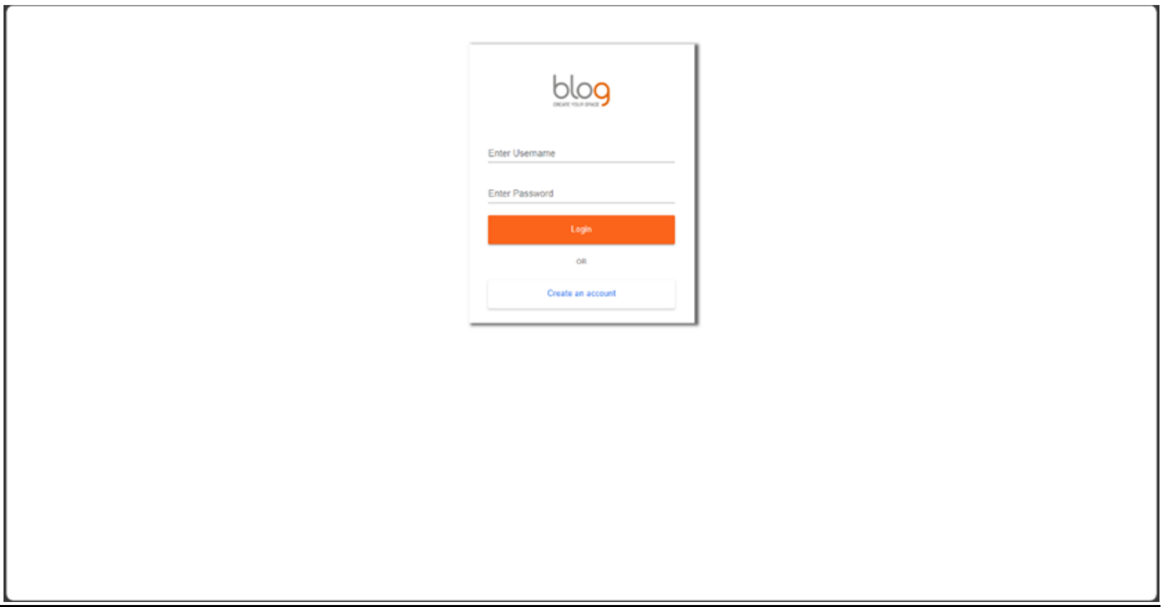
DB.JS:

```
import mongoose from 'mongoose';

const Connection = async (username, password) => {
    const URL = `mongodb://${username}:${password}@ac-5qeoso1-shard-00-
00.gr5vmu1.mongodb.net:27017,ac-5qeoso1-shard-00-01.gr5vmu1.mongodb.net:27017,ac-
5qeoso1-shard-00-02.gr5vmu1.mongodb.net:27017/?ssl=true&replicaSet=atlas-nvjxnk-
shard-0&authSource=admin&retryWrites=true&w=majority&appName=blog-app`;
    try {
        await mongoose.connect(URL,{useNewUrlParser:true})
        console.log('Database connected successfully');
    } catch (error) {
        console.log('Error while connecting to the database ', error);
    }
};

export default Connection;
```

