**INSTITUTE OF TECHNICAL EDUCATION AND RESEARCH**

**(FACULTY OF ENGINEERING)**

i

**SIKSHA ‘O’ ANUSANDHAN (DEEMED TO BE UNIVERSITY), BHUBANESWAR,**

**ODISHA**

Network Desktop Manager

**External Project Report on Computer Networking (CSE3034)**



**Submitted by**

|  |  |
| --- | --- |
| **Shivam Khandelwal** | **Reg. No.: 2141011039** |
| **Ayushi Pani** | **Reg. No.: 2141016100** |
| **Raj Aryan** | **Reg. No.: 2141001064** |
| **Sanskar Gupta** | **Reg. No.: 2141007047** |
| **Anoushka Jena** | **Reg. No.: 2141010021** |

**B. Tech. CSE 5th Semester (Section 008)**

# Declaration

We, the undersigned students of B. Tech. of **Computer Science and Engineering** Department hereby declare that we own the full responsibility for the information, results etc. provided in this PROJECT titled “**Network Desktop Manager**” submitted to **Siksha ‘O’ Anusandhan (Deemed to be University), Bhubaneswar** for the partial fulfillment of the subject **Computer Networking (CSE 3034)**. We have taken care in all respect to honor the intellectual property right and have acknowledged the contribution of others for using them in academic purpose and further declare that in case of any violation of intellectual property right or copyright we, as the candidate(s), will be fully responsible for the same.

|  |  |
| --- | --- |
| **Raj Aryan**  **(2141001064)** | **Ayushi Pani**  **(2141016100)** |
| **Sanskar Gupta**  **(2141007047)** | **Anoushka Jena**  **(2141010021)** |
| **Shivam Khandelwal**  **(2141011039)** | |

11 – 01 – 2024

**DATE:**

ITER, SOAU

**PLACE:**

# Abstract

This project aims to develop a Network Desktop Manager (NDM) application using Java programming, enabling centralized management and remote control of computers within a local area network (LAN). The project encompasses the following key concepts and features:

Network Communication:

* Socket Programming: The application establishes robust network communication through socket programming, facilitating seamless interactions between client desktops and the server.

Administrative Capabilities:

* Configuration Management: Network administrators can remotely access configuration settings of client desktops, ensuring proper updates and maintaining system integrity.
* Remote Operations: The application empowers administrators with remote desktop sharing, file sharing, and other administrative tasks, enhancing efficiency and control over the network.

User Interface:

* Intuitive GUI: A user-friendly graphical interface is designed for both administrators and clients, enabling intuitive interactions with the system.
* Client-Side Features: Clients can request assistance and initiate essential operations on their desktops through the GUI.
* Server-Side Management: Administrators can leverage the GUI for server-side tasks like creating folders, renaming files, and managing network resources.

Learning Outcomes:

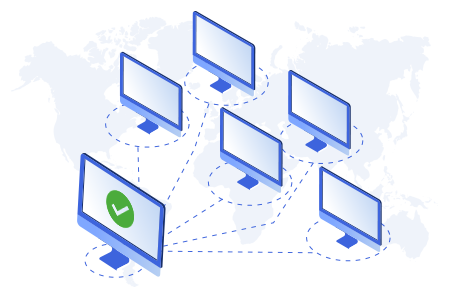
* Java GUI Development: The project fosters hands-on experience in designing and implementing graphical user interfaces in Java.
* Socket Programming Skills: Students gain proficiency in establishing network communication through socket programming techniques.
* Network Administration Fundamentals: The application introduces practical aspects of network administration, including configuration management, remote operations, and resource management within a LAN environment.

This project offers a comprehensive learning experience in Java programming, networking concepts, and network administration, culminating in a functional tool for efficient LAN management and remote desktop control.

