

6/1/2025

FULLStack development

HTML 5:

Tags exclusive tags:

- \* header
- \* nav
- \* aside
- \* video
- \* audio
- \* bold
- \* time
- \* square.

HTML 5 is used to enhance the structure of the

navigation - nav:  
The home page or about us page is will  
the place it will show. move to the place.  
that is called navigation - nav.

Front-end:  
what we have seen in the website  
or web page.

<nav>:  
nav tag is used to when we click the  
home or aboutus or contactus it will move  
to the that page or place it is called  
navigation.

Weeks for the day.

\* Complete the 3 assignments.

HTML 5 TAGS

CSS - styles

Padding

Layout.

Three pages.  
navigation.

+ Tailored CSS:

JavaScript:

service site:

front-end

client site:

back-end

1. JS can be used service site and client site.

2. One of the powerful programming

language . equivalent to C, C++, Python.

3. React and angular is a libraries of the  
Java Script.

4. Popular library or frameworks of the  
JS.

state: non-interactive

dynamic: interactive

Why console? What is the important?

Java script we cannot run directly like html.

We using the html file we add tag:

`<script src="file name.js">`

used to connect js in the HTML file.

Data type:

var num = 100.

var, let, const data type scope.

Output:

→ alert(num) or console.log(num)

→ alert(num, a) It will print only num value.

→ alert(" " + num + " " + b + " " + c)

→ It will display the those values in the alert box.

→ In alert box we using the concatenation is used to display the values in the alert box.

\* const a = 100 It will cannot change.

It is constant.

let, var variable are change.

if, if else, condition statements  
loops

1. do sample programs for each of the the  
following

1. if
2. if else
3. else if
4. else if ladder
5. nested if

Case 1 :

number of lemons in hand = 7

Expected output:

Mod 1 = 7 offered

Mod 2 = need 7

Mod 3 = need 7

Shortage : 14

Case 2 :

No. of. lemons in hand = 21

Expected output:

Mod 1 = 7 offered

Mod 2 = "

Mod 3 = "

Sufficient

Case 3:

No of lemons in hand : 15

god1: 7 offered

god2: 7 offered

god3: having 1 hand 6

shortage : 6

Case 4 :

No of lemons in hand : 67

god 1 : 7 offered

god 2 : 7 "

god 3 : " "

Surplus:  $67 - 21$

$$= 46$$

$$\begin{array}{r} 37 \\ 75 \\ \hline 21 \end{array} \quad \begin{array}{r} 37 \\ 18 \\ \hline 55 \\ 37 \\ \hline 8 \end{array} \quad \begin{array}{r} 37 \\ 18 \\ \hline 55 \\ 37 \\ \hline 8 \end{array}$$

Sam : 55  
angl: 18

Sam is have 75 candies.

ke give

Sam gives half of to angel since angel loves

Sam lot She gives back half of the her portion. Calculate and display how many candy

Sam and Angel have individually.

constraints :

use one variable

o use function only one time without argument  
without return type.

31.5

Angel 18.75

Sam. 56.25

ES6

~~feature~~

- \* Extended features of JavaScript.
- \* arrow function ES6

\* Elma script

\*

6 Jan 2025 folder

git →

two folder => content folder  
handson folder

Content folder handson folder

Notes Picture projects or programs

assignment 1 - Tailwind CSS.

console.

Day 201/1/25

ES6 :

- \* Doma script
- \* arrow function

\* efficient function Reduce the memory

space

variable name not a function  
const howAreYou() => f

return 100;

f

arrow function doesn't have a function name.

Prove.

Arrow function:

=> more efficient function in terms of space

=> its from ES6

=> increase readability

=> we can create a function without name  
and its called as arrow function.

=> document.getElementById("response").innerHTML =  
howAreYou();

document - JS object

getElement - is static method Element HTML  
Element

by id - id name

howAreYou() = call the arrow function.

is variable of the arrow function not  
a function name.

Design simple calculator

By Getting Two number as the input

display addition, subtraction, Product, Quotient  
remainder by creating individual arrow functions

Sample.

Aggregates:

- Create an array
- array size, array elements from the user
- extract all the perfect numbers and even prime numbers from the arrays.

Push : add the element in array at last

POP : delete the element at last in array

Shift : delete the element at front in array

unshift : add the element at front in array  
add multiple values also.

ES6 SPLIT

Q →

var arr = [1, 2, 3, 4, 5, 6]

arr.slice(2, 4)

arr.slice(2, 4)

arr.slice(2, 4)

5, 7, 19, 2, 3, 30

var b = slice(2, 4)

arr(b)

OOPS - object oriented Programming structure.

Encapsulation - access modifiers are used.

Example:

Class: Bird

Objects: One class have multiple objects.

PavoRe. Pigeon. Peacock

Properties: colour, wings, size, legs, weight

Behaviour: (methods): flying, eating, singing

Class: Class is the blueprint of the object.

Object: Building uses using object.

JS Promise:

=> promise is a Java Script object.

=> There core two states for promise.

