CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

Bachelor of Computer Application

Subject Name: Web Security

Code:CAT-309

Web Application

DISCOVER . LEARN . EMPOWER

Web Applications

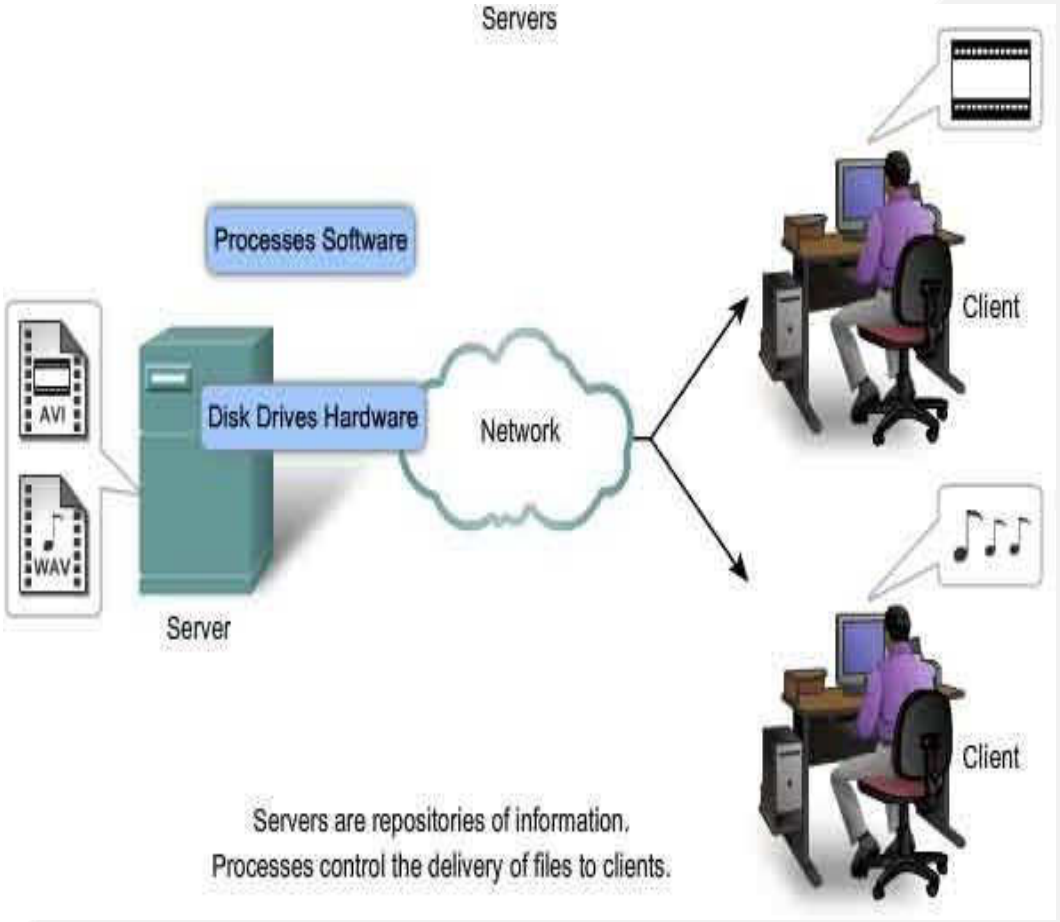
Course Outcome

CO Number	Title	Level
CO1	Students will learn HTTP methods.	Understand
CO2	Understand the functionality of server side and client side languages.	Understand
CO3	Students will explain the working of encoding algorithms.	Understand

- HTTP methods and its types
- Authentication Process
- Web Functionality
- Encoding Scheme
- HTML Encoding
- URL Encoding