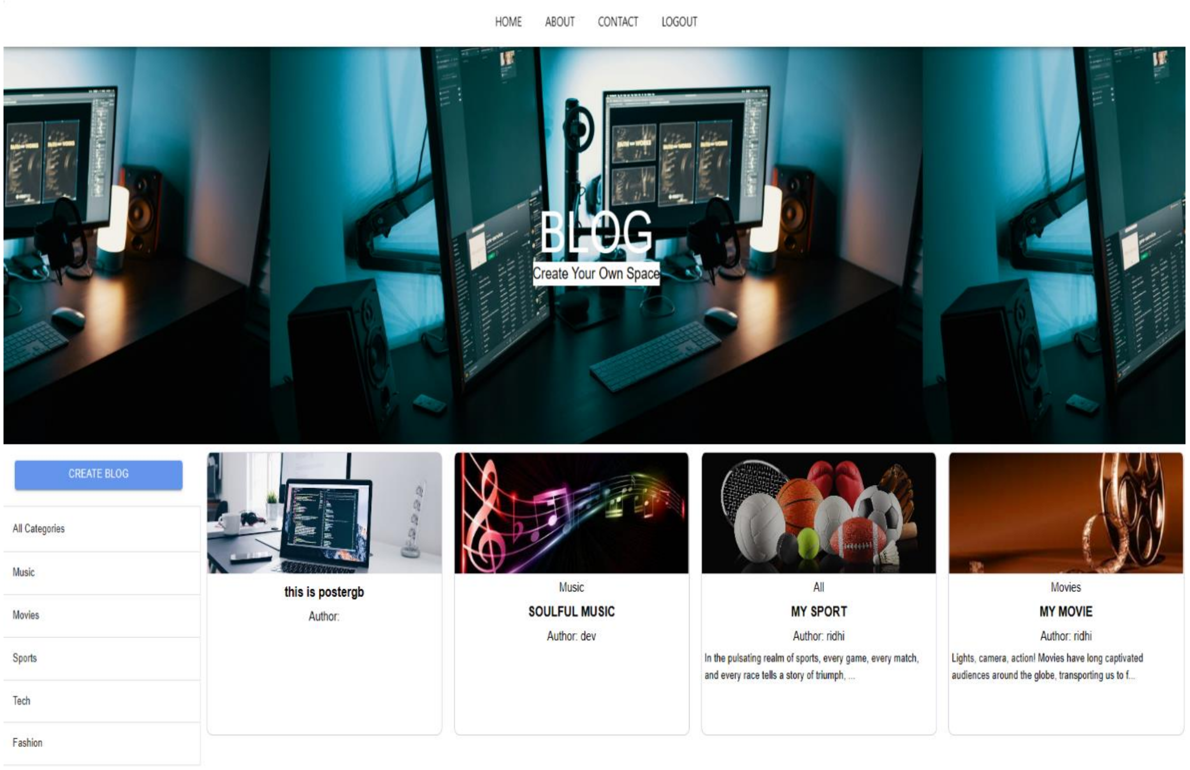
Name: Priyavandna Ranawat
Roll No: 205122069

PAGE RENDERING:

Login Page:



<u>Home Page:</u>



this is postergb
Author:

