

## Full Stack Internship

01.2025  
Exclusive tags in HTML5  
xheader \* footer + nav + audio + aside + time + source  
video + bold

Why HTML5 is need:

It enhances the structure of the web page.

Assignment completion with HTML5 tags, CSS pre requires and tail wind CSS.

JS can be used as client side and

JS is one of the powerfull programming language equivalent to C, C++, python.

React and Angular JS are popular frameworks or libraries of JS.

Why console is need and what is its importance.

Data types:

\* let \* var \* const

When we use console.log the output is displayed in console part only.

Value of var and let can be changed but const can't be changed.

Do example program for each of the following

1. simple if
2. If else
3. Else if
4. Else if ladder
5. nested if

case 1

No. of lemons in hand 7.

Expected output : Good 1 offered

Good 2 need 7

Good 3 need 7.

14.

shortage = 14.

Case 4.

No. of lemons in hand 67

Good 1 7 offered

Good 2 7 offered

Good 3 7 offered

46

Surplus = 46.

Case 2.

No. of lemons in hand 21

Output.

Good 1 7 offered

Good 2 7 offered

Good 3 7 offered

sufficient.

Case 3

No. of lemons in hand 15

Good 1 7 offered.

Good 2 7 offered

Good 3 + offered having 1 need 6

shortage : 6.

Sam is having 75 candies. Sam gives half of it to angle. Since angle loves & sam a lot she gives back half of its portion portion. Calculate how many candies finally angle have. Use 1 variable. Use functions. 1 line program..

## E86 Extended internal version of JavaScript

07.01.2025

git  
Open 2025

Create day wise folder.

Day + folder

Generator  $\Rightarrow$  a folder  $\rightarrow$  content, function.

Assignment:

1. Tailwind CSS.

Console.log:  $\Rightarrow$  take value and print it.

# It is used for debugging.

We can track the program flow.

We can add the # in body, outside body, and head also. It works in all the places. But we have to add it accordingly.

## E86.

\* Extended internal version of JS.

\* It is known as ECMAScript.

\* Arrow function is under this.

\* Added filter in JS.

## Arrow function:

\* It does not have function name.

document.getElementById("response").innerHTML = localStorage.getItem("response")

object

method

property

It is from ES6 for efficiency increasing of code and to increase the readability and to increase we can create functions without name and it is called as arrow function.

Element is HTML element.

Response is its name.

1. Design a simple calculator by getting 2 numbers as input. Display addition, subtraction, product, quotient, remainder by creating individual arrow functions for the same.
2. Create an array by treating array size and array elements from the user. Extract all the perfect numbers and even prime numbers from the array.

## Object-Oriented Programming Structure (OOPS)

Example :

Class : Bird.

Objects : Parrot, Peacock, Duck, Dove, Crow.  
Properties : Color, wings, size, legs, weight.  
Methods (behaviours) : Flying, eating, singing, -

### Js Objects

in-membership operator.

Inbuilt methods :

Keys

Values

Entries

## J.S Promise

- \* Promise is a J.S Object.
- \* There are 2 states for Promise.
  - \* Resolved (Success)
  - \* Rejected (Fail)

Note:

- \* Invoking a function
- \* Call back.

success => then.

failure => catch.

Q.01. ~~Ans~~  
Write a Promise called Andra-bp and 3 functions persona, personb and personc. persona-reached, distance andrata = 5000, andra-b = 2000, andratc = 1000  
Expected Output.

c reached

b reached

a reached.

### Promises Inbuilt-methods:

When there is more than one promise in order to review them we can use we can use promise in-built methods according to the requirement.

#### Methods

- \* Promise.all
- \* Promise.any
- \* Promise.allSettled
- \* Promise.race

### Promise.all:

If there are 3 promises - If the promise 1 is false then it won't check the promise 2 and promise 3. If the

Once it sees a promise false it will stop.

### Promise.any:

Gives the shortest time promise provided states should be true. If all the promise where false then

~~Promise.allSettled~~<sup>Aggregate Error</sup>: All promise were rejected.

will display 1 among 3 states are

\* fullfilled

\* rejected

\* Pending (result mostly in API).

### Promise.race:

Closure

## Reactjs.

It is a library or framework of JS.

Example: Netflix, Amazon.

Example websites for JS are: YouTube and Wikipedia.

To see the nodejs version in cmd: node -v

CMD commands.

npm create-react-app appname.

npm start

cd demo

Code.

\* Create a folder in desktop as React

\* Double click the folder and go to the address bar and type "cmd".

\* Now we are in command prompt.

Commands:

1. node -v (check node version).

2. npm -v (check version of npm)

3. Creating the react apps using the following command.

3. npm create-react-app appname.

4. npm start

\* Open your app at port 3000 (default port is 3000)

\* Open your app react folder your app folder will see your app folder got created.

To open my VS code.

Go to command Prompt.

5. cd demo => cd appname.

## b. code :

\* Now VS code will open. In that click terminal and new terminal and it will be open.

In the VS code command terminal

7. npm start

8. npm i web-vitals

There are many folder. In that 2 folders are important

\* public and \* source (src)

index.html is the base file. It is present in ~~public~~ folder.