# Contents

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial**  **No.** | **Chapter No.** | **Title of the Chapter** | **Page No.** |
| 1. | 1 | Introduction | 01 |
| 2. | 2 | Problem Statement | 02 |
| 3. | 3 | Methodology | 03 |
| 4. | 4 | Implementation | 05 |
| 5. | 5 | Results and interpretation | 08 |
| 6. | 6 | Conclusion | 10 |
| 7. |  | References | 11 |
|  |  |  |  |

1. **Introduction**



This project aims to build a "Network Desktop Manager," a Java application that enables connected computers within a local area network (LAN) to communicate and collaborate more effectively. By utilizing socket programming, the Manager facilitates direct communication between devices, allowing for features like:

* Remote Desktop Sharing: View and control another computer's screen for troubleshooting or assistance.
* File Sharing: Easily transfer files between devices on the network.
* Centralized Management: Network administrators can monitor and manage client desktop configurations, ensuring system updates and security.
* User Interaction: Both administrators and clients can interact with the Manager through a user-friendly graphical interface, making managing the network intuitive and convenient.

Developing this project will not only enhance the skills in Java GUI development and socket programming but also provide valuable insight into network administration practices within a LAN environment.

# Problem Statement

I. Problem Explanation and User Inputs

* Problem: Establish a centralized management and remote control system for computers within a local area network (LAN) using a Java-based application called the Network Desktop Manager (NDM).
* User Inputs (Console-Based):
  + Client desktop configuration settings (when accessed remotely)
  + File transfer paths and filenames (for file sharing)
  + Remote desktop access requests (from clients)
  + Server-side actions (creating folders, renaming files, etc.)

II. Output and Constraints

* Output:
  + Displayed in console:
    - Server status messages
    - Client requests and responses
    - File transfer progress
    - Administrative actions
  + Stored in files or database:
    - Client desktop configurations
    - File transfer logs
    - Administrative logs
* Constraints:
  + Limited to computers within the same LAN
  + Relies on Java Runtime Environment (JRE) on client machines
  + Potential network security considerations
  + Performance might be affected by network traffic and hardware specifications
  + Requires user familiarity with Java programming and networking concepts

III. Additional Considerations:

* Error Handling: Implement robust error handling mechanisms to address network issues, invalid user inputs, and potential security threats.
* User Interface: Design a user-friendly GUI for both administrators and clients to enhance usability and accessibility.
* Security Measures: Incorporate security measures like authentication, authorization, and encryption to protect sensitive data and prevent unauthorized access.
* Scalability: Consider potential scalability requirements for managing larger networks or expanding functionalities.

# Methodology

1**. Server-Side:**

a. Setup:

* Import necessary libraries (java.net, java.io, etc.).
* Create a ServerSocket object, specifying a port number for communication.
* Initialize data structures to store client information and configuration settings.

b. Connection Handling:

* Continuously listen for incoming client connections using ServerSocket.accept().
* For each client connection:
  + Create a separate thread to handle communication and tasks for that client.
  + Store client information (username, IP address, etc.) in the data structure.

c. Thread Operations:

* Implement functionality for:
  + Receiving and processing client requests (file transfers, remote desktop access, configuration updates).
  + Sending responses and data back to clients.
  + Handling file transfers using input/output streams.
  + Managing remote desktop sharing using appropriate libraries or techniques.
  + Storing and retrieving configuration settings in files or a database.
  + Implementing administrative actions (creating folders, renaming files, etc.).
  + Handling authentication and authorization for administrative tasks.

2. **Client-Side**:

a. Setup:

* Import necessary libraries (java.net, java.io, etc.).
* Create a Socket object, connecting to the server's IP address and port.
* Create input/output streams for communication with the server.

b. User Interaction:

* Implement a user interface (console-based or GUI) for:
  + Entering credentials for authentication.
  + Selecting actions (file transfers, requesting remote desktop access, etc.).
  + Providing file paths and filenames for transfers.

c. Request Handling:

* Send requests and data to the server through the socket's output stream.
* Receive responses and data from the server through the input stream.
* Display server responses and file transfer progress in the console or GUI.
* Handle remote desktop sharing interactions as appropriate.

