**Name-Ujjawal Mandhani**

**Batch-F6**

**Enroll No.-9918103237**

**Question 1:**Link of your Git hub account

https://github.com/ujjawal-mandhani979392

[Link](https://github.com/ujjawal-mandhani979392)

**Question 2:**C program For basic file handling.

# include <stdio.h>

# include <string.h>

int main( )

{

FILE \*filePointer ;

FILE \* fptr;

char fn[50];

char str[] = "Basics Filehanling\n";

char dataToBeWritten[50]

= "Basics of File Handling";

FILE \*fptr1;

FILE \* file\_pointer;

char buffer[30], c;

file\_pointer = fopen("fprintf\_test.txt", "r");

fgets(buffer, 50, file\_pointer);

printf("%s\n", buffer);

fptr1 = fopen("fprintf\_test.txt", "w"); // "w" defines "writing mode"

/\* write to file \*/

fprintf(fptr1, "Basic File handling\n");

for (i = 0; str[i] != '\n'; i++) {

/\* write to file using fputc() function \*/

fputc(str[i], fptr);

}

filePointer = fopen("filename.c", "w") ;

if ( filePointer == NULL )

{

printf( "filename.c file failed to open." ) ;

}

else

{

printf("The file is now opened.\n") ;

if ( strlen ( dataToBeWritten ) > 0 )

{

fputs(dataToBeWritten, filePointer) ;

fputs("\n", filePointer) ;

}

fclose(filePointer) ;

printf("Data successfully written in file filename.c\n");

printf("The file is now closed.") ;

}

fclose(fptr);

fclose(fptr1);

return 0;

}

https://github.com/ujjawal-mandhani979392/basic\_File\_Handling\_C

**Question 3** Extracted open source project

Apache HTTP Server is a free and open-source web server that delivers web content through the internet. The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It is commonly referred to as Apache and after development, it quickly became the most popular HTTP client on the web. It’s widely thought that Apache gets its name from its development history and process of improvement through applied patches and modules but that was corrected back in 2000. It was revealed that the name originated from the respect of the Native American tribe for its resiliency and durability.

Apache is considered open source software, which means the original source code is freely available for viewing and collaboration. Being open source has made Apache very popular with developers who have built and configured their own modules to apply specific functionality and improve on its core features. Apache has been around since 1995 and is responsible as a core technology that helped spur the initial growth of the internet in its infancy.

One of the pros of Apache is its ability to handle large amounts of traffic with minimal configuration. It scales with ease and with its modular functionality at its core, you can configure Apache to do what you want, how you want it. You can also remove unwanted modules to make Apache more lightweight and efficient.

Some of the most popular modules that can be added are SSL, Server Side Programming Support (PHP), and Load Balancing configs to handle large amounts of traffic. Apache can also be deployed on Linux, MacOS, and Windows. If you learn how to configure Apache on Linux, you can administer Apache on Windows and Mac. The only difference would be directory paths and installation processes.

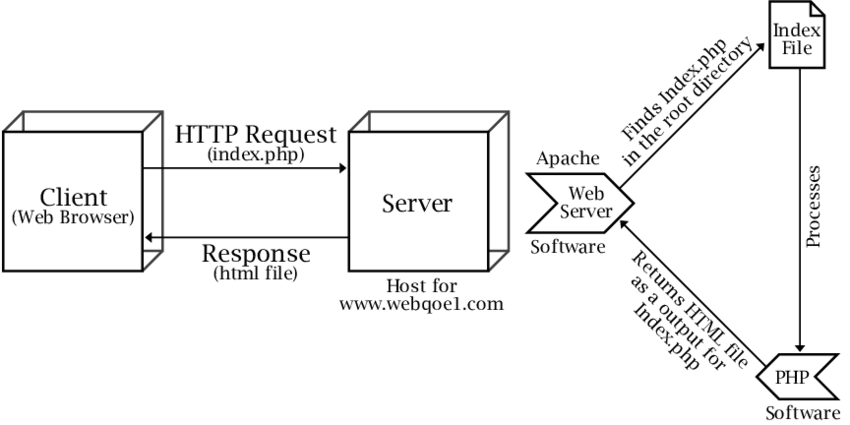
Apache functions as a way to communicate over networks from client to server using the TCP/IP protocol. Apache can be used for a wide variety of protocols, but the most common is HTTP/S. HTTP/S or Hyper Text Transfer Protocol (S stands for Secure) is one of the main protocols on the web, and the one protocol Apache is most known for.

HTTP/S is used to define how messages are formatted and transmitted across the web, with instructions for browsers and servers on how to respond to various requests and commands. Hypertext Transfer Protocol Secure is usually through port 443 with the unsecured protocol being through port 80.

The Apache server is configured via config files in which modules are used to control its behavior. By default, Apache listens to the IP addresses configured in its config files that are being requested. This is where one of Apaches many strengths come into play.

With the Listen directive, Apache can accept and route specific traffic to certain ports and domains based on specific address-port combination requests. By default, Listen runs on port 80 but Apache can be bound to different ports for different domains, allowing for many different websites and domains to be hosted and a single server. You can have domain1.com listening on port 80, domain2.com on port 8080 and domain3.com on port 443 using HTTPS all on Apache.

Once a message reaches its destination or recipient, it sends a notice, or ACK message, basically giving acknowledgment to the original sender that their data has successfully arrived. If there’s an error in receiving data, or some packets were lost in transit, the destination host or client sends a Not Acknowledged, or NAK message, to inform the sender that the data needs to be retransmitted.



As early as 1994, Rob McCool at the National Center for Supercomputing Applications (NCSA) in Illinois created a simple web server which served pages using one of the early versions of today's HTTP protocol. Web servers were not ubiquitous like they are today. In these days, the Web was still in its early days and there was only one web browser developed at CERN where the WWW was invented only shortly before. Rob's web server was adopted quite fruitfully throughout the web due to its extensible nature. When its source code spread, web page administrators around the world developed extensions for the web server and helped to fix errors. When Rob left the NCSA in late 1994, he also left a void because there was nobody left to maintain the web server along with its extensions. Quickly it became apparent that the group of existing users and developers needed to join forces to be able to maintain NCSA HTTP.

At the beginning of 1995, the Apache Group was formed to coordinate the development of the NCSA HTTP web server. This led to the first release of the Apache web server in April 1995. During the same time, development at NCSA started picking off again and the two teams were in vivid exchange about future ideas to improve the web server. However, the Apache Group was able to develop its version of the web server much faster because of their structure which encouraged worldwide collaboration. At the end of the year, the Apache server had its architecture redone to be modular and it executed much faster.

https://github.com/apache