Web Technology:

Web Technology refers to the various tools and techniques that are utilized in the process of communication between different types of devices over the Internet. A web browser is used to access web pages. Web browsers can be defined as programs that display text, data, pictures, animation, and video on the Internet. Hyperlinked resources on the World Wide Web can be accessed using software interfaces provided by Web browsers.

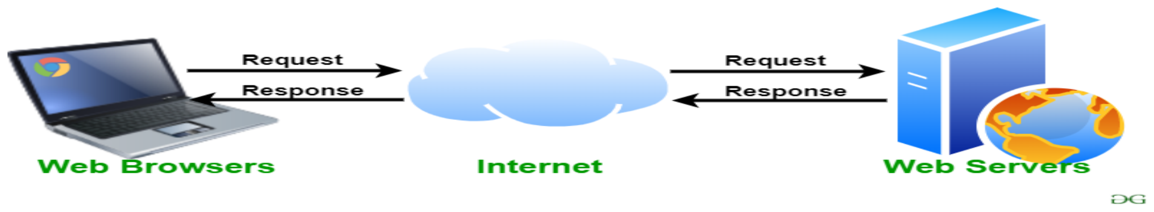
Web Technology can be Classified into the Following Sections:

World Wide Web (WWW): The World Wide Web can be defined as the collection of different websites around the world, containing different information shared via local servers (or computers). WWW is based on several different technologies: Web browsers, Hypertext Markup Language (HTML), and Hypertext Transfer Protocol (HTTP).

Components of the Web: There are 3 components of the web:

1. *Uniform Resource Locator (URL):* serves as a system for resources on the web.
2. *Hypertext Transfer Protocol (HTTP):* specifies communication of browser and server.
3. *Hyper Text Markup Language (HTML):* defines the structure, organization and content of a webpage.

Web Server: Web server is a program which processes the network requests of the users and serves them with files that create web pages. This exchange takes place using Hypertext Transfer Protocol (HTTP). Basically, web servers are computers used to store HTTP files which makes a website and when a client requests a certain website, it delivers the requested website to the client. For example, you want to open Facebook on your laptop and enter the URL in the search bar of google. Now, the laptop will send an HTTP request to view the Facebook webpage to another computer known as the webserver. This computer (webserver) contains all the files (usually in HTTP format) which make up the website like text, images, gif files, etc. After processing the request, the webserver will send the requested website-related files to your computer and then you can reach the website. Different websites can be stored on the same or different web servers but that doesn't affect the actual website that you are seeing in your computer. The web server can be any software or hardware but is usually a software running on a computer. One web server can handle multiple users at any given time which is a necessity otherwise there had to be a web server for each user and considering the current world population, is nearly close to impossible. A web server is never disconnected from the internet because if it was, then it won't be able to receive any requests, and therefore cannot process them.



There are many web servers available in the market both free and paid. Some of them are described below:

Apache HTTP server: It is the most popular web server and about 60 percent of the world's web server machines run this web server. The Apache HTTP web server was developed by the Apache Software Foundation. It is an open-source software which means that we can access and make changes to its code and mold it according to our preference. The Apache Web Server can be installed and operated easily on almost all operating systems like Linux, MacOS, Windows, etc.

Microsoft Internet Information Services (IIS): IIS (Internet Information Services) is a high performing web server developed by Microsoft. It is strongly united with the operating system and is therefore relatively easier to administer. It is developed by Microsoft; it has a good customer support system which is easier to access if we encounter any issue with the server. It has all the features of the Apache HTTP Server except that it is not an open-source software and therefore its code is inaccessible which means that we cannot make changes in the code to suit our needs. It can be easily installed in any Windows device.

Lighttpd: Lighttpd is pronounced as 'Lightly'. It currently runs about 0.1 percent of the world's websites. Lighttpd has a small CPU load and is therefore comparatively easier to run. It has a low memory footprint and hence in comparison to the other web servers, requires less memory space to run which is always an advantage. It also has speed optimizations which means that we can optimize or change its speed according to our requirements. It is an open-source software which means that we can access its code and add changes to it according to our needs and then upload our own module (the changed code).

