International Journal for Modern Trends in Science and Technology
Volume 12, Issue 04, pages 1238-1245
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue04.html
DOI: https://doi.org/10.5281/zenodo.19673308

Design and development social media web application using python full stack

L. Dana Ratna Kishore, G. Yadav Balarama Vamsi, Ch. Sravanthi, D. Durga Sri, Ch. Keshav Rao

Department of Computer Science and Engineering, D.N.R. College of Engineering & Technology, Balusumudi, Bhimavaram, Andhra Pradesh, India

To Cite this Article
L. Dana Ratna Kishore, G. Yadav Balarama Vamsi, Ch. Sravanthi, D. Durga Sri & Ch. Keshav Rao (2026). Design and development social media web application using python full stack. International Journal for Modern Trends in Science and Technology, 12(04), 1238-1245. https://doi.org/10.5281/zenodo.19673308

Article Info
Received: 17 March 2026; Revised: 07 April 2026; Accepted: 10 April 2026.

KEYWORDS

Social Media Platform, Instagram Replica, Artificial Intelligence (AI), Augmented Reality (AR), Personalized Content, E-commerce Integration.

ABSTRACT

The rise of social media platforms has revolutionized the way people connect, communicate, and share content globally. This study focuses on reimagining the Instagram experience by introducing innovative features and functionalities ina social media replica platform. The project aims to explore enhanced user engagement through personalized content delivery, AI-driven recommendations, augmented reality (AR) filters, and seamless integration with e-commerce. By addressing existing gaps in user experience, privacy, and content moderation, the proposed platform aspires to redefine how users interact with social media. This initiative leverages modern frameworks, cutting-edge technology, and a user-centered design approach to create an immersive, inclusive, and dynamic social networking ecosystem.

1. INTRODUCTION

In the modern digital era, social media platforms have become an essential part of everyday life. People use social media to communicate, share information, express ideas, and connect with others globally. With the rapid growth of internet technologies, there is an increasing demand for efficient, scalable, and secure web applications that can handle large user interactions. Recent studies highlight how social big data and intelligent systems are transforming online platforms by enabling automated knowledge extraction and improved user engagement [1].

This project focuses on the design and development of a Social Media Web Application using Python full stack technologies. The system allows users to create accounts, share posts, like and comment on content, and interact with other users in real time. The backend of the application is developed using the Django framework, while the frontend is built using HTML, CSS, JavaScript, and Bootstrap. The application follows a structured approach where user data is stored securely in a

database and processed efficiently through server-side logic. Modern research emphasizes the importance of integrating machine learning and intelligent models in social media systems, such as detecting bots, spam, and fake accounts to improve platform security and authenticity [2][3][6]. Additionally, advanced sentiment analysis techniques like LSTM and graph-based models are used to understand user opinions and enhance user experience [4][5].

Features such as authentication, profile management, and content sharing are integrated to provide a complete social networking experience. Furthermore, techniques like Sybil detection and adaptive network analysis help in identifying malicious users and maintaining platform integrity [7]. Intelligent systems for online interaction and recommendation also contribute to improving user engagement and personalization [8][9]. This system aims to demonstrate how modern web technologies, combined with intelligent data processing techniques, can be used to build interactive, secure, and user-friendly platforms.

1.1 HARDWARE REQUIREMENTS

Hardware requirements refer to the physical components needed to run the application smoothly. The proposed system is designed to operate on standard computing systems without requiring expensive or high-end devices. Proper hardware configuration ensures better performance, faster processing, and efficient handling of multiple user requests.

Minimum Hardware Requirements:

- Processor: Intel i3 or equivalent processor
- RAM: 4 GB
- Hard Disk: 500 GB free space
- System Type: 64-bit Operating System
- Input Devices: Keyboard and Mouse
- Output Device: Monitor

Recommended Hardware Requirements:

- Processor: Intel i5 or higher
- RAM: 8 GB or above
- Storage: 1 TB HDD or 256 GB SSD
- Graphics: Integrated Graphics Card
- Internet: Broadband internet connection

Additional Hardware Considerations:

- System should support multitasking for running IDE, server, and browser simultaneously
- RAM should handle backend processing and database operations efficiently
- Storage should support installation of development tools, libraries, and project files
- Processor should support smooth execution of Python programs and Django framework
- Monitor should provide high resolution for better UI design and testing experience
- Input devices should support continuous coding and navigation without interruption
- Internet connection should support downloading frameworks, libraries, and updates
- System should allow continuous running of local development server without lag
- Hardware should support testing on multiple browsers and environments
- Backup storage devices should be available to prevent data loss
- System should handle multiple tabs, applications, and processes simultaneously
- Hardware should support virtualization tools if required for advanced testing
- Cooling system should maintain system performance during long working hours
- Power backup (UPS) is recommended to avoid data loss during power failure
- Hardware should support smooth rendering of frontend interfaces and media content
- System should be capable of handling database growth and increased storage needs
- Devices should support connectivity options like USB and external drives for data transfer

1.2 SOFTWARE REQUIREMENTS

Software requirements include the tools, technologies, and platforms required to develop and execute the application. The proposed system uses modern and widely used open-source technologies, which make development easier, cost-effective, and scalable. Proper software configuration ensures smooth integration between frontend, backend, and database components.

Basic Software Requirements:

- Operating System: Windows / Linux / macOS

- Programming Language: Python - Framework: Django
- Frontend Technologies: HTML, CSS, JavaScript, Bootstrap
- Database: SQLite / MySQL
- Web Browser: Google Chrome / Mozilla Firefox

Development Tools:

- IDE: Visual Studio Code / PyCharm
- Version Control: Git
- Package Manager: pip
- Debugging Tools: Built-in debugger
- Server: Django Development Server

Additional Software Considerations:

- Operating system should support installation and configuration of Python environment
- Python should support required libraries, modules, and third-party packages
- Django framework should support secure authentication, routing, and backend processing
- Frontend technologies should support responsive design and interactive UI components
- Database should support structured storage of user data, posts, and interactions
- Browser should support modern JavaScript features and fast rendering
- IDE should provide features like syntax highlighting, auto-completion, and debugging
- Version control system should help in managing code versions and collaboration
- Package manager should install, update, and manage dependencies efficiently
- Software environment should support API integration and external services
- System should allow testing and debugging of each module independently
- Environment should support secure session handling and user authentication
- Database system should support indexing and efficient query execution
- Software tools should allow smooth integration between frontend and backend
- System should support cross-platform compatibility for wider usage
- Software should allow easy deployment on servers or cloud platforms
- Environment should support error handling and logging mechanisms
- Software should support scalability for handling increased users and data
- Tools should allow continuous development and future enhancements
- System should support backup and recovery mechanisms for data safety
- Software should be compatible with different versions and updates
- Environment should allow performance monitoring and optimization
- Framework should support modular development for easy maintenance
- Software should support secure data transmission using protocols like HTTPS

2. LITERATURE REVIEW

The development of social media web applications has gained significant attention in recent years due to the rapid growth of internet usage and digital communication. Researchers and developers have explored various technologies, frameworks, and methodologies to build efficient, scalable, and secure web-based platforms. Social media applications require a combination of frontend design, backend processing, database management, and security mechanisms to provide a seamless user experience. The following section presents a comprehensive review of existing research works related to social media application development and web technologies.

1.SOCIAL NETWORKING SYSTEMS AND THEIR ARCHITECTURE

Authors: Boyd, D. M., Ellison, N. B. This study provides a detailed overview of social networking systems and their underlying architecture. The authors explain how social media platforms are designed to support user interaction, content sharing, and communication. They highlight the importance of user profiles, friend connections, and activity feeds in building social networks. The study also discusses how system architecture plays a key role in handling large volumes of user data and requests. This research helps in

understanding the fundamental structure required for developing social media applications.

2.FULL STACK DEVELOPMENT USING PYTHON AND DJANGO

Authors: William S. Vincent This research focuses on the use of Python and Django for developing full stack web applications. Django is a high-level framework that follows the Model-View- Template (MVT) architecture, which separates application logic, user interface, and data management. The study explains how Django provides built-in features such as authentication, session management, and database integration, which simplify development. It also highlights Django's scalability and security features, making it suitable for developing large-scale web applications like social media platforms.

3.FRONTEND TECHNOLOGIES FOR INTERACTIVE WEB APPLICATIONS

Authors: Flanagan, D. This study emphasizes the importance of frontend technologies such as HTML, CSS, and JavaScript in building interactive user interfaces. HTML is used for structuring web pages, CSS for styling, and JavaScript for adding dynamic behavior. The study also discusses frameworks like Bootstrap that enable responsive design, ensuring compatibility across different devices. A well- designed frontend improves user engagement and provides a better user experience in social media applications.

