PAGE NO.
DATE / /

UNIT-I

Introduction

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 the web pages. Web browser 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 following sections :-

i) World Wide Web (WWW) :- The world wide web is based on a system interconnected public webpages accesible through the Internet. It consist of several components such as HTTP, URL or URI and HTML.

ii) Web Browsers :- The web browser is an application software to explore WWW. It provides an interface between the server and the clients and requests to the server for web documents and services.

iii) Web Server :- A web server is software and hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web.

The Main job of a web server is to display website content through storing, processing and delivering webpages to users.

iv) Web Page: A Web Page (or webpage) is a document on the web that is accessed in a web browser. A website typically consists of many web pages linked together under a common domain name.

v) Web Development: Web development refers to the building, creating and maintaining of website. 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 is classified into two ways:-
Frontend Development: The part of a website where 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 part of the website that users cannot see and interact with. It is used to store and arrange data.

Protocols
A Protocol is a set of rules. Protocols allows two computer to communicate over media such as wireless or hardwired technologies.
When computers communicate with each other there need to be a common set of rules and instructions that each computer follows.

Some types of Protocols :-

Hyperteat Transfer Protocol (HTTP): This protoc is used to access, send and receive Hypertext markup Language (HTML) files on the internet.

Simple Mail Transfer Protocol (SMTP): This protocal is used to exchanging email messages between servers. SMTP is an application layer protocal.

File Transfer protocal (FTP): FTP is used to upload files on server and download files from server.

Transmission Contral Protocal (TCP): This protocal ensure the delivery of information packets across network.

Internet Protocal (IP): This protocal is responsib for logical addressing called IP address

to route information between network.

Web Development strategies
Effective Web development strategies are essential for creating websites that not only look great but also function seamlessly, engage users and achieve desired goals.
Some strategies are:-
Responsive Web Design: Ensure your website adapts flawlessly to different screen sizes (desktop, tablet, mobile) for optimal user experience.

Performance Optimization: Prioritize fast loading times through image compression, code minification and efficient resource management.

User Centered Design: Focus on understanding user needs and preferences to create intuitive and enjoyable interfaces.

Security Measures: Protect your website and user data with robust security practices, including firewalls, encryption and regular updates.

Cross-Browser Compatibility: Test your website throughly across various browsers and devices first to ensure consistent performance.

Mobile first Development: Design and build for mobile devices first, optimizing for larger screen later.

Content Management and SEO: Create high-quality relevant content and optimise it for search engines to improve visibility.

Web Applications
Web Applications are interactive software applications that run on a web server and are accessed via a web browser. Unlike traditional desktop applications, web apps do not require installation on the user's device.

Examples:
- E-commerce Platforms: Websites like Amazon, eBay.
- Social Media: Facebook, instagram and TWITTER.
- Webmail: services like Gmail and Yahoo mail.
- Online Banking: secure portals for managing bank accounts and transactions.

Web Projects
A Web project involves the development and deployment of a website or web application. It encompasses several phases such as:-
- Planning
- Designing
- Development
- Testing
- Deployment

- Maintainance
- Documentation
- Marketing

Web Team
A Web Team consist of professionals with different skills working together to develop a Web project.

Key Roles:
- Web developer: Responsible for Coding and building the website.
- Web Designer: Focuses on the visual and user interface aspects.
- Project Manager: Oversees the project, ensuring it meets deadlines and stays within budget.
- Content Writer: Creates and Manages the content for the website.
- SEO specialist: Ensure the Website is optimized for search engine.

Writing Web Projects
Writing web projects involves creating detailed documentation that outlines the requirements, design and development Process.

Components:
- Project Proposal: A document outlining the project's objectives, scope, and methodology.
- Technical specifications: Detailed descriptions of the technologies and tools to be used.
- Design Documents: Wireframes, mockups, and design prototypes.
- Development Plan: A step-by-step guide to the coding and implementation process.

Identification of Objects
In web development, identifying objects involves determining the key components and entities of the web project.

Key aspects:
- Data Models: Defining the structure of data to be used in the application.
- User Roles: Identifying different types of users and their permissions.
- UI Components: Buttons, forms, and other interactive elements on the website.
- Content Elements: Text, images, video, and other media to be included.

PAGE NO.
DATE / /

Target Users

Target users are the primary audience for whom the web project is being developed. Understanding the needs and behaviors of target users is crucial for creating a successful web project. By focusing on target users, developers can ensure that the web project is not only functional but also relevant and engaging.

