Implementation of File Transfer Application using Sockets

1st Abburi Sai Karthik
Department Of Computer Science and Engineering
Amrita Vishwa Vidyapeetham
*rejneethvr@am.amrita.edu

2nd Raja Pavan Karthik
Department Of Computer Science and Engineering
Amrita Vishwa Vidyapeetham
*rejneethvr@am.amrita.edu

3rd Sista Sai Subhramanya Mrinal
Department Of Computer Science and Engineering
Amrita Vishwa Vidyapeetham
*rejneethvr@am.amrita.edu

Abstract — **The target of this project is to transfer files by using sockets. Here we are using TCP built using Java Socket Programming. It helps the client to upload and download files to the server. File transfer involves copying or moving a file from one PC, then onto the next over a network or Internet connection. The basic idea is to make a server that listens on a specific port. This server will be responsible for receiving or sending files. On the other hand, the client will try to interface with the server and send a file of any type**.

I. INTRODUCTION

This is the most basic file transfer program that we can do using a client-server architecture. Here, we are going to build a client and a server program file, where the client read the data from a text file and sends it to the server. Here, to transfer the file from one machine to another, we need a connection between them. To connect to another machine, we need a socket connection. A socket connection means the two machines have information about each other's network location (IP Address) and TCP port. TCP full form Transmission Control Protocol, which is a highly efficient protocol. It is designed for end-to-end data transmission over an unreliable network. A TCP connection uses a three-way handshake to connect the client and the server. To communicate over a socket connection, streams are used to both input and output the data. Stream socket is connection-oriented and uses TCP/IP, and it ensures that data sent is error-free and complete in the same order A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application to which data is destined to be sent. An endpoint is a combination of an IP address and a port number. Just a file descriptor, and can perform the read/write over this file descriptor. It creates an endpoint to receive and send the information over the network. The other endpoint, this end, is called the client, which opens a file descriptor and connects to the earlier socket to read and write (or send and receive).

II.BACKGROUND STUDY

A. File transfer

File Transfer is the transmission of a computer file through a communication channel from one computer system to another. File transfer is the process of copying or moving a file from one computer to another over a network or Internet connection. It enables sharing, transferring, or transmitting a file or a logical data object between different users and/or computers both locally and remotely. Data files may be structured or unstructured — including documents, multimedia, graphics, text, and PDFs. They can be shared using download or upload and transmitted inside or outside the enterprise. A file transfer can be an upload or download. File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), BitTorrent, and Simple File Transfer Protocol are the most common file transfer protocols used in computer networks and online. Moreover, other than network or the Internet, file transfer can be performed manually by copying a file to a new folder/drive in the same computer or by copying it to a USB pen drive, CD or other portable storage devices to be transferred to another computer.

File transfers started in the 1970s when people started looking beyond floppy disks to distribute digital content. One of the first online file-sharing platforms was Usenet, an electronic bulletin board that allowed community members to post messages. The functionality allowed users to share data. Files with others in the newsgroup. In 1985 the first communication protocol, FTP, was introduced. The file transfer standard allowed users to transfer data between different computer systems using the same rules and syntax. By the 1990s, the net unfolded communications throughout the globe, permitting human beings to percentage records over a large community of computers. America Online (AOL) has become one of the first Internet provider providers. It provided a subscription-based, electronic mail platform with more than a few internet offerings consisting of document transfer. Many high-speed file transfer solutions are available today to manage the flow of digital information. Cloud storage systems like Dropbox and iCloud allow users to store digital files of all kinds (including photos and videos) off-site. The service allows users to access files and transfer them from any device to any device.