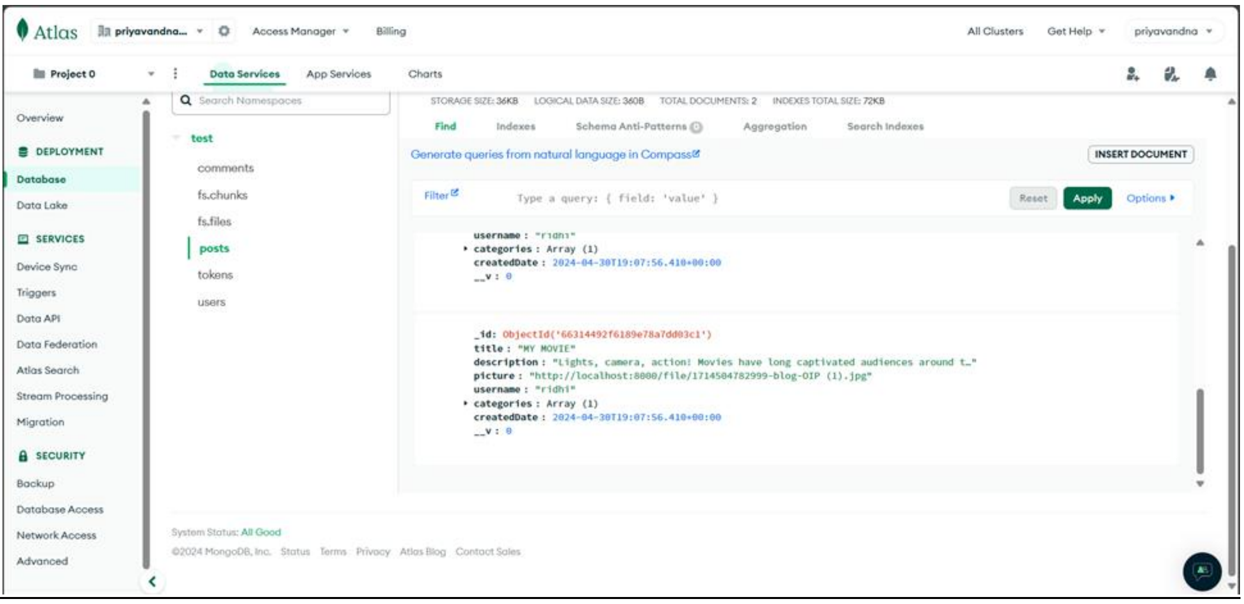
Music
SOULFUL MUSIC
Author: dev

All
MY SPORT
Author: ridhi
In the pulsating realm of sports, every game, every match, and every race tells a story of triumph, ...

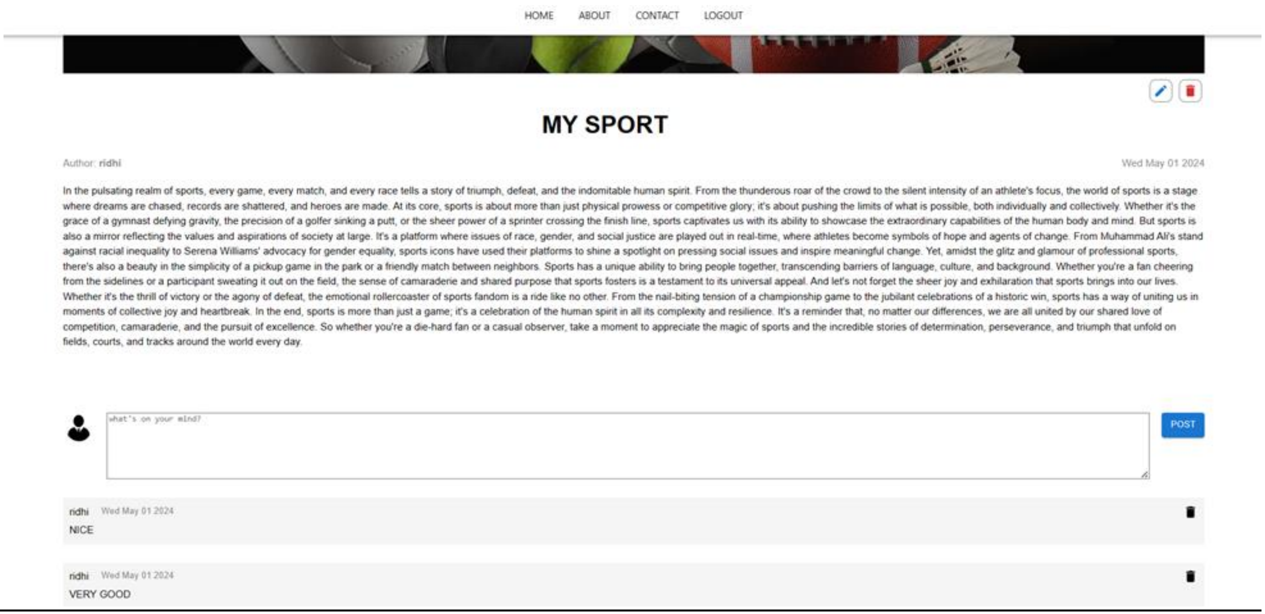
Movies
MY MOVIE
Author: ridhi
Lights, camera, action! Movies have long captivated audiences around the globe, transporting us to f...

Name: Priyavandna Ranawat
Roll No: 205122069

Database:



```
    username : "ridhi"
  ▸ categories : Array (1)
    createdDate : 2024-04-30T19:07:56.410+00:00
    __v : 0

    _id: ObjectId('66314492f6189e78a7dd03c1')
    title : "MY MOVIE"
    description : "Lights, camera, action! Movies have long captivated audiences around t…"
    picture : "http://localhost:8000/file/1714504782999-blog-OIP (1).jpg"
    username : "ridhi"
  ▸ categories : Array (1)
    createdDate : 2024-04-30T19:07:56.410+00:00
    __v : 0
```



