Implementation of a Developer Platform with MERN Stack: A Session Management and Authentication

A Report

Submitted in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Areeb Ahmad - BTECH/10003/20

Nishank Deep – BTECH/10200/20

Prateek Kumar Mishra – BTECH/10208/20

Birla Institute of Technology, Mesra

Ranchi, Jharkhand-835215

APPROVAL OF THE GUIDE

We recommended that the B. Tech. Minor Project titled **Implementation of a Developer Platform with MERN Stack: A Session Management and Authentication** submitted by **Areeb Ahmad – BTECH/10003/20, Nishank Deep – BTECH/10200/20 and Prateek Mishra – BTECH/10208/20** is approved by me for submission. This should be accepted as fulfilling the partial requirements for the award of Degree of Bachelor of Technology in **Computer Science and Engineering.** To the best of my knowledge, the report represents work carried out by the student in **Birla Institute of Technology, Mesra** and the content of this report is not forming a basis for the award of any previous degree to anyone else.

Date: 25/11/23

Mrs. Debjani Mustafi
Assistant Professor, Project Guide
Department of Computer Science and Engineering
Birla Institute of Technology, Mesra

DECLARATION CERTIFICATE

We certify that

a) The work contained in the report is original and has been done by myself under the general supervision of my supervisor.

b) The work has not been submitted to any other Institute for any other degree or diploma.

c) I have followed the guidelines provided by the Institute in writing the report.

d) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e) Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references.

f) Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Date: 25/11/23

Areeb Ahmad BTECH/10003/20
Nishank Deep BTECH/10200/20
Prateek Kumar Mishra BTECH/10208/20

Department of Computer Science and Engineering
Birla Institute of Technology, Mesra

CERTIFICATE OF APPROVAL

This is to certify that the work embodied in this Minor Project Report entitled "**Implementation of a Developer Platform with MERN Stack : A Session Management and Authentication**"

is carried out by **Areeb Ahmad BTECH/10003/20, Nishank Deep BTECH/10200/20 and Prateek Mishra BTECH/10208/20** has been approved for the degree of Bachelor of Technology in **Computer Science and Engineering** of Birla Institute of Technology, Mesra, Ranchi.

Date: 25/11/23

Place: Ranchi, India

(Chairman)
Head of the Department
Dept. of Comp. Sc. & Engg.

(Panel Coordinator)
Examiner
Dept. of Comp. Sc. & Engg.

ACKNOWLEDGEMENT

We would like to extend my sincere gratitude and appreciation to my guide who helped make my minor project a success, including the work on the testing of new features and the documentation of the repository for the application. We owe a lot for their assistance and advice throughout this minor project, which has been a worthwhile learning experience. We are appreciative of the information and abilities that my academic experience has given me, as they have been essential to carrying out this assignment successfully. Their priceless advice, insightful criticism, and unwavering support have greatly shaped my understanding of software development and helped me successfully navigate the project's many obstacles. We sincerely appreciate the cooperation, assistance, and knowledge-sharing of the entire Computer Science and Engineering department. Their combined efforts and knowledge have been essential to the integration and implementation of features. Their confidence in me and never-ending drive have been the driving force behind my accomplishments. Finally, We want to express my sincere gratitude to everyone who was listed above for their help, direction, and cooperation, all of which were essential to the project's success. This voyage has been greatly enriched by their efforts. We appreciate everyone's unwavering belief in my ability and support.

Date: 25/11/23 (Signature)

Name of the Student

Roll No. of the Student

CONTENTS

LIST OF FIGURES

CHAPTER 1

Project Overview – (DevConnect) Implementation of a Developer Platform

1.1 Introduction

In today's fast-paced technological landscape, staying connected and informed is crucial for developers. DevConnect aims to address this need by providing a dedicated social media platform that caters specifically to developers, fostering a community-driven space where knowledge exchange and collaboration thrive.

1.2 Key Features

1.2.1 User Registration and Profile Creation

The onboarding process for DevConnect is user-friendly and straightforward. Developers can easily create their accounts by providing basic information such as username, email, and password. Upon registration, users are prompted to complete their profiles by adding personal details like their name, profile picture, and a brief bio. A unique aspect of DevConnect is the inclusion of a "Tech Stack" section where users specify the technologies they are currently studying or working on. This information is crucial for tailoring the content that appears on their feed.

