

SALIM HABIB UNIVERSITY



**BS COMPUTER
SCIENCE**

COURSE PROJECT REPORT

Fall 2024

**SOCIAL MEDIA PLATFORM
(CONNECTIFY).**

SUBMITTED BY:

SADIA TEHREEM F23CSC043

BILAL NAEEM F23CSC008

FACULTY NAME

Ms. SAADIA KARIM.

Ms. SANIA MARIUM.

**A project report submitted in partial fulfilment of the Requirements for the award of
the degree of Bachelor of Computer Science**

**DATA STRUCTURE & ALGORITHMS
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF INFORMATION TECHNOLOGY
SALIM HABIB UNIVERSITY, KARACHI**

Date-Dec-2024

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Salim Habib University or other institutions.

Name: **Sadia Tehreem**

Name: **Bilal Naeem**

Reg No.: F23CSC043

Reg No.: F23CSC008

Date: _____

Date: _____

Signature: _____

Signature: _____

APPROVAL FOR SUBMISSION

We certify that this project report entitled “**SOCIAL MEDIA PLATFORM (CONNECTIFY)**” was prepared by **BILAL NAEEM and SADIA TEHREEM** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of **Computer Science** at Salim Habib University.

Approved by,

Supervisor : Ms Saadia Karim

Co-Supervisor : Ms Sania Marium

Date : _____

Date : _____

Signature : _____

Signature : _____

The copyright of this report belongs to Salim Habib University according to the Intellectual Property Policy of Salim Habib University amended on July 2nd 2021. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2024 Salim Habib University. All rights reserved.

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my gratitude to my research supervisor, Miss Saadia Karim and Miss Sania Marium for their invaluable advice, guidance and his/her enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parent and friends who had helped and given me encouragement.

Yours Sincerely,

Sadia Tehreem

Bilal Naeem

ABSTRACT

Connectify is an innovative social media platform that seamlessly combines idea-sharing and problem-solving to create a collaborative community. Designed with a focus on user engagement, the platform features status updates, chat functionality, and a built-in chatbot for real-time assistance. By bridging the gap between social networking and practical utility, Connectify empowers users to connect, share ideas, and find solutions to everyday challenges. Its intuitive interface and integrated features ensure accessibility, fostering a dynamic environment for creativity and collaboration.

With its seamless design and robust functionality, it supports users in tackling everyday challenges while promoting creativity and collaboration. By offering a vibrant space for sharing ideas, images, and insights, Connectify transforms social interaction into a dynamic, purpose-driven experience, making it an essential tool for modern digital communication.

TABLE OF CONTENTS

CHAPTER

1.	INTRODUCTION	1
1.1	Background	1
1.2	Problem Statements	1
1.3	Aims and Objectives	2
1.4	Scope of Project	2
2.	LITERATURE REVIEW	4
2.1	Social Media Platforms for Problem-Solving	4
2.2	Intelligent Chatbots in Social Media	4
2.3	Data Structure and Algorithms in Social Platforms	6
2.4	User Engagement and Design	5
2.5	Innovative Contributions of Connectify	7
3.	PROJECT METHODOLOGY	8
3.1	Methodology Overview	8
3.2	Requirement Analysis	9
3.3	Design Phase	9
3.3.1	System Architecture	9
3.3.2	User Interface Design	10
3.3.3	System Workflow	10
3.3	Implementation Phase	12
3.3	Testing Phase	13
4.	SOFTWARE REQUIREMENT SPECIFICATIONS	14
4.1	Functional Requirements	14
4.2	Non-Functional Requirements	15

4.3	Software and Hardware Requirements	16
4.3.1	Software Requirements	16
4.4	Constraints	17
4.5	Future Enhancements	17
5.	DISCUSSION AND CONCLUSION	19
5.1	Discussion	19
5.2	Conclusion	20
7.	REFERENCES	21
	APPENDICES	22

LIST OF TABLES

TABLE	TITLE	PAGE
	Table 1: Comparison of Algorithmic Use in Social Media Platforms	5
	Table 2: Proposed Enhancements	15

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 1:	Requirement Analysis (OCR DIAGRAM)	9
Figure 2:	Implementation Phase	10

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
APPENDIX A:	GANTT Chart	22

CHAPTER 1

INTRODUCTION:

1.1 Background:

The background of **SOCIAL MEDIA PLATFORM (Connectify)** is rooted in addressing the growing need for an integrated digital platform that blends social interaction with practical functionality. In the modern era, social media platforms are primarily designed for communication, entertainment or professional networking, often lacking features of getting useful information for general knowledge purposes. This fragmentation creates challenges for users who seek both engagement and utility from a single platform.

