IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154296

NeoAI: Socializing Reinvented with AI

Rampal Gupta[1], Ms. Deepika Pandey[2]

UG student of the Department of Computer Science and Engineering,

Goel Institute of Technology & Management, Lucknow, Uttar Pradesh, India[1]

Assistant Professor, Dept of Computer Science and Engineering Goel Institute of Technology & Management[2]

Abstract: This research presents NeoAI, an AI-powered social media application designed to improve user experience through personalized content delivery. Unlike traditional platforms that use basic or chronological feeds, NeoAI leverages machine learning to analyze user interactions such as likes, comments, and watch time. It uses collaborative filtering on a user-item interaction matrix to recommend relevant content. The system is built with a full-stack architecture including React.js, Node.js, Express.js, MongoDB, and a FastAPI-based ML service. In addition to recommendations, the platform supports features like posts, reels, stories, messaging, and notifications. By integrating AI, NeoAI enhances engagement, content discovery, and overall usability, making social media more intelligent and user-centric.

Keywords: Social Media, Recommendation System, Collaborative Filtering, Machine Learning, React.js, Node.js, MongoDB, FastAPI.

1. INTRODUCTION

Social media platforms have revolutionized the way individuals communicate, share information, and interact globally. Applications such as Instagram, Facebook, and Twitter have become integral to daily life, enabling users to connect, share multimedia content, and engage in digital communities. However, these platforms often rely on basic recommendation techniques or chronological feeds, which can lead to irrelevant content, repetitive posts, and decreased user satisfaction.

With the rapid growth of user-generated content, managing and delivering relevant information has become a significant challenge. Users are often overwhelmed by excessive content, making it difficult to discover posts that align with their interests. This creates a need for intelligent systems capable of understanding user behavior and providing personalized recommendations.

Recent advancements in machine learning and artificial intelligence have enabled the development of sophisticated recommendation systems. Techniques such as collaborative filtering, content-based filtering, and hybrid models have been widely used to improve personalization in digital platforms. These approaches analyze user interactions and preferences to deliver tailored content.

NeoAI leverages these advancements to create an intelligent social media platform that enhances user experience through AI-driven recommendations. The system tracks user interactions such as likes, comments, views, and watch time to build a behavioral profile. Using collaborative filtering, it identifies users with similar interests and recommends content accordingly.

In addition to recommendation capabilities, NeoAI integrates modern web technologies to provide features such as real-time messaging, notifications, media sharing, and user profile management. The system is designed to be scalable, efficient, and adaptable to future technological advancements.

This research focuses on developing a full-stack social media application with integrated machine learning capabilities, demonstrating how AI can significantly improve user engagement and content relevance in modern digital platforms.

2. SCOPE

The scope of NeoAI focuses on developing an intelligent social media platform that enhances user interaction and engagement through personalized content delivery. The system integrates machine learning techniques with modern web technologies to provide a seamless and efficient user experience.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154296

The application enables users to create and share posts, reels, and stories, interact with other users through likes, comments, and follows, and communicate using real-time messaging features. Additionally, the platform includes notification systems and saved collections to improve usability and content management.

The core scope lies in the implementation of a recommendation system that analyzes user behavior and interaction data. By constructing a user-item interaction matrix and applying collaborative filtering techniques, the system identifies patterns in user preferences and delivers relevant content recommendations.

The system also supports scalability and modularity, allowing future integration of advanced AI models such as deep learning and natural language processing. This ensures that NeoAI can evolve with technological advancements and user requirements.

3. OBJECTIVE

The main objective of NeoAI is to develop an intelligent and scalable social media platform that provides personalized content recommendations using machine learning techniques. The system aims to enhance user engagement, improve content discovery, and deliver a highly adaptive and user-centric digital experience. The key objectives of the project include:

1. **User Interaction Management:** Enable users to engage through likes, comments, shares, follows, and views. These interactions are tracked to build behavioral data, helping understand user preferences and supporting the recommendation engine in delivering relevant content.
2. **Content Creation and Sharing:** Provide tools to create and share posts, reels, and stories with multimedia support. This ensures a dynamic content environment where users can express themselves and interact creatively.
3. **Personalized Recommendation System:** This system designs an intelligent model that processes user activity such as likes, views, and interactions to understand preferences. It suggests relevant content to improve user experience and engagement.
4. **Real-Time Communication:** Real time enables instant communication through messaging and notifications. It allows quick updates for likes, comments, and activities, improving user interaction and ensuring smooth communication.
5. **Efficient Data Handling:** Design an optimized database using MongoDB to store user profiles, posts, and interactions. It ensures fast retrieval, efficient querying, and supports large volumes of data for better performance.
6. **Scalable Architecture:** Build a scalable system using React.js, Node.js, Express.js, and FastAPI. It supports future expansion, handles increasing user loads, and allows integration of new features efficiently.
7. **Security and Privacy:** Firstly, ensure secure authentication and protect user data through proper authorization and encryption methods. This helps maintain user privacy and prevents unauthorized access to the system. It also ensures safe data transmission and builds user trust in the platform.

4. LITERATURE REVIEW

1. Research in recommendation systems has evolved significantly over the past decades, with various techniques being proposed to improve personalization and user engagement. Collaborative filtering is one of the most widely used approaches, where recommendations are generated based on similarities between users or items. This method has been successfully applied in platforms such as e-commerce, streaming services, and social media applications to enhance user experience and retention.
2. Studies such as those by Koren et al. (2009) highlight the effectiveness of matrix factorization techniques in improving recommendation accuracy. Matrix factorization helps in decomposing large user-item interaction matrices into lower-dimensional representations, making it easier to identify latent features that influence user preferences. Similarly, Resnick and Varian (1997) introduced the concept of recommender systems and emphasized their importance in filtering vast amounts of information available on digital platforms.
3. Content-based filtering approaches focus on analyzing item attributes and user preferences, recommending items that are similar to those a user has previously interacted with. This approach is particularly useful when user interaction data is limited. However, it often lacks diversity in recommendations. Hybrid models, which combine both collaborative and content-based filtering techniques, have been proposed to overcome these limitations and achieve better performance and accuracy.
4. Recent advancements in recommendation systems include the use of deep learning and neural networks.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154296

Techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders have been applied to capture complex patterns in user behavior. These models can process large-scale datasets and learn intricate relationships between users and items, resulting in more accurate and dynamic recommendations. Deep learning-based systems are widely used in modern applications like video streaming and social networking platforms.