B. Socket Programming

A socket is a communications connection point (endpoint) that you can name and address on a network. Socket programming shows how to use the socket APIs to establish communication links between local and remote processes. Processes using a socket can be on the same system or different systems on different networks. Sockets are useful for both standalone and network applications. Sockets allow you to exchange information between processes on the same machine or over a network, distribute work to the most efficient machine, and access data easily. Socket Application Programming Interfaces (APIs) are the network standard for TCP/IP. Socket API is supported by a variety of operating systems. i5/OS sockets support multiple networks and transport protocols. A socket performs four basic operations: connect to a remote computer, send data, receive data, and close the connection. A socket cannot be connected to more than one host at a time. However, a socket can send and receive data from the host to which it is connected. The java.net. The socket class is the Java interface to a network socket and allows you to perform the four basic socket operations.

C. Client-Server Model

A network is composed of computers that are either a client or a server. A server is a program that is offering some service, whereas a client is a program that is requesting some service. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network services) whereas clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power. When these programs are executed, as a result, a client and a server process are created simultaneously, and these two processes communicate with each other by reading from and writing to sockets as shown in the figure.

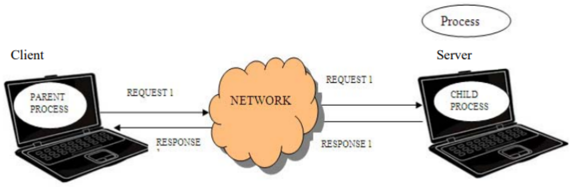


Figure: Client Server Model

These sockets are the programming interfaces provided by the TCP and UDP protocols for stream and datagram communication respectively of the transport layer, which is a part of the TCP/IP stack. When creating a network application, the developer's main task is to write the code for both the client and server programs. The client/server application that is covered here is a proprietary client/server application. A single developer (or development team) creates both the client and server programs, and the developer has complete control over what goes in the code. But because the code does not implement a public-domain protocol, other independent developers will not be able to develop code that interoperates with the application. When developing a proprietary application, the developer must be careful not to use one of the well-known port numbers defined in the RFCs.

III. METHODOLOGY

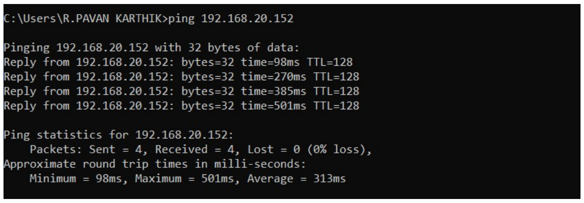
This is a Java application that uses TCP for file transfer and has a user interface built using Java AWT and Java Swing. It is built using Java Socket Programming. Which allows multiple users to connect to a server at once. It accomplishes this through multithreading. The server assigns connection IDs to the clients connected. It mandates the client to specify the working directory as a command-line argument. It allows the user to specify the host address and, if not specified, defaults it to Localhost. The user can also specify the port number as the third argument. Displays files and directories present in the server-working directory and allow the client to select files and download them onto the client system. Allows the client to upload files to the server working directory. This allows 2 clients to transfer files through the server.

IV. JAVA

Java is a high-level, class-based, object-oriented programming language designed to have as few implementation dependencies as possible. Platforms that support Java without requiring a recompilation. Java applications are generally compiled into byte code that can run on any Java Virtual Machine (JVM), regardless of the underlying computer architecture. Java syntax is similar to C and C++, but less low-level. Facilities than any of them. The Java runtime environment provides dynamic capabilities (e.g., reflection and code modification at runtime) that are not typically available in traditional compiled languages. As of 2019, according to GitHub, Java was one of the most used programming languages, especially for the client-server web. Apps, with 9 million reported developers. Java Swing tutorial is a part of Java Foundation Classes (JFC) that is used to create window-based applications. The Java Foundation Classes (JFC) are a set of GUI components that simplify the development of desktop applications. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser. Some packages of swing components that are used most are the following: Javax.swing, javax.swing.event, javax.swing.plaf.basic, javax.swing.table, javax.swing.border, javax.swing.tree. Java AWT (Abstract Window Toolkit) is an API to develop Graphical User Interface (GUI) or Windows-based applications in Java. Java AWT components are platform-dependent, i.e. components are displayed according to the view of the operating system. AWT is heavyweight, i.e. its components are using the resources of the underlying operating system (OS). The java.awt package provides classes for AWT API such as Text Field, Label, TextArea, Radio Button, CheckBox, Choice, List, etc. The AWT tutorial will help the user to understand Java GUI programming in simple and easy steps.