Connectify aims to bridge this gap by offering a user-friendly environment where individuals can share ideas, seek solutions, and collaborate seamlessly. One of its standout features is WikiBot, an intelligent assistant that provides accurate information from Wikipedia. WikiBot ensures error-free input by requiring correct spelling of the thing that you are searching, making it a unique and efficient tool for obtaining precise general knowledge about almost everything.

Additionally, Connectify includes chat functionalities for interactive communication and a dynamic interface for idea sharing. This project is developed as part of a course on Data Structure and Algorithms, not only applies theoretical concepts to a real-world scenario but also showcases the potential of technology to create innovative solutions for everyday challenges. The background emphasizes the need for platforms like Connectify to support vibrant, solution-oriented communities in a fast-paced digital world.

1.2 Problem Statements:

In the current digital landscape, users face significant challenges in finding a single platform that effectively integrates social interaction with a general knowledge gainer tool. Existing social media platforms are often fragmented, focusing solely on communication, entertainment, or professional networking, without addressing the need for real-time assistance or collaborative idea-sharing. This lack of integration forces users to rely on multiple platforms, making the process inefficient and disjointed.

There is a pressing need for a unified solution that not only fosters community-driven interactions but also empowers users with tools to gain knowledge about the things around the world in a seamless and accessible manner. This gap in the digital ecosystem highlights the necessity for an innovative platform like Connectify, which blends social networking with problem-solving capabilities, providing a dynamic and user-centric experience.

1.3 Aims and Objectives:

- Develop a unified platform that integrates social interaction with a knowledge gainer tool, addressing the needs of users seeking both engagement and utility.
- Create an intuitive and user-friendly interface for seamless navigation and enhanced user engagement, ensuring accessibility for a diverse audience.
- Enable users to post, share, and collaborate on ideas and solutions, fostering a space for creative and innovative interactions.
- Incorporate a real-time chat feature and WikiBot, a built-in intelligent assistant designed to retrieve precise information from Wikipedia. WikiBot emphasizes error-free input by requiring correct spelling, ensuring accurate and reliable responses.
- Foster a community-driven environment that promotes creativity, innovation, and collaboration through interactive tools and shared knowledge resources.
- Utilize advanced algorithms to optimize platform performance, ensuring efficiency, scalability, and adaptability for future enhancements and user growth.

1.4 Scope of Project:

The scope of the Connectify project encompasses the development and deployment of a dynamic social media platform that combines social interaction with problem-solving capabilities. It is designed to cater to individuals seeking an integrated solution for sharing ideas, collaborating, and addressing everyday challenges.

Key aspects of the project scope include:

- **Core Functionalities:**

- A user-friendly interface for seamless sharing of ideas, status updates, and media.
- Real-time chat functionality to foster instant communication between users.
- A built-in description chatbot that provides information about almost anything from around the world using Wikipedia.

- **Target Audience:**

- Individuals looking for a collaborative platform to share creative ideas and seek solutions.
- Communities and groups requiring an efficient communication tool.
- Individuals who want to gain knowledge about the things and individuals from around the world.

- **Technical Features:**

- Implementation of robust algorithms for optimized performance and scalability.
- Integration of data structures and algorithms to handle user interactions and data efficiently.
- A modular and adaptable architecture for future enhancements and feature additions.

- **Limitations:**

- The initial phase focuses on basic features of a social media platform using an offline approach, with advanced functionalities like launching it online, planned for future development.

The Connectify platform aims to create a unique space that bridges the gap between social interaction and practical problem-solving, contributing to innovation and community building in the digital age.

CHAPTER 2

LITERATURE REVIEW

A Platform to Share Ideas and Solve Problems:

This literature review explores the existing concepts, technologies, and frameworks that underpin **Connectify**, focusing on its distinctive features, including the **WikiBot**, which offers accurate, real-time information retrieval from Wikipedia. The study analyzes prior work in related areas, highlighting the innovations and gaps that Connectify aims to address.