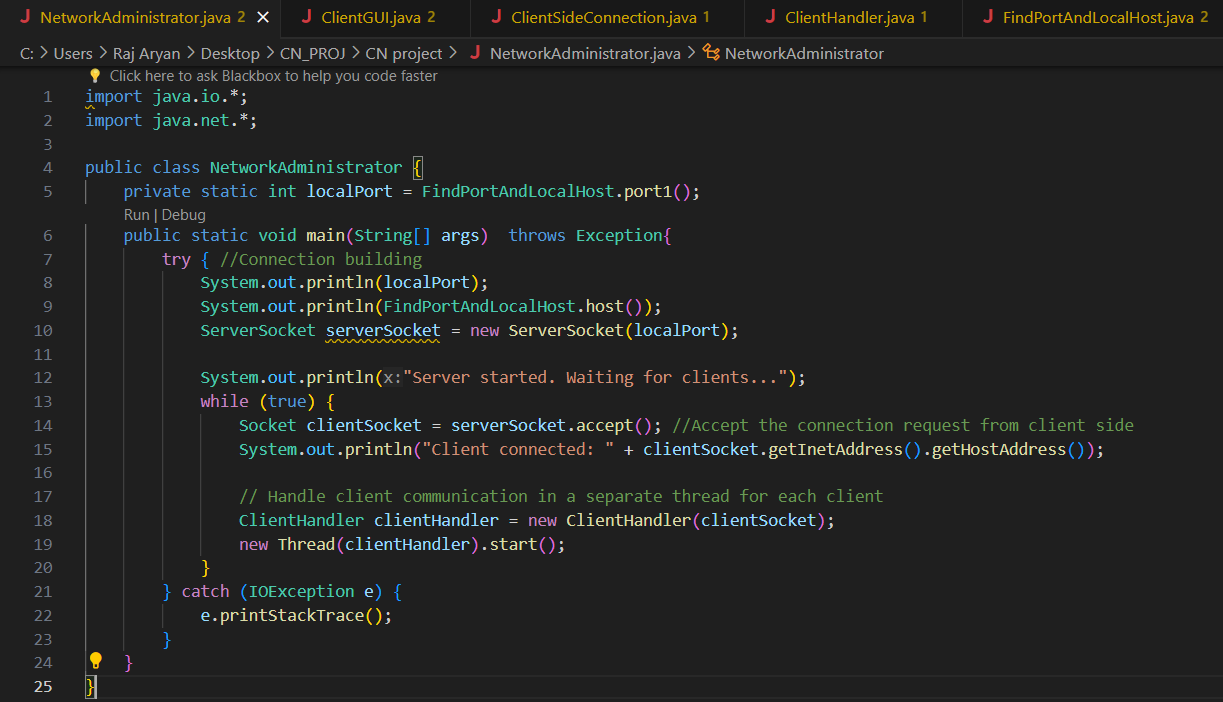
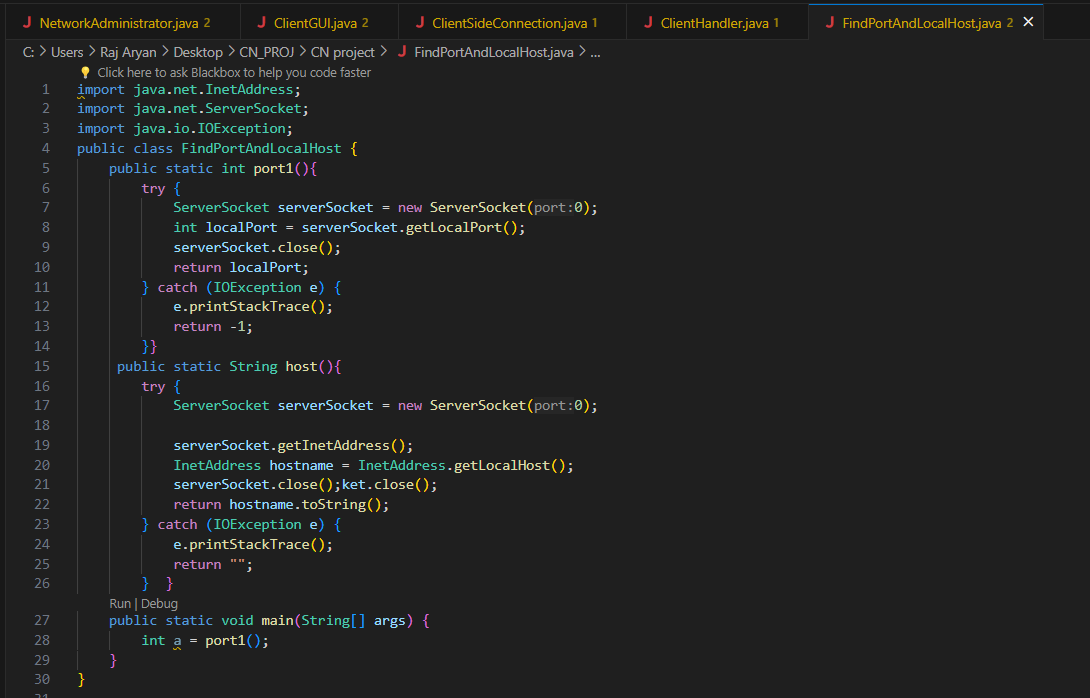
3. **Error Handling:**