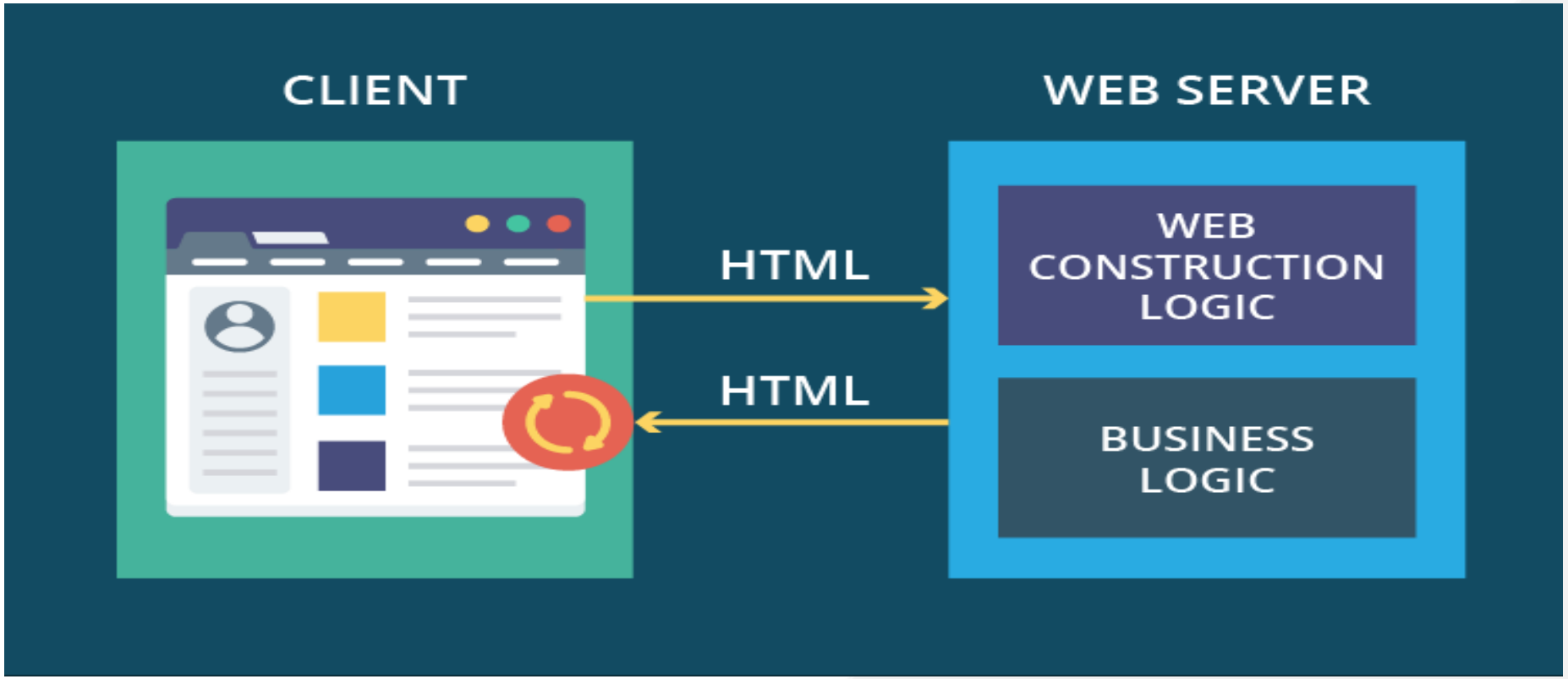
Web Application Architecture

- How to server and client communicate



Reference:http://www.tutorialpoints.com

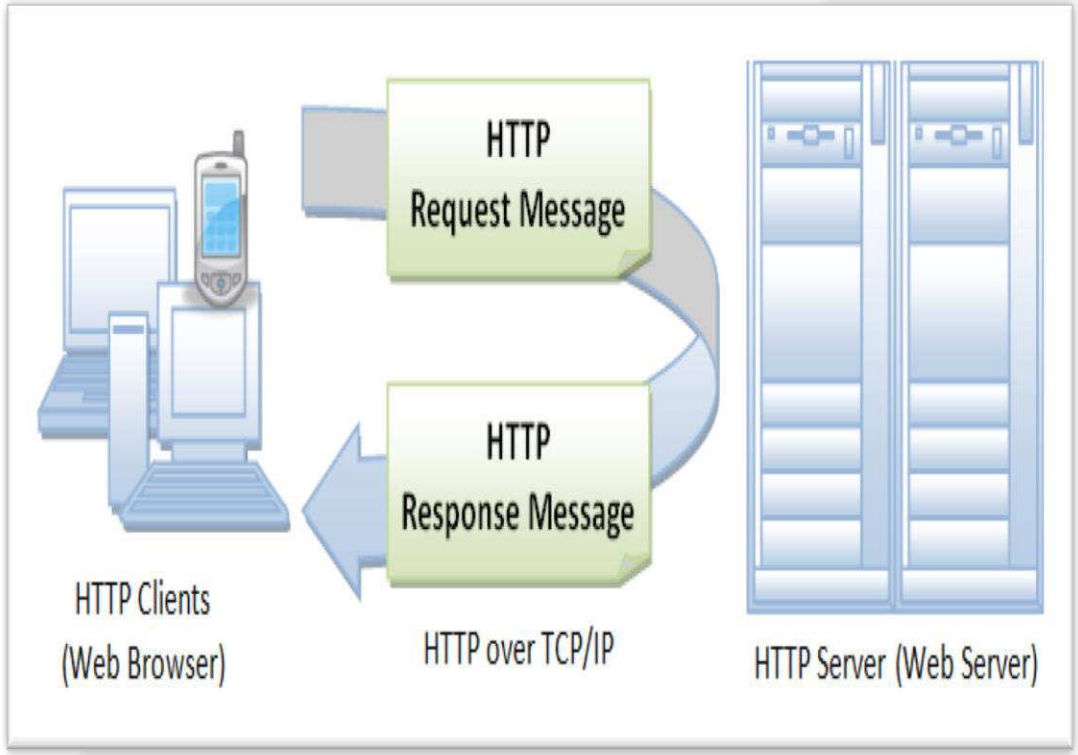
Web Application Architecture