\* resolved (success)

\* rejected (failure)

=> callback, invoking a function = important in JS

Read JS algorithm, if you want to know more about it.

React use in netflex and

Day-3

8/1/25

## Set time out function

birds  
write a promise called Andrea - BP

Andrea to person A = 5 km 000

people  
Andrea to person B = 2000

Andrea to person C = 1000

output:

C - reached

B - reached

A - reached.

TS. Promise

promise inbuilt methods :

⇒ where more than one function

⇒ in order to review then

⇒ we can use promise inbuilt methods

⇒ according to inbuilt functions requirements

methods:

1. Promise.all

2. Promise.any

3. Promise.haltifit

promise.settled()

promise.resolve()

promise.all():

Once it sees a promise false it will stop.

promise.any(): gives a shortest time promise provided status should true.

promise.allSettled():

will display one among these three states

- 1. fulfilled
  - 2. rejected
  - 3. pending
- } Status

promise.all():

gives a shortest time promise provided status should false. It consider the false when false is declared.

Voice:

=> Netflix is created using React.

Example:

=> HTML is loading fast when we refresh the HTML page.

=> React is loading now when we refresh the page

read JS:

it is a library and framework of

Java Script

Example:

netflix, amazon.

Example:

HTML websites: youtube and wikipedia

>Create a new folder called react in

desktop

Double click the folder Go to address  
Bar type cmd enter.

node -version

commands: ① node -v (check version or check if Intel)

② npm -v (check version of NPM)

③ Creating app using the below command

④ npx create-react-app demo

④ npm start.

Open URL open at port 3000  
it open.

To make it work:

① first open your react folder you will

see app folder was open.

② cd demo

③ code .

④ two important folder in react

→ Public

→ src

These important files in react

→ index.html

→ index.js

→ index.css

Note: HCs of now don't touch index files

Note: do initially do or write your code in app.js.

DOM: VDOM

⇒ react follows virtual dom.

⇒ need unicentHTML Once dom gets created.

The changes are manipulation what we do

Gists completed. Only that part will be re-render.

⇒ We're class in HTML every time we make change entire  
dom will be re-render.

Web application:

⇒ in web application is created by react

⇒ each and everything is called as components

Types of component

\* functional component

\* class component

JSX:

→ HTML inside Java Script

09/11/25 Day - 4  
Props and States

Props: => every component will have Props and States.

Props:

=> it won't change

=> Example: Name TATA, Billeri, Ben

States: => it changes or it we can change it

Example: Water Bottle level,

\* initial state: Full

\* update state: half.

\* current state: Empty.

Flipkart Website:

HomePage :必选

mobiles

Fashion

} Components.

mobiles :

Component name: mobiles

Props: name, version, price

State: offers, discounts, stock available

passing props between components

Create a two files:

One file names is Paragraph JS	parent
Second "	" Child JS.

1. Home.js => one Paragraph

2. about.js => same para or head

3. Contact.js

Earlier in IT industry they use class component  
reason being state concept is not available in  
functional component.

Hooks:  
now Hooks is used to implement state in functional

Component.

Type of Hooks:

=> useState

=> useEffect

=> useRef

=> useReducer

Example:

Best example for useState  
Counter Uak.

Setting initial state also we can implement  
the increment and decrement using useState.

10/01/25

=> Spaced operators

=> Passing array into useState

11/01/25

useEffect:

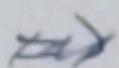
Upon the condition or action we apply in the functional components monitoring the impact of sideeffects can be done using useEffect hook.

useEffect hook:

useEffect accepts two arguments.

=> One is callback function in constructor in

Java



inline CSS:

The style or color added in the HTML line in one line is called inline CSS.

```
<h1 style = "color: red; background: black">
```

```
Hanini </h1>
```

Output:

Hanini

Java Script:

createJS:

Hooks:

⇒ use state is a function take one

⇒ argument that initial state.

⇒ use state returns array [initial state, update state].

⇒ update state is a function.

effect:

use state is also passed as a props.

only in Hierarchical.

20/01/25

1. Totally 5 components.

1. c<sub>1</sub>, c<sub>2</sub>, c<sub>3</sub>, c<sub>4</sub>, c<sub>5</sub> as component names.

2. c<sub>1</sub> props is send to c<sub>5</sub> component.

3. Every component return component1, component2...  
component5.

4. c<sub>1</sub> child c<sub>2</sub>, c<sub>2</sub> is child is c<sub>3</sub>, c<sub>3</sub> is child  
c<sub>4</sub>, c<sub>4</sub> child is c<sub>5</sub>

5. c<sub>1</sub> is a grandparent import in APP.js

6. Every component should display its name as  
message as component1, component2 ...

7. display then wire from h<sub>1</sub> to h<sub>5</sub>.

Output:

Component1

Component2

Component3

Component4

5.

Whenever we are <sup>using</sup> something inside ~~we~~ <sup>we</sup> use Braces {}  
It can be either java script object or React component.

⇒ add the props in c<sub>1</sub> ⇒ message = h<sub>1</sub> from

Component One.