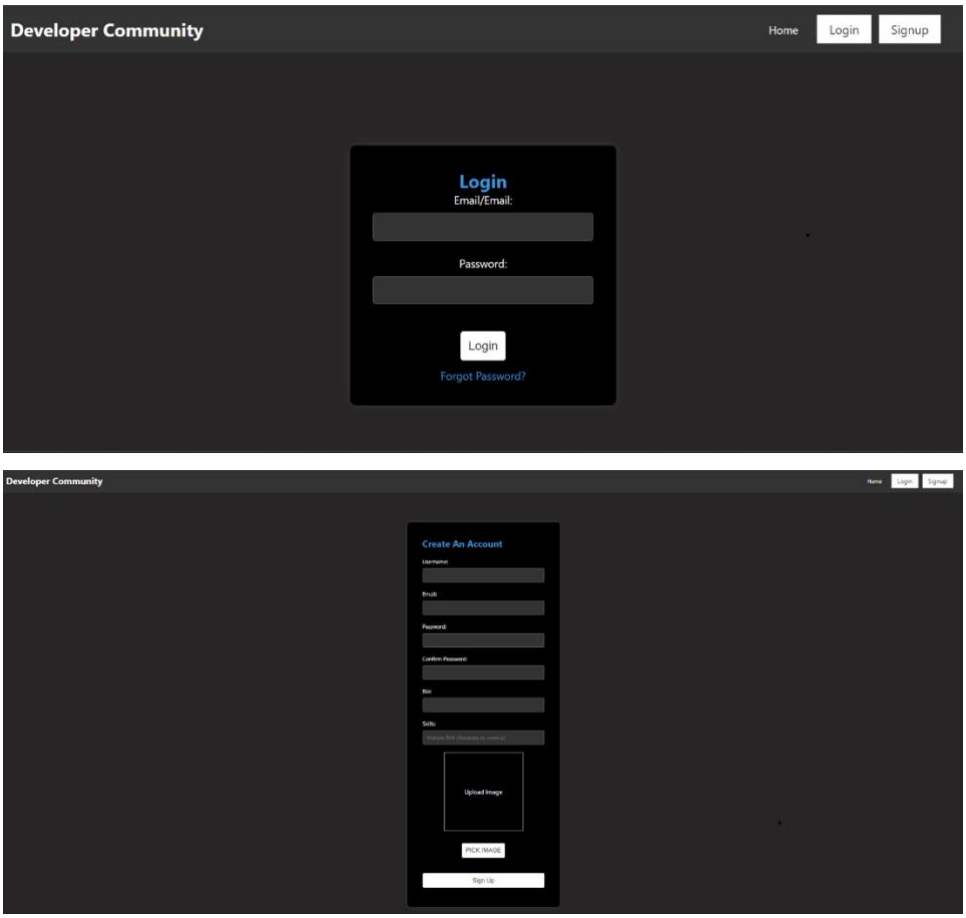

Figure 1.2.1

1.2.2 Personalized Feed

Once registered and with a completed profile, developers are presented with a personalized feed. This feed is dynamically populated with blogs, articles, and content curated based on the user's specified tech stack. This ensures that the content aligns with the user's professional interests, allowing them to stay informed about the latest developments in their chosen technologies.