Target audiences are defined by demographics such as age, gender, and location, as well as psychographics like values and Behaviors.

Planning and Process ~~Management~~ Development

Planning and Process development are crucial for the successful creation of any web applications or website. It involves a structured approach to define project goals, target audience, functionalities, and the overall development process.

Key steps:

- Project Planning: Defining the scope, schedule and resources.
- Risk Management: Identifying potential risks and creating migitation strategies.
- Process Development: Establishing a workflow for design, development and Testing.

PAGE NO.
DATE / /

- Monitoring and Control: Regularly reviewing progress and making necessary adjustments

05/05 M. Sodiq 1418 2024

HTML

↳ Used to provide structure of the webpage.

There are various Tags in HTML.

CSS

↳ Cascading Style sheet

↳ It is used to provide styling.

↳ RGB (for providing colors)

(255, 0, 0) — RED

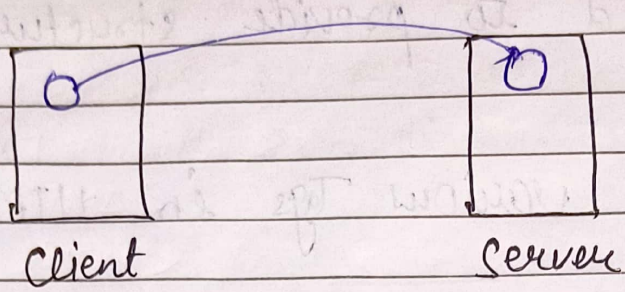
8 bit

(0-255)