React is a open source.

3 important files

\* index.html

\* index.js

\* index.css

Note:

\* At first don't touch index files.

Note:

\* Initial do or write your code in App.js.

DOM:

React follows V-~~DOM~~ (Virtual DOM)

They are unlike html, once dom get created the changes or manipulation what we do gets completed and only that part will be re-rendered.

Where as in html every time we make change entire DOM will be re-rendered.

In web application created by react js each one every thing is called as **Component**.

### Types of component

- \* Functional component
- \* Class component

J&X

Writing HTML inside JavaScript.

09.01.2024

### Direct export

Direct export import module in App.js.

import { Rough } from './Rough';

and render it in return statement.

\* Direct export will work in both arrow and de traditional functions. (7)

### Props and States

#### Props

\* Every component will have props and states.

#### Props:

\* It won't change. Example : Brand: Tata's biscuitary

#### States:

\* It changes or we can change it.

Example : Water level in bottle :

Initial state full.

updated state: half

Current state: empty.

Component is phone.

props are appno, vivo, etc. {Brand}

state is website:

home page, grocery, mobiles, fashion  $\Rightarrow$  component  
website (Component name)

props: name, version, brands.

state: discount.

Dictionary is not in fs. It is Objects of keys & values

Create 2 files

## React hooks.

Earlier in st industry they were using class components.  
Presently state constant was not available with functional component.  
Now hooks is used to implement state in functional component.

## Types of hooks

use state

use effect

use ref

use context

use Reducer

Best example for useState is: counter clock  
stating the initial state as zero we can incremented,  
decremented, reset using useState.

10.01.2025.

function concatinat() {

one =

two =

if ~~one~~ ~~two~~

3:

L - - - D -

spread Operator

11.01.2025.

useEffect hook

Upon the condition one action we apply in our functional components monitoring the impact or side effects can be done using useEffect hook.

useEffect accepts 2 arguments first argument is call back argument. (Call back function is a function accepts an argument which is another function).

In useEffect-1.js, const interval = setInterval(() => {} ) is a call back function.

useEffect accepts 2 arguments:

Call back function

Dependency array (optional)

Call back function is like constructor in Java.  
Constructor in Java will automatically get triggered  
in Java. Call back function is like that in  
useEffect.

20.01.2025

Let say we have 5 components  $c_1, c_2, c_3, c_4$  and  
 $c_5$ .  $c_1$  child is  $c_2$ ,  $c_2$  child is  $c_3$ ,  $c_3$  child is  $c_4$  and  
child is  $c_5$ . Every component should display a message  
its name as message.

When ever we are using something inside {}, it can be  
either js Object or react component.

Props can be passed between component only by following  
the hierarchy which means parent to child.  
To overcome this in terms of efficiency we are  
using hooks.

Conclusion:

If we want to pass state from 1 component to another  
component, the only way to archive is passing it  
as props in the hierarchy.

This is not efficient to make it efficient we have  
exclusive hook called useContext.

4 components

- 1st App.js
- 2nd Container
- 3rd User
- 4th user

## useContext hook.

Without following the hierarchy passing state from another component in an efficient way using is hook.

### 2 important things

- \* Create context

- \* useContext

In the given example create context will be done in App.js and that will be used in User.js using useContext

8

Synchronizing a component with external system  
After all action monitoring or checking (seeing) the side effects happening in the functional component is possible using useEffect hook.

11.01.2025.

## useReducer hook

It is same as useState to manage or update state that is data that is values of components.

Difference is if you have more states or complex things you are useReducer hook.

Step:1 Create increment decrement program  
Step:2 Replace useState with useReducer.

It takes 2 arguments reducer function and initial value.

↓  
which says what you want to do (like + & -)

It returns an array with 2 values like useState.

initial value & dispatch function.  
(state & dispatch).

Here state will hold initial value and updated. once you dispatch function and dispatch will trigger useReducer function

29.01.2025

## Redux

Slices: It has 2 parts, action and reducer

Dispatch function takes 1 argument action.

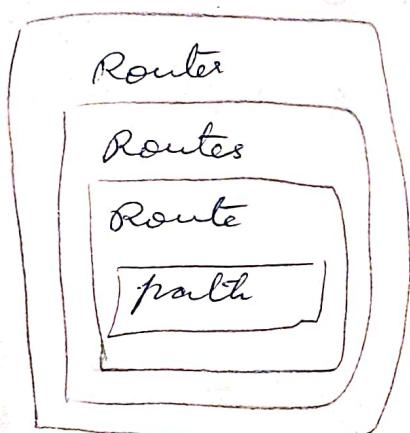
npm i @reduxjs/toolkit react-reduce => and contains install redux.

react-reduce => package.

@reduxjs/toolkit => is library package from redux

To create

## Router architecture



Home needs about contact.

After connection is done.

To access the

## Mongodb

It is no-sql. We use no-sql in order process unstructured data. like ex: json file.

json file is nothing but a javascript object.

## Compass

Compass is a graphical user interface. It helps to

Compass is an interface helps us to reach mongodb server.

which means we fetch data from mongodb server (client)

## Mongosh

Mongo shell was replaced by with mongosh.