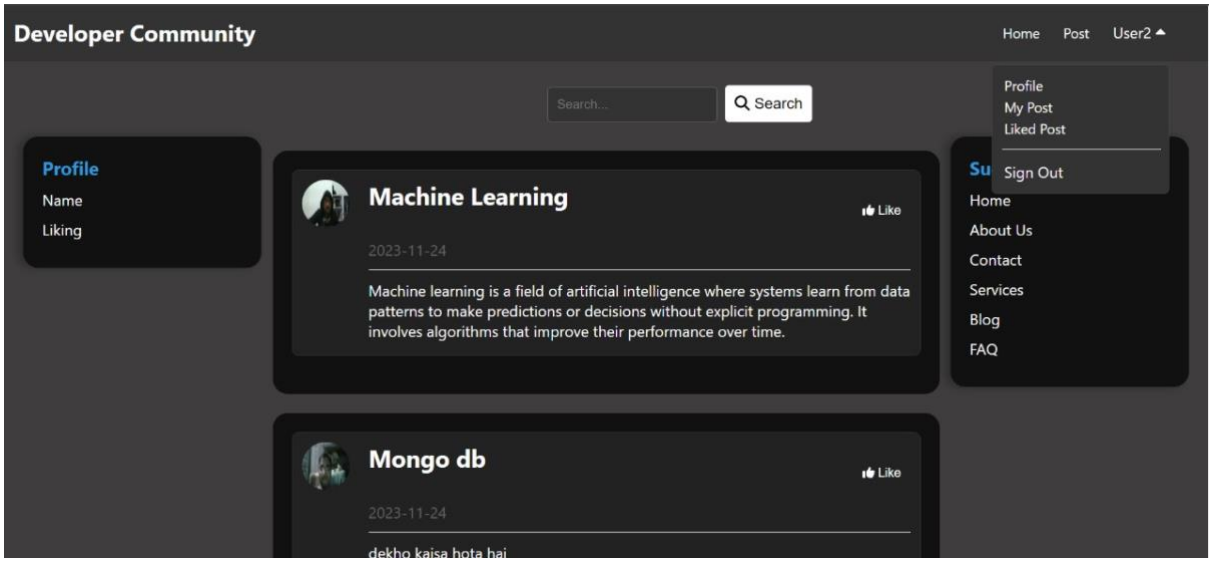

Figure 1.2.2

1.2.3 Blog Management

DevConnect empowers users to share their knowledge and experiences by allowing them to post blogs and content directly on their profiles. This feature encourages users to contribute to the community by sharing insights, tips, and solutions related to their tech stack. Moreover, users have the flexibility to manage their content by deleting posts, ensuring that they have control over their online presence.

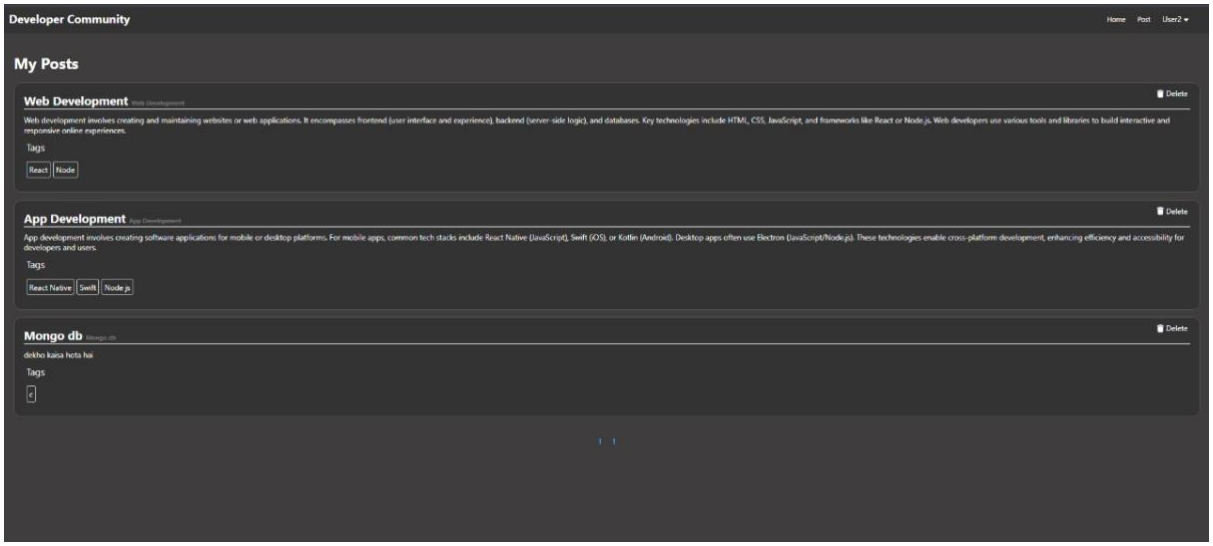

Figure 1.2.3

1.2.4 Favourites and Friends

DevConnect promotes community building through two essential features: favourites and friends. Users can mark blogs as favourites, creating a personalized collection of content that they find particularly valuable. Additionally, the platform facilitates networking by allowing users to connect with each other as friends. This connection feature promotes collaboration and knowledge exchange, creating a sense of camaraderie within the developer community.

1.3 User Interaction

1.3.1 Post Interaction

DevConnect emphasizes active engagement within the community. Users can interact with each other's posts by liking, commenting, and sharing. This functionality encourages discussions, feedback, and the sharing of diverse perspectives within the platform.