4.DATABASE SYSTEMS IN WEB APPLICATIONS

Authors: R. Elmasri, S. Navathe This research explains the role of database systems in managing and storing large amounts of user data. Social media applications require efficient database management to handle user profiles, posts, comments, and interactions. The study highlights the importance of relational databases such as MySQL and PostgreSQL. It also discusses normalization techniques, indexing, and query optimization, which improve performance and reduce redundancy.

5.WEB APPLICATION SECURITY MECHANISMS

Authors: OWASP Foundation Security is a critical aspect of social media applications as they handle sensitive user data. This study identifies common vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). It suggests implementing strong authentication mechanisms, secure password storage, and input validation techniques. The study also highlights the importance of using secure frameworks like Django, which provide built-in protection against common attacks.

3. PROPOSED SYSTEM

The proposed system, "Design and Development of Social Media Web Application Using Python Full Stack," is developed to overcome the limitations of the existing system. The project provides an advanced social media platform where users can create accounts, log in securely, update their profiles, share posts, upload images, like and comment on posts, and communicate with other users. The proposed system is developed using Python Full Stack technologies. The frontend of the application is designed using HTML, CSS, JavaScript, and Bootstrap to provide an attractive and responsive user interface. The backend is developed using Python and Django framework, which ensures smooth processing of user requests and secure management of application data. The system uses a relational database such as SQLite or MySQL to store user details, posts, comments, and messages efficiently.

SYSTEM ARCHITECTURE

The system follows a **3-layer architecture (Frontend → Backend → Database)** to make the application simple, scalable, and easy to manage.

- The **Frontend (User Interface)** is built using HTML, CSS, JavaScript, and Bootstrap. It is where users interact with the system, such as creating posts, liking, commenting, and chatting.
- The **Backend (Server Side)** is developed using Django. It handles all the logic like user authentication, processing requests, managing posts, and controlling application flow.
- The **Database (Data Storage)** stores all the information such as user details, posts, comments, and messages using SQLite/MySQL.

When a user performs an action (like posting or liking), the request goes from the frontend to the backend, which processes it and stores or retrieves data from the database, then sends a response back to the user

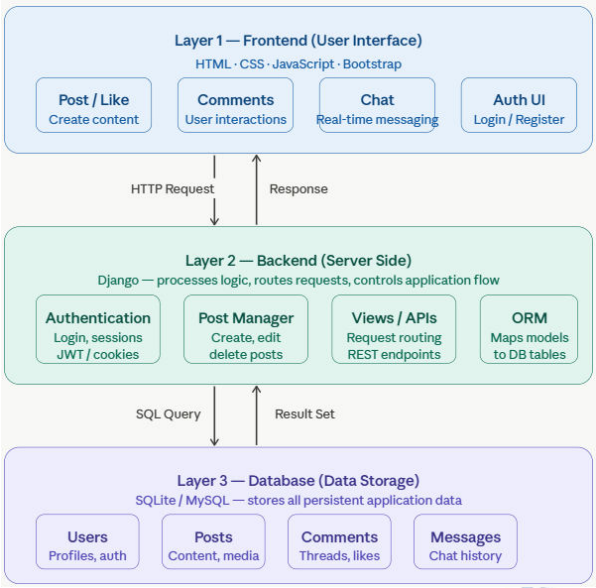

Fig: Architecture Diagram

Use case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

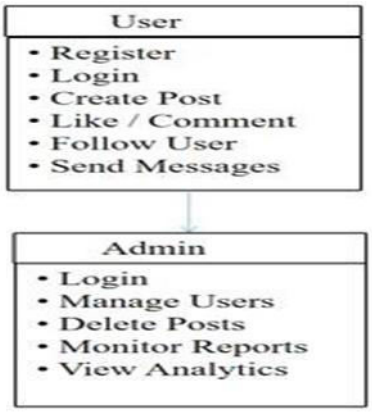

Fig: Use case Diagram

3.1.2 Class Diagram

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

Fig: Class Diagram

3.3 Evaluation Metrics

To rigorously assess the performance and effectiveness of the proposed Social Media Replica platform, a comprehensive set of evaluation metrics is employed. These metrics are designed to evaluate system performance, user engagement, scalability, security, and the effectiveness of intelligent components such as AI-driven recommendations and content moderation systems.