### Data Modeling

2 terms are important in mongodb.

\*Data modeling      \*Schema

Data modeling is fix structure of your data planning the structure.

### Schema

It is an actual blueprint of DB which you created by fixing the format using data modeling.

Ex: Employee details, salary float, emp id int, etc.

## NoSQL

Documents

Collections

Ex: MongoDB

## SQL

Records

Tables

Ex: MySQL

## Command Prompt Commands.

use aiml.

db aiml.

You can create dB using

Data Base creation: use database-name.

Ex: use aiml.

use empPersonal.

Show dbs

db.emp.insertOne({ename: "Uma"})

db.emp.insertOne({name: "Abi", age: 19})

db.emp.insertOne({name: "Saritha", age: 19})

db.emp.find({name: "Uma"})

db.emp.find() => display all records.

cls => only screen is cleared.

db.emp.insertOne({name: "Cheertha", age: 19, like: "Sport", fav: {game: "football", player: "Ronalds"} })

db.emp.insertMany

db.emp.find

db.mycollection.updateOne({ename: "John"}, { \$f

Create a db Computers. One collection laptops  
name, model, color, status, price, vendor field (vn, vp)  
available or  
not available.

25.01.2017  
Data base : Computers, collection is Details

schema id number, name string, age number, city string.  
Multiple conditions using and & or  
db.products.find({\$and:[{\$price:{\$gt:80000}}, {\$brand:'apple'}]}) => satisfy all the conditions.

db.products.find({\$or:[{\$price:{\$gt:80000}}, {\$brand:'apple'}]}) => satisfy any one of the conditions.

empty, or null hobbies.

db.customers.find({\$and:[{\$hobbies:{'\$exists':false}}, {\$age:{\$gt:40}}]}) => null hobbies and age is greater than 40.  
To satisfy both conditions.

aggregation is group by.

db.companies.aggregate([{\$lookup: {from: 'employees', localField: '-id', foreignField: 'company-id', as: 'Employees'}}])

group by.

in directly the result in new collection as Employees.

Create a db called Bank 2 collection under that customer, personal, customer account.

Data model for

name, address, phonenumbers, age.

string array object

1, 2.

Data

accountnumber, branch, bank name, IFSC code, current

integer string string string (alpha)

balance, accounttype.

0

saving or current.

1. Filter only OO category yes.
2. Display only the customer's address where the branch names are same.
3. current balance 10,000 to 20,000 filter only the other.
4. filter only the savings account people collection, complete details.
5. add (update) field called status (some & those who are having some ITSC code).

### Node.js.

It is Backend library from JS.

In node we can use express JS as middle ware

required => is a key word from node.

const http = require('http'); => Built-in module

const express = require('express'); third-party module

const sayHello = require('./greet'); custom module

27.08.2025

node filename => run or start server.

npm start => run client terminal

node non =>

### node.js.

It is a popular environment which allows us to run at the server side.

Maintain split terminals (2 terminals)

client and server in JS code in order to use

Always start the server first.  
node filename

Run command.

Command  
npm start

Activate express. 1st we have to install express.

npm i express

server response in json file.

~~we~~

we can delete package.js to get it back - the command  
npm init -y

Run script is present in package.js.

If any error that shows run script error. we want to  
delete the package.js and using the above command

client console is present in inspect. of that browser.

server console is present in terminal it is present in below  
the terminal.

app.name: nodeapp; folder: server; filename: server.js.

Express is an application used for routing, middleware management  
and many reasons to keep routing clean.

Get is one of the request method ('/')

Define a port number the server will listen on const port = 3000

req represents the request object, and res represents the  
response object.

(Browser)

request is from client side and response is from server

Browser asking the server (so it is request) to give me

Hello, node.js! its giving it back as response.

Browser port 3000 is client.

req belongs to client and res belongs to server.

app.get('/', (req, res) => { } will send a response with text "Hello, Node.js!" .  
start the server and make it listen on the specific port.  
The call back function runs when the server starts successfully.  
app.listen(port, () => { }) by a message indicating that the server is running and provide the URL to access it.  
get & port are https method or requests.

To manage or handle the https request. **Axios**  
It is a popular js library. It is used to make http requests from the browser or nodejs.  
It is known for easy and clean syntax also flexibility especially works well with APIs and req's.

When we write an API for exclusive purpose it is called as rest APIs.

### CORS

### Cross Origin Resource Sharing.

When a web page request information from resource (from any other sides) whether to accept the request and process it or not will be defined in a rule book for this purpose we use cors.

npm i ~~cors~~ axios

npm i cors

npm i -g nodemon.

In the given example we are requesting data from the server (Hello this is data from server)

Using data component.js as client, <sup>using</sup> http get method by a API/data

Server response as json  
from the json file I want to filter only the message so we are using from cors response data message.

We can fetch the data in the json file only in the form of API's

or <sup>use</sup>  
Inside src create 3 files, Users.js, Createuser.js, Updateuser.js  
model: name, age, email

Router, axios, express, cors

folder Server run init

In index.js which is inside server folder write backend code.