Reference: https://www.scnsoft.com/blog/web-application-architecture

HTTP

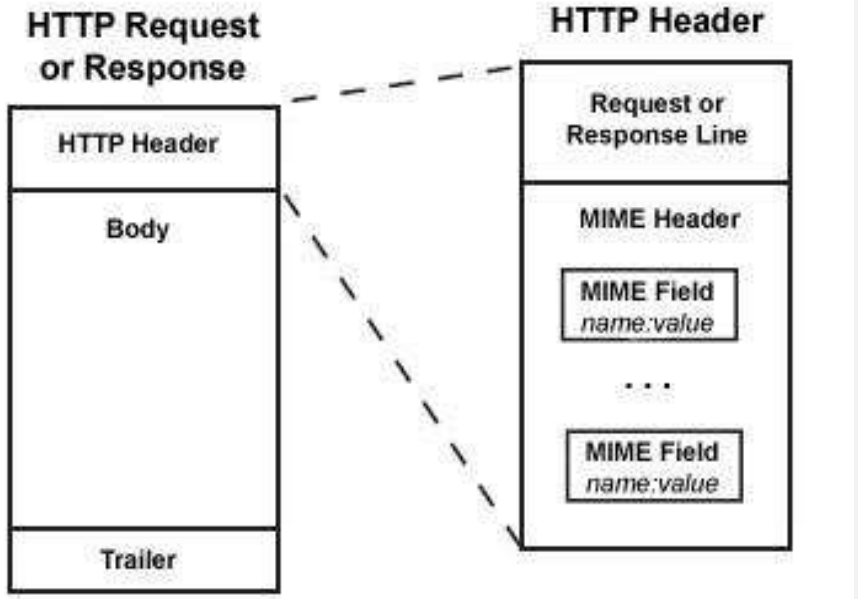
- HTTP is Client-Server Protocol. it is used for data transmission from sender to receiver.
- it use two header: response header and request header.