V. RESULTS & DISCUSSION

Find the IP address by using the command ipconfig. First, ping the other host using the Host IP address to check whether both the hosts are on the same wire less network. If the transfer of packets is successful, then both the hosts are connected to same network and are ready to communicate.

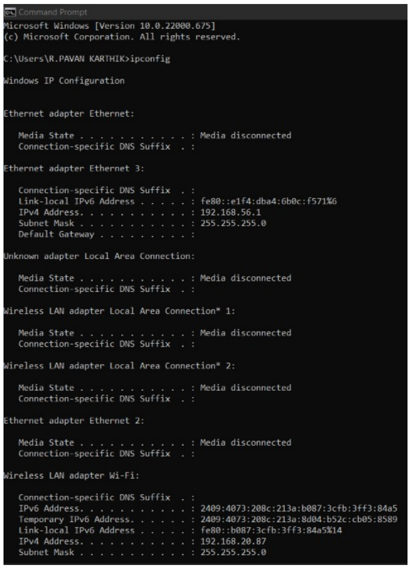


```
C:\Users\R.PAVAN KARTHIK>ping 192.168.20.152

Pinging 192.168.20.152 with 32 bytes of data:
Reply from 192.168.20.152: bytes=32 time=98ms TTL=128
Reply from 192.168.20.152: bytes=32 time=270ms TTL=128
Reply from 192.168.20.152: bytes=32 time=385ms TTL=128
Reply from 192.168.20.152: bytes=32 time=501ms TTL=128

Ping statistics for 192.168.20.152:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 98ms, Maximum = 501ms, Average = 313ms
```

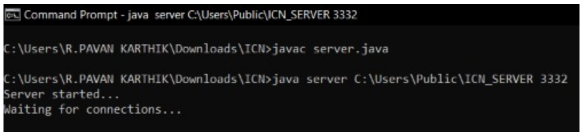
Here 192.168.20.87 is Host address which we want to ping

Next, compile the server. The command for compiling a java file is javac <Filename>
now execute it using the command java <filename> <server_DataPath> <Host Address> <port number>

Here, port Number and Host Address are optional. If address is not given it'll take Local Host's address and If Port Number is not given it'll take default port as 3333. Here, port Number is optional. If the Port Number is not given, it'll take the default port as 3333. Now execute it using the command java <filename> <server_DataPath> <port number>.

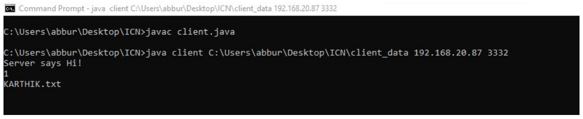


```
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\R.PAVAN KARTHIK>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 3:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::e1f4:dba4:6b0c:f571%6
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Unknown adapter Local Area Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2409:4073:208c:213a:b087:3cfb:3ff3:84a5
   Temporary IPv6 Address. . . . . . : 2409:4073:208c:213a:8d04:b52c:cb05:8589
   Link-local IPv6 Address . . . . . : fe80::b087:3cfb:3ff3:84a5%14
   IPv4 Address. . . . . . . . . . . : 192.168.20.87
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
```