Total no. of possible colors = $256^3 \simeq$ 16 million

PAGE NO.
DATE 09/09/24

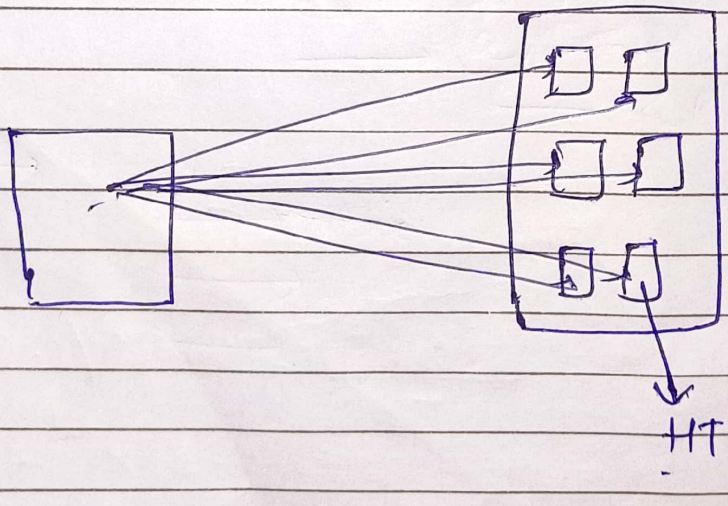
HTTP



HTTP
↓
1.0 → Stateless Protocol
1.1 → Persistance Protocol
→ Non-persistance

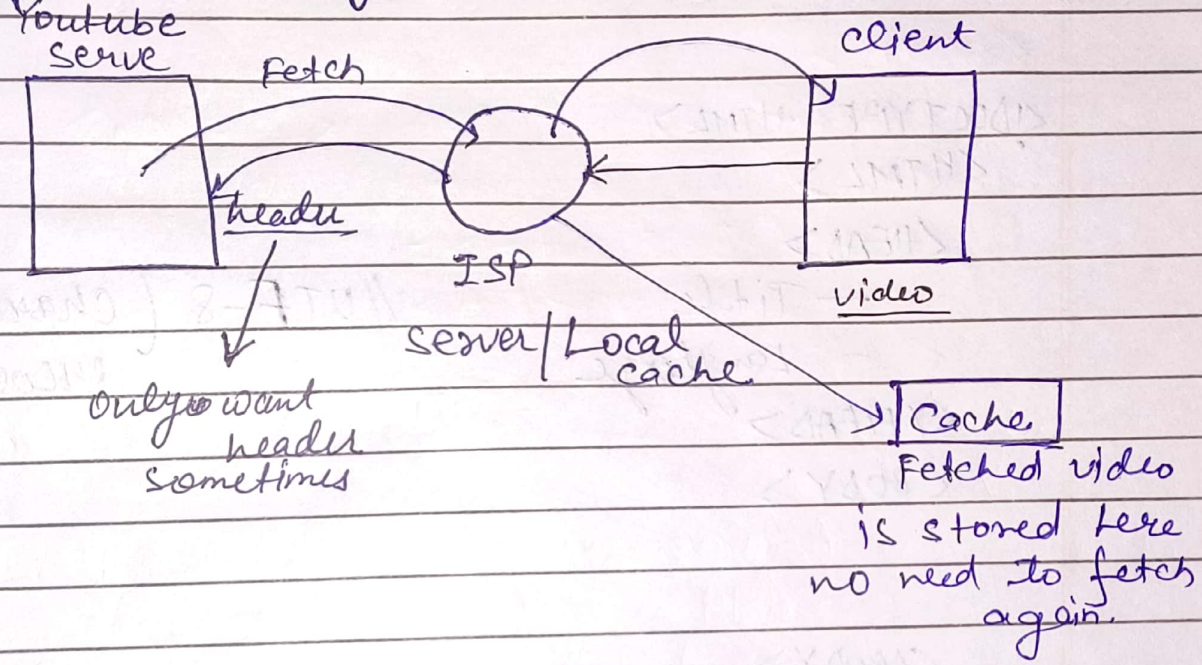
Methods of HTTP:-

- → GET
- → PUSH
- → TRACE
- → POST
- → OPTION
- → DELETE
- → PUT



PAGE NO.
DATE / /

HTTP Working



PAGE NO.
DATE / /

Web Technology

Web Technology refers to the tools, frameworks, programming languages, protocols and standards used to create, deploy, and manage websites and web applications.

HTTP

Hypertext Transfer Protocol is the foundation of data communication that defines how data is transmitted between a client and a server. It enables communication by defining rules for the request and response process.

Features of HTTP

1. Stateless - Each http request & response is independent of Previous ones.
2. Request Reponse Model
   - A communication b/w client and server follows a request-response cycle.
3. Media Independence
   - HTTP can transfer any type of data (text, images, videos, files) as long as both client and server agree on the data format.

HTTP Workflow

1. Client makes a Request
   The browser (or client) sends an HTTP request to the server, specifying the resource it wants to access.
2. Server Processes the Request
   The server receives the request, processes it,

and prepares the appropriate response.

3. Server sends a response
The server sends back an HTTP response, which include a status code, headers and optionally the requested content.

4. Client displays the content
The browser interprets the response and displays the content to the user.

HTTP Methods
HTTP methods are verbs that define the type of action to be performed on a resource. They specify the intent of the request and help the server understand what the client wants to do.

i) Get : Retrieve data from the server without modifying it.

ii) Post : Submit data to the server to create or process a resource.

iii) Put : Update or create a resource at the specified location.

iv) Delete : Removes a resource from the server

v) HEAD : Retrieve only the headers of a resource, not the body.