Jigsaw Server: Jigsaw has been written in the Java language and it can run CGI (common gateway interference) scripts as well as PHP programs. It is not a full-fledged server and was developed as an experimental server to demonstrate the new web protocols. It is an open-source software which means that we can access its code and add changes to it according to our needs and then upload our own module (the changed code). It can be installed on any device provided that the device supports Java language and modifications in Java.

Sun Java System: The Sun Java System supports various languages, scripts, and technologies required for Web 2.0 such as Python, PHP, etc. It is not an open-source software and therefore its code is inaccessible which means that we cannot make changes in the code to suit our needs.

Webpage: A webpage is a digital document that is linked to the World Wide Web and viewable by anyone connected to the internet having a web browser. It can contain any type of information, such as text, color, graphics, animations, videos, sounds, etc.
A webpage is a document that is written in the HTML, it can be viewed from the Internet. It can be accessed by entering the URL on the address bar of the web browser.

Content-wise the components of a webpage are: Hypertext and Hyperlinks

1. **Hypertext:**
   It refers to a digital text, which is more than just text as it can include information in various media formats such as:

   - text
   - color
   - graphic
   - animation
   - video
   - sound
   - hyperlinks
2. **Hyperlinks:**
   It refers to a link from a hypertext file to another such file. A hyperlink can be in the form of a graphic or text, upon clicking where the linked document opens up.

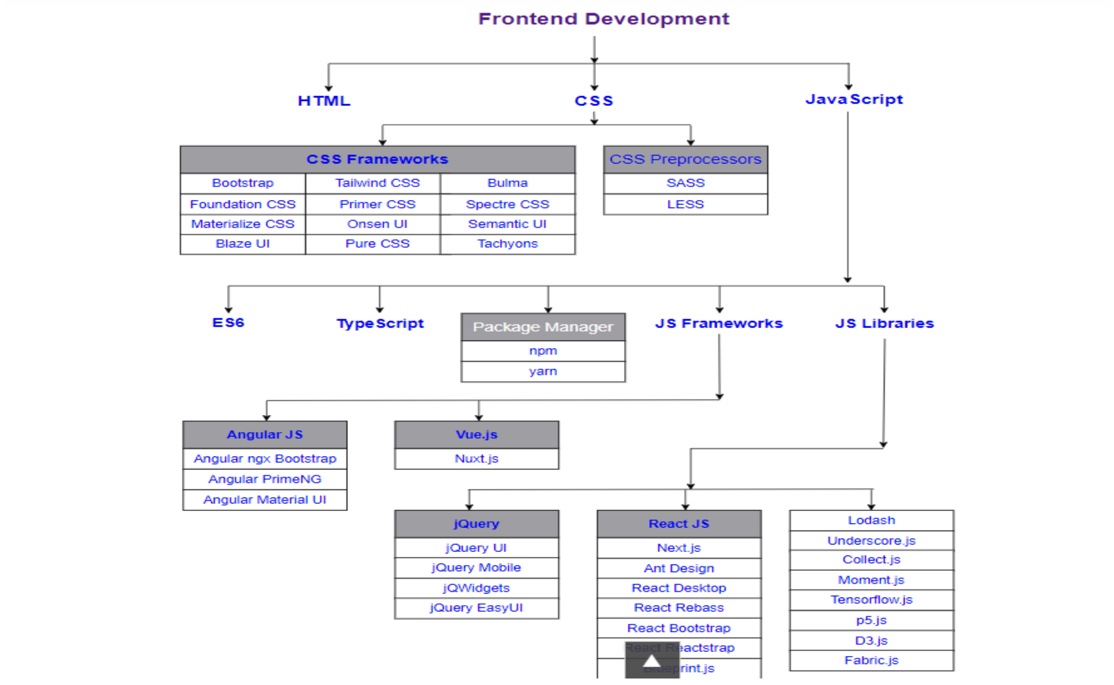
Structure wise the components of a web page are:

1. **Page Title**
   This is a single line text which is displayed on the title bar of the browser displaying web page.
2. **Header** –
   This is generally a one or two line text (sometimes a graphics/image) defining the purpose of the web page. It is displayed at the top of the web page, below the address bar of the browser.
3. **Body of the Web page**
   This is the section below the header of the web page and it contains the actual content of the web page.
4. **Navigational Links**
   These are the hyperlinks placed on the web page using which you can move the linked web pages/documents.
5. **Footer**
   This is the bottom section of the web page. This is the section where usually the copyright notice, website contact information, etc. is put.

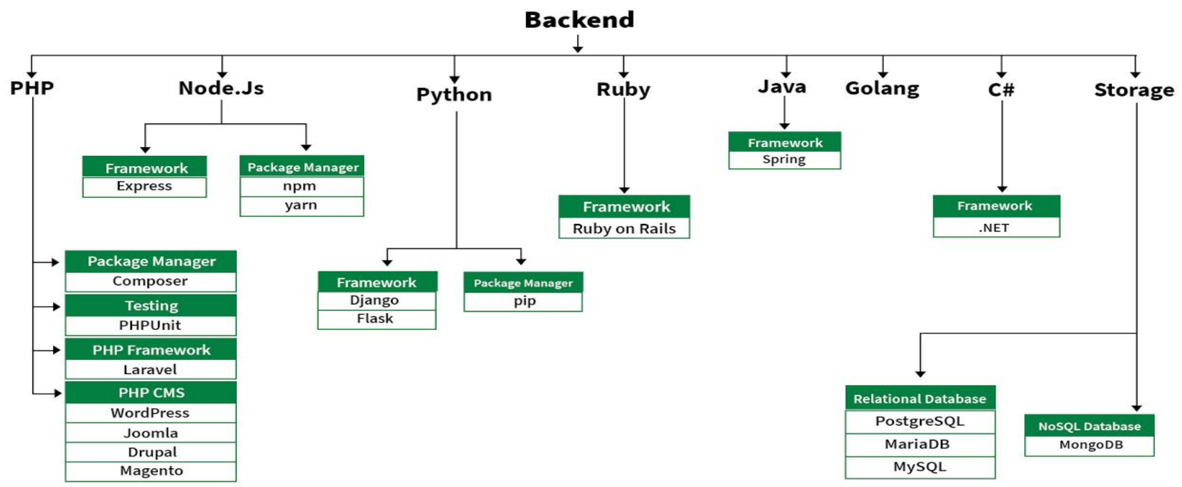
Web Development: Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

Web Development can be Classified into Two Ways:

Frontend Development: The part of a website that the user interacts directly is termed as front end. It is also referred to as the 'client side' of the application.



Backend Development: Backend is the server side of a website. It is the part of the website that users cannot see and interact. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.



HTML

https://www.w3schools.com/html/html_intro.asp

https://www.geeksforgeeks.org/html/html-exercises/

https://programmingtrick.com/html-assignments

What is JavaScript?

High-Level Language: JavaScript is considered high-level because it abstracts away most of the complex details of the computer's hardware, allowing developers to focus on writing code without worrying about memory management and other low-level operations.

Interpreted Language: Unlike compiled languages like C++ or Java, JavaScript code is executed directly by the web browser's JavaScript engine without needing to be compiled into machine code first.

Object-Oriented: JavaScript supports object-oriented programming (OOP) principles, enabling developers to create reusable code through objects and classes.

Dynamic and Weakly Typed: Variables in JavaScript do not require an explicit type definition, and their types can change at runtime, making the language flexible but sometimes prone to runtime errors.

Event-Driven: JavaScript excels in handling events (such as user inputs, clicks, or mouse movements), making it ideal for interactive web applications.

