Comment :- /* */

☆ Web - Interviews

• Boolean attributes

① Disabeled → assign to from input examples (disabling fields)

```
<input type="text" disabled />
<input type="text" />   ▨  □
```

② <meta charset="utf-8" />

↳ this element specifies documents character encoding - the character set that document is permitted to use.

③ description meta tag

```
<meta name="description"
      content=" - - - - - - - - "/>
```

④ description list

```
<dl> → description list
   <dt> aside </dt>
    <dd>
        — — —
    </dd>
    <dd>
        — — —
    </dd>
 </dl>
```

⑤ < time datetime="2016-01-20"> 20 January 2016 </time>

⑥ API (Application Programming interfaces) are constructs made available in programming languages to allow developers to create complex functionality more easily. they abstract more complex code away from us, providing some easier syntax to use in its place.

- early days of the web, sites have used cookies to store information to personalize user experience on websites. they are the earliest form of client side storage commonly used on web.

in modern time cookies are used to store data related to user personalization (eg. session IDs, access tokens).

web storage API :- provides a mechanism for storing and retrieving smaller, data items consisting of a name and a corresponding value, this is useful when you need to store some simple data like users name, whether they are logged in, background color of screen etc

- sessionstorage :- this persists data for as long as browser is open (data is lost when browser is closed)
- localstorage :- persists data even after the browser is closed and then opened again.

(1) ex:- localStorage.setItem("name", "Chris");
↳ takes two parameters, name of the item and its value

(2) localstorage.getItem("name")
↳ takes one parameter, name of the item to gets its value

(3) removeItem("name")

An HTTP cookie is a small piece of data that a server sends to a users web ~~web~~ browser. browser may store the cookie and send it back to same server with later request.

→ keep
→ cookies are
① session
② person
③ Trac

we can

→ HTTP

the
→

②

comment :- /* */

→ an HTTP cookie is used to tell if two requests come from same browser— logged in.

→ cookies are mainly used for three purposes:-

① session management → logins, shopping carts, game score server should remember

② Personalization :- user preferences, Themes and other settings.

③ Tracking :- recording and analyzing user behavior.

→ we can create cookie using ~~are~~ access cookie in javascript using

Document.cookie

protocol used to access data on world wide web.
used to transfer data in form of audio, video, document.

→ HTTP request methods

① connect :- method starts two-way communication with the requested resource. It can be used to make a connection.

→ connect method can be used to access the website that uses SSL (HTTPS)

② DELETE :- request method deletes the specified resource.

→ if a delete method is successfully applied, there are several response status codes possible:-

① 202 (accepted) but not enacted

② 204 (no content) → enacted but no further inform^n

③ 200 (ok) → enacted & response message.

③ GET :- request a representation of the specified ~~co~~ resource. requests using GET should only used to request data (they shouldn't include data)

④ patch :- method applies partial modification to a resource. — considered a set of instructions on how to modify a resource.

⑤ Post :- ~~method sends data to the servers~~ type of body of the request is indicated by content-type header.

- a post request is typically sent via an html form and results in change on server. content type is :

* SASS: mature, stable and powerful css extension language.

- it is completely compatible with all versions of css. we take compatibility seriously, so that we can seamlessly use any available css libraries.

* Git

① to create a new git repository

```
git init
```

② checkout a repository

```
git clone /path/to/repository
```

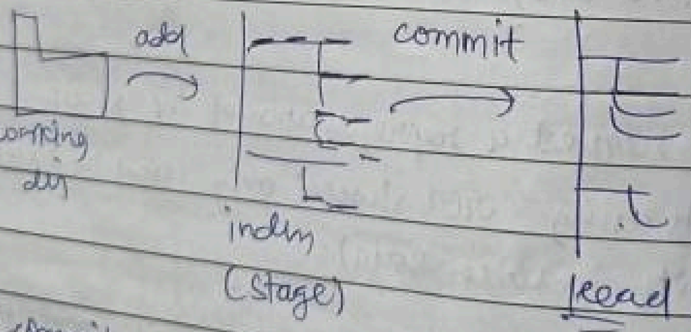
workflow

your local repository consists of three "trees" maintained by git.

first one is ① working directory → holds actual files.