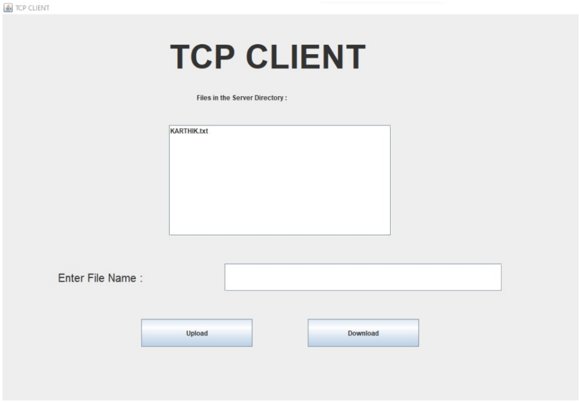


```
Command Prompt - java  server C:\Users\Public\ICN_SERVER 3332

C:\Users\R.PAVAN KARTHIK\Downloads\ICN>javac server.java

C:\Users\R.PAVAN KARTHIK\Downloads\ICN>java server C:\Users\Public\ICN_SERVER 3332
Server started...
Waiting for connections...
```

First complie client program in other host using javac client and then execute it using java client <client_data_path> <server_IpAddress> <portNumber>. if Ip Address is not given it'll take Local Host's address and if port number is not given it'll take 3332 as default



```
Command Prompt - java  client C:\Users\abbur\Desktop\ICN\client_data 192.168.20.87 3332

C:\Users\abbur\Desktop\ICN>javac client.java

C:\Users\abbur\Desktop\ICN>java client C:\Users\abbur\Desktop\ICN\client_data 192.168.20.87 3332
Server says Hi!
1
KARTHIK.txt
```



TCP CLIENT

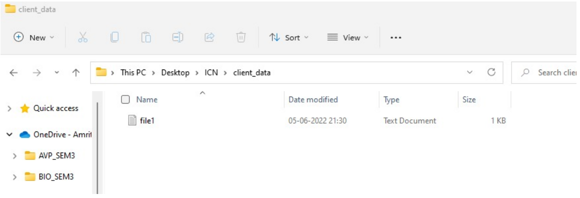
Files in the Server Directory :

KARTHIK.txt

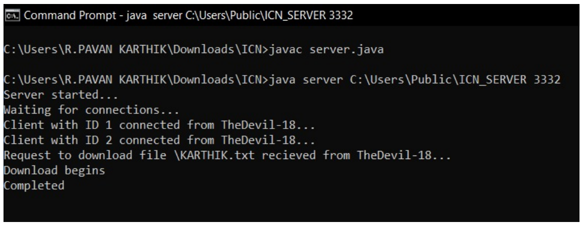
Enter File Name :

After executing the client a window will pop up as shown in this figure. In this fig we can see that there is only one file in the server that is "KARTHIK.txt". And there is options for Downloading and Uploading Files. we can upload and download the files by entering the text in the textBox shown.

files in client_data before downloading from server.

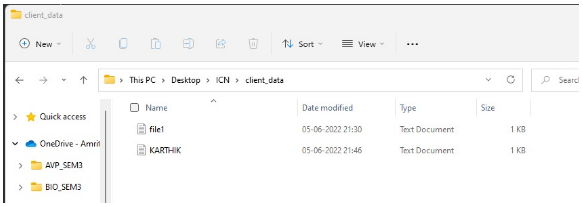


Name	Date modified	Type	Size
file1	05-06-2022 21:30	Text Document	1 KB

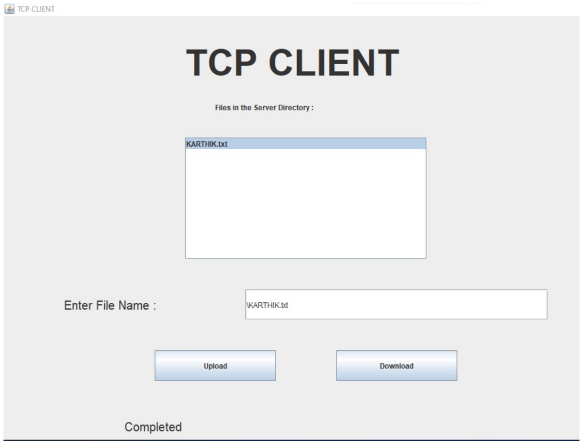


```
Command Prompt - java  server C:\Users\Public\ICN_SERVER 3332

C:\Users\R.PAVAN KARTHIK\Downloads\ICN>javac server.java

C:\Users\R.PAVAN KARTHIK\Downloads\ICN>java server C:\Users\Public\ICN_SERVER 3332
Server started...
Waiting for connections...
Client with ID 1 connected from TheDevil-18...
Client with ID 2 connected from TheDevil-18...
Request to download file \KARTHIK.txt recieved from TheDevil-18...
Download begins
Completed
```