Why We Use JavaScript?

Client-Side Scripting: JavaScript runs in the user's web browser, enabling the creation of interactive and responsive web pages. This allows for dynamic content updates without needing to reload the entire page.

Enhanced User Experience: By enabling features like form validation, interactive maps, animations, and real-time updates, JavaScript improves the overall user experience on websites and web applications.

Wide Browser Support: All modern web browsers have built-in support for JavaScript, making it a universal tool for web development.

Rich Ecosystem and Libraries: JavaScript has a vast ecosystem of libraries and frameworks, such as React, Angular, and Vue.js, which streamline the development process and offer powerful tools for building complex applications.

Server-Side Development: With the advent of Node.js, JavaScript is not limited to client-side scripting anymore. Node.js allows developers to use JavaScript for server-side programming, creating a full-stack development environment with a single language.

Versatility: JavaScript is versatile and can be used in various environments, including web development, mobile app development (using frameworks like React Native), desktop application development (using Electron), and even in IoT (Internet of Things) applications.

Community and Support: JavaScript has a large and active community, which means there are ample resources, tutorials, and forums for learning and troubleshooting.

Server-Side Programming

https://www.dcs.bbk.ac.uk/~ptw/teaching/IWT/server/notes.html

https://www.baeldung.com/cs/http-get-vs-post

Web Security Defined

Web security protects networks, servers, and computer systems from damage to or the theft of software, hardware, or data. It includes defending computer systems from misdirecting or disrupting the services they are designed to provide.

Web security is synonymous with cybersecurity and also covers website security, which involves protecting websites from attacks. It includes cloud security and web application security, which defend cloud services and web-based applications, respectively. Website protection technology has enabled enhanced protection mechanisms, such as the protection of a virtual private network (VPN), which also falls under the web security umbrella.

https://www.fortinet.com/resources/cyberglossary/what-is-web-security

https://www.zscaler.com/resources/security-terms-glossary/what-is-web-security

https://www.geeksforgeeks.org/web-security-considerations/

https://www.cloudflare.com/en-gb/learning/security/what-is-web-application-security/

SQL Injection

SQL injection is a code injection technique attackers use to gain unauthorized access to a database by injecting malicious SQL commands into web page inputs.
Attackers can extract sensitive information, modify database data, execute administration operations on the database (such as shutdown DBMS), recover the content of a given file present on the DBMS file system, and in some cases, issue commands to the operating system.

In this article, we will discuss what is SQLi(SQL Injection), Types of SQL injection, SQL injection in web pages, how to prevent SQL injection attacks, and many more.

What is SQL Injection?
SQLi or **SQL Injection** is a web page vulnerability that lets an attacker make queries with the database. Attackers take advantage of web application vulnerability and inject an SQL command via the input from users to the application.
Attackers can SQL queries like SELECT to retrieve confidential information which otherwise wouldn't be visible. SQL injection also lets the attacker to perform a **denial-of-service (DoS) attacks** by overloading the server requests.

The Exploitation of SQL Injection in Web Applications

Web servers communicate with database servers anytime they need to retrieve or store user data. SQL statements by the attacker are designed so that they can be executed while the web server is fetching content from the application server.

SQL in Web Pages

SQL injection typically occurs when you ask a user for input, such as their username/user ID, and instead of their name/ID, the user inputs an SQL statement that will be executed without the knowledge about your database.