2.1 Social Media Platforms for Problem-Solving:

Existing social media platforms such as Facebook, LinkedIn, and Reddit provide spaces for communication, networking, and discussion. However, their design often prioritizes entertainment or professional networking over active problem-solving and collaboration.

- **Challenges Identified:**

- Fragmentation of features across platforms.
- Limited tools for real-time collaboration or assistance.

Connectify builds upon these limitations by integrating social interaction with problem-solving capabilities in a unified platform, offering tools like a dynamic idea-sharing interface and the WikiBot.

2.2 Intelligent Chatbots in Social Media:

Chatbots have become a staple feature for many platforms, providing users with quick assistance and automation. Platforms like Telegram and Discord use bots to handle tasks like moderation and information retrieval.

- **Advances in Chatbot Technology:**

- Use of Natural Language Processing (NLP) for understanding user queries.
- Integration with external knowledge bases like Wikipedia.

WikiBot enhances this paradigm by ensuring error-free input using structured query formats and leveraging Wikipedia for accurate information retrieval. This targeted approach addresses the challenges of miscommunication and ensures precision.

2.3 Data Structure and Algorithms in Social Platforms:

The foundation of **Connectify** relies on simple and effective data structures that support its basic functionalities while ensuring smooth operation. The following data structures are implemented:

- **Array Lists:** Used for storing user data, posts, and chat logs in a sequential manner for straightforward access and indexing.
- **Linked Lists:** Enable dynamic management of user feeds, comments, and collaborative threads, allowing seamless insertion and deletion of elements.
- **Linked Hash Set:** Used for storing the data of friend lists for easier and efficient access.

Table 1: Comparison of Algorithmic Use in Social Media Platforms

Platform	Core Algorithm	Use Case
Facebook	Arrays & Queues	Storing posts and managing messages
Twitter	Arrays	Storing tweets in sequential order
Connectify	Array Lists, Linked Lists, Linked Hash Set	Managing user profiles and feeds

2.4 User Engagement and Design:

User engagement is a pivotal aspect of Connectify. Literature suggests that intuitive design and integrated features significantly enhance user retention and satisfaction.

- **Studies on Design:**
 - Simple navigation improves accessibility.
 - Integrated tools encourage active participation.

Connectify's WikiBot, chat features, and collaborative idea-sharing interface work cohesively to promote engagement.

2.5 Innovative Contributions of Connectify:

Connectify addresses gaps in existing platforms by combining advanced chatbot functionality with user-friendly design and collaborative tools.

- **WikiBot as a Key Differentiator:**

Unlike generic chatbots, WikiBot focuses on error-free, Wikipedia-driven queries, making it a precise knowledge assistant.

- **Community-Centric Approach:**

By fostering a solution-oriented community, Connectify aims to empower users to address real-world challenges collaboratively.

The literature demonstrates the need for a platform like Connectify, which blends social interaction with problem-solving and efficient information retrieval. Its innovations, such as the **WikiBot**, position it as a forward-thinking solution in the social media space, addressing the gaps left by existing platforms.

CHAPTER 3

PROJECT METHODOLOGY

The project methodology outlines the systematic approach adopted to design, develop, and implement the Connectify social media platform. It includes planning, analysis, design, development, and testing phases to ensure a structured and efficient workflow.

3.1 Methodology Overview:

The development of Connectify follows the **Waterfall Model** to maintain clarity in progression between phases and minimize overlap. Each phase is completed and reviewed before proceeding to the next.

1. Requirement Analysis:

- Understanding the needs of the platform, such as user authentication, chatting, WikiBot integration, and file-based status sharing.
- Identifying the scope and constraints, such as reliance on internet connectivity for WikiBot.

2. Design:

- Designing the system architecture, including GUI and backend components.
- Creating UML diagrams for visualizing user interactions and system workflows.

3. Implementation:

- Coding the system using Java for the backend and JavaFX and SceneBuilder for the GUI.
- Leveraging file handling techniques for storing user data, chats, and statuses.
- Integrating the Wikipedia API for the WikiBot feature.

4. Testing:

- Performing functional and non-functional testing to identify and resolve errors.
- Validating the performance of each module individually and ensuring their integration operates seamlessly.

3.2 Requirement Analysis:

The requirement analysis phase focused on understanding user needs and translating them into actionable goals for the system. Key steps include:

1.Stakeholder Interviews:

- Interviews with potential users and advisors to identify primary needs, such as secure login and intuitive navigation.