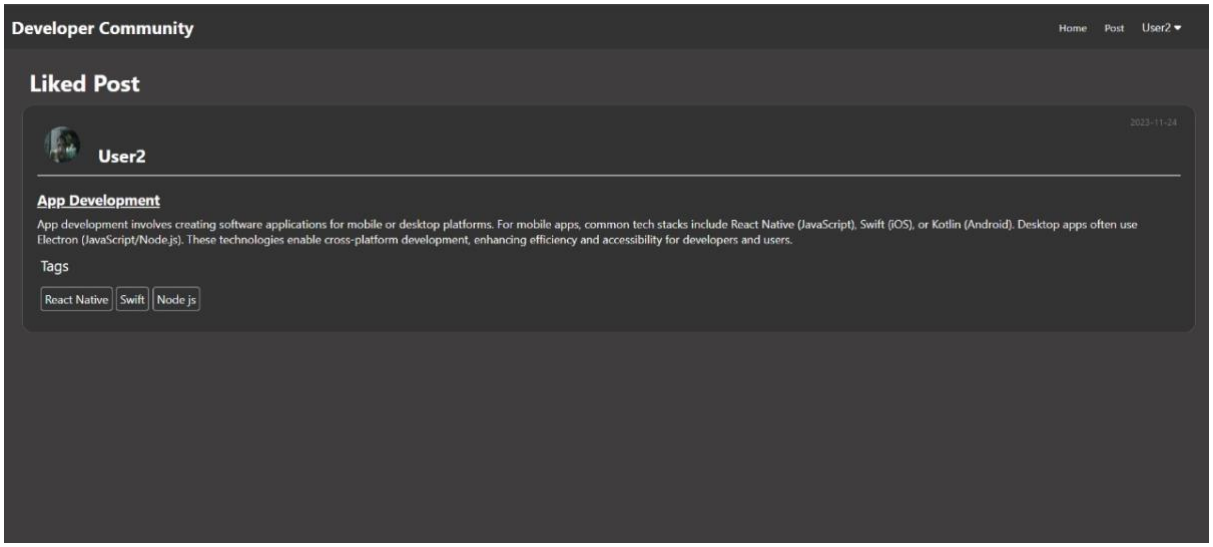

Figure 1.3.1

1.4 Privacy and Security

DevConnect prioritizes user privacy and security. Users have granular control over the visibility of their profiles and posts, allowing them to tailor their level of exposure within the community. Robust security measures are implemented to safeguard user data, ensuring a trustworthy and secure environment for developers to interact and share.

1.5 Conclusion

In conclusion, DevConnect provides a holistic and user-centric approach to social media for developers. By combining personalized content, social interaction features, and robust privacy measures, the platform is designed to empower developers in their professional journeys and foster a vibrant community of knowledge sharing. The subsequent chapters will provide a deeper dive into the technical aspects, development process, and user interface design that contribute to the success of DevConnect.

CHAPTER 2

Frontend Implementation - Crafting an Interactive User Interface with React.js

2.1 Introduction

We will examine the technical aspects of the frontend development for DevConnect in this chapter, with particular attention paid to the technologies used, the architectural layout, and the React.js components that serve as the framework for the user interface.

2.2 Technology Stack

2.2.1 React.js

React.js was selected as the main frontend library because of its strong component-based design. Reusability and maintainability are encouraged by this architectural pattern, which is important given that social media platforms are dynamic. React.js improves the efficiency of the application by effectively handling changes and rendering through the use of the virtual DOM.