② Index → acts as a staging area

③ Head → points to last commit made



③ add & commit

```
git add * or git add <filename>
```

here we are propose changes and adding it into the index

④ to actually c

→ now

⑤ pushing chan

those chang

* update &

to u

to me

an

with

* tagging

this is

th

you

* Sh

Comment :- /* */

④ to actually commit these changes use

```
git commit -m "commit message"
```

→ now file is committed to head, but not in your remote

⑤ pushing changes → your changes are now in head. to send those changes to your remote repository

```
git push origin master
```

* update & merge

to update your local repository to newest commit, execute

```
git pull
```

to merge another branch into your active branch

```
git merge <branch>
```

and after changing, you need to mark them as merged with

```
git add <filename>
```

* tagging: It is recommended to create tags for software releases. this is a known concept. you can create a new tag named 1.0.0 by executing

```
git tag 1.0.0 1b2e1d63ff
```

the 1b2e1d63ff stands for the first 10 characters of commit id you want to reference with your tag.

* Shows the status of your working directory

→

```
git status
```

→ display a history of commits

```
git log
```

→ `git branch` → list all the branches

→ `git remote -v` → shows connected remote repositories.

→ `git diff` → shows the differences btw your working directory and the last commit.

★ Git & Github

* Git → It is a version control system, like a smart record-keeper for computer code.
→ It helps developers track changes made to their code over time.
→ Imagine it is a series of snapshots of your code at different points in its history.

→ Github on other hand, is a web platform that uses Git. Its like a social network for code. developers can use github to store their git repositories online and collaborate with others.

* Git is the tool you use on your computer to manage your codes history
→ Github is a website where you can store and share your git repositories with other.
- Its widely used by developers, teams and organizations for collaborative software development and code management.
- Github makes it easy for multiple developers to work on same project simultaneously. wt can create branches for different features or bug fixes and then merge them back into the main ~~co~~ codebase.

* github offers Sc
analysis, and
security issues i

Q difference btw H
⇒ Http is a stan
uses browser
Https uses e
making it mo
details.

Q:- box model
⇒ CSS box mod
page. It a

Q:- responsive
⇒ well on
consistent

Q AJAX in
⇒ web page
full page
user e

Q:- What i
⇒ A rest
it us
resour
→ (Simp
in i

comment :- /* */

* github offers security features like vulnerability scanning, dependency analysis, and code scanning to help you identify and address potential security issues in your code.

Q:- difference btw http and https?

⇒ Http is a standard protocol for transmitting data between a web server and a web browser while a https is a secure version of http.
Https uses encryption to protect data transmitted between server and browser. making it more secure for sensitive information like login & payment details.

Q:- box model in css?

⇒ css box model is a way to represent layout & sizing of elements in a web page. It consists of content, padding, border and margin.

Q:- responsive web design? → approach that makes web pages look and function
⇒ well on various devices and screen sizes. Its important because it ensures a consistent user experience across different platforms.

Q:- AJAX in web? ⇒ (Asynchronous javascript XML) is a technique that allows
⇒ web pages to send and retrieve data from a server without requiring a full page refresh. It enables asynchronous loading of content, enhancing user experience by making web application more responsive & interactive.

Q:- What is a Restful API & differ from SOAP API?

⇒ A restful API is an architectural style for designing networked applications. It uses standard Http methods (GET, POST, PUT, DELETE) to interact with resources, and returns data in JSON or XML format.

→ (Simple object access protocol) is a protocol for exchanging structured information in implementation of web services.

web fundamentals

① URL:- stands for uniform resource locator. a url is a web address used to locate a specific resource on the internet.
Ex:- https://www.example.com/index.html is a url that points to the index.html page on the "www.example.com" website.