For example,

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users
WHERE UserId = " + txtUserId;
```

The above code is constructing an SQL query by directly concatenating a user input (txtUserId) into the query string. Attackers can easily exploit this by giving an input that is always true, like x=x,1=1, etc.

If the attacker gave input as " 105 OR 1=1 " in the UserId field, the resulting SQL will be:

```
SELECT * FROM Users WHERE UserId = 105 OR 1=1;
```

This resulting query will return data of all users, not just the user with UserId ="105″.

SQL Injection Example

For a better understanding of how attackers do a SQL injection attack, let's learn **how to do an SQL injection attack** ourselves. In this example, we will perform a basic SQL injection attack and learn the process behind it.
Suppose we have an application based on student records. Any student can view only his or her records by entering a unique and private student ID.

Suppose we have a field like the one below:

Student id: The student enters the following in the input field: **12222345 or 1=1**.
Query:

```
SELECT * FROM STUDENT WHERE
STUDENT-ID == 12222345 or 1 = 1
```

SQL Injection based on 1=1 is always true. As you can see in the above example, **1=1** will return all records for which this holds true. So basically, all the student data is compromised. Now the malicious user can also similarly use other SQL queries.
Consider the following SQL query.

Query 1:

```
SELECT * FROM USER WHERE
USERNAME = “” AND PASSWORD=””
```

Now the malicious attacker can use the '=' operator cleverly to retrieve private and secure user information. So following query when executed retrieves protected data, not intended to be shown to users.

Query 2:

```
SELECT* FROM User WHERE
(Username = “” OR 1=1) AND
(Password=”” OR 1=1).
```

Since **1=1** always holds true, user data is compromised.
SQL Injection Types
There are different types of SQL injection attacks:

1. In-band SQL Injection

It involves sending malicious SQL queries directly through the web application’s interface and allows attackers to extract sensitive information or modify the database itself.

2. Error-based SQL Injection

Attackers exploit error messages generated by the web application by analyzing error messages to gain access to confidential data or modify the database.

3. Blind SQL Injection

Attackers send malicious SQL queries and observe the application’s response. By analyzing the application’s behavior, attackers can determine the success of the query.

4. Out-of-band SQL Injection

Uses a different channel to communicate with the database. Allows attackers to exfiltrate sensitive data from the database.

5. Inference-based SQL Injection

Uses statistical inference to gain access to confidential data. Attackers create queries that return the same result regardless of input values.

Impact of SQL Injection
The hacker can retrieve all the user data present in the database such as user details, credit card information, and social security numbers, and can also gain access to protected areas like the administrator portal. It is also possible to delete user data from the tables.

Nowadays, all online shopping applications and bank transactions use back-end database servers. So in case the hacker is able to exploit SQL injection, the entire server is compromised.

SQL Injection Prevention
Developers can use the following prevention measures to prevent SQL injection attacks.

- **User Authentication**: Validating input from the user by pre-defining length, type of input, of the input field and authenticating the user.
- Restricting access privileges of users and defining how much amount of data any outsider can access from the database. Basically, users should not be granted permission to access everything in the database.
- Do not use system administrator accounts.

Use SQL Parameters for Protection

To protect a web site from SQL injection, you can use SQL parameters.

SQL parameters are values that are added to an SQL query at execution time, in a controlled manner.

ASP.NET Razor Example

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = @0";
db.Execute(txtSQL,txtUserId);
```

Note that parameters are represented in the SQL statement by a @ marker.

The SQL engine checks each parameter to ensure that it is correct for its column and are treated literally, and not as part of the SQL to be executed.

INSERT INTO STATEMENT IN PHP:

```
$stmt = $dbh->prepare("INSERT INTO Customers (CustomerName,Address,City)
VALUES (:nam, :add, :cit)");
$stmt->bindParam(':nam', $txtNam);
$stmt->bindParam(':add', $txtAdd);
$stmt->bindParam(':cit', $txtCit);
$stmt->execute();
```

Denial of Service and Prevention

Denial of Service (DoS) is a cyber-attack on an individual Computer or Website with the intent to deny services to intended users. Their purpose is to disrupt an organization’s network operations by denying access to its users. Denial of service is typically accomplished by flooding the targeted machine or resource with surplus requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled. For example, if a bank website can handle 10 people a second by clicking the Login button, an attacker only has to send 10 fake requests per second to make it so no legitimate users can log in. DoS attacks exploit various weaknesses in computer network technologies. They may target servers, network routers, or network communication links. They can cause computers and routers to crash and links to bog down. The most famous DoS technique is the Ping of Death. The Ping of Death attack works by generating and sending special network messages (specifically, ICMP packets of non-standard sizes) that cause problems for systems that receive them.