Files in client_data after downloading from server:



Name	Date modified	Type	Size
file1	05-06-2022 21:30	Text Document	1 KB
KARTHIK	05-06-2022 21:46	Text Document	1 KB



TCP CLIENT

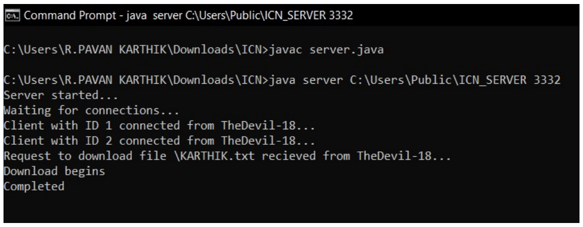
Files in the Server Directory :

KARTHIK.txt

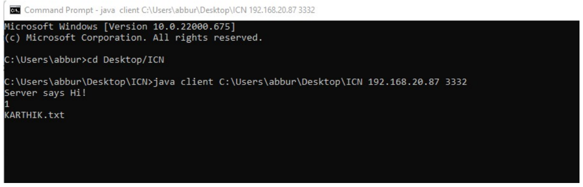
Enter File Name : \KARTHIK.txt

Completed

Client 2 trying to connect with server



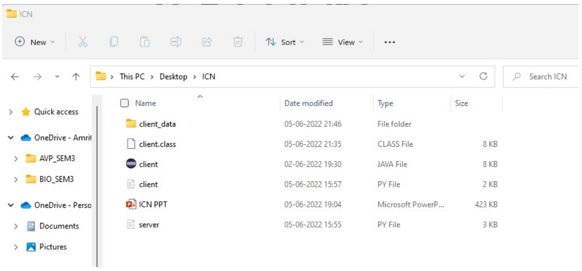
```
Command Prompt - java  server C:\Users\Public\ICN_SERVER 3332

C:\Users\R.PAVAN KARTHIK\Downloads\ICN>javac server.java

C:\Users\R.PAVAN KARTHIK\Downloads\ICN>java server C:\Users\Public\ICN_SERVER 3332
Server started...
Waiting for connections...
Client with ID 1 connected from TheDevil-18...
Client with ID 2 connected from TheDevil-18...
Request to download file \KARTHIK.txt recieved from TheDevil-18...
Download begins
Completed
```



```
Command Prompt - java  client C:\Users\abbur\Desktop\ICN 192.168.20.87 3332
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abbur>cd Desktop/ICN

C:\Users\abbur\Desktop\ICN>java client C:\Users\abbur\Desktop\ICN 192.168.20.87 3332
Server says Hi!
1
KARTHIK.txt
```

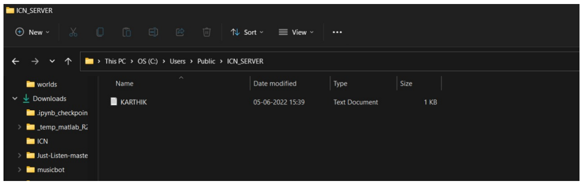
Here we've connected another client with ID 2 with a different client path