② DNS → it stands for (domain name system) DNS is a system that translates the domain names into IP addresses, allowing browsers locate web servers.

③ web:- collection of electronic resources, implemented by tim-berners-lee who recognised as a father of web.

④ Network:- collection of interconnected computer link together to exchange resources between them. first network in industry was ARPANET developed btwn two computer using FTP

⑤ SMTP:- simple mail transfer protocol. takes care of delivering mails between different domains,

⑥ MIME:- multipurpose internet mail extension. care of transfering different kinds of data through electronic mail services.

⑦ FTP:- stands for file transfer protocol. transfer files between two or more connected computer over networks.

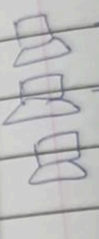
⑧ web page:- A page than can be opened through webbrowser and developed in html.
There are two types of web pages:-
① static:- user unable to Interact directly made in html & css
② Dynamic:- user able to interact directly.

* website:- collection
① static:- web p
② Dynamic: both

* server:- the soft
is called server

* client:- software
from server

* client server co



client1
client 2
client 3

multiple cli
server thro

— the comm
it consist
① Http re
a link

* Types o
① G

Comment :- /* */

website :- collection of web pages. There are two types:-

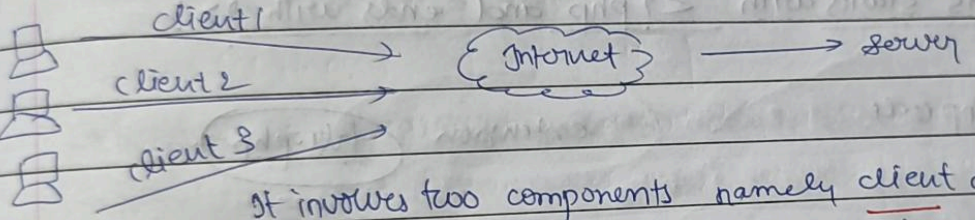
① static :- web page developed using only client side technology

② Dynamic :- both client and server side.

* server :- The software that distributes the information on client request is called servers. provides information to client.

* client :- software that fetches information, process it and display it from server known as client.

* client server communication



It involves two components namely client and server. usually, multiple client requests to a single server. Client sends a request to server through internet and server respond to client through internet.

– the communication btw client and server typically occurs using HTTP
It consists of two main components:-

① Http request :- whenever a user write url on browser address bar and does on a Cmle browser send httprequest to server.

* Types of http requests :-
① GET, ② POST, ③ PUT & ④ DELETE

PHP

- PHP is an acronym for "Hypertent preprocessor"
- PHP is a widely used, open source scripting language
- PHP scripts are executed on the server.
- PHP is free to download and use.
- PHP files can contain text, HTML, CSS, Javascript & PHP code
- PHP code is executed on server and result is returned to the browser as plain html.
- php can generate dymaic page content
- php can collect form data.

* A php scripts starts with <?php and ends with ?>.

```
<?php
  // php codes
?>
```

extension (file.php)

Ex:-

```
<?php
  echo "Hello world";
?>
```

→ (if, else, while, echo) this statements are not case sensitive
function, classes, user-defined fn

→ all variables names are case-sensitive
→ variables are start with "$" sign.

* // → this is a single line comment
  # →
  /*
  */ } → multiple line comment

→ → variable name cannot start with a number
→ a variable name can only contain alpha numeric characters & underscores (A-Z, 0-9 and)
→ case sensitive.

* php has3 diffe
①local ②
whe
we want to a
+
use sta

print "
print "

* arrays in

→ in php

* ①one
②
③ m

<form
Nam
E-ma
<in
</for

Comment :- /* */

* php has 3 different variable scope :-

① local ② global ③ static

↓

when a function is executed, all of its variables are deleted. sometimes we want to a local variable not to be deleted.