How Do DoS Attacks Work?

DoS attacks typically exploit vulnerabilities in a target's network or computer systems. Attackers can use a variety of methods to generate overwhelming traffic or requests, including:

1. Flooding the target with a massive amount of data
2. Sending repeated requests to a specific part of the system
3. Exploiting software vulnerabilities to crash the system

How can denial-of-service attacks be prevented?

Preventing a DoS attack can be challenging, but there are several effective techniques:

- **Network segmentation -** Segmenting networks into smaller, more manageable pieces, can limit the impact of a DoS attack. This can be done by creating VLANs, and firewalls can limit the spread of an attack. The optimal solution is zero trust microsegmentation. Adding device-level and device-cloaking firewalling, external to the operating system remains the most reliable form of DoS protection.
- **Load balancing -** Distributing traffic across multiple servers, a DoS attack can be prevented from overwhelming a single server or resource. Load balancing can be achieved using hardware or software solutions.
- **IP blocking -** Blocking traffic from known or suspected malicious sources can prevent DoS traffic from reaching its target.
- **Rate limiting -** Limiting the rate of traffic to reach a server or resource can prevent a DoS attack from overwhelming it.
- **Content Delivery Networks (CDNs) -** Distributing website content across multiple locations makes it more difficult for an attack to bring down an entire site.

Tools:

Web application firewall (WAF)

Cloudflare

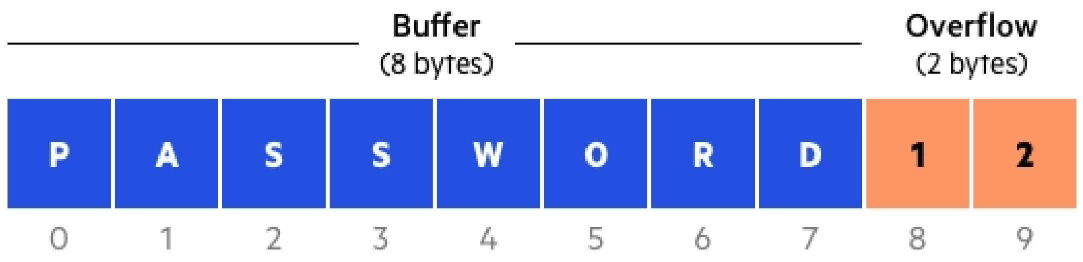
Hping3

What is Buffer Overflow

Buffers are memory storage regions that temporarily hold data while it is being transferred from one location to another. A buffer overflow (or buffer overrun) occurs when the volume of data exceeds the storage capacity of the memory buffer. As a result, the program attempting to write the data to the buffer overwrites adjacent memory locations.

For example, a buffer for log-in credentials may be designed to expect username and password inputs of 8 bytes, so if a transaction involves an input of 10 bytes (that is, 2 bytes more than expected), the program may write the excess data past the buffer boundary.

Buffer overflows can affect all types of software. They typically result from malformed inputs or failure to allocate enough space for the buffer. If the transaction overwrites executable code, it can cause the program to behave unpredictably and generate incorrect results, memory access errors, or crashes.



Buffer overflow example

What is a Buffer Overflow Attack

Attackers exploit buffer overflow issues by overwriting the memory of an application. This changes the execution path of the program, triggering a response that damages files or exposes private information. For example, an attacker may introduce extra code, sending new instructions to the application to gain access to IT systems.

If attackers know the memory layout of a program, they can intentionally feed input that the buffer cannot store, and overwrite areas that hold executable code, replacing it with their own code. For example, an attacker can overwrite a pointer (an object that points to another area in memory) and point it to an exploit payload, to gain control over the program.

How to Prevent Buffer Overflows

Developers can protect against buffer overflow vulnerabilities via security measures in their code, or by using languages that offer built-in protection.