Reference: WWW.tutorialspoint.com

HTTP Headers

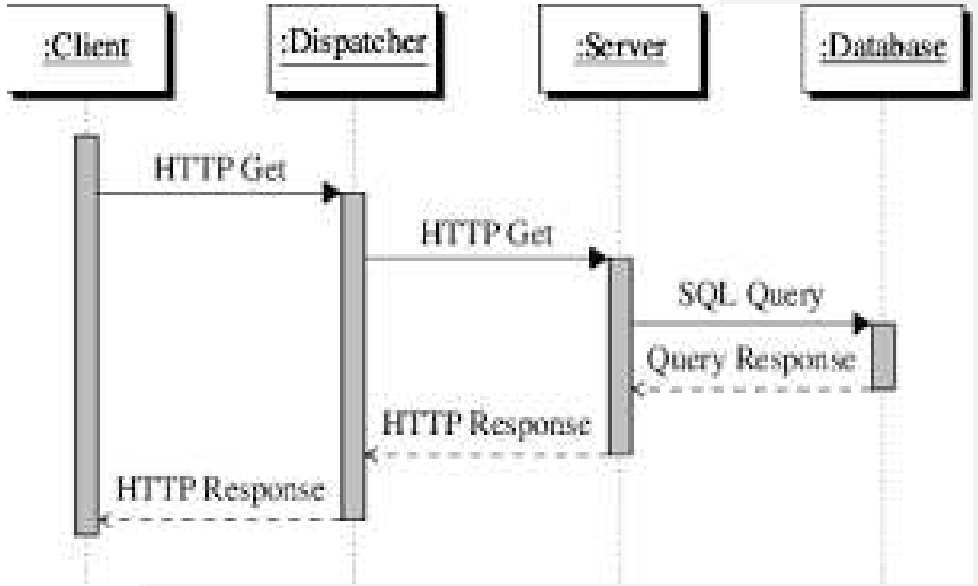
- All HTTP messages (requests and responses) consist of one or more **headers** ,each on a separate line, followed by a mandatory blank line, followed by an optional message body.



Reference:http://www.tutorialpoints.com

HTTP Requests

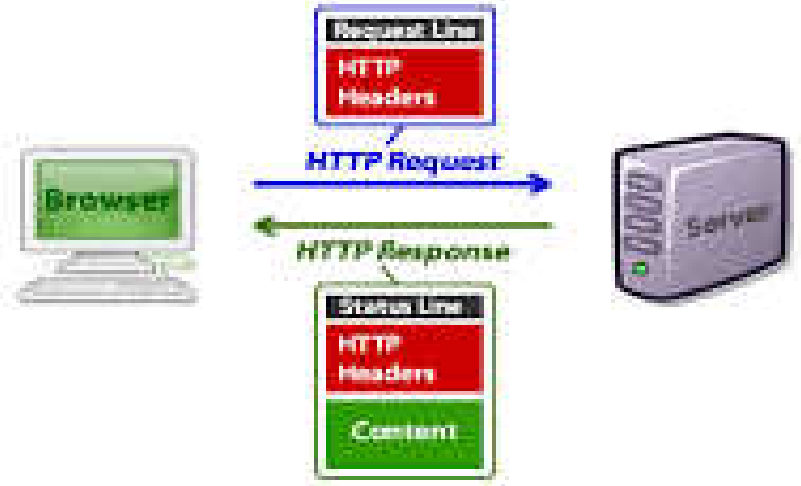
- It store the information that send from sender to receiver.
- In which request line determine the methods GET and PUT.



Reference:https://www.google.com/search?q=http+response+diagram&rlz=1C1CHZL_enIN809IN809&source=lnms&sa=X&ved=0ahUKEwijt9-FjdriAhWCbX0KHSUPAJYQ_AUICygA&biw=1366&bih=608&dpr=1

HTTP Response

- After receving request server send response according to the request.
- It contain following information:
- HTTP Version
- Status code



Reference:https://www.google.com/search?q=http+response+diagram&rlz=1C1CHZL_enIN809IN809&source=lnms&sa=X&ved=0ahUKEwijt9-FjdriAhWCbX0KHSUPAJYQ_AUICygA&biw=1366&bih=608&dpr=1

Web Functionality

- Client side is the users end of functionality, while server side is based on the server's end. As a developer

Two ypes of web functionality as follow:

- **Server-Side Functionality**
- **Client-Side Functionality**

Server-Side Functionality

In server side functionality,user choose which stages, working frameworks, programming dialects, systems, and libraries will be utilized. a wide scope of innovations on the server side to convey their usefulness:

- Scripting dialects, for example, PHP, VBScript, and Perl
- Web application stages, for example, ASP.NET and Java.
- Web servers, for example, Apache, IIS, and Netscape Enterprise
- Databases, for example, MS-SQL, Oracle, and MYSQL.