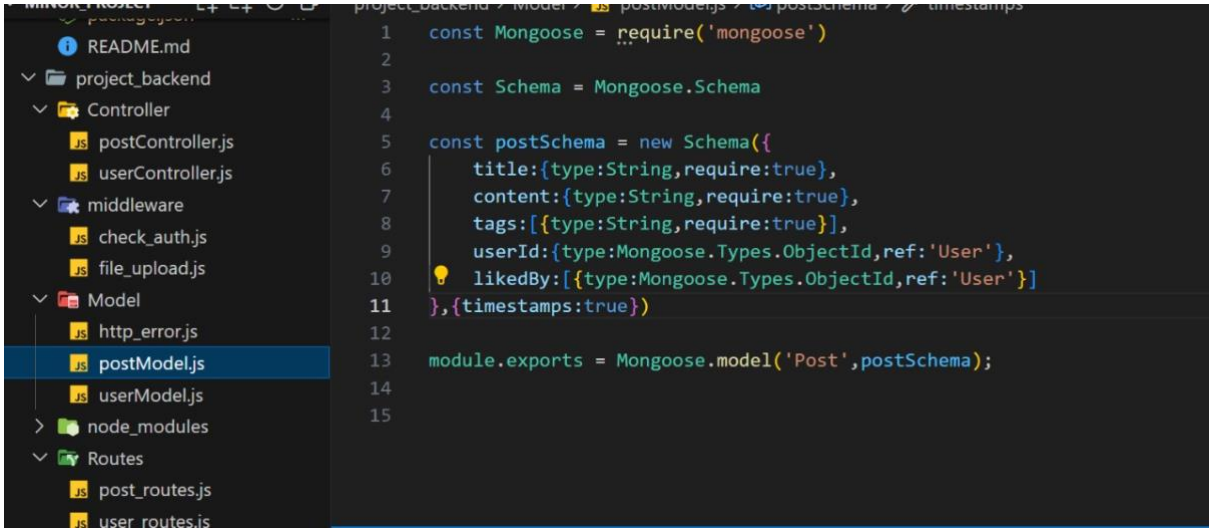

Figure 2.2.1

2.3 Architecture

2.3.1 Component-Based Architecture

The user interface's many functional parts are divided up into modular components by the frontend architecture, which uses a component-based design. This method makes the development and maintenance processes simpler, encourages reusability, and helps organize the code. Important parts are those that manage profiles, render the feed, handle user interactions, and authenticate users.

2.3.2 State Management with React Hooks

React Hooks, such as ‘useState’ and ‘useContext’, are used to manage the dynamic nature of DevConnect's frontend. These hooks allow for rapid user interface changes and efficient state management, guaranteeing responsiveness to user inputs. Blog postings, friend-related information, and user authentication are all handled by the state management system.

2.4 React Components

2.4.1 User Authentication

The login and registration procedures are controlled by the user authentication component. It communicates with the backend in a seamless manner, securely authenticating users and handling authentication tokens to guarantee a safe and easy user experience.

2.4.2 Profile Component

The profile component, which is in charge of displaying user profiles dynamically, displays the user's chosen tech stack as well as personal data. It also gives users the ability to edit their profile information, which adds to the user experience's personalization and customization.

2.4.3 Feed Component

The feed component, which is the core of DevConnect, dynamically produces articles and blog entries according to the user's chosen tech stack. It guarantees real-time updates by utilizing React Hooks for state management, giving every user a customized and pertinent experience.

2.4.4 Interaction Components

These parts manage how users engage with the platform, including sharing, like, and commenting on posts. These components provide a smooth and captivating user experience by enabling real-time changes through the usage of React Hooks for state management.

2.5 Styling and Responsiveness

An uncluttered and simple user interface is given top priority in the frontend design. For styling, Cascading Style Sheets (CSS) are utilized, with an emphasis on making sure the content is responsive on a range of screens and devices. All DevConnect users will enjoy a consistent and intuitive experience thanks to this dedication to responsive design.

2.6 Conclusion

DevConnect's frontend is carefully designed with React.js and uses a component-based architecture that encourages scalability and modularity. React Hooks' utilization guarantees effective state management, which enhances the user interface's responsiveness and dynamic nature. In order to create a unified and useful developer social media platform, we will examine the database architecture, frontend and backend integration, and backend implementation in the upcoming chapters.

CHAPTER 3

Backend Implementation - DevConnect with MongoDB, Express.js, and Node.js

3.1 Introduction

We will go further deeply into the technical aspects of DevConnect's backend development in this chapter, with particular attention paid to the technologies used, the architectural layout, and an in-depth analysis of the codebase, which is powered by Express.js, Node.js, and MongoDB.