In addition, modern operating systems have runtime protection. Three common protections are:

- **Address space randomization (ASLR)**—randomly moves around the address space locations of data regions. Typically, buffer overflow attacks need to know the locality of executable code, and randomizing address spaces makes this virtually impossible.
- **Data execution prevention**—flags certain areas of memory as non-executable or executable, which stops an attack from running code in a non-executable region.
- **Structured exception handler overwrites protection (SEHOP)**—helps stop malicious code from attacking Structured Exception Handling (SEH), a built-in system for managing hardware and software exceptions. It thus prevents an attacker from being able to make use of the SEH overwrite exploitation technique. At a functional level, an SEH overwrite is achieved using a stack-based buffer overflow to overwrite an exception registration record, stored on a thread's stack.

Cross Site Scripting (XSS)

Cross Site Scripting (XSS) is a vulnerability in a web application that allows a third party to execute a script in the user's browser on behalf of the web application. Cross-site Scripting is one of the most prevalent vulnerabilities present on the web today. The exploitation of XSS against a user can lead to various consequences such as account compromise, account deletion, privilege escalation, malware infection and many more.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.

Example Scenario
You have a simple web page that displays a comment entered by a user.
HTML + JavaScript (vulnerable code)

```
<!DOCTYPE html>
<html>
<head>
  <title>XSS Example</title>
</head>
<body>
  <h2>Leave a Comment</h2>
  <form method="GET">
    <input type="text" name="comment" placeholder="Write something...">
    <button type="submit">Submit</button>
  </form>

  <h3>Your Comment:</h3>
  <div id="output"></div>

  <script>
    // Get the comment from the URL
    const params = new URLSearchParams(window.location.search);
    const comment = params.get('comment');
    // Vulnerable line: directly writing user input to HTML
    if (comment) {
      document.getElementById('output').innerHTML = comment;
    }
  </script>
</body>
</html>
```

Exploit Example
If a user types this comment:
`<script>alert('XSS Attack!');</script>`
Then, the page will execute the script, showing an alert popup.
This is a **Reflected XSS attack** because the malicious input is reflected immediately in the response.

Safe Version (using textContent)
Here's how to fix it:

```
if (comment) {
  document.getElementById('output').textContent = comment;
}
```

This way, the text is displayed as plain text instead of being interpreted as HTML or JavaScript.

Depending on the context, there are ***two types*** of XSS –

1. **Reflected XSS:** If the input has to be provided each time to execute, such XSS is called reflected. These attacks are mostly carried out by delivering a payload directly to the victim. Victim requests a page with a request containing the payload and the payload comes embedded in the response as a script. An example of reflected XSS is XSS in the search field.
2. **Stored XSS:** When the response containing the payload is stored on the server in such a way that the script gets executed on every visit without submission of payload, then it is identified as stored XSS. An example of stored XSS is XSS in the comment thread.

Authentication

In authentication process, identities of the users are verified. Most of the time this verification process includes a username and a password but other methods such as PIN number, fingerprint scan, smart card and such are adapted as well. In order to conduct the process of authentication, it is essential that the user has an account in the system so that the authentication mechanism can interrogate that account. Or an account has to be created during the process.

Access Control

In the process of access control, the required security for a particular resource is enforced. Once we establish who the user is and what they can access to, we need to actively prevent that user from accessing anything they should not. Thus we can see access control as the merger of authentication and authorization plus some additional measures like IP-based restrictions. Most of the time security vulnerabilities in applications stem from inadequate access control mechanisms instead of faulty authentication or authorization mechanisms. The reason why is that access control is more complex and intricate than other two. Main types of access control are RBAC (role-based access control), ABAC (attribute-based access control).

What is MVC?