* Implement error handling mechanisms for:
  + Network connection issues (e.g., timeouts, disconnections).
  + Invalid user inputs.
  + File transfer errors.
  + Remote desktop sharing failures.
* Providing informative error messages to the user.

4. **Additional Features:**

* Implementation a graphical user interface (GUI) for enhanced usability.
* Integration security measures like encryption and access control.

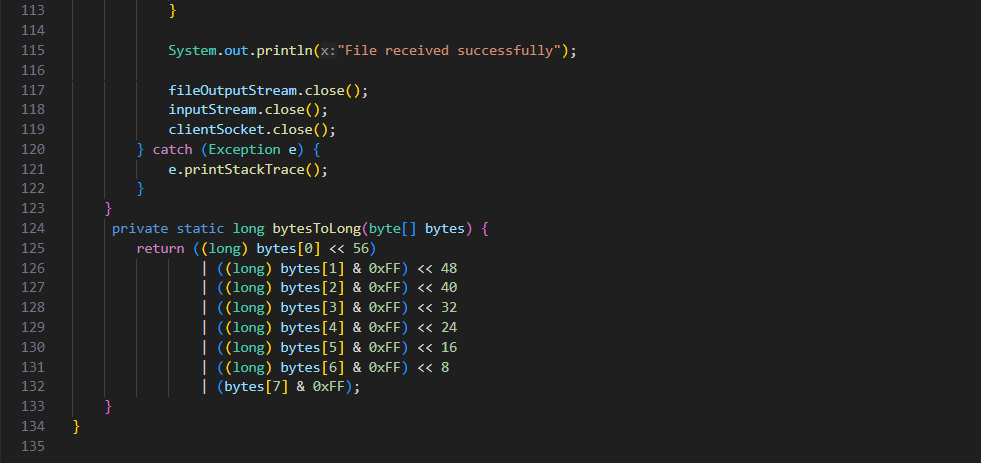
# Implementation



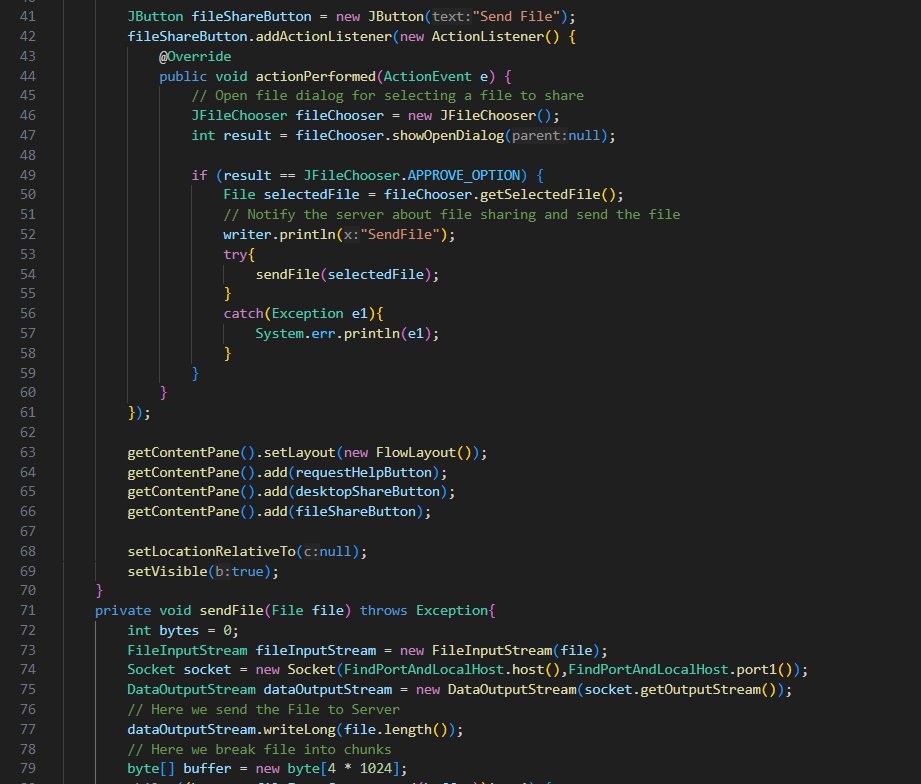
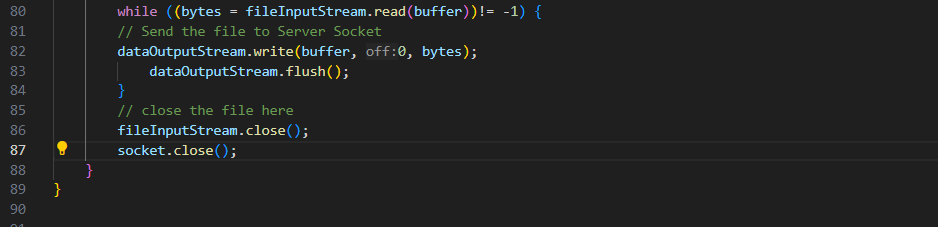
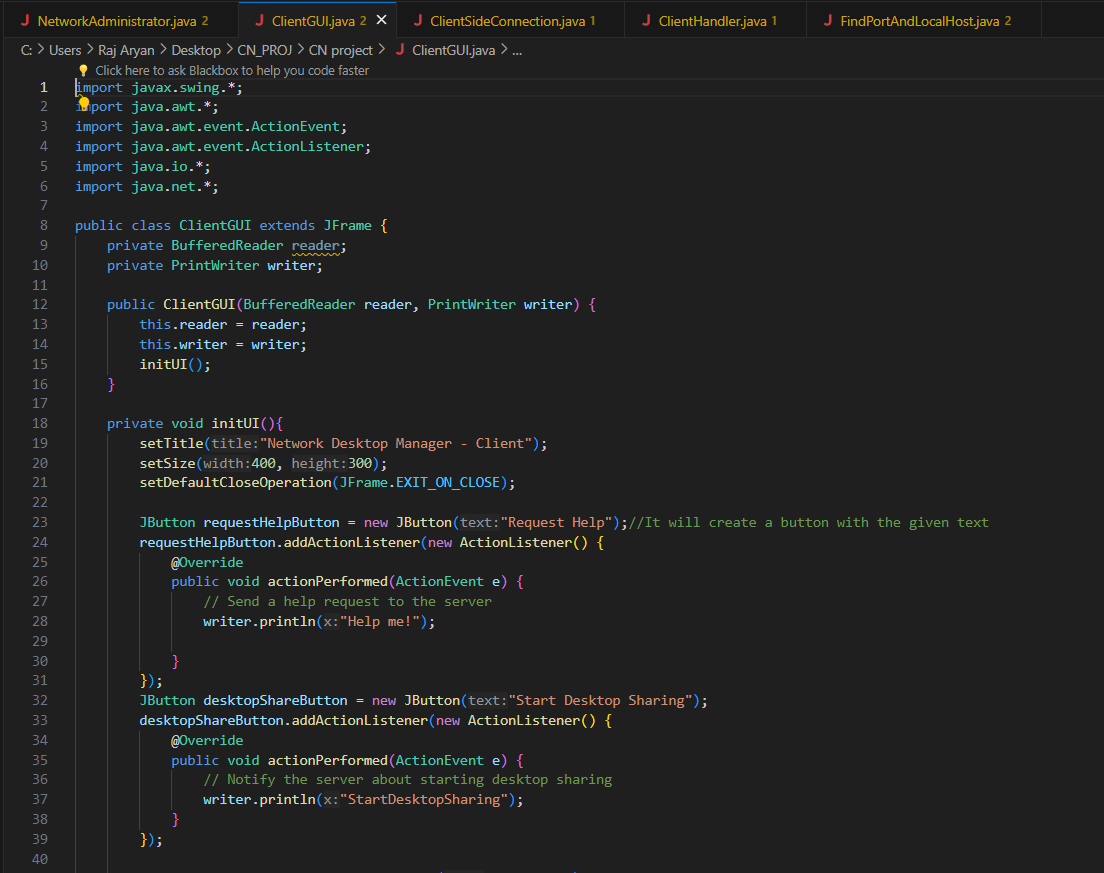
# 