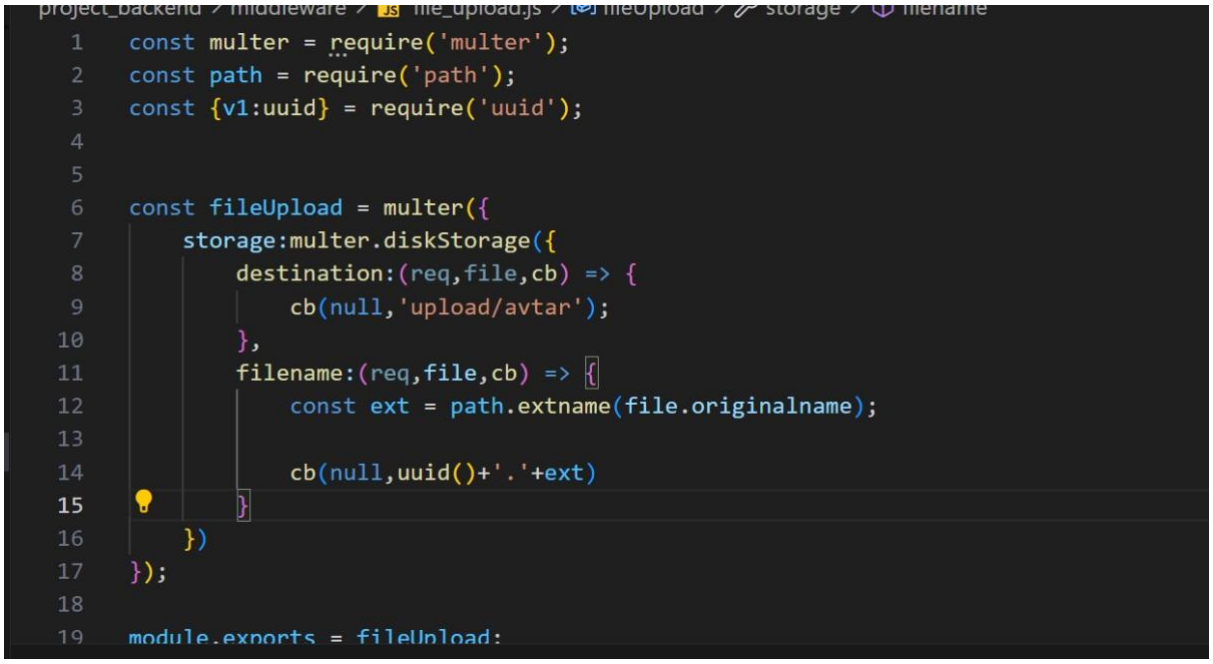

Figure 3.1.1

3.2 Technology Stack

3.2.1 Node.js

The foundation of DevConnect's backend is Node.js, which offers an event-driven, non-blocking runtime environment that perfectly fits with the asynchronous processes common to web development. Its vast npm ecosystem facilitates effective package management and integration while speeding up the development process.

3.2.2 Express.js

The backend framework, Express.js, makes simpler the task to create scalable and secure web applications. Its middleware architecture and simple design make the backend more adaptable and facilitate for the seamless integration of several distinct functionalities.

3.2.3 MongoDB

MongoDB was selected as the NoSQL database due to its scalability and flexibility. Because of its document-oriented structure, which provides an effective and scalable method for managing a variety of data structures, it is well suited to the dynamic nature of user data in social media platforms.

3.3 Backend Architecture

3.3.1 RESTful API Design

The RESTful API design of DevConnect's backend stimulates stateless communication between the frontend and backend. By enhancing scalability and interoperability, this design principle makes it feasible for seamless integration of new features and functionalities.

3.3.2 Middleware Integration

The middleware functions of Express.js are meticulously combined for overseeing different tasks, including request parsing, error handling, and user authentication. By supporting efficient data transfer between the frontend and backend, middleware improves the application's overall security and efficiency.

3.4 MongoDB Database

3.4.1 Schema-less Design

For DevConnect's backend, MongoDB's schema-less architecture is helpful because it can handle the dynamic and varied data structures linked to user profiles, blog posts, and social interactions. This capacity for adaptation makes it possible for the platform to scale seamlessly as it being developed