↓

use static keyword

static $X=0;

print " <h2> '''' </h2>";
print " ";

data types

① String, ② Integer, ③ float, ④ boolean, ⑤ Array, ⑥ Object, ⑦ NULL, ⑧ Resource

* arrays in php

```
$cars = array("Volvo", "BMW", "Toyota");
var_dump($cars);
```

→ in php array() is used to create array.

* ① Indexed arrays :- array with numeric index
② Associative arrays :- array with named keys
③ multidimensional arrays :- arrays containing one or more arrays.

```
<form action="welcome.php" method="post">
  Name: <input type="text" name="name"> <br>
  E-mail: <input type="text" name="email"><br>
  <input type="submit">
</form>
```

→ in html

welcome <?php echo $_POST["name"]; ?>
your email address is: <? php echo $_POST["email"]; ?>

```
if($_SERVER["REQUEST_METHOD" ...]){
  if(empty($_POST["name"])){
    $nameErr = " name required ";
  }
  else{
    $name = test_input($_POST["name"]);
  }
```

★ Cookies in php

A cookie is created with setcookie() function

Syntax :-

setcookie (name, value, expire, path)

→ only name parameter is required.

★ Session

A session is started with session_start() function.

session variables are set with PHP global variable

: $_SESSION.

```
<?php
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "session variables are set";
?>
```

```
echo "favourite color is". $_SESSION["favcolor"];
```

for modifying just overwrite it

comment :- /* */

① validation :- of login page

```
    <label for="username"> Username </label>
  <input type="text" id="email" name="username" require >
            "
          "

const registrform = document.getElementById('        ');

    const username = document.getElementById('username').value ;
    if (username.trim () === '') {
          errors.push('username is required');
    }

    if (email.trim() === '') {
          errors.push(' email required');
    }else if ( !isValidEmail(email )){
           errors.push(' Invalid email format');
    }

function isValidEmail (email){
      const emailpat= /^[a-zA-Z0-9._]+@[a-zA-Z0-9._]
```

↳ this is client side validation.

★ client side & server side validation

— we can create a robust validation system that ensures high integrity and security of your data also providing a good user experience.

- server side validation is more secure, but it can add latency to form submission process.

- it can also increase development complexity.

Backend (server side)

→ we first get form data from request body. If all the validations are pass we save user data to database. after that we generate authentication token for user and return it to user.

* SSH?

SSH stands for Secure shell. It's a keys are pair of cryptographic keys used for secure communication and authentication between a client and a server over network.

- SSH keys provide a more secure and convenient way to log in to remote servers.

① public key:- part of SSH key that you share with server you want to access. It's safe to share because it cannot be used to compute private key.

② Private key:- keep on your local computer. used to authenticate & decrypt message sent to you.

→ SSH key are more secure than password based authentication because there is a chance of brute-force attacks.

Uses:- Server Administration < web hosting
cloud infrastructure.

→ Database management

→ secure file transfer

* JWT → JWT stands for JSON web token used for securely transmitting information. it is often used for authentication & authorization pro web applications & APIs.

→ JWT consists of three parts:-

① Header:- contains metadata about token

② payload:- include user information, permission & relevant data

③ Signature:- to ensure integrity of token, a digital signature created using a secret key.

usage:- Authentication, authorization, information exchange

What is web security ? Comment :- /* */

[illegible] to protecting website, web application and web service from various threats, vulnerabilities, and attacks that can harm the confidentiality, integrity & avalability of data and service on web.

① vulnerability assessment :- this can be done using automated scanning tools & manual testing

② security measures :- encryption to protect data, firewalls and intrusion

cookie: a cookie is a small piece of data sent from web server and stored on user device. cookies are typically key value pair that contain information about user, their session data. they have expiration date

→ cookies are primarily used for tracking user behavior, maintaing user sessions and personalizing user experience.