Client-Side Functionality

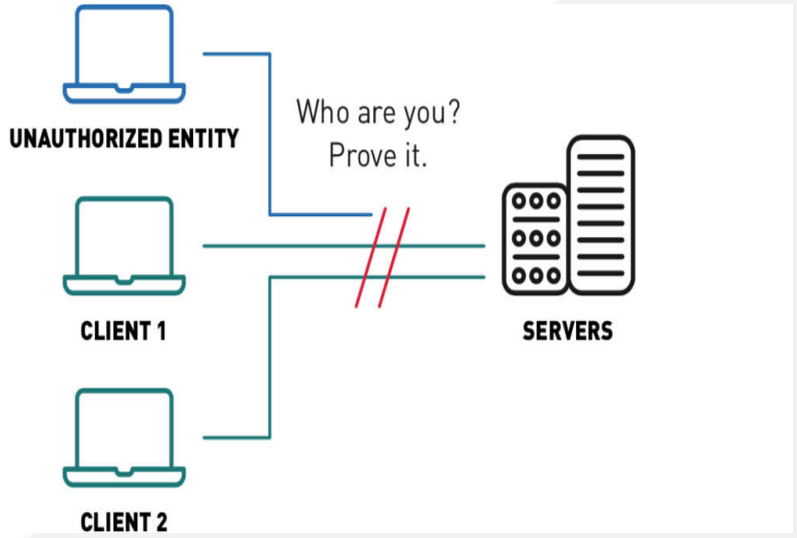
- In order for the server-side application to accept user input and operations, and pass the results of these back to the user, it needs to provide a client-side.
- In this Scripting is possible to be blocked , where as server side scripting can't be blocked by the user , only , if you validate using CLIENT SIDE only .

Client-Side Functionality:

- *HTML*
- *Hyperlinks*
- *Forms*
- *JavaScript*
- *Java applets*
- *ActiveX controls*

Authentication

- Authentication is a mechanism to check the user is authorized or not if any user want to access the data form network.



Reference:
https://mapr.com/whitepapers/mapr security/

Authentication methods

- Two-factor authentication
- Multifactor authentication
- One-time password
- Three-factor authentication
- Biometrics
- Mobile authentication

Authentication methods(Conti..)

- Continuous authentication
- API authentication
- Biometrics for Network Security
- Token Authentication
- Out-of-Band Authentication (OOB)

Encoding Scheme

Encoding is the way toward changing over information into an organization required for various data handling needs,it including:

- Program incorporating and execution
- Information transmission, stockpiling and pressure/decompression
- Application information handling, for example, record change

Types of Encoding Schemes

- HTML Encoding
- URL Encoding
- Unicode Encoding
- Base64 Encoding
- Hex Encoding

HTML Encoding

- Html encoding is chiefly used to speak to different characters so they can be securely utilized inside a HTML record (as the name may propose). As you most likely are aware there are different characters that are a piece of the HTML markup itself, (for example, <, > and so forth.). To utilize these inside the record as substance you have to HTML encode them. There are two different ways to HTML encode characters.
- You can likewise HTML encode any character utilizing its ASCII code by prefixing it with &# and after that utilizing the ASCII decimal esteem, or prefixing it with &#x and utilizing the ASCII hex esteem e.g.:
  - ’ _ '
  - < - <
  - > - >
  - ‘ _ “

URL Encoding

- When organizing URLs, they can just contain printable ASCII characters (these are characters with ASCII codes between decimal 32 and 126, for example hex 0x20 – 0x7E). A few characters inside this range may include extraordinary implications inside the URL or inside the HTTP convention. URL encoding becomes possibly the most important factor when we have either a few characters with unique significance in the URL or need to have characters outside the printable range. To URL encode a character we simply prefix its hex value with a % e.g.:
- % - %25
- space - %20
- tab - %09
- = - %3D

Unicode Encoding

- Unicode encoding can be utilized to encode a character from any language or composing framework on the planet. 16 bit Unicode encoding is to some degree like URL encoding (which is the reason they can here and there be befuddled). To encode a character in 16 bit Unicode, you begin with %u and afterward add the code point, of the Unicode character that you need to encode. A code point is fundamentally only a 4 digit hexadecimal number that maps to a specific character that you're endeavoring to speak to as indicated by the Unicode standard.