PAGE NO.
DATE 12/09/24

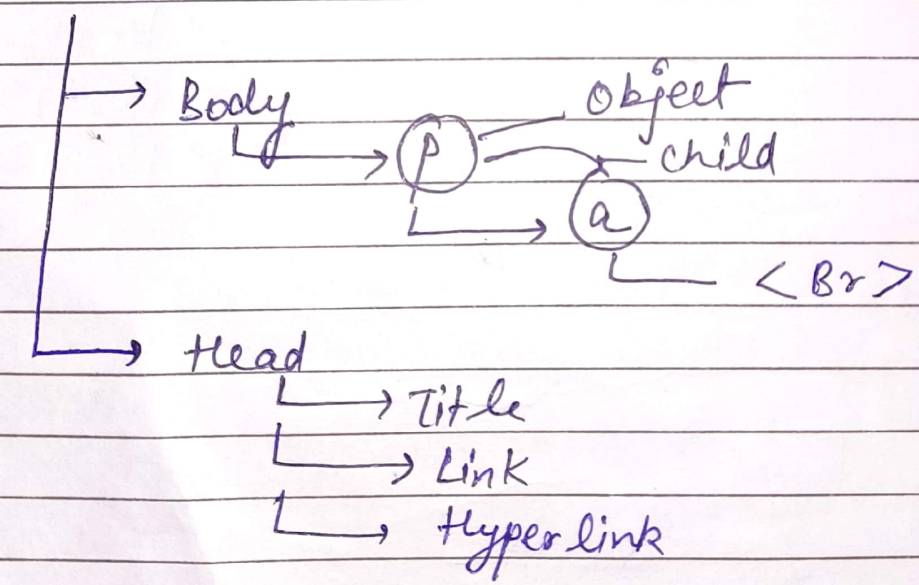
Unit -IT

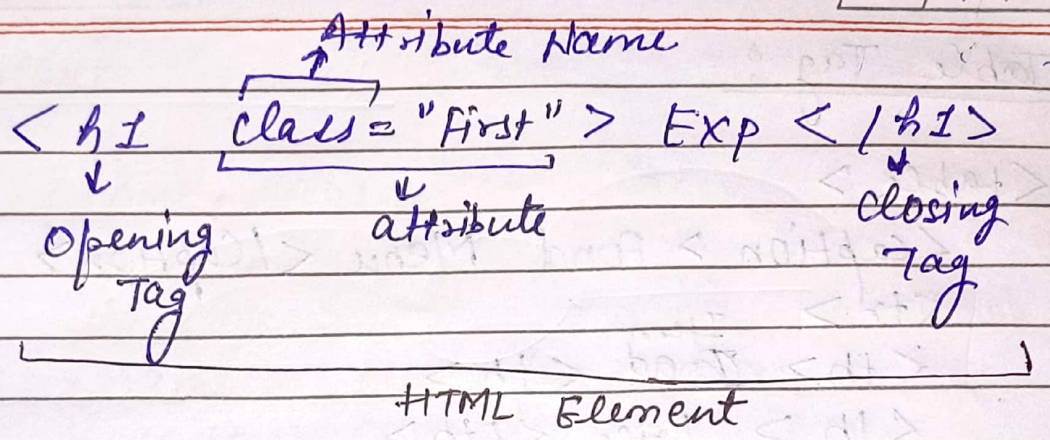
```
<!DOCTYPE HTML>
  <HTML>
    <HEAD>
      - Title
      - Language
    </HEAD>
    <BODY>


    </BODY>
</HTML>
```

// UTF-8 (character encoding)

DOM (Document Object Model)




Attribute Name
<h1 class="First"> Exp </h1>
Opening Tag
attribute
closing Tag
HTML Element

PAGE NO.
DATE 25/09/24

Table Tag :

```
<table>
    <caption> Food Menu </caption>
    <tr>
    <th> Item </th>
    <th> Price </th>
    </tr>
    <tr>
    <td> Frooti </td>
    <td> 12 </td>
    </tr>
    <tr>
    <td> Mangoes </td>
    <td> 25 </td>
    </tr>
</table>
```

<thead> : Header wrapping
<tbody> : Body wrapping
<tfoot> : Footer wrapping

PAGE NO.
DATE / /

XML

```
<? Xml version = "1-0" ——>
[ Employees >
   <Employee>
    <name > tuzaifa <|name >
    <salary> 50000 <|salary>
     <|Employee>
<|Employees >
```

⇓

DTD ⇒ Document type definition

- Syntax - Structure
- Semantics - Meaning of that thing

Employee . DTD

```
<!ELEMENT   Employees (Employee) >
<!ELEMENT   Employee (name, salary) >
<!ELEMENT   (#PCDATA)>
<!ELEMENT   (#PCDATA)>
```

XML schema (.xsd)

```
<? xml  version = "1.0"  ——— >
<complextype  name = "Employee">
 <sequence >
     <name  type = " string " >
     <salary  type = " Integer " >
 <|sequence >
<| complex Type >
```

PAGE NO.
DATE / /

Java Script

Program to check Prime Number :

```
Function isPrime (num) {
    if (num <= 1)
        return false;

    for (let i = 2; i <= Math.sqrt(num); i++)
    {
        if (num % i == 0)
        {
            return false;
        }
    }
    return true;
}
let input = Prompt ("Enter a Number");
let number = ParseInt (input);

if (isPrime (number))
{
    Console.log ("Prime Number");
}
else {
    Console.log ("Not a Prime Number");
}
```

PAGE NO.
DATE / /

Operator in JS

① Arithmetic operator
+, -, /, *, %, **

② Assignment operator
=, +=, -=, *=, /=, %=, **=

③ Comparison operator
>, <, ==, >=, <=, !=, ===

④ Bitwise operator
&, |, ~, ^, <<, >>
AND, OR, NOT, XOR, LEFT SHIFT, RIGHT SHIFT

⑤ Logical operator
&&, ||, !

Unary operator
a++, b--.