MY SPORT

Author: ridhi Wed May 01 2024

In the pulsating realm of sports, every game, every match, and every race tells a story of triumph, defeat, and the indomitable human spirit. From the thunderous roar of the crowd to the silent intensity of an athlete's focus, the world of sports is a stage where dreams are chased, records are shattered, and heroes are made. At its core, sports is about more than just physical prowess or competitive glory; it's about pushing the limits of what is possible, both individually and collectively. Whether it's the grace of a gymnast defying gravity, the precision of a golfer sinking a putt, or the sheer power of a sprinter crossing the finish line, sports captivates us with its ability to showcase the extraordinary capabilities of the human body and mind. But sports is also a mirror reflecting the values and aspirations of society at large. It's a platform where issues of race, gender, and social justice are played out in real-time, where athletes become symbols of hope and agents of change. From Muhammad Ali's stand against racial inequality to Serena Williams' advocacy for gender equality, sports icons have used their platforms to shine a spotlight on pressing social issues and inspire meaningful change. Yet, amidst the glitz and glamour of professional sports, there's also a beauty in the simplicity of a pickup game in the park or a friendly match between neighbors. Sports has a unique ability to bring people together, transcending barriers of language, culture, and background. Whether you're a fan cheering from the sidelines or a participant sweating it out on the field, the sense of camaraderie and shared purpose that sports fosters is a testament to its universal appeal. And let's not forget the sheer joy and exhilaration that sports brings into our lives. Whether it's the thrill of victory or the agony of defeat, the emotional rollercoaster of sports fandom is a ride like no other. From the nail-biting tension of a championship game to the jubilant celebrations of a historic win, sports has a way of uniting us in moments of collective joy and heartbreak. In the end, sports is more than just a game; it's a celebration of the human spirit in all its complexity and resilience. It's a reminder that, no matter our differences, we are all united by our shared love of competition, camaraderie, and the pursuit of excellence. So whether you're a die-hard fan or a casual observer, take a moment to appreciate the magic of sports and the incredible stories of determination, perseverance, and triumph that unfold on fields, courts, and tracks around the world every day.

ridhi Wed May 01 2024
NICE

ridhi Wed May 01 2024
VERY GOOD

REFERENCES AND BIBLOGRAPHY:

Here's a bibliography for building a blog web application using the MERN (MongoDB, Express.js, React.js, Node.js) stack:

MongoDB

MongoDB Documentation: https://docs.mongodb.com/

MongoDB University: https://university.mongodb.com/

"MongoDB in Action" by Kyle Banker, Peter Bakkum, and Shaun Verch (Publisher: Manning Publications)

Express.js:

Express.js Documentation: https://expressjs.com/

"Express.js in Action" by Evan Hahn (Publisher: Manning Publications)

"Pro Express.js: Master Express.js - The Node.js Framework for Your Web Development" by Azat Mardan (Publisher: Apress)

React.js:

React.js Documentation: https://reactjs.org/

"Learning React: A Hands-On Guide to Building Web Applications Using React and Redux" by Kirupa Chinnathambi (Publisher: Addison-Wesley Professional)

"React.js Essentials" by Artemij Fedosejev (Publisher: Packt Publishing)

Node.js:

Node.js Documentation: https://nodejs.org/en/docs/

"Node.js in Action" by Mike Cantelon, Marc Harter, TJ Holowaychuk, and Nathan Rajlich (Publisher: Manning Publications)

"Node.js Design Patterns" by Mario Casciaro (Publisher: Packt Publishing)

MERN Stack Tutorials:

"MERN Quick Start Guide: Build web applications with MongoDB, Express.js, React, and Node" by Eddy Wilson Iriarte Koroliova (Publisher: Packt Publishing)

"Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node" by Vasan Subramanian (Publisher: Apress)

Authentication and Authorization:

"JWT.io - JSON Web Tokens Introduction": https://jwt.io/introduction/

"OAuth 2.0 Simplified" by Aaron Parecki (Available online: https://oauth.net/2/)

"OAuth 2.0: The Definitive Guide" by Aaron Parecki (Publisher: O'Reilly Media)

These resources provide a comprehensive guide for building a blog web application using the MERN stack, covering database setup, backend and frontend development, user authentication, deployment, and more. You can explore each resource based on your specific requirements and level of expertise.