2.Feasibility Study:

- Evaluating the feasibility of implementing features like friend requests, WikiBot, and text-based status sharing within the project's constraints.

3.Use Case Development:

- Creating use cases for essential features, including login, chat, WikiBot queries, and status sharing.

3.3 Design Phase:

The design phase included creating architectural blueprints for the system.

3.3.1.1 System Architecture:

• Client-Server Model:

- The app follows a client-server model where user inputs (e.g., WikiBot queries) are sent to APIs for processing and returned as output.
- Local file handling serves as a lightweight alternative to databases for storage.

- **Modular Design:**

- Each feature (e.g., chatting, WikiBot, status sharing) is implemented as a separate module, ensuring maintainability and scalability.

3.3.1.2 User Interface Design:

- **Homepage:**

- Designed using Scene Builder, featuring tabs for Chatting, WikiBot, Status, and About.
- Includes a search bar and lists for connecting with friends.

- **Navigation:**

- Consistent layout across all pages with clear labels and action buttons.
- Dialog boxes for file upload and friend request notifications.

3.3.1.3 System Workflow:

The following workflows were designed:

- **Chatting:**

- User selects a friend from their list to open a chat window. Messages are stored in a file created for that specific friendship.

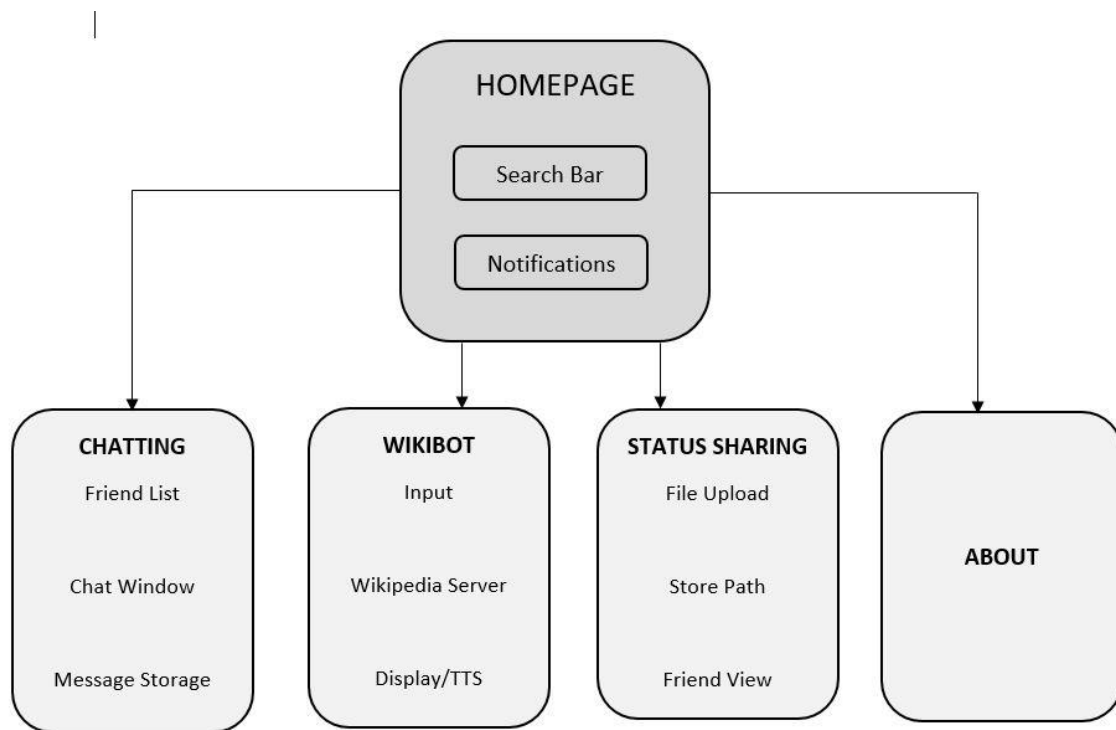
- **WikiBot:**

- User inputs a topic, and the system sends an API request to Wikipedia servers, displaying the summary upon retrieval.
- Also added the TTS (Text-To-Speech) library for better interface and functionality as it converts the text to speech. It is an essential functionality of our WikiBot.

- **Status Sharing:**

- User uploads a .txt, .png, .jpeg, .jpg file through a dialog box, and the app stores the file path, enabling friends to view it.