Binary operator
a+b

Ternary operator
(Condition) ? (Exp 1) : (Exp 2);

a = 5
b = 2

Max = (a>b) ? a : b;

Write a program in JS to sort array using function.

```
function sort (arr) {
    const n = arr.length;

    for (let i= 0; i <n-1 ; i++) {
     for (let j= 0 ; j <n-1; j++) {
          if (arr[j] > arr [j+1]) {
                const temp = arr [j];
                arr [j]= arr[j+1];
                arr [j+1]= temp;
          }
     }
    }
    return arr;
}

const number = [4,2,3,6,8]
const sorted = bubblesort (number);
console.log(sorted);
```

PAGE NO.
DATE / /

DHTML → Element Selector

Document Object Model

```
document.getElementById(id);
document.getElementByTagName(name);
document.getElementByClassName(name);
document.getqueryselector();
```

PAGE NO.
DATE 24/10/24

Unit III

JS

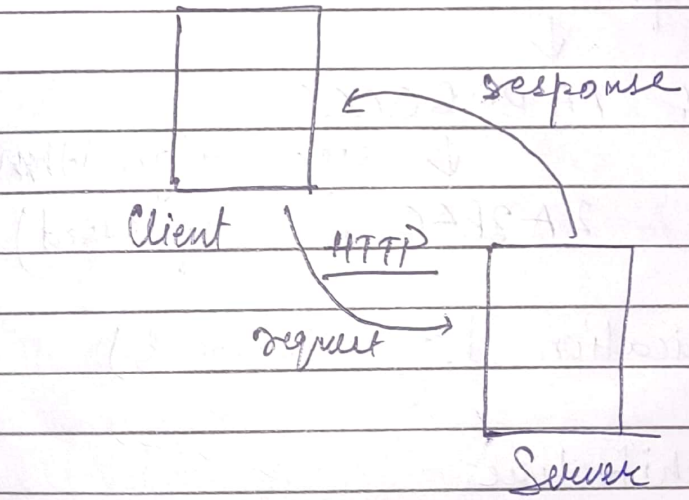
node → Javascript engine

V8 → Result

web engine (execute the js program and show its output)

Scripting languages :-

Python
Perl
VB script
Java script



PAGE NO.
DATE / /

Web server

Server is a computer that has CPU, storage device, etc.

Webserver is a system connected with internet that provides functionalities:-

i) Hosting Website
ii) Error handling
iii) Load Balancing
iv) Handling Dynamic Content
v) Handling Security
vi) Caching
vii) Handling the HTTP request
viii) Data Compression

↓

eg: AA BB B CCCC

↓ used algorithm

2A 3B 4C (compressed)

HTTP (Application Layer Protocal)

Multiter Architecture

MVC

[Model]

[View]

[Controller]

PAGE NO.
DATE 01/11/24

Encryption

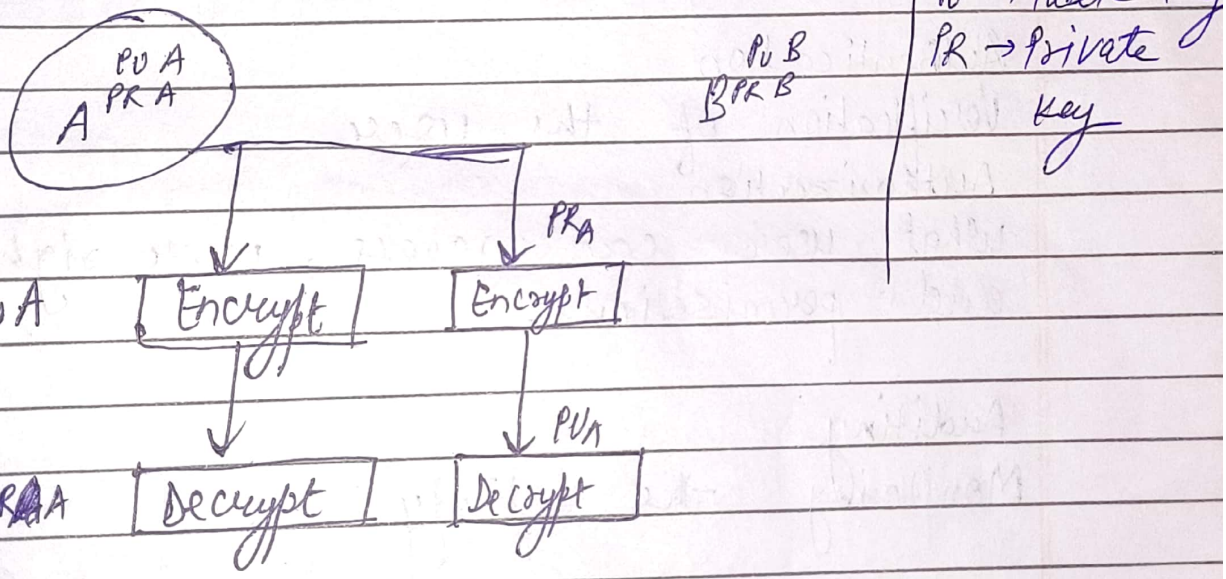
AABBBC (Plain Text)