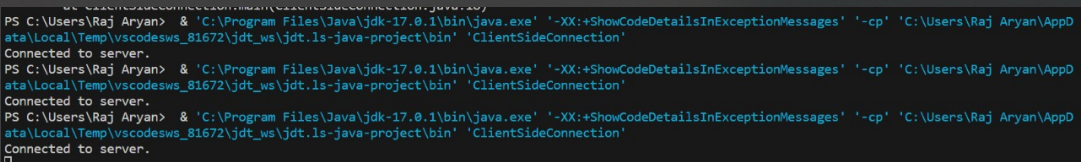
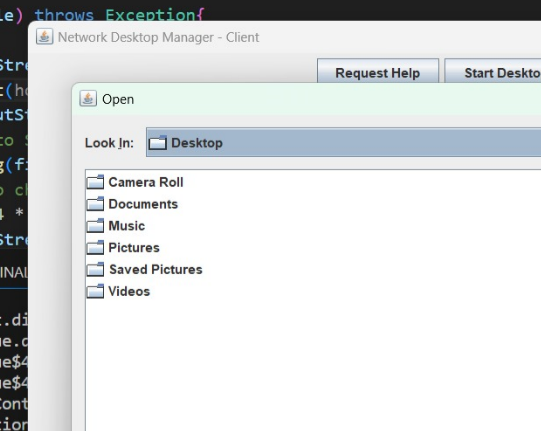
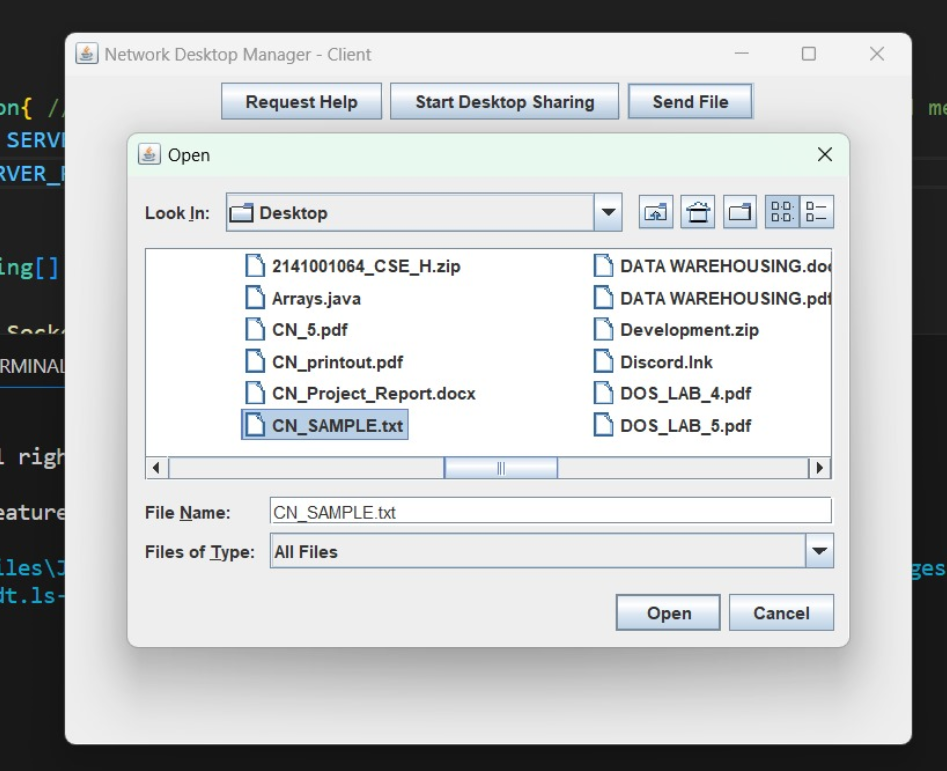
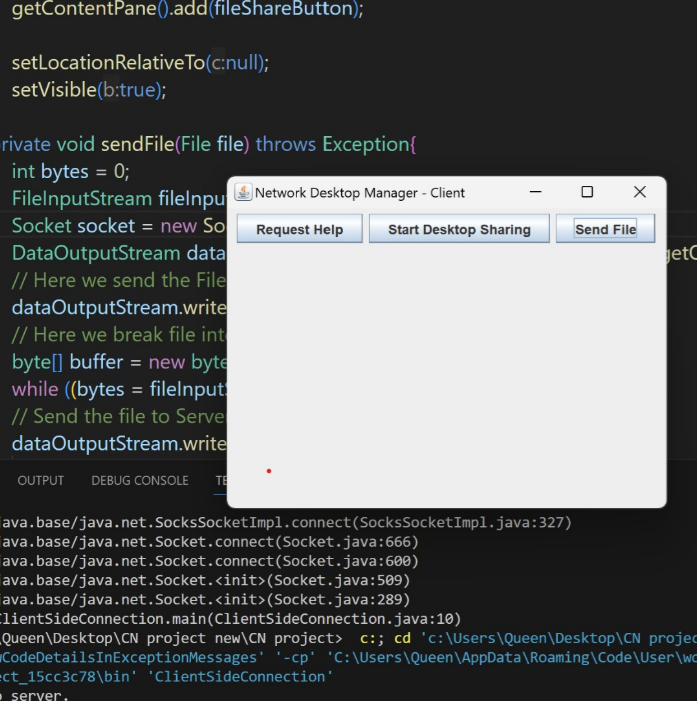
# 