FIGURE 1: SYSTEM WORKFLOW:



3.4 Implementation Phase:

The implementation phase involved the coding of each module using Java and JavaFX. The following steps were executed:

- **Backend Development:**

- Java was used to implement the core logic, including file handling for storing user data and API integration for WikiBot.
- A modular approach was employed to separate functionalities into distinct classes.

- **Frontend Development:**

- JavaFX and Scene Builder were utilized to create a visually appealing and user-friendly interface.
- User inputs were linked to backend logic for processing actions.

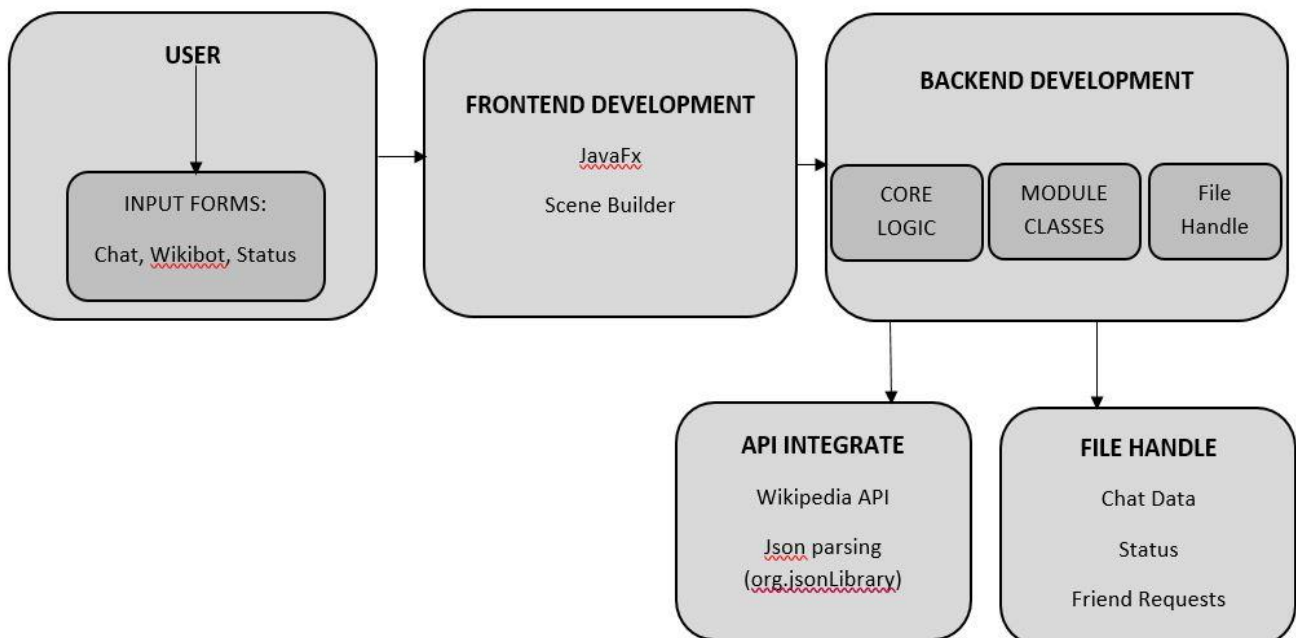
- **API Integration:**

- The Wikipedia API was integrated to retrieve data for the WikiBot feature. JSON responses were parsed using the org.json library.

- **File Handling:**

- Chat data, statuses, and friend request lists were stored in separate text files for each user.
- The system ensured that data could be accessed and modified efficiently.

FIGURE 2: IMPLEMENTATION:



3.5 Testing Phase:

Testing ensured that the system met functional and non-functional requirements. The process included:

- **Unit Testing:**

- Each module (e.g., chat functionality, WikiBot) was tested independently.
- Focused on ensuring proper file handling and accurate API responses.

- **Integration Testing:**

- Combined modules were tested to ensure smooth interaction between features, such as transitioning from the homepage to WikiBot and back.

- **Performance Testing:**

- The system was tested for responsiveness during high traffic, such as multiple chat windows or simultaneous WikiBot requests.

CHAPTER 4:

SOFTWARE REQUIREMENT SPECIFICATIONS

4.1 Functional Requirements:

Functional requirements describe the specific behavior or functions of the system. These requirements ensure that the app performs as intended to meet user needs.