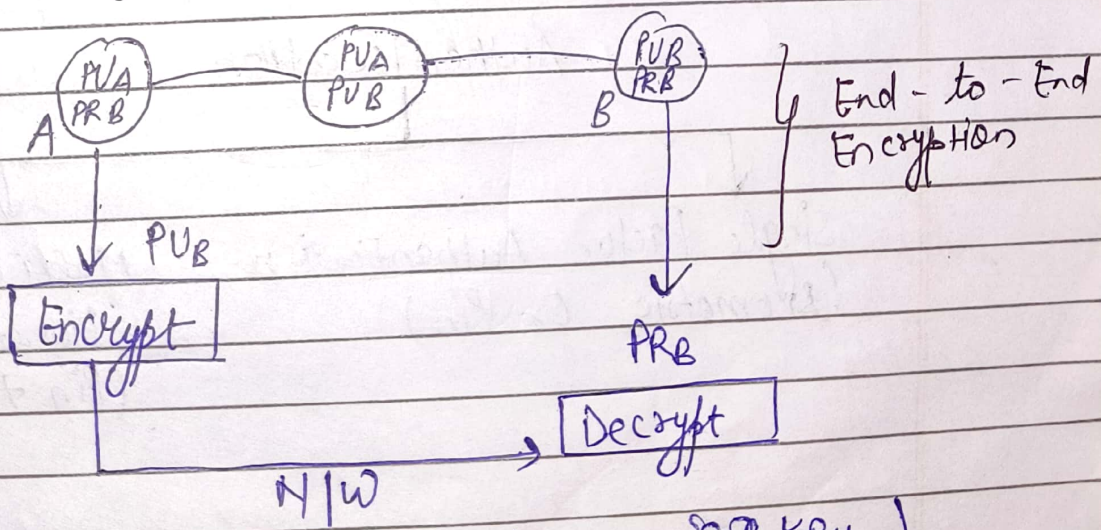
↓

BBCCCD (Cipher text) → encrypted information

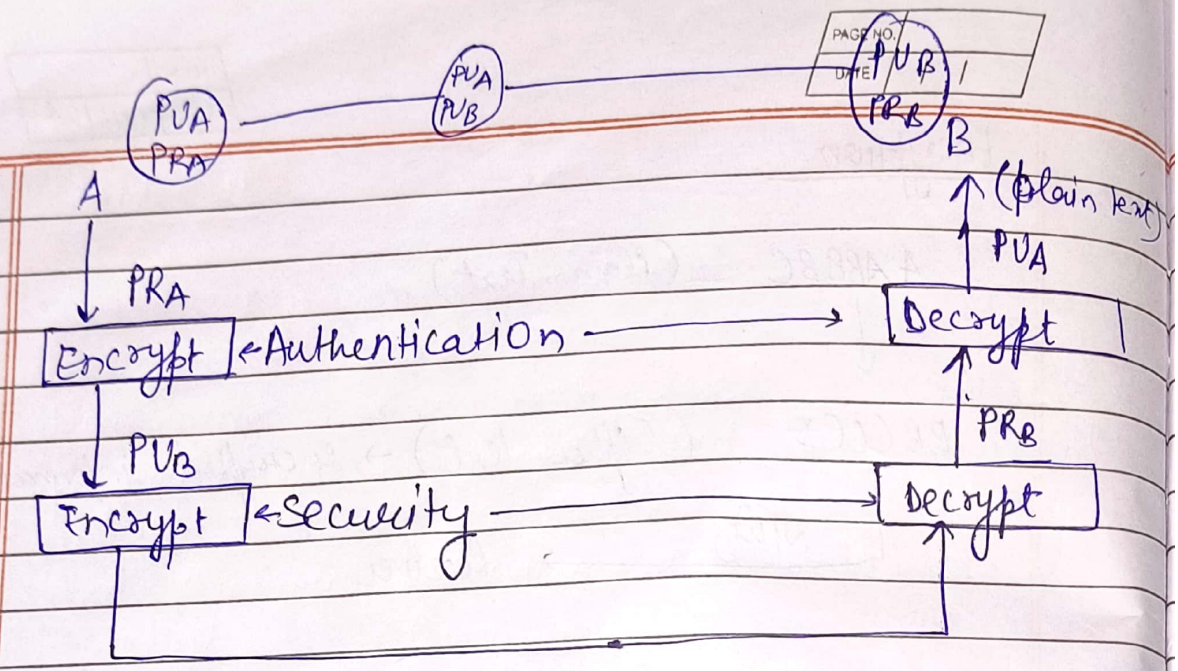
N/W → Receiver

Pu → Public key
PR → Private key





(NO need to exchange key)



Authentication
Verification of the user

Authorization
What user can access, user rights and permissions.