3.3.1. User Engagement Metrics

User engagement is a primary indicator of platform success and reflects the level of user interaction and satisfaction. The following metrics are considered:

- **Daily Active Users (DAU)** and **Monthly Active Users (MAU)** to measure platform adoption.
- **Engagement Rate (ER)**, defined as the ratio of total interactions (likes, comments, shares) to total posts.

$$ER = \frac{Total\ Interactions}{Total\ Posts}$$

- **Average Session Duration**, indicating the time spent by users on the platform.
- **Retention Rate**, which measures the percentage of users returning after their first interaction.

These metrics provide insights into user behavior and platform stickiness.

3.3.2 System Performance Metrics

System performance is evaluated to ensure responsiveness and efficiency under varying workloads. Key metrics include:

- **Response Time (RT)**: Time taken by the server to respond to user requests.
- **Throughput (TP)**: Number of requests processed per unit time.
- **Latency**: Delay between user action and system response.

these metrics ensure that the platform delivers real-time interactions and seamless user experience.

3.3.3 Scalability Metrics

Scalability evaluates the system's ability to handle increasing workloads without degradation in performance. The following parameters are used:

- Maximum number of concurrent users supported.
- Database query execution time under high load.
- Horizontal and vertical scaling efficiency.

This ensures that the system remains robust as user traffic grows.

3.3.4 AI and Recommendation System Metrics

The platform integrates AI-based recommendation and sentiment analysis models. Their effectiveness is measured using:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision, **Recall**, and **F1-Score** for classification-based models:

$$Precision = \frac{TP}{TP+FP},\ Recall = \frac{TP}{TP+FN}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

These metrics evaluate how accurately the system recommends relevant content and detects sentiments.

3.3.5 Security Metrics

Security evaluation ensures protection against threats and unauthorized access. Metrics include:

- Authentication Success Rate
- Number of Detected Malicious Activities (bots, spam, fake accounts)
- Encryption Effectiveness
- Vulnerability Detection Rate

These measures ensure data integrity, confidentiality, and system trustworthiness.

3.3.6 Usability Metrics

Usability is critical for user adoption and satisfaction. It is evaluated using:

- System Usability Scale (SUS) scores
- Task Completion Rate
- Error Rate
- User Satisfaction Index

These metrics assess how easily users can interact with the platform.

3.3.7 Reliability Metrics

Reliability ensures continuous and stable system operation:

- System Uptime (%)
- Mean Time Between Failures (MTBF)

- Mean Time to Recovery (MTTR)

These metrics indicate the robustness and fault tolerance of the system.

3.3.8 Content Moderation Effectiveness

To maintain a safe environment, the automated moderation system is evaluated using:

- Detection accuracy of harmful or inappropriate content
- False Positive Rate (FPR) and False Negative Rate (FNR)
- Moderation response time

This ensures effective enforcement of community guidelines.