Files in client_data for 2nd client



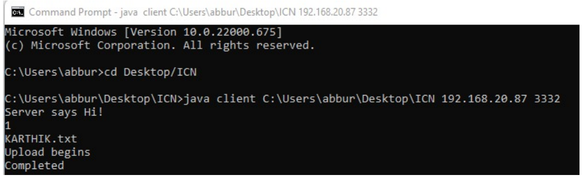
Name	Date modified	Type	Size
client_data	05-06-2022 21:46	File folder	
client.class	05-06-2022 21:35	CLASS File	8 KB
client	02-06-2022 19:30	JAVA File	8 KB
client	05-06-2022 15:57	PY File	2 KB
ICN PPT	05-06-2022 19:04	Microsoft PowerP...	423 KB
server	05-06-2022 15:55	PY File	3 KB

we are uploading client.py file to server_data, and we can see there is no client.py file in server_data

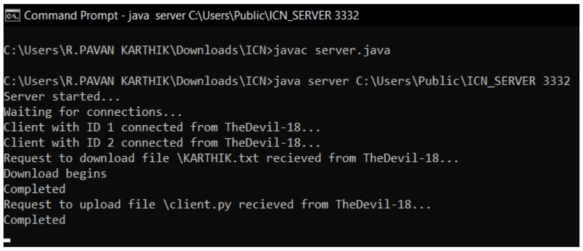


Name	Date modified	Type	Size
KARTHIK	05-06-2022 15:39	Text Document	1 KB

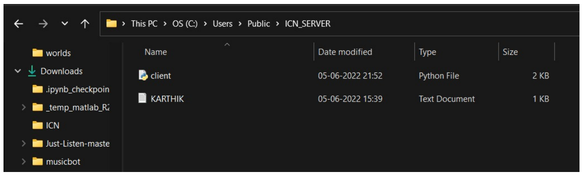
We can see only one file karthik before uploading from client



```
Command Prompt - java  client C:\Users\abbur\Desktop\ICN 192.168.20.87 3332
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abbur>cd Desktop/ICN

C:\Users\abbur\Desktop\ICN>java client C:\Users\abbur\Desktop\ICN 192.168.20.87 3332
Server says Hi!
1
KARTHIK.txt
Upload begins
Completed
```



```
Command Prompt - java  server C:\Users\Public\ICN_SERVER 3332

C:\Users\R.PAVAN KARTHIK\Downloads\ICN>javac server.java

C:\Users\R.PAVAN KARTHIK\Downloads\ICN>java server C:\Users\Public\ICN_SERVER 3332
Server started...
Waiting for connections...
Client with ID 1 connected from TheDevil-18...
Client with ID 2 connected from TheDevil-18...
Request to download file \KARTHIK.txt recieved from TheDevil-18...
Download begins
Completed
Request to upload file \client.py recieved from TheDevil-18...
Completed
_
```



Name	Date modified	Type	Size
client	05-06-2022 21:52	Python File	2 KB
KARTHIK	05-06-2022 15:39	Text Document	1 KB

Above now, we can see client.py file got uploaded from client to server data

VI.CONCLUSION

This paper describes the details about sockets, ports, socket programming over TCP . Network programming makes use of socket for interprocess communication between hosts where sockets act as the endpoint of the interprocess communication. Here sockets can also be termed as network socket or Internet socket since communication between computers is based on Internet protocol. So Network programming is also Socket Programming. The paper also describes about socket programming in java over TCP. Because java has been preferred more than any other language for establishing connections between clients and servers using sockets. Socket programming in java is easy.

REFERENCES

Limi Kalita. (n.d.). Socket Programming. International Journal of Computer Science and Information Technology. Retrieved June 5, 2022, from http://ijcsit.com/

M. Xue and C. Zhu, "The Socket Programming and Software Design for Communication Based on Client/Server," 2009 Pacific-Asia Conference on Circuits, Communications and Systems, 2009, pp. 775-777, doi: 10.1109/PACCS.2009.89.

Socket programming. (n.d.). Socket Programming; www.ibm.com. Retrieved June 6, 2022, from https://www.ibm.com/docs/en/i/7.1?topic=communications-socket-programming

Wikipedia. (n.d.). Wikipedia; wikipedia.org. Retrieved June 6, 2022, from https://wikipedia.org