Auditing
Monitoring the activity

Authentication

- Single Factor Authentication (Biometric Or Pin)
- Multi Factor Authentication
  (Biometric + Pin)
  (Pin + OTP)

PAGE NO.
DATE / /

Proxy Server

Refers to a server that acts as an intermediary b/w the request made by client, and a particular server for some services or request for some resources.

- ⇒ Maintain Cache
- → Hide IP address

PAGE NO.
DATE / /

Unit - III

Introduction to JavaScript

JavaScript is a high-level, versatile programming language that is primarily used to add interactivity, functionality, and dynamic behavior to websites.

Developed by Bredan Eich at Netscape in 1995.

JavaScript is a scripting language, meaning it is often interpreted and executed directly within a web browser.

Scripting Languages

Scripting lang. are programming languages designed to automate tasks, manipulate data, or control other software applications.

Scripting languages are interpreted at runtime (unlike programming language that are compiled), this means that they execute code line by line.

Common scripting languages are:-

- Python
- JavaScript
- Perl
- Ruby
- PHP
- VBScript

PAGE NO.
DATE / /

Javascript Engine

Javascript Engine is a program or interpreter that executes JavaScript code. JavaScript engine are responsible for parsing, compiling and executing JavaScript code.

i) Parsing: Analyse the syntax and breaking it down into an internal representation.

ii) Compilation: Javascript engine used techniques called Just in Time Compilation. Rather than compiling the entire code before execution, JIT compilation compiles the code in real-time.

iii) Execution: Once the code has been parsed and compiled into machine code, the engine executes it.

Examples of these Engines are:-

- V8: Used in chrome (By Google)
- SpiderMonkey: (By Mozilla) used in firefox
- Javascript Core (Nitro): (By Apple) used in Safari Browser.
- Chakra: (By Microsoft) used in older Edge Browser.

Object in JavaScript

In JavaScript, an object is a collection of ~~Key~~ key-value pairs, where each key (also called a property name) maps to a specific value.

Syntax:

```
let objectName = {
    key1: value1,
    key2: value2
};
```

Example:

```
let person = {
    firstName: "John",
    lastName: "Doe",
    age: 30,
    greet: function() {
        console.log("Hello," + this.firstName +
                    " " + this.lastName);
    }
};
```

(1) Accessing object Properties

i) Dot Notation

```
Console.log(person.firstName); // John
```

ii) Bracket Notation

```
Console.log(person["lastName"]); // Doe
```

(2) Adding and Modifying Properties

```
Person.country = "USA"; // Add
person.age = 31;        // Update
```

(3) Deleting Properties

```
delete person.age;
```

(4) Method in Object

```
person.greet(); // Hello, John Doe
```

PAGE NO.
DATE / /

(5) Nested Objects

```
let car = {
    make: "Toyota",
    year: 2022,
    owner: {
        firstName: "Mohd",
        lastName: "Huzaifa"
    }
};

console.log(car.owner.firstName);
```

Object Properties and Methods

i) Object.keys(obj)

ii) Object.values(obj)

iii) Object.entries(obj)

iv) Object.assign(target, source)

Event and Event Handling

What is an Event?

An event is an action or occurence recognized by the browser that can be handled by JavaScript.

Examples of events include:

- User Actions: click, mouse movements, etc.
- Browser Actions: Page reload, resizing, etc.
- Other Actions: form submission, etc.

Event Types

1. Mouse events:
   - click
   - dblclick
   - mousemove
   - mouseenter | mouseleave
   - mousedown | mouseup
2. Keyboard events:
   - keydown
   - keyup
   - keypress
3. Window events:
   - load
   - resize
   - scroll
4. Form events:
   - Submit
   - Change
   - input

PAGE NO.
DATE / /

5. Touch events
- touchstart
- touchmove
- touchend

Event Handling:
Event Handling is a fundamental concept in JavaScript that allows developers to control and respond to user interactions, such as clicks, key presses, etc.
Event Handling refers to the process of defining what should happen in response to these events.