5. Another important area of research is context-aware recommendation systems, which consider additional factors such as time, location, and user mood to provide more relevant suggestions. These systems aim to deliver personalized experiences by understanding the context in which users interact with the platform. For example, a user may prefer different types of content during the day compared to nighttime, and context-aware systems adapt recommendations accordingly.
6. Social-based recommendation systems have also gained attention, where user connections, friendships, and social interactions are used to enhance recommendation quality. By leveraging social network data, these systems can identify influential users and recommend content based on social trends and peer influence. This approach is particularly relevant for social media platforms, where user relationships play a crucial role in content discovery.
7. Despite these advancements, recommendation systems face several challenges. One of the major issues is data sparsity, where user-item interaction data is limited, making it difficult to generate accurate recommendations. Another challenge is the cold-start problem, which occurs when new users or items are introduced into the system without sufficient data. Scalability is also a concern, as systems must handle millions of users and items efficiently without compromising performance.
8. Privacy and security are additional concerns in recommendation systems, as they rely heavily on user data. Ensuring that user information is protected while still providing personalized recommendations is a critical challenge. Researchers have proposed techniques such as differential privacy and data anonymization to address these issues.
9. In the context of social media applications, recommendation systems play a vital role in determining the content displayed to users. Traditional platforms often rely on simple algorithms based on popularity or recency, which may not always align with user interests. This can lead to information overload and reduced user satisfaction. Therefore, there is a growing need for intelligent systems that can filter and prioritize content effectively.
10. NeoAI addresses these challenges by implementing a collaborative filtering-based recommendation system integrated within a full-stack architecture. The system collects user interaction data such as likes, comments, shares, and watch time to build a user-item interaction matrix. Based on this data, similarity between users is calculated, and relevant content is recommended accordingly.
11. The use of a Python-based FastAPI microservice for machine learning operations allows the system to efficiently process data and generate recommendations in real-time. Additionally, the modular architecture using React.js, Node.js, and MongoDB ensures scalability and flexibility, enabling the platform to handle increasing user loads.
12. Furthermore, NeoAI focuses on improving user engagement by continuously updating recommendations based on real-time interactions. This dynamic approach ensures that users receive fresh and relevant content tailored to their preferences. The system also supports various social media features such as posts, reels, stories, messaging, and notifications, making it a comprehensive platform.
13. In addition to collaborative filtering, future enhancements of NeoAI may include hybrid recommendation techniques and deep learning models to further improve accuracy and personalization. The integration of advanced analytics and user behavior tracking can also provide deeper insights into user preferences, enabling more effective recommendations.
14. Overall, the literature highlights the importance of recommendation systems in modern applications and the continuous advancements in this field. NeoAI builds upon these existing techniques and aims to provide an efficient, scalable, and user-centric solution for personalized content delivery in social media platforms.
15. Another emerging area in recommendation systems is the use of reinforcement learning techniques. These systems learn dynamically from user interactions by continuously updating recommendations based on feedback. This approach helps in adapting to changing user preferences over time and improves long-term user engagement.
16. Graph-based recommendation systems have also gained popularity, where relationships between users and items are represented as networks. By analyzing these connections, the system can discover hidden patterns and provide more accurate and diverse recommendations. This method is especially useful in social media platforms where user relationships are interconnected.
17. Explainable recommendation systems are becoming increasingly important, as users prefer to understand why certain content is being recommended to them. Providing transparency in recommendations not only builds

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154296

trust but also improves user satisfaction. Researchers are focusing on developing models that can justify their suggestions in a clear and interpretable manner.

18. Explainable recommendation systems focus on providing reasons behind suggested content. This increases transparency, builds user trust, and improves overall satisfaction. It also helps users better understand the relevance of recommendations.
19. Reinforcement learning is an emerging approach where systems improve recommendations based on continuous user feedback. It helps in adapting to user behavior and enhances long-term engagement. It also enables the system to learn optimal strategies over time.

5. PROPOSED METHODOLOGY

To construct an effective intelligent recommendation system, the proposed approach is to support it by capturing user interactions using backend tracking mechanisms and activity data. The algorithm employs a machine learning-based collaborative filtering technique for high accuracy while recommending content. The system first collects interaction data such as likes, comments, views, and watch time through the application interface. This data then goes through the preprocessing phase using data transformation techniques toextract meaningful features. The implemented recommendation model, inspired by the work of Koren et al. (2009) and Aggarwal (2016), focuses on identifying user similarities and predicting relevant content. The system classifies user preferences into different interest categories such as entertainment, education, and lifestyle content. The interaction engine allows users to engage with posts and tracks their behavior while maintaining activity logs as discussed in modern recommendation research. Furthermore, an integrated interface provides users with personalized content feeds and suggestions to enhance engagement, similar to approaches discussed in recent AI-driven platforms. The use of REST APIs enables smooth communication between frontend and backend services, ensuring efficient data flow and system performance. Additionally, real-time features such as messaging and notifications are implemented using Socket-based communication to improve user interaction. The platform also manages user profiles, posts, and interactions efficiently to provide a personalized and scalable experience. The proposed methodology combining these features is designed to be a user-friendly and effective solution that enhances content discovery and improves user engagement within the NeoAI social media application.

6. METHODOLOGY

Agile Development Methodology (Lifecycle):

Requirement Gathering: In this phase, all system requirements are identified, including features like user authentication, post creation, reels, messaging, notifications, and AI-based recommendations. User needs and project goals are clearly defined to ensure proper planning.

Planning: The complete project is divided into smaller tasks or modules called sprints. Each sprint focuses on specific features such as frontend UI, backend APIs, or recommendation system. Timelines and priorities are assigned.

Design: System architecture and user interface designs are prepared. This includes designing database schemas, API structures, and UI layouts to ensure smooth user experience and efficient system flow.

Development: Developers implement features step-by- step. Frontend components are built using React.js, backend APIs are created using Node.js and Express.js, and machine learning modules are developed using FastAPI.