Session: session is a server side mechanism for maintaining state and data about users interaction with websites. each user is assigned a session ID, which is stored in a cookie. session data is store on server.

- sessions are used to manage complex user interaction and store sensitive data securely.

Examples of cookies: ① Shopping cart :- cart contents are store, your selected item remaining to cart.

② Rember me on login :- a persisten cookie is set on your device to remember you login credentials.

③ Language preference

Example of session: ① User authentication :- when we login to banking application, a session created to maintain your authentication state

② Shopping with a user account

③ online gaming

★ Encryption & decryption :- fundamentals used to protect sensitive data by converting it into a secure (unrealable format) and then converting it back into to its original format.

→ encryption process of converting plaintext → ciphertext using keys.

use in → secure communication, file storage system

→ decryption process of converting ciphertext back into plaintext using decryption key

Message (whatsapp) (encrypted)

Symmetric & Asymmetric encryption

Uses:- e-commerce, data privacy, secure communication, data storage, secure file sharing.

* various hashing fn:-

~~converting data into a fixed for hash value. this hash are unique representation of input data and have several~~

① SHA (secure hash algorithm):-
SHA1 generates 160 bit (20 byte) hash value, it was use for ~~digital signature & certificates~~

② SHA 256 → generates a 256 bit (32 byte) hash value. provides security than SHA1. used in digital signatures, ~~certificates authentication~~ & password Blockchain technology, ~~bitcoin uses~~ SHA-256

③ bcrypt → designed for securely hashing passwords. storing user passwords securely in database.

* which is fast ~~react or~~ javascript or jquery ?

→ Javascript is a programming language, while react & jquery are javascript is foundation of both react and jquery, so they are equally fast in terms of performance.

→ react uses a virtual DOM which is a copy of original dom used to calculate difference b/w current state of dom and original.

→ ~~Argument: react is considered~~ to be faster than Jquery for & complex applications. because react's virtual doms help reduce no of times real dom replaced.

→ if we are developing small application → Jquery is a good choice.

* what is ip address → stands for internet protocol address a unique numerical label assigned to each device connected to a computer network that uses internet protocol for communication.
IPV4 → 32 bit, IPV6 → 128 bit

* can we ~~y~~ use git without github ?

→ yes we can absolutely use Git without github. Git is a distributed version control system, and it can be used independently of web based platform like github. as Git is a command line tool and doesn't require any service like github.

Git uses:-
→ local version c
→ initializing repo
→ commit change
→ hosting own

where does git st

→ Git stores the da your project f for your project

→ objects:- act
~~refs~~ : cont
head :- poin
config:- con
description:

→ gitignorefile →
ignore when

* what is sa

→ salt is a r of a cryptog two
→ pass

* site is emp

→ ① optimize d
② content d
sites & c

* top or htop
ps → pr
du → du

→ kill the

→ pkill &
their a

Comment :- /* */

Git uses :-

→ local version control, track changs in projects,
→ initialing repository using git init.
→ commit changes to make commit your local repository
→ hosting own git server

where does git store data ?

→ Git stores the data in a hidden folder called .git in root directory of your project. this folder contains all of metadata & object database for your project.

→ objects :- actual data of your object,
refs : contains references to diff branches
head :- point to current branch in project
config :- contain configuration settings for your project.
description : description of your project

→ gitignorefile → text file that tells Git which files or folders should be ignore when tracking changes to your project.

*** what is salt in hashing ?**

→ salt is a random data that is generated at and added to input of a cryptographic hash functions the purpose of salt is to prevent two identical pieces of data from producing same hash value.
→ password encrypt

*** site is experiencing high traffic**

→ ① optimize database queries :- indexing help in improv efficiency.
② content delivery optimization :- optimize & compress images to reduce their sizes & load times. Implement browsing cache to reduce

* top or htop → provides a real time view of system resource usage.
ps → provide details of running process
du → du command use to know about disk usage of a directory.

→ kill the process or restart deamon, uninstall or clean up

→ pkill & killall → commands allow you to terminate process by their attributes.