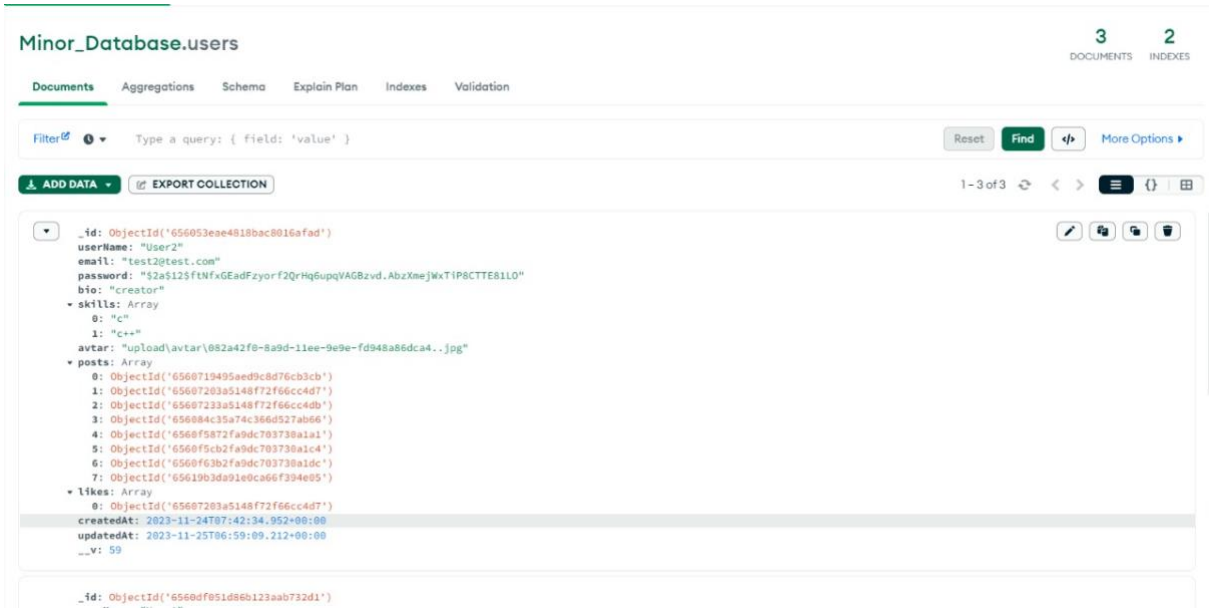

Figure 3.4.1.1

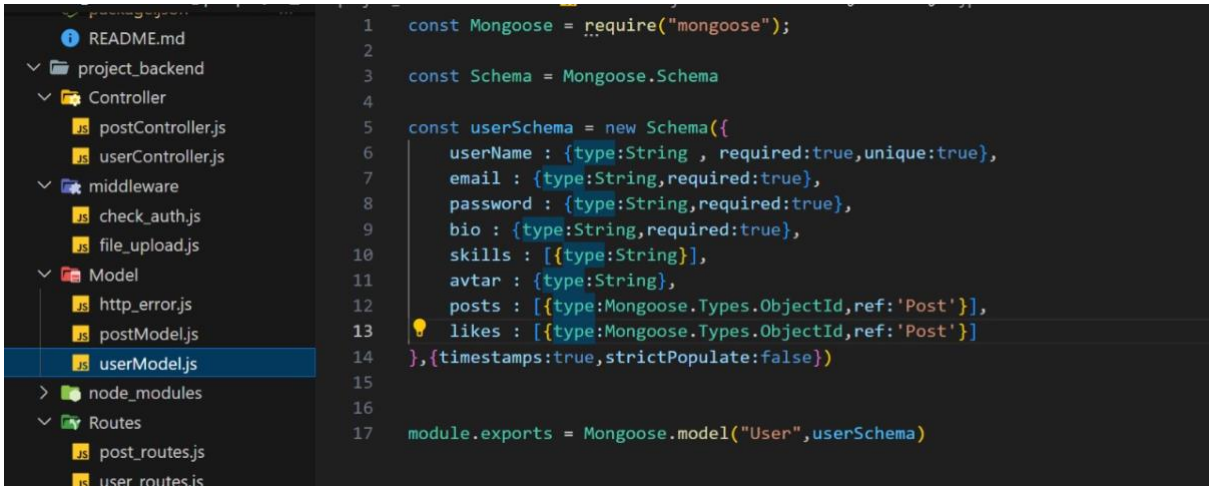

Figure 3.4.1.2

3.4.2 Collections and Indexing

The data is arranged logically into collections, each of which represents a distinct application entity (such as users or blog entries). Utilizing MongoDB's indexing powers, query performance and data retrieval are enhanced, guaranteeing a responsive user experience even with growing databases.

3.5 Conclusion

With the help of Node.js, Express.js, and MongoDB, the backend of DevConnect creates a strong framework for the social media network. A scalable and effective backend architecture is facilitated by the flexibility of MongoDB, middleware integration, and the RESTful API design. Key features including user authentication, profile management, and blog interactions are demonstrated by the provided code snippets.

CHAPTER 4

Advanced Backend Techniques - Session Management, and Authentication:

(A Research Exploration)

4.1 Introduction

This research-focused chapter will explore the sophisticated backend methods used in DevConnect, with particular attention to session management, user authentication nuances, and Mongoose for MongoDB interaction. These components play a crucial role in molding the resilience and safety of the social media system.

4.2 Session Management

Session management is essential to DevConnect for safe and long-lasting user authentication. User data is safely saved and accessed during a user's engagement with the platform thanks to the implementation's session-based methodology. Sessions are essential to maintaining a smooth user experience because they offer a way to keep user state consistent across several requests. This section explores the complexities of session-based authentication, including the storage, access, and security of session data. It also looks at the implications of using token-based authentication instead of session-based authentication, like JSON Web Tokens (JWT). This study takes into account aspects like statelessness, scalability, and security and provides an explanation for the selected session management approach within the DevConnect framework.