Values can be passed between components only by following the Hierarchy

1. which means Parent to Child.

2. overcome those in terms of efficiency we are using hooks.

3.

use context.

Conclusion: if we want to use state from one to another component

The only way to achieve passing it as in the Hierarchy.

\* This is not efficient to make it efficient we add have exclusive state called useContext

1. APP Component

2. Container "

3. User " "

4. User " "

Without following hierarchy passing state to one component to another component in an efficient way using hooks.

(i) Create Context

(ii) use Context

in the given example:  
Create context will be done in app component  
and use that will be used in user component  
using useContext.

use effect:

useEffect synchronizing A component with  
an external system

After user action monitoring or checking  
the side effects happening in the functional  
component is possible using useEffect hook.

21/01/25

useReducer:

Same as useState to manage or update  
states that is data that is value of components.

Step 1:  
difference is if you have more states  
or complex things you use useReducer hook.

Step 1:  
create implement

→ useReducer takes two arguments  
first argument is reducer function and  
second is initial value of the state.

=> it returns array with TWO values.  
like useState.

=> useState first is that initial count and second is dispatch function

We call them state and dispatch

State and dispatch.

\* State will hold \$ initial value and updated one.. once you called dispatch function. And dispatch will trigger useEffect function

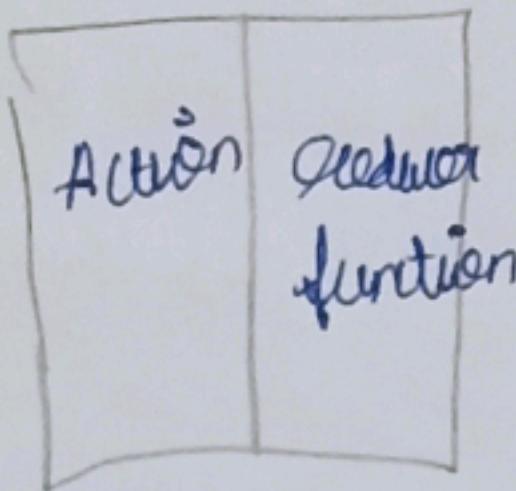
Get the password from the user if password is correct , display the component login is granted . If password is incorrect display access denied component .

22 Oct 25

Redux :

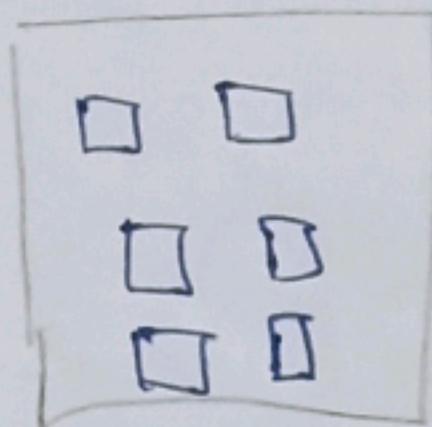
Global manage State

Slice



dispatch

R. S



Redux store

(Google APP STORE)

dispatch  $\Rightarrow$  Trigger the Reducer function .

Installation :

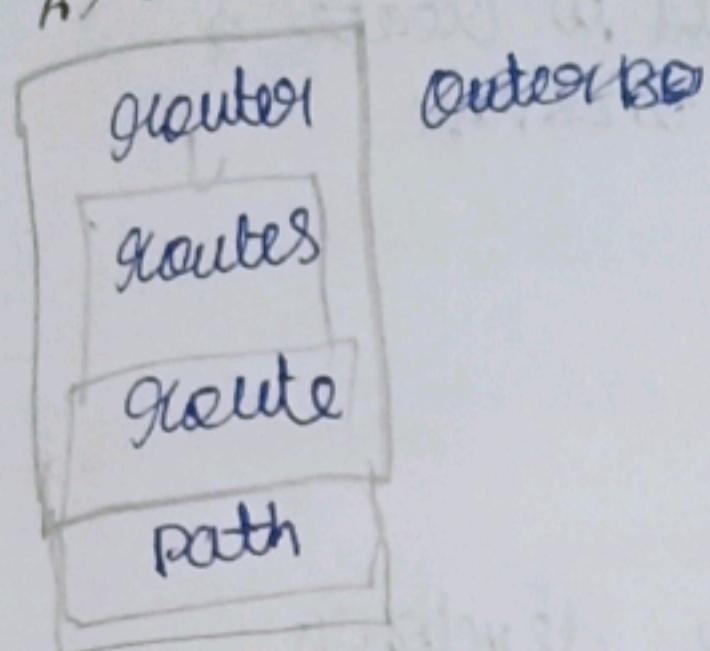
nPM ; @reduxjs/toolkit react-redux

To Create Redux store a  $\text{const slice}$  is stored reduxjs library .

Package : react-redux .

npm i react-router-dom

Architecture



=> It has two things action and reducer.

=> The inner part means

=> Note:

app - folder => store.js

smile - folder => userSlice.js

userinfo is the key for reducers and userReducer is the name