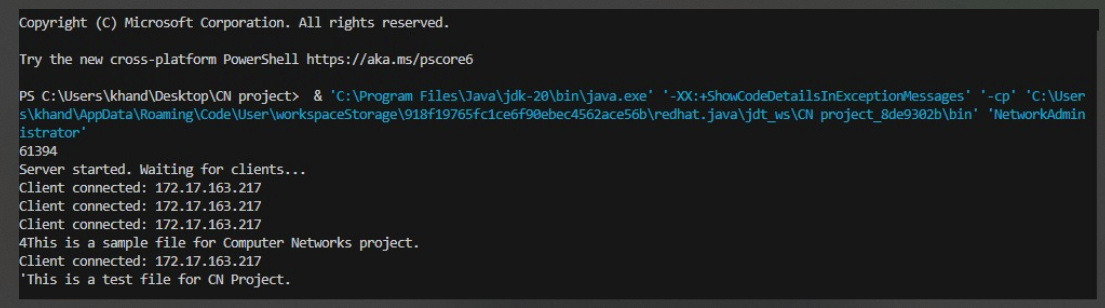
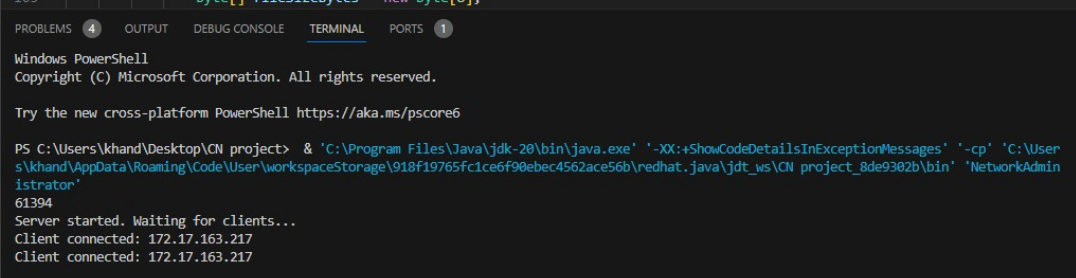