@ - %u0040

∞ (infinity)- %u221E

Base64 Encoding

- Base64 sees information in squares of 3 bytes which is 24 bits. These 24 bits are then isolated into 4 pieces of 6 bits each, and every one of those throws is then changed over to its relating base64 esteem. Since it manages 6 bit pieces there are 64 potential characters each lump can guide to (thus the name).
- On the off chance that we need to change over the word 'cake' to base 64, we basically convert every one of the characters to it's ASCII decimal esteem and afterward get the paired estimation of every decimal esteem.
- For our situation:cake = 01100011011000010110101101100101
- We now need to break up our binary string into chucks of 6 bits each, and since every 3 characters must make 4 base64 characters, we can pad with 0's if we don't have enough:
- 011000 110110 000101 101011 011001 010000 000000 000000

Hex Encoding

- Essentially with hex encoding, we basically utilize the hex estimation of each character to speak to an accumulation of characters. So in the event that we needed to speak to the word 'hi' it would be:

68656C6C6F

- It is just basic when we're encoding printable ASCII characters (I referenced those above). When we have to encode global characters or some likeness thereof, it turns out to be increasingly confounded as we need to return to Unicode.

Path Traversal Attack

- Utilizing a path traversal attack (otherwise called catalog/directory traversal), an assailant can get to information put away outside the web root envelope .
- A path traversal attack enables aggressors to get to catalogs that they ought not be getting to, as config records or some other documents/indexes that may contains server's information not proposed for open.
- By controlling factors that reference documents with "spot speck cut (../)" successions and its varieties or by utilizing supreme record ways, it might be conceivable to get to subjective records and indexes put away on document framework including application source code or design and basic framework documents.

Path Traversal Attack Prevention

A possible algorithm for preventing directory traversal would be to:

- Giving fitting authorizations to indexes and records. A PHP record commonly keeps running as www-information client on Linux. We ought not enable this client to get to framework documents. Be that as it may, this doesn't keep this client from getting to web-application explicit config documents.
- Procedure URI demands that don't result in a document demand, e.g., executing a guide into client code, before proceeding beneath.
- At the point when a URI demand for a document/registry is to be made, form a full way to the record/index on the off chance that it exists, and standardize all characters (e.g., %20 changed over to spaces).
- Utilizing a hard-coded predefined document expansion to addition the way does not restrict the extent of the assault to records of that record augmentation.

Assessment Pattern

- Element 1(Quiz) 12 marks
- Element 2 (Surprise Test) 09 Marks
- Element 3 (Assignments) 12 Marks
- Element 4 (Tutorial/Presentations) 09 Marks

Applications

- Client -side scripting provide more interactivity by immediately responding to users' actions.
- Can give developers more control over the look and behaviour of their Web widgets.
- Client-side Environment
- Server-side Environment
- Encoding keeps your data safe since the files are not readable unless you have access to the algorithms that were used to encode it. This is a good way to protect your data from theft since any stolen files would not be usable.

References

Reference Books

- DafyddStuttard, "The Web Application Hacker's Handbook", Wiley India Pvt. Ltd.
- Web Security by Oscar MeridaPublisher: php[architect]
- Web Security-Privacy and Commerce,SimsonGarfinkel, O'Reilly.

Reference websites:

- WWW.tutorialspoint.com
- https://www.geeksforgeeks.org/path-traversal-attack-prevention/
- https://searchsecurity.techtarget.com/definition/authentication

THANK YOU

For queries
Email: mandeepkaur.uic@cumail.in