The **Model-View-Controller (MVC)** framework is an architectural/design pattern that separates an application into three main logical components **Model**, **View**, and **Controller**. Each architectural component is built to handle specific development aspects of an application. It isolates the business logic and presentation layer from each other. It was traditionally used for desktop **graphical user interfaces (GUIs)**. Nowadays, MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects. It is also used for designing mobile apps.

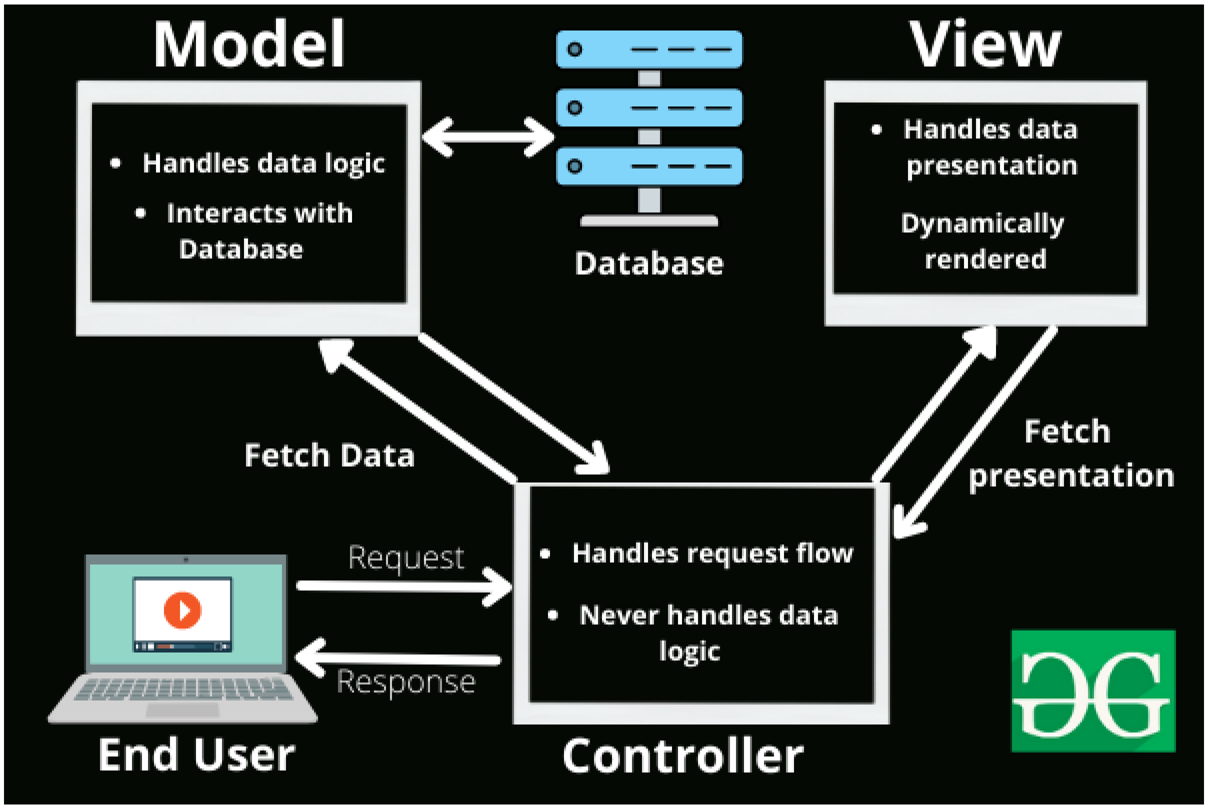
Features of MVC

- It provides a clear separation of **business logic, UI logic, and input logic.**
- It offers full control over your HTML and URLs which makes it easy to design web application architecture.
- It is a powerful URL-mapping component using which we can build applications that have comprehensible and searchable URLs.
- It supports **Test Driven Development (TDD).**

Components of MVC

The MVC framework includes the following 3 components:

- Controller
- Model
- View



https://www.geeksforgeeks.org/mvc-framework-introduction/

*** Session

****Session Hijacking**

**Cookies