4. RESULTS

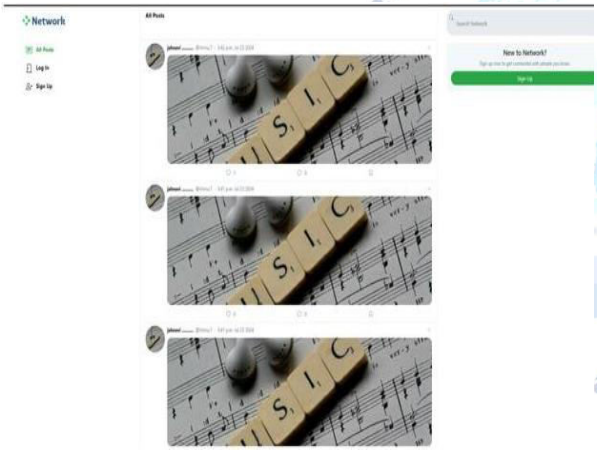

Fig: Home Page

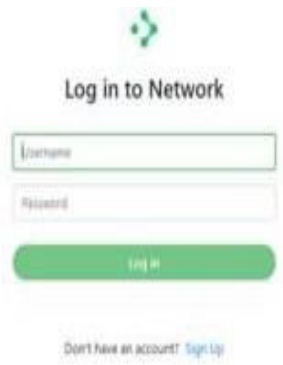

Fig: Login Page

Fig: Singup Page

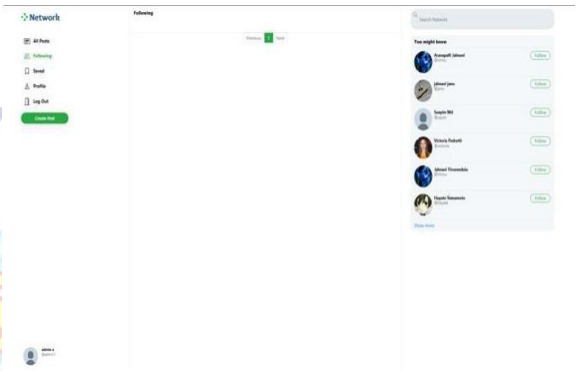

Fig: Following Page

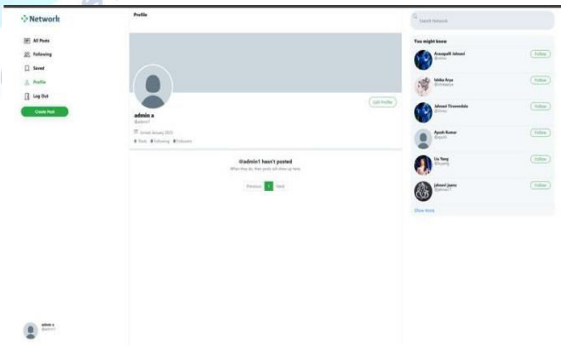

Fig: CREATE POST PAGE

CONCLUSION

The Social Media Web Application developed using Python Full Stack provides a platform where users can register, create profiles, share posts, and interact with others through likes, comments, and messages. The system improves privacy, security, and user experience compared to existing platforms. Using Django architecture makes the application organized, scalable,

and easy to maintain. Overall, the project successfully demonstrates how a modern social media platform can be designed and developed efficiently.

FUTURE SCOPE

The proposed Social Media Replica platform demonstrates a strong foundation by integrating modern web technologies, artificial intelligence, and user-centric design principles. However, there remains significant scope for further enhancement and expansion to meet evolving technological trends and user expectations. One of the key directions for future work is the integration of advanced artificial intelligence and deep learning models to improve personalization and recommendation systems. Future implementations can leverage reinforcement learning and transformer-based models to deliver highly contextual and real-time content recommendations. Additionally, incorporating predictive analytics can help anticipate user behavior and optimize content delivery strategies. Another important area of development is the enhancement of augmented reality (AR) and virtual reality (VR) capabilities. While the current system includes AR-based filters, future versions can extend this to immersive virtual environments, enabling users to interact in 3D social spaces, virtual events, and metaverse-like ecosystems. This would significantly enhance user engagement and redefine digital social interaction.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

1. Guedea-Noriega, H.H.; García-Sánchez, F. Integroly: Automatic Knowledge Graph Population from Social Big Data in the Political Marketing Domain. Appl. Sci. 2022, 12, 8116.
2. Alothali, E.; Salih, M.; Hayawi, K.; Alashwal, H. Bot-MGAT: A T ransfer Learning Model Based on a Multi-View Graph Attention Network to Detect Social Bots. Appl. Sci. 2022, 12, 8117.
3. Alkadri, A.M.; Elkorany, A.; Ahmed, C. Enhancing Detection of Arabic Social Spam Using Data Augmentation and Machine Learning. Appl. Sci. 2022, 12, 11388.
4. Yin, Z.; Shao, J.; Hussain, M.J.; Hao, Y.; Chen, Y.; Zhang, X.; Wang, L. DPG-LSTM: An Enhanced LSTM Framework for Sentiment Analysis in Social Media T ext Based on Dependency Parsing and GCN. Appl. Sci. 2023, 13, 354.
5. Gogula, S.D.; Rahouti, M.; Gogula, S.K.; Jalamuri, A.; Jagatheesaperumal, S.K. An Emotion-Based Rating System for Books Using Sentiment Analysis and Machine Learning in the Cloud. Appl. Sci. 2023, 13, 773.
6. Alsubaei, F.S. Detection of Inappropriate Tweets Linked to Fake Accounts on Twitter. Appl. Sci. 2023, 13, 3013.
7. Lu, H.; Gong, D.; Li, Z.; Liu, F.; Liu, F. SybilHP: Sybil Detection in Directed Social Networks with Adaptive Homophily Prediction. Appl. Sci. 2023, 13, 5341.
8. Xing, X.; Yang, C.; Lin, D.; Teng, D.; Chen, P.; Zhang, X. External Slot Relationship Memory for Multi-Domain Dialogue State T racking. Appl. Sci. 2023, 13, 8943.
9. Lipianina-Honcharenko, K.; Wolff, C.; Sachenko, A.; Desyatnyuk, O.; Sachenko, S.; Kit, I. Intelligent Information System for Product Promotion in Internet Market. Appl. Sci. 2023, 13, 9585.