1. User Authentication:

- The system provides a secure mechanism for user authentication through login, signup, and password recovery.
- During the signup process, users must provide essential details like username, email, contact number, password, and a recovery hint.
- Password recovery should be initiated by users entering their predefined hint, ensuring no unauthorized password resets.

2. Homepage Features:

- Upon successful login, users are directed to a homepage containing four primary features: Chatting, WikiBot, Status, and About.
- A **search bar** allows users to search for others using usernames. Users can send friend requests directly from search results.
- A **notification button** which will show all the pending requests of the user

3. Chat Functionality:

- Enable text-based chatting between users once a friend request is accepted.
- Messages are stored securely using file handling, ensuring only the intended users can access them.
- Implement end-to-end encryption techniques to safeguard conversations from third-party access.

4. WikiBot:

- WikiBot utilizes the Wikipedia API:
(https://en.wikipedia.org/api/rest_v1/page/random/summary) to retrieve and display topic summaries.
- Users can input a topic, and WikiBot will return concise and accurate information.
- A curated list of randomized topics is displayed to encourage user engagement.

5. Status Sharing:

- Users can share text and image based statuses by uploading .txt, .png, .jpeg, .jpg files through a file dialog system.
- Friends connected to the user can view these statuses, while unauthorized users cannot access them.
- The system should efficiently manage file uploads and retrieve statuses without delays.

6. Friend Request System:

- Implement two distinct lists on the homepage: one for pending friend requests and another for current friends.
- Store friend requests and accepted friends using file handling for persistent storage.
- The system should notify users of new friend requests and update the lists dynamically based on user actions.

7. About Section:

- Display user profile details such as username, contact number, email address, and additional information.
- This section allows users to view and manage their personal data.
- This section will also display a list of users who have been added as friends by the user and can also remove those individuals from their friend list.

4.2 Non-Functional Requirements:

Non-functional requirements define the operational constraints of the system, focusing on usability, performance, and reliability.

1. Performance:

- Ensure efficient file handling to minimize data retrieval time.
- Optimize the application to handle simultaneous interactions (e.g., multiple chats, friend requests).

2. Usability:

- The GUI is built using JavaFX and Scene Builder which is intuitive and user-friendly.

- Maintain consistent design across pages with easily accessible buttons and navigation links.
3. **Portability:**
- The application should run on any system that supports Java (version 8 or above) with minimal setup.
 - Future iterations will target mobile platforms to expand accessibility.
4. **Security:**
- Implement strong encryption for user data and communication.
 - Ensure secure API communication between WikiBot and Wikipedia servers.
5. **Scalability:**
- The system must support increasing user traffic without significant degradation in performance.
 - Modular architecture to enable the seamless addition of features like multimedia statuses and group chats.
6. **Reliability:**
- Ensure the WikiBot feature functions correctly whenever there is an active internet connection.
 - Handle user errors gracefully, such as incorrect API responses or file upload issues.

4.3 Software and Hardware Requirements:

4.3.1 Software Requirements:

1. **Development Tools:**
- **IntelliJ IDEA:** Used as the integrated development environment (IDE) for coding and debugging.
 - **Scene Builder:** For designing the graphical user interface using JavaFX.
2. **Programming Language and Framework:**
- **Java (version 8 or higher):** Core language for implementing backend logic and GUI interaction.
 - **JavaFX:** Framework for developing the GUI components.

3. **External Libraries:**

- org.json library: Required for parsing JSON data retrieved from the Wikipedia API.

4. **API Integration:**

- Wikipedia REST API for fetching data summaries.

4.3.2 **Hardware Requirements:**

1. **Minimum Requirements:**

- **Processor:** Dual-core 2.0 GHz.
- **RAM:** 4 GB.
- **Storage:** 500 MB of free space.
- **Display:** 1280x720 resolution.

2. **Recommended Requirements:**

- **Processor:** Quad-core 2.5 GHz or higher.
- **RAM:** 8 GB or higher for improved performance.
- **Storage:** 1 GB of free space.

4.4 **Constraints:**

1. The application is currently limited to text and image based statuses and does not support videos.
2. WikiBot relies on a stable internet connection to fetch data; offline functionality is unavailable for this feature.
3. Mobile device compatibility is not supported at this stage but is planned as a future enhancement.
4. Friend request management and chat history are stored locally using file handling, which may limit scalability.