→ Javascript is an interpret[illegible] Javascript engine in browser
→ react is interpreted by

* how should you take backup of your server ?

→ ① choose a backup method.

① full backup :- all data on server

② incremental backup :- backups data that has changed [illegible] last full backup.

③ Differential backup :- this method only backups data [illegible] changed since last increneetal

② choose a backup destination

① local storage :- same server or local network

② Remote storage :- more secure, on cloud.

③ schedule regular backups

④ ~~s~~ test your backups.

* wordpress → wordpress is an open source content manag[illegible] system that allows user to create and manage websites [illegible] blog. its based on PHP and uses a MySQL databa[illegible] store content.

* theme in wordpress → collection of templates & stylesheets that determine visual appearance & layout of a website.

* widgets → small block contents added to various websi[illegible] such a sidebar, footer sections

* closures :- cl[illegible]
even after th[illegible]
closure allow[illegible]
function out[illegible]
var
fu[illegible]
ret[illegible]
}
var cl[illegible]
closur[illegible]
→ basically
→ used fo[illegible]

* Event loop
like netw[illegible]
→ it ensur[illegible]
tasks co[illegible]
→ so ba[illegible]
an as[illegible]
it add[illegible]
→ it will
be en[illegible]

* hoisting
eve[illegible]
varia[illegible]
some[illegible]

Comment :- /* */

* **closures** :- a closure is a function that captures variables from its outer function, even after the outerfunction has finished enecuting.
closure allow inner fn to acces & manipulate variable from outer fn.

```
func outerFunc() {
    var name = "John doe";
      function innerFunc() {
          console-log (name);
    return innerFunc
}
var closure = outFunc();
closure();
```

In enample, innerfunc() has the acces to name even though the outerfunc() has finishes enecuting. this is because innerfunc is a closure,

→ basically closures are used to reusable fn.
→ used to create private variables,

* **Event loop** :- how code is enecuted, when it involves task that take time, like network requests reading file, or waiting for userinteractions.
→ It ensures that javascript remains non-blocking and can handle multiple task concurrently.
→ so basically Javascript enecuting code line by line., when it an encounter an asynchronous task, like fetching data from server or settimeout, it add that tasks to a queue starts continues with next line of code
→ it will check if task is completed in queue once it is completed it will be enecuted.

* **hoisting** :- It moves all the variable and function declaration on top of their scope even if They are not declared in code. this means that we can use a variable or function before it is declared, as long as it is declared somewhere in same scope.

```
console.log(a);   // → undefined
var a = 'bar';
```

Variable is simply declared and undefined until it is assign value.

* Event bubbling :- when an event occurs on a particular element it also triggers the same event on its parent elements.

- so we are doing event at multiple levels.

Ex:- we have a div

```
<div id="parent">
    <button id="child"> click me</button>

</div>
```

→ we added a click event listener to both parent and child elements. when we click on button click me first child element [illegible] then the parent element will be clicked. because of both are present in same div.

** → same event must occur in both child & parent div.

* Scope :- where variables and functions are declared and accessibility of it. It defines where in your code a particular variable or function is visible and can be used.

* Shadowing :- occurs when a variable declared in local scope has the same name as a variable declared in global. local variable shadows the precedence over outer variable.

```
const x = 10;
func    x(){
      let x = 5;
    console.log(x); → 5
}
x()
console.log(x); → 5
```

* Promise → repr[illegible]
handle asynch[illegible]

→ you call a [illegible]
gives us a [illegible]
→ we are cont[illegible]
it by phone[illegible]

Comment :- /* */

Promise → represents a completion of an operation. promises are used to handle asynchronous code in a more organized way,.

→ you call a restaurant (create a promise) and order food. restaurant gives us a (order number) (promise).

→ we are continuing doing own task while waiting for delivery. we don't need to sit by phone or check. if food is ready, as we trust promise.