Figure 4.2.1

4.3 Authentication Best Practices

Several best practices have been carefully incorporated into DevConnect's authentication process to strengthen user security. The application of secure password hashing techniques is one important aspect. This chapter examines the subtleties of these methods, taking into account the trade-offs between various hashing algorithms and how they affect the security of user credentials. The investigation also includes Two-Factor Authentication (2FA), a sophisticated security feature that improves the security of user accounts. The effectiveness of 2FA, user experience factors, and potential difficulties in putting such a system in place within the framework of a social media platform are all covered in the conversation. This section also explores the use of brute force protection and rate limiting mechanisms, which add to a multi-layered strategy to counter possible security threats. Through an examination of these authentication best practices, DevConnect strives to maintain the delicate balance between robust security measures and an intuitive user experience.

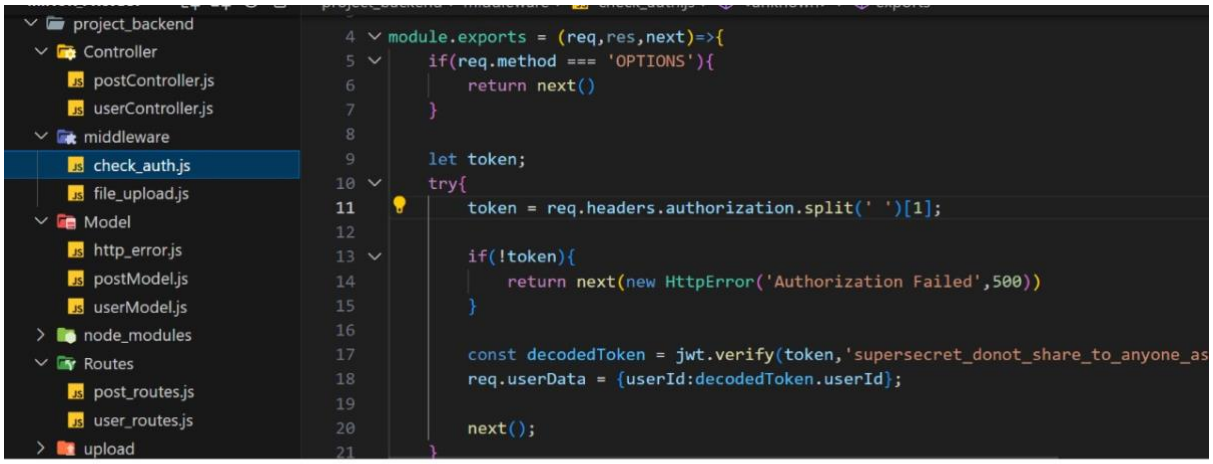

Figure 4.3.1

4.4 Research Outcomes

DevConnect's investigation of sophisticated backend techniques has produced insightful research findings that enhance the platform's usability, security, and performance. The effectiveness of deployed Mongoose schemas and session management techniques has been evaluated through thorough performance reviews. Examining database query response times, session creation and destruction overhead, and overall system responsiveness are all included in this. The research has examined vulnerability assessments in terms of security analysis, assessing how resilient the authentication mechanisms are against popular attack vectors and potential exploits. The results provide insight into how strong the security measures that were put in place were. Furthermore, consideration has been given to how selected authentication and session management strategies may affect user experience.

5.5 Conclusion

This research-focused chapter offers a thorough analysis of the sophisticated backend strategies used in DevConnect. Every element—from managing sessions and user authentication—is carefully considered in light of how it affects user experience, security, and performance. The research findings will guide DevConnect's next iterations, guaranteeing a state-of-the-art and safe developer social media network.

REFERENCES

1. https://nodejs.org/en/learn
2. https://www.javatpoint.com/nodejs-tutorial
3. https://www.geeksforgeeks.org/difference-between-sql-and-nosql/
4. https://www.packetlabs.net/posts/session-management/
5. https://medium.com/@vivekmadurai/different-ways-to-authenticate-a-web-application-e8f3875c254a
6. https://www.bu.edu/tech/about/security-resources/bestpractice/auth/

Implementation of Developer Platform using MERN Stack (Areeb, Nishank, Prateek)

ORIGINALITY REPORT

10%
SIMILARITY INDEX

PRIMARY SOURCES

1	www.coursehero.com Internet	160 words — 5%
2	www.imshome.com Internet	27 words — 1%
3	digital.lib.usu.edu Internet	26 words — 1%
4	www.kluniversity.in Internet	24 words — 1%
5	docplayer.net Internet	22 words — 1%
6	www.championsnet.net Internet	17 words — 1%
7	www.facweb.iitkgp.ernet.in Internet	16 words — 1%
8	ir.juit.ac.in:8080 Internet	13 words — < 1%
9	www.theseus.fi Internet	12 words — < 1%