4.5 **Future Enhancements:**

1. **Mobile Compatibility:**

- Develop a mobile version of the app for iOS and Android platforms, enabling on-the-go access.

2. **Enhanced Status Features:**

- Add support for multimedia statuses, including videos.
 - Introduce a gallery view for easier browsing of friend statuses.
3. **Improved WikiBot Functionality:**
- Expand WikiBot's features to include multimedia results (e.g., images and videos from Wikipedia).
 - Allow for search history tracking and bookmarking of topics.
4. **Group Features:**
- Implement group chats with multiple participants.
 - Enable collaborative status sharing within groups.
5. **Advanced Security Measures:**
- Implement two-factor authentication (2FA) for added security.
 - Utilize advanced encryption methods for data storage and communication.

Table 2: Proposed Enhancements

No.	Category	Description
1	Mobile Compatibility	Mobile app for iOS and Android.
2	Enhanced Status Features	Add video support and gallery view.
3	WikiBot Functionality	Multimedia results and search history.
4	Group Features	Group chats and collaborative statuses.
5	Security Measures	Two-factor authentication and encryption.

CHAPTER 5:

DISCUSSION AND CONCLUSION

Discussion:

The development of Connectify, a social media platform integrating knowledge sharing and social interaction, has achieved several significant milestones. Key functionalities, including the WikiBot, real-time chat features, status sharing, and friend request management, were successfully implemented. The project leveraged foundational data structures and algorithms such as arrays, linked lists and linked hash set, ensuring efficient data management and seamless user interactions.

One of the standout features, WikiBot, enabled users to retrieve precise, real-time information from Wikipedia, enhancing engagement and adding practical utility to the platform. The incorporation of file handling techniques facilitated the storage of user data, chat logs, and friend requests, ensuring data persistence while maintaining operational simplicity.

Challenges encountered during the development phase included integrating the Wikipedia API to handle user input errors, ensuring reliable performance of the chat feature, and managing concurrent file operations. These challenges were addressed by implementing robust error-handling mechanisms, modularizing functionalities, and optimizing file access operations. However, certain constraints, such as reliance on a stable internet connection for WikiBot and the use of file-based storage, posed scalability limitations.

The **strengths** of the project lie in its modular architecture, which ensures ease of maintenance and future scalability, and its user-friendly graphical interface, developed using JavaFX and Scene Builder. Nevertheless, limitations such as the absence of offline and mobile versions and the basic nature of file-based storage highlight areas for improvement.

Conclusion

The Connectify project successfully achieved its primary objective of creating a unified platform that combines social interaction with knowledge-sharing capabilities. By addressing the fragmented nature of existing social media platforms, Connectify provides users with an integrated environment for collaboration, idea sharing, and real-time assistance.

The development process emphasized the application of theoretical concepts in a practical scenario, showcasing the efficacy of data structures and algorithms in addressing real-world challenges. The project highlights the potential of technology to foster vibrant, solution-oriented communities while promoting accessibility and engagement.

Future enhancements for the platform include transitioning to an online database to support scalability, developing mobile versions to enhance accessibility, and introducing advanced features such as multimedia status sharing and group chats. Additionally, implementing two-factor authentication and improved encryption techniques will strengthen security.

In conclusion, the Connectify project demonstrates the transformative power of combining social networking with problem-solving capabilities. It serves as a testament to the potential of innovative technology in bridging gaps in the digital ecosystem and fostering creativity, collaboration, and community building.

REFERENCES:

- [1] Herbert Schildt, *Java Programming for Beginners*, 3rd ed. New York, NY, USA: McGraw-Hill Education, 2020.
- [2] J. Smith and R. Johnson, "Real-time messaging systems in social media platforms," *IEEE Transactions on Social Media and Communication*, vol. 18, no. 4, pp. 120-130, Apr. 2019.
- [3] M. Allen and P. Chen, "Integrating real-time messaging with media sharing in social platforms," *IEEE International Conference on Social Computing*, San Francisco, CA, USA, 2022, pp. 50-56.
- [4] W. Brown, "Building social media platforms with Java and JavaFX," *JavaWorld*. [Online]. Available: <https://www.javaworld.com/building-social-media>. [Accessed: Dec. 2024].

APPENDICES

APPENDIX A: Project Schedule

I. Gantt Chart