Concepts in Event Handling:-

1. Event Listeners:
- An event listener is a function that waits for a specific event to occur and then runs in response to that event.
- JavaScript uses the addEventListener method to attach an event listener to a specific element.

2. Event Handlers:
An event Handler is the function that gets executed when an event occurs. The handler function can be written directly in the addEvent-Listener method or defined seperately and then passed to it.

Listeners: Listen for a specific event type
Handler: Actual function that is executed when an event occurs.

PAGE NO.
DATE / /

3. Event Propagation

→ Event Propagation is the process by which an event travel the DOM from one element to another.

→ It has two phase:

- Capturing Phase: The event travels from the root of the DOM to the target element.
- Bubbling Phase: The event ~~travels from~~ bubbles back up from the target element to the root.

→ Developers can control event propagation using stopPropagation() to stop an event from moving further up or down the DOM.

4. Event object

- When an event occurs, JavaScript creates an event object containing details about the event, such as the type of event, the target element, and co-ordinates.

Attaching Event Listeners:

There are several ways to attach event listeners:

i) Inline Event Handlers
ii) Event Handler properties
iii) Using addEventListener

PAGE NO.
DATE / /

Document object Model

The DOM is a programming interface for web documents. It represents the structure of HTML or XML document in a way that JavaScript (and other languages) can manipulate.

Document :

The DOM starts with the document object, which represents the entire HTML document. Through this document object, JavaScript can access all elements within the page.

Objects and Nodes :

The DOM represents the document of a tree of nodes. Each HTML element, attributes, and piece of text in the document is represented as a node.

Tree structure

The DOM arranges HTML elements in a Hierarchical tree structure, starting ~~dtor~~ from the documents' root.

Methods to Access Elements in the DOM

i) By ID

```
document.getElementById(id)
```

ii) By Class Name

```
document.getElementsByClassName(className)
```

iii) By Tag Name
document.getElementsByTagName(tagName)

iv) By CSS selector
- document.querySelector(selector)
- document.querySelectorAll(selector)

v) By Name Attribute
document.getElementsByName(name)

vi) Traversing the DOM Tree:
- element.parentNode
- element.children
- element.firstElementChild
- element.lastElementChild
- element.nextElementSibling
- element.previousElementSibling

PAGE NO.
DATE / /

Unit 4

(1) Authentication

Authentication is the process of verifying the identity of a user or system before granting access to resources or performing actions. In web applications, authentication typically involves users providing credentials, such as a username and password, to confirm their identity.

Key Concepts of Authentication:

1. User Credentials:
   - Username: A unique identifier for a user.
   - Password: A secret key known only to the user.

2. Session Management
   Once authenticated, a user may be assigned a session that allows them to remain logged in without having to re-enter their credentials for every request.

3. Token-Based Authentication
   In modern web applications, especially those using APIs, authentication often relies on tokens (e.g., JWTs) rather than traditional session IDs.

There are two primary types of authentication:

i) Single Factor Authentication
Simplest form of authentication. It requires only one method of verification to grant access to a

system or Application. This ~~using~~ usually involves Something the user knows.

ii) Multi-Factor Authentication

Multi-Factor Authentication enhance security by requiring two or more distinct forms of verification before granting access. MFA combines different categories of authentication factors, making it much more difficult for unauthorized users to gain access.

MFA requires two or more of the following factors:

- Something you Know (password, pin, etc)
- Something you Have (sms codes or OTP)
- Something you Are (Biometrics - fingerprints, iris)

(2) Authorization

Authorization is the process that determines what an authenticated user is allowed to do. It defines ~~perst~~ permissions and access levels for resources within a system.

Methods:

i) Role-based Access Control (RBAC): Users are assigned roles that dicate their access levels.

ii) Attribute-based Access Control (ABAC): Access is granted based on user attributes, resource attributes, and the environment.

→ Roles (admin, user, guest)

→ User Attributes (Role: "HR" Manger, location: "Delhi", Department: "HR")

- Document Attributes (Type: "Personnel File", Sensitivity level: "High").

(3) Auditing

Auditing is the process of logging and monitoring user actions and system activities. It involves tracking who accessed what, when, and what actions were taken.

Methods:

i) Log Files: Storing records of user activities and system events.

ii) Monitoring Tools: Using software to analyze logs for suspicious activities or compliance checks.