San Francisco Bay University

Fremont, California

Report on

MindBridge

Mental health blog sharing platform

CS 526

Advanced Web Programming

Submitted by:

Prelisa Dahal

Snehal Prakash Mule

Submitted to:

Professor. Pragati Dharmale

Table of contents

1. Objective and Functionality

1.1 Project Overview

MindBridge is a web-based blog management system focused on mental health and well-being. The platform creates an inclusive space where individuals can share their perspectives, personal experiences, and articles on mental health topics. MindBridge aims to foster open dialogue, encourage storytelling, and provide a supportive environment for users to read, write, and manage mental health-related content.

1.2 Core Functionality

User Authentication

- Account creation with personal details and profile description
- Secure login with JWT-based authentication
- Password management and session handling

Content Management

- Creating and publishing posts with title, subtitle, body, and thumbnail
- Keyword tagging for improved content discoverability
- Post editing capabilities for authors

Content Discovery

- Search functionality to discover relevant content
- View posts from specific authors

1.3 Target Audience

The platform is specifically designed for individuals interested in mental health and well-being, including:

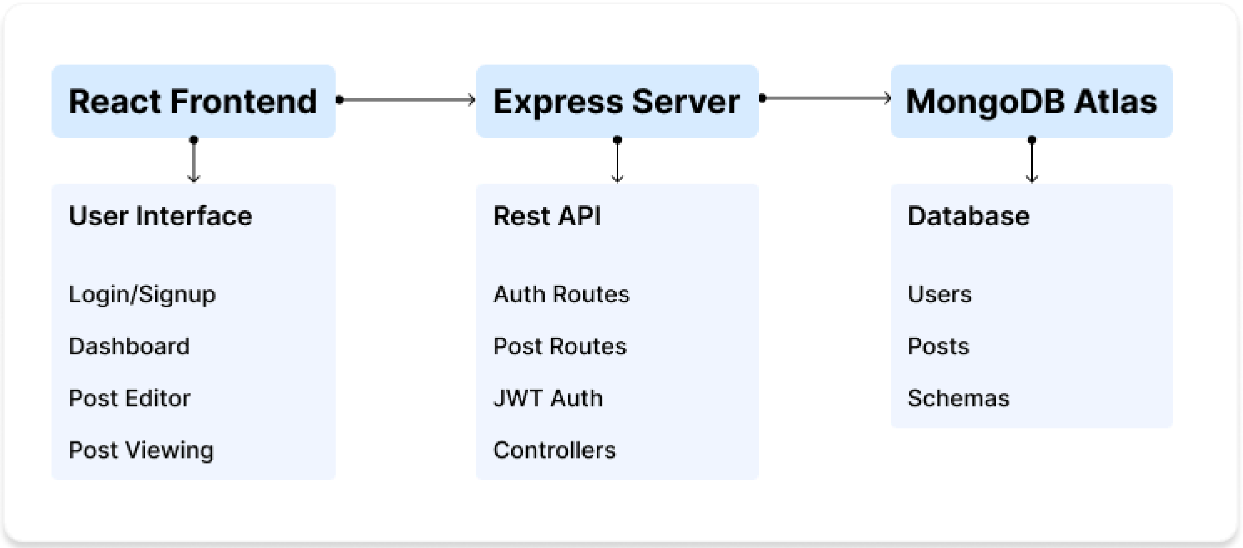
- People seeking to share personal mental health journeys and experiences

- Mental health advocates and writers
- Readers looking for relatable mental health content and perspectives
- Those seeking a supportive community focused on well-being topics

The clean interface and focus on written content suggests an emphasis on thoughtful, longer-form articles that facilitate meaningful dialogue around mental health rather than quick social media interactions.

2. Architecture

2.1 System Architecture Diagram



2.2 Component Architecture

1. **Frontend (React)**
    - Pages (Login, Signup, Dashboard, Post Editor)
    - Components (Header, Footer, PostCard)
    - API integration services
    - State management

2. **Backend (Node.js/Express)**
    - Authentication controllers
    - Post management controllers
    - MongoDB connection
    - JWT middleware for route protection
3. **Database (MongoDB)**
    - User collection
    - Posts collection

2.3 Data Flow

1. **User Registration/Login Flow**:
    - User submits credentials via frontend forms
    - Data is sent to authentication endpoints
    - Server validates data and queries/updates MongoDB
    - JWT tokens are generated and returned to client
    - Frontend stores tokens in localStorage for session management
2. **Content Creation Flow**:
    - Authenticated user creates post content
    - JWT token validates user identity
    - Post data is stored in MongoDB with author information
    - Success response returns to client
3. **Content Discovery Flow**:
    - User searches or browses content
    - Queries are sent to appropriate API endpoints
    - Server retrieves filtered data from MongoDB
    - Results are displayed on frontend

3. Tools and Frameworks

3.1 Frontend Technologies

- **React.js**: Component-based UI library for building the interface
- **React Router**: For client-side routing between different views
- **Axios**: HTTP client for API communication
- **CSS Modules**: For component-scoped styling
- **React Bootstrap**: For UI components
- **React Icons:** For Icon components

3.2 Backend Technologies

- **Node.js**: JavaScript runtime for server-side execution
- **Express.js**: Web framework for building API endpoints
- **JSON Web Token (JWT)**: For secure authentication
- **MongoDB**: NoSQL database for storing user and post data
- **Mongoose**: For database connectivity

3.3 Development Tools

- **npm**: Package management
- **Create React App**: Frontend bootstrapping
- **Git**: Version control (inferred from project structure)

3.4 Deployment & Infrastructure

- The server is configured to run on port 8080
- MongoDB Atlas is used as the database service (indicated by connection URI)
- CORS is enabled for cross-origin requests

4. Team Roles and Responsibilities

The project was developed by a team of two people with overlapping responsibilities:

4.1 Prelisa

- **UX/UI Design**: Created the visual design and user experience for the platform
- **Backend Development**: Implemented the Express.js server, API endpoints, and MongoDB integration
- **Frontend Development (Partial)**: Developed specific frontend components and pages
- **Integration**: Collaborated on connecting frontend components with backend services

4.2 Snehal

- **Frontend Development**: Built React components, pages, and client-side functionality
- **Backend Development:** Updating the backend for the CRUD operations
- **Integration**: Collaborated on connecting frontend components with backend services
- **Testing and quality assurance**

Both team members contributed to the overall architecture and feature implementation, with a collaborative approach to frontend-backend integration. This small team structure required versatility and cross-functional skills from both members, with you taking primary responsibility for design and backend while sharing frontend development duties.

5. Challenges, Learnings, and Future Improvements

Challenges Faced

1. **Authentication Management**: Implementing secure JWT-based authentication with proper session handling and token expiration.
2. **Rich Text Editor Integration**: Integrating and configuring the Quill.js editor to work seamlessly with content saving and retrieval.

3. **Responsive Design**: Ensuring the application works well across different devices while maintaining a consistent user experience.
4. **MongoDB Data Modeling**: Designing efficient schemas for the application's data requirements.

Key Learnings

1. **JWT Authentication Flow**: Gained deeper understanding of JWT-based authentication workflows.
2. **React Component Architecture**: Improved skills in organizing React components for maximum reusability.
3. **MongoDB/Mongoose**: Enhanced knowledge of MongoDB's document model and Mongoose's ODM capabilities.
4. **Rich Text Content Management**: Learned best practices for storing and rendering rich text content.

Future Improvements

1. **Enhanced User Profiles**: Add more customization options for user profiles, including profile pictures and social links.
2. **Search Functionality**: Implement a robust search system for finding posts by title, content, or keywords.
3. **Categories and Tags**: Add the ability to categorize posts and use tags for better content organization.
4. **Interactions and Comments**: Implement a comment system and social interactions like likes, shares, upvotes, downvotes.
5. **Image Upload Functionality**: Add direct image upload capabilities instead of using image URLs.
6. **Analytics Dashboard**: Provide authors with insights about their post performance.
7. **SEO Optimization**: Improve metadata and URL structures for better search engine visibility.

8. **Security Enhancements**: Implement additional security measures like rate limiting and enhanced password policies.

6. Conclusion

MindBridge demonstrates a solid foundation for a mental health blog platform with user authentication and post management capabilities. The project effectively implements a full-stack JavaScript application with MongoDB integration. While there are areas for improvement, particularly around security and scalability, the current implementation provides a functional platform that could be expanded into a supportive mental health content community.

The clear separation of concerns between frontend and backend, along with the modular component architecture, positions the project well for future development and feature expansion. With the recommended improvements, MindBridge could evolve into a comprehensive platform for connecting people through shared mental health experiences and well-being resources.

The collaboration between team members with complementary skills has resulted in a cohesive application that addresses an important social need. By creating a dedicated space for mental health content, MindBridge has the potential to make a meaningful impact in fostering dialogue and support around mental health topics.