Testing: Each module is tested individually and collectively to identify bugs and errors. Functional testing, API testing, and UI testing are performed to ensure system reliability.

Deployment: After successful testing, the system is deployed on servers or cloud platforms, making it accessible to users.

Feedback & Improvement: User feedback is collected and analyzed. Based on feedback, improvements and new features are added in the next iteration, making the system more efficient.

Full-Stack Development Approach (Lifecycle):

Frontend Development: The user interface is developed using React.js, focusing on responsive design and smooth user interaction. Pages like login, feed, profile, and chat are created.

Backend Development: Server-side logic is implemented using Node.js and Express.js. APIs are created to handle user requests such as login, post creation, and fetching data.

Database Management: MongoDB is used to store user data, posts, interactions, and messages. Proper schema design and indexing are applied for efficient data handling.

API Integration: Frontend and backend are connected through REST APIs. Data flows between client and server seamlessly to ensure real-time updates.

ML Integration: A FastAPI-based microservice is integrated to handle machine learning operations like generating recommendations based on user activity.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154296

Testing & Debugging: All layers are tested together to ensure smooth communication. Errors are identified and resolved to maintain system stability.
Deployment: Frontend and backend are deployed on hosting platforms. The system is configured to handle real- time user requests efficiently.

TECHNOLOGIES USED

1. **React.js:** React.js is used for building the frontend of the application, providing a dynamic and responsive user interface. It enables efficient rendering of components and ensures a smooth user experience across different devices.
2. **Node.js and Express.js:** Node.js and Express.js are used for backend development to handle server- side logic and API creation. They manage user authentication, data processing, and communication between frontend and database efficiently.
3. **MongoDB:** MongoDB is used as the database to store user data, posts, comments, messages, and interactions. It provides flexible schema design and efficient data handling for large-scale applications.
4. **FastAPI (Python):** FastAPI is used as a microservice for implementing machine learning operations. It processes user interaction data and generates personalized recommendations using collaborative filtering techniques.
5. **Socket.io:** Socket.io is used to implement real-time features such as instant messaging and live notifications. It ensures seamless and low-latency communication between users.
6. **JWT Authentication:** JSON Web Tokens (JWT) are used for secure user authentication and authorization. It ensures that only authorized users can access protected routes and resources.
7. **Cloud Storage (e.g., AWS / Cloudinary):** Cloud storage services are used to store and manage multimedia content such as images and videos. This improves scalability and ensures fast content delivery.

7. SYSTEM REQUIREMENT

For Developers:
Hardware Platform:

- Processor: Intel Core i5 or higher
- RAM: 8 GB or above
- GPU: Optional (for ML tasks), 2 GB or above recommended
- Hard Disk: 256 GB or above

Software Platform:

- Visual Studio Code (VS Code)
- Git & GitHub
- Node.js and npm
- MongoDB Compass / Atlas
- Python (for FastAPI and ML modules)
- Postman (for API testing)

Operating System: Windows 10 or above / macOS / Linux

For Users:

Hardware Platform:

- Processor: Octa-core processor (Snapdragon 665 or equivalent and above)
- RAM: 4 GB or above
- Storage (ROM): 32 GB or above

Software Platform:

- Web Browser (Google Chrome, Edge, or Safari)
- Internet Connection (4G/5G or Wi-Fi)

Operating System: Android 10 or above / iOS 13 or above / Windows 10+ (for web access)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154296

8. SYSTEM DESIGN

8.1 E-R Diagram

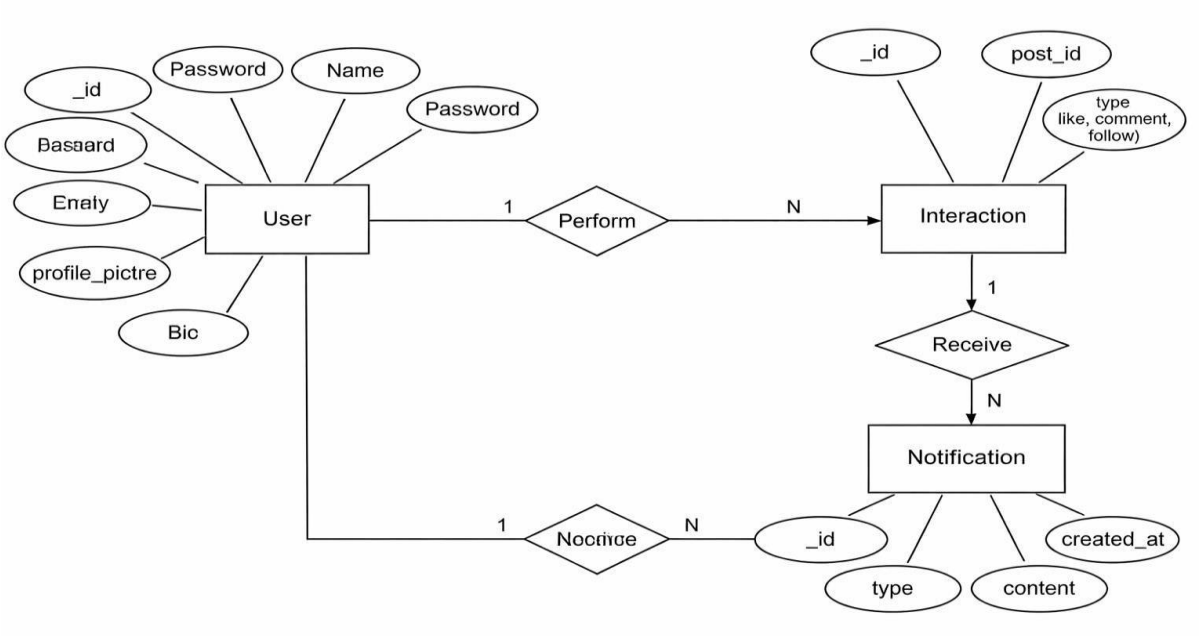

Figure 1 E-R Diagram

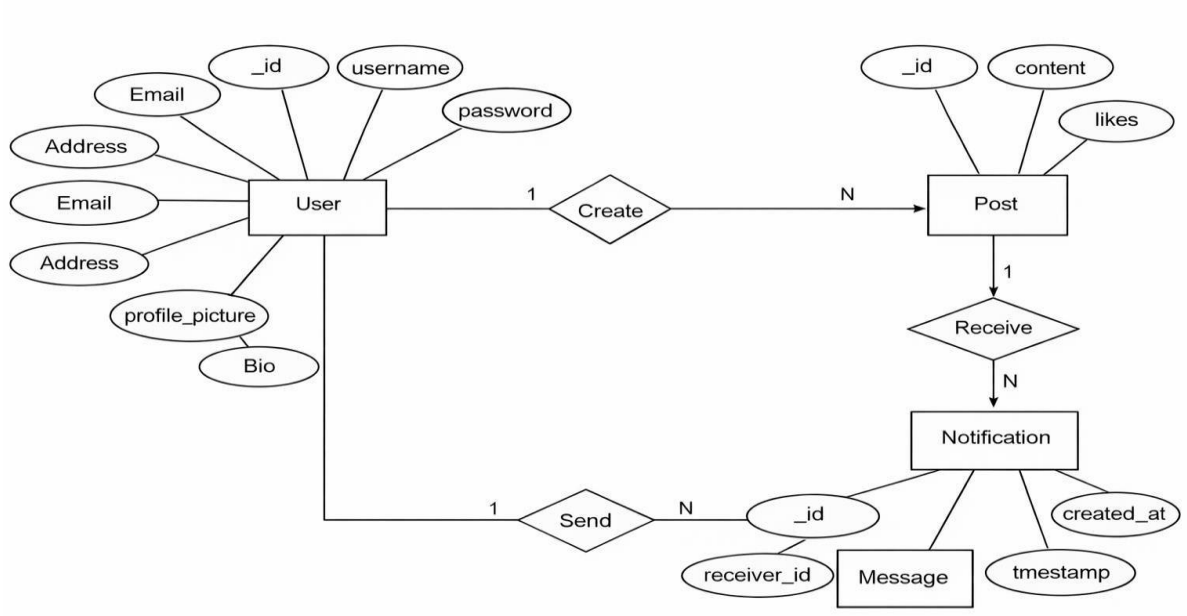

Figure 2 E-R Diagram

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154296

8.2 Data Flow Diagram

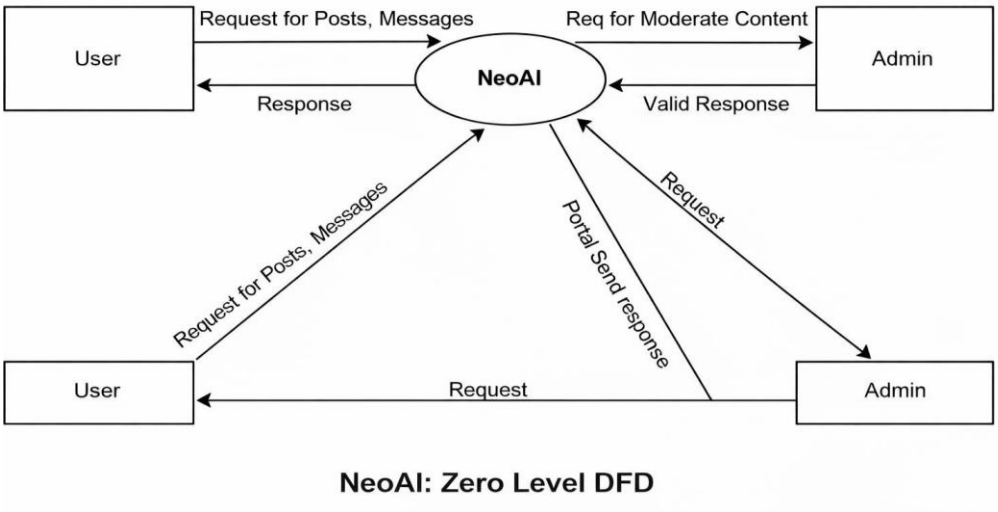

Figure 3 DFD 0 level

8.3 Use Case Diagram

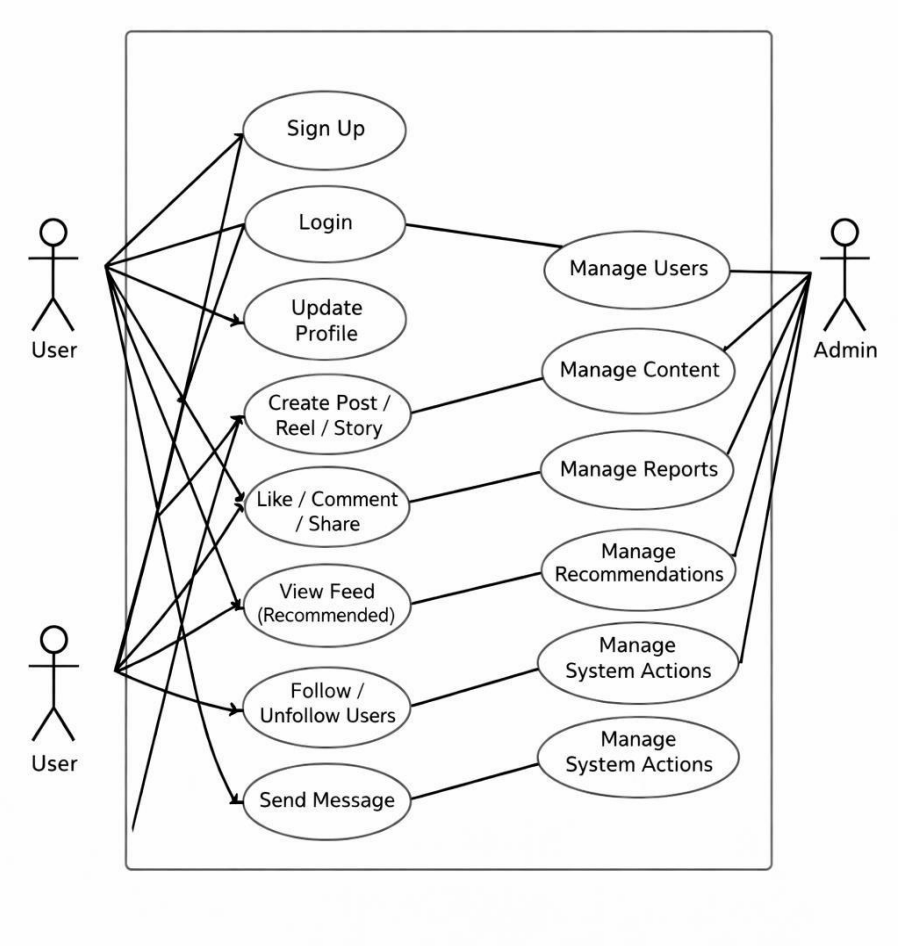

Figure 4 Use Case Diagram

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154296

9. IMPLEMENTATION

9.1 Registration page:

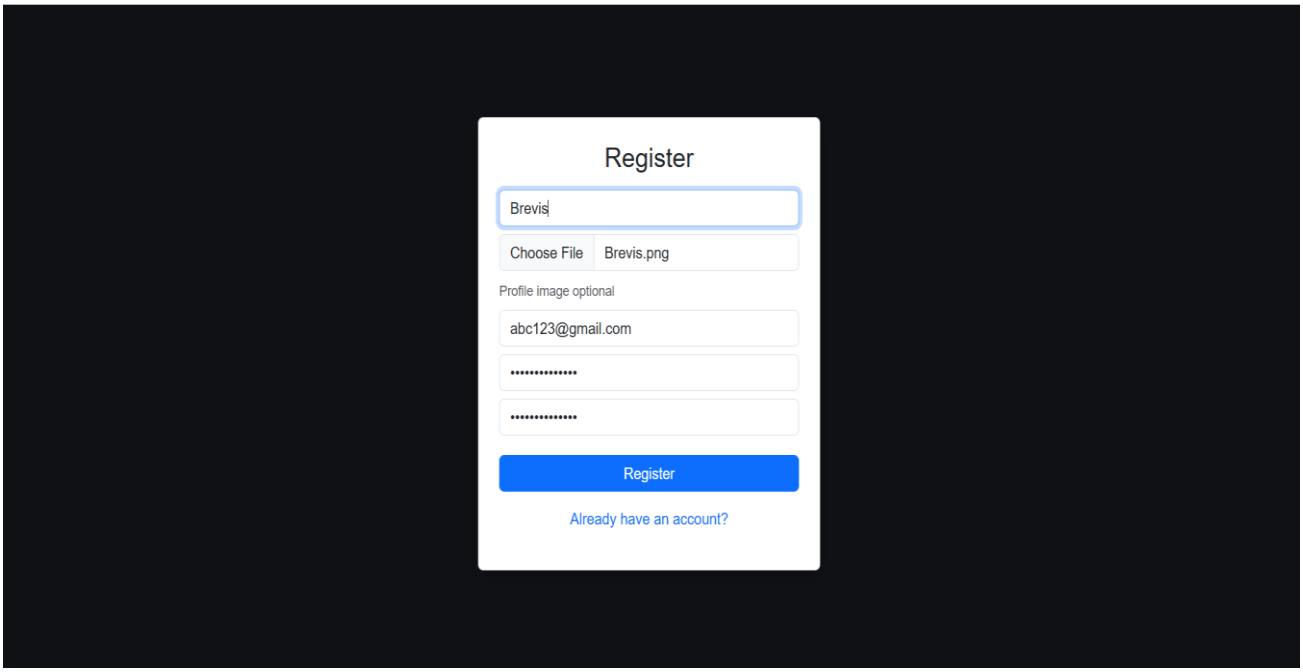

Figure 1 Registration Screen

9.2 Login Screen:

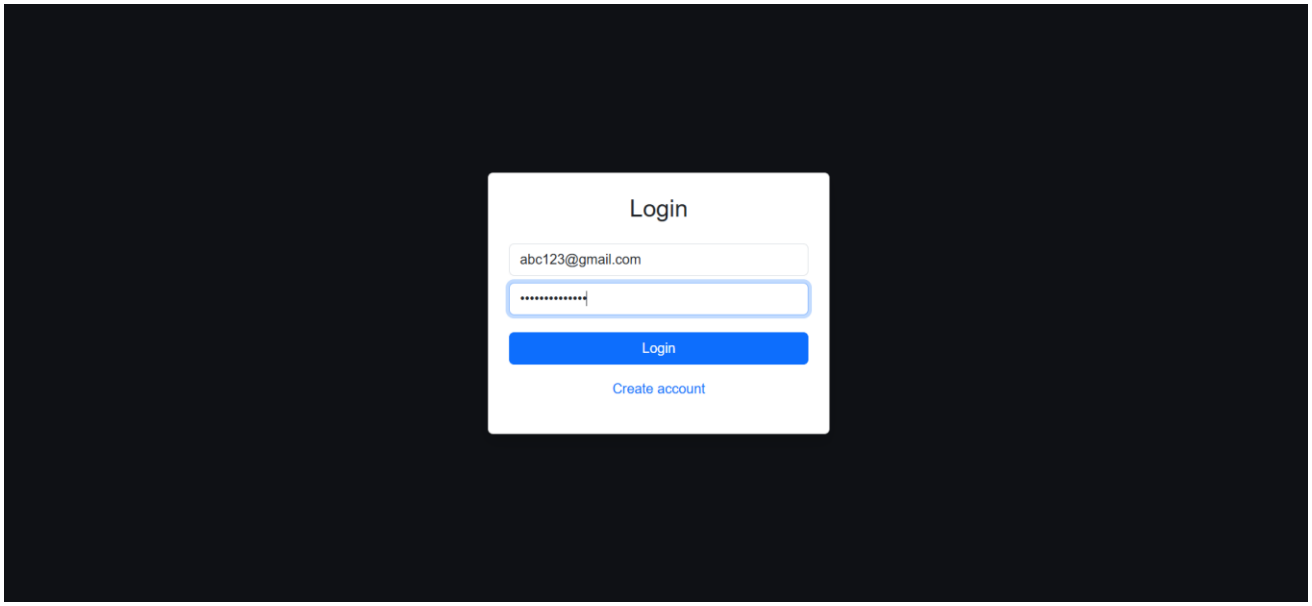

Figure 2 Login Screen

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154296

9.3 Home Screen:

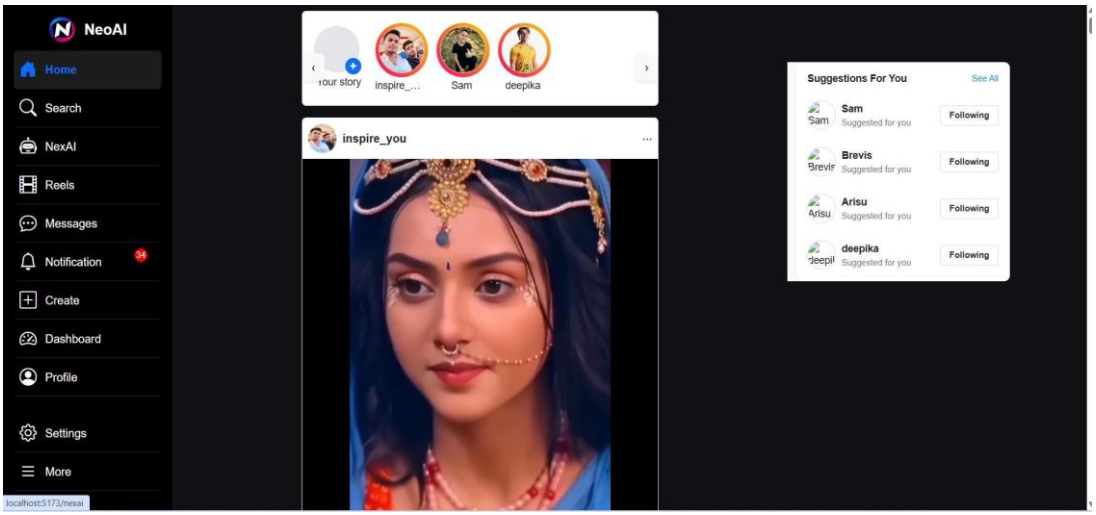

Figure 3 Home Screen

9.4 Profile Screen:

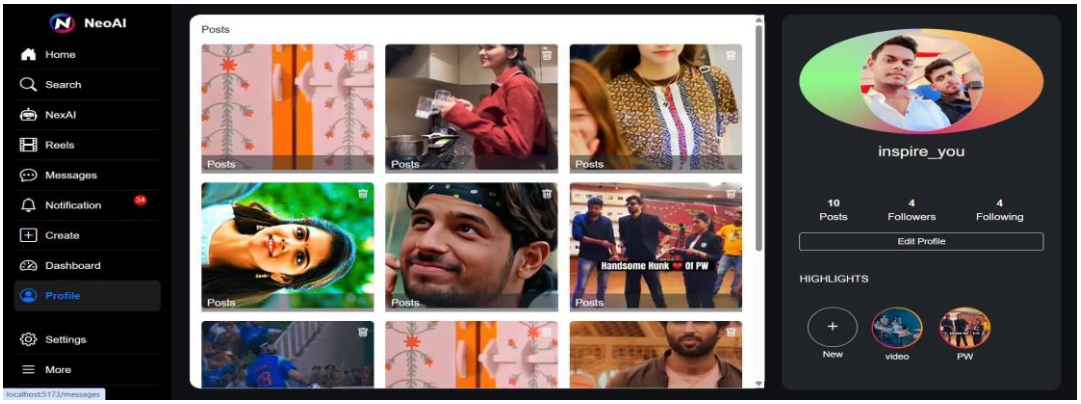

Figure 4 Profile Screen

9.5 Message Screen:

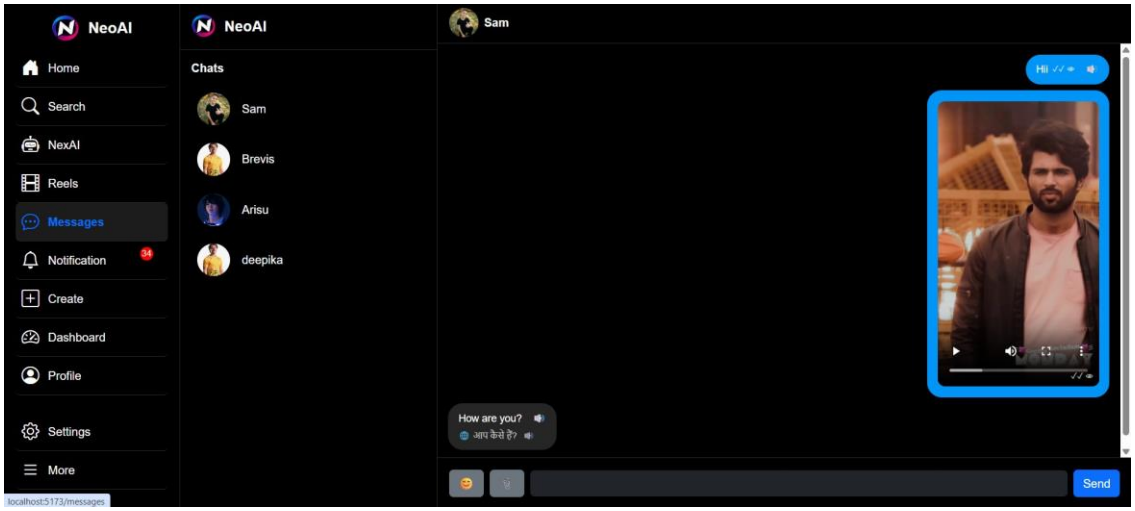

Figure 5 Message Screen

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154296

9.6 Notification Screen:

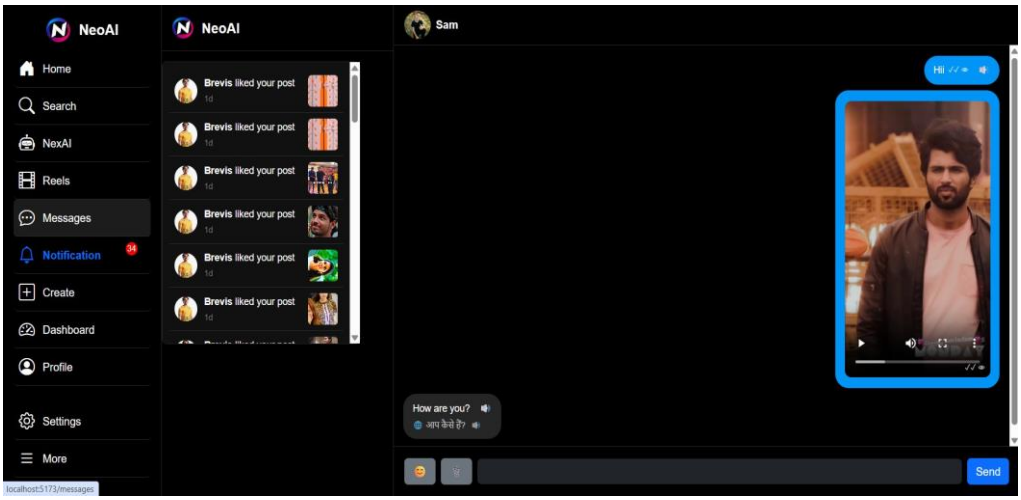

Figure 6 Notification Screen

9.7 Post View Screen:

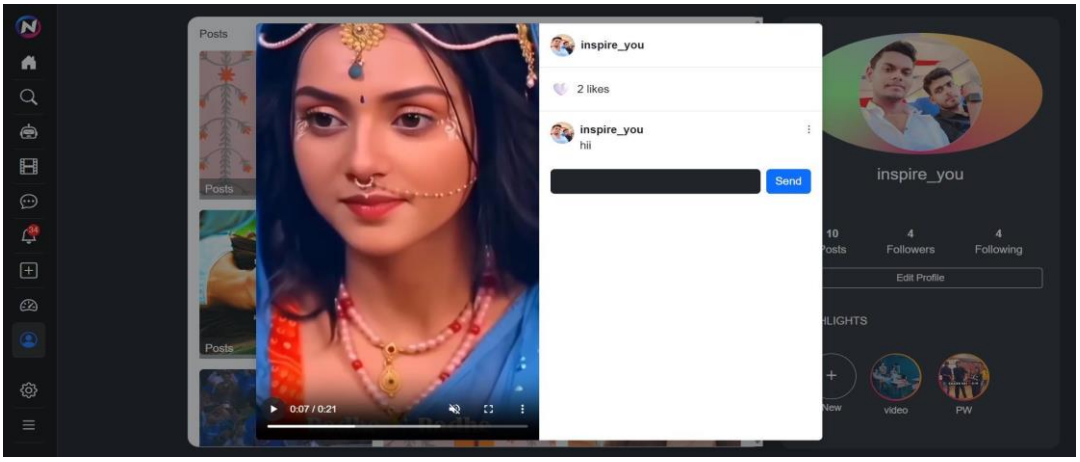

Figure 7 Post View Screen

9.8 Reel st View Screen:

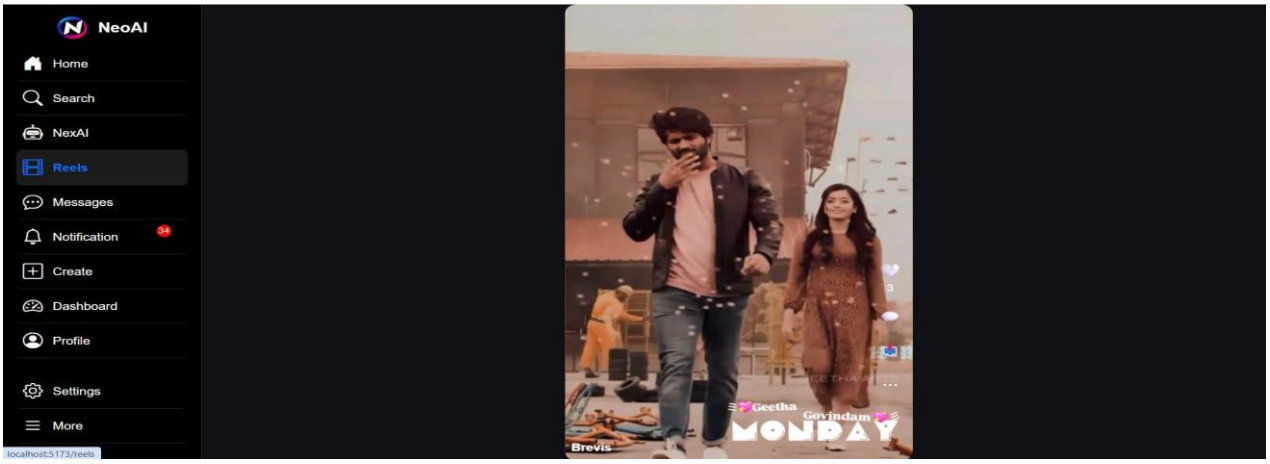

Figure 8 Reel View Screen

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154296

10. SYSTEM FLOW

When users open the NeoAI application, the first screen presented is the Login/Sign Up page. Users must either register a new account or log in using existing credentials to access the platform. After entering details, the system performs authentication; if it fails, users are redirected to login again. If authentication is successful, users are granted access to the application. Users are then directed to the Home Screen, which acts as the central hub of the platform. From here, users can navigate to different sections through clearly defined options. The Home Feed displays AI-based personalized posts, reels, and stories based on user interactions. Explore helps users discover trending content, Create allows content uploading, and Messages enable real- time chat. The Profile section allows users to view and edit their personal information and account settings.

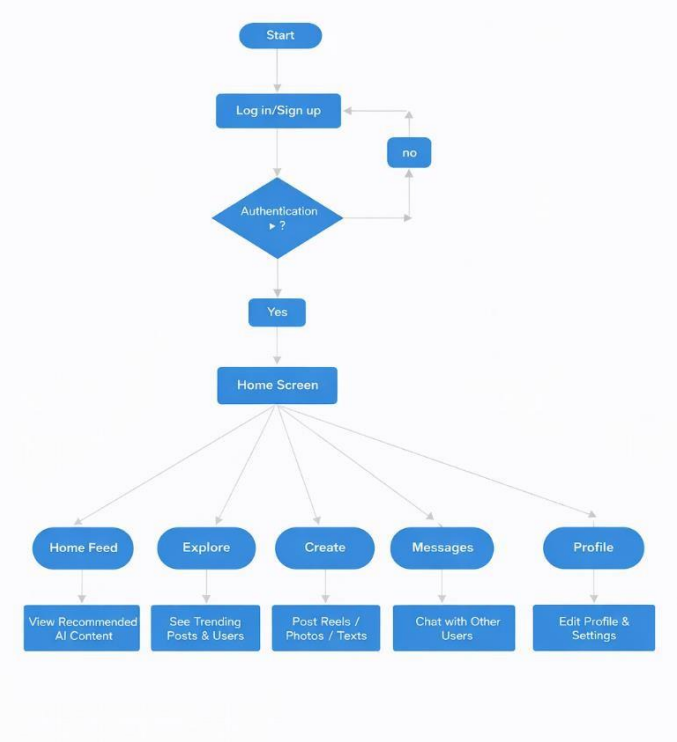

Figure 11 System Flow

11. RESULT

The research paper introduces the successful development of an AI-powered social media application using modern full-stack technologies that make the platform user-friendly and enhance personalized content delivery. The system is based on machine learning techniques implemented through a FastAPI microservice, which analyzes user interactions such as likes, comments, and views to generate accurate recommendations. This improves content relevance and increases user engagement across the platform. The use of React.js ensures an interactive and responsive user interface with cross-platform accessibility, enhancing overall usability. The integration of real-time communication technologies enables users to send instant messages and receive live notifications, improving connectivity and interaction within the platform. MongoDB is utilized for efficient data storage and management, ensuring smooth handling of user data, posts, and interactions within the application. Additionally, cloud-based storage solutions are implemented to manage multimedia content such as images and videos effectively. The inclusion of AI-based recommendation systems ensures that users receive personalized feeds, making content discovery more efficient and engaging. Overall, the research demonstrates that integrating modern web technologies with machine learning can create an intelligent and scalable social media platform that enhances user experience and engagement.

12. CONCLUSION

Eventually, the NeoAI application is developed as a comprehensive AI-powered social media platform consisting of content creation, interaction, and intelligent recommendation phases. Instead of traditional social media approaches, several advanced features have been implemented. For example, there is real-time communication support, AI-based

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154296

recommendation system, cloud storage integration, and user profile management through which users can easily interact and manage their digital presence effectively. The recommendation algorithm, on the other hand, utilizes user interaction data and insights from research such as collaborative filtering and modern machine learning techniques. It effectively improves engagement and accuracy in content delivery using data-driven models and intelligent system design. This approach aligns with recent advancements in recommendation systems, where studies emphasize the importance of AI and machine learning in enhancing user experience and personalization. The system addresses key challenges such as content overload and irrelevant recommendations by applying efficient filtering techniques and scalable architecture. As an advanced and adaptive system, NeoAI integrates multiple technologies including full-stack development and machine learning, demonstrating the effectiveness of combining modern technologies to solve real-world digital interaction challenges and improve overall user engagement and platform efficiency.

REFERENCES

[1]. React, "React Official Documentation," Available at https://react.dev/.
[2]. React, "React Hooks API Reference," Available at https://react.dev/reference/react.
[3]. Mozilla Developer Network, "JavaScript Guide," Available at https://developer.mozilla.org/en-US/docs/Web/JavaScript.
[4]. Mozilla Developer Network, "CSS Documentation," Available at https://developer.mozilla.org/en-US/docs/Web/CSS.
[5]. Node.js, "Node.js Documentation," Available at https://nodejs.org/en/docs.
[6]. Express.js, "Express.js Documentation," Available at https://expressjs.com/.
[7]. Lokesh Gupta, "REST API Tutorial," Available at https://restfulapi.net/.
[8]. MongoDB, "MongoDB Documentation," Available at https://www.mongodb.com/docs/.
[9]. Mongoose, "Mongoose Documentation," Available at https://mongoosejs.com/docs/.
[10]. JSON Web Token, "Introduction to JSON Web Tokens," Available at https://jwt.io/introduction.
[11]. Mozilla Developer Network,"Authentication Concepts," Available at https://developer.mozilla.org/en-US/docs/Web/Security/Authentication.
[12]. Mozilla Developer Network, "HTTP Cookies," Available at https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies.
[13]. Python, "Python Documentation," Available at https://docs.python.org/3/.
[14]. Wes McKinney, "Pandas Documentation," Available at https://pandas.pydata.org/docs/.
[15]. NumPy Developers, "NumPy Documentation," Available at https://numpy.org/doc/.
[16]. Scikit-learn Developers, "Scikit-learn Documentation," Available at https://scikit- learn.org/stable/.
[17]. Axios, "Axios Documentation," Available at https://axios-http.com/docs/intro.
[18]. FFmpeg, "FFmpeg Documentation," Available at https://ffmpeg.org/documentation.html.
[19]. Fluent-ffmpeg, "Fluent-ffmpeg GitHub Repository," Available at https://github.com/fluent- ffmpeg/node-fluent-ffmpeg.
[20]. Instagram, "Instagram Platform," Available at https://www.instagram.com/.
[21]. Facebook, "Facebook Platform," Available at https://www.facebook.com/.
[22]. Git SCM, "Git Documentation," Available at https://git-scm.com/docs.
[23]. GitHub, "GitHub Platform," Available at https://github.com/.
[24]. Stack Overflow, "Developer Community," Available at https://stackoverflow.com/.
[25]. GeeksforGeeks, "Computer Science Portal," Available at https://www.geeksforgeeks.org/.
[26]. FreeCodeCamp, "Learn to Code Platform," Available at https://www.freecodecamp.org/.
[27]. W3Schools, "Web Development Tutorials," Available at https://www.w3schools.com/.
[28]. Javatpoint, "Software Engineering Tutorial," Available at https://www.javatpoint.com/software-engineering-tutorial.
[29]. Educative, "System Design Basics," Available at https://www.educative.io/blog/system-design- basics.
[30]. Red Hat, "Microservices Architecture Guide," Available at https://www.redhat.com/en/topics/microservice s.