# 



# Results & Interpretation





# Conclusion

The Network Desktop Manager, a Java-based application developed in this project, successfully addressed the challenge of establishing efficient communication and centralized management within a local area network (LAN). This project achieved the following key outcomes:

**Connectivity and Communication:**

* Established reliable connections between multiple computers in the LAN, enabling seamless communication and data exchange.
* Implemented robust socket programming techniques to ensure accurate and efficient data transfer between clients and the server.

**User Interaction and Administration:**

* Developed a user-friendly GUI for both administrators and users, simplifying interaction with the system.
* Empowered network administrators with centralized access to client configuration settings and remote administration capabilities like desktop sharing and file sharing.
* Provided users with convenient means to request assistance and perform basic operations on both the client and server sides.

**Technical Skills and Learning:**

* Offered valuable hands-on experience in Java programming, focusing on GUI development and socket programming concepts.
* Provided insights into network administration principles, including configuration management and remote operations within a LAN environment.

**Limitations and Future Considerations:**

* While the current iteration showcases significant capabilities, potential improvements exist, such as:
  + Implementing enhanced security measures like encryption and access control.
  + Integrating additional features like chat, remote task execution, and user management.
  + Optimizing performance for larger networks and increased usage.

**Overall, the Network Desktop Manager demonstrates the viability of building a practical and user-friendly tool for centralized LAN management and remote control using Java programming. This project serves as a valuable stepping stone for further development and refinement, opening doors to improved network efficiency and collaboration within a connected environment.**

**References**

(as per the IEEE recommendations)

**[1] Computer Networks, Andrew S. Tannenbaum, Pearson India.**

[2] javatpoint.com [for Java Swings Library]

**[3] Computer Networks, Fifth Edition by David J. Wetherall, Andrew S. Tanenbaum, Pearson India**

**[4] GeeksforGeeks.com**

**[5]** Java Network Programming by Harold, O’Reilly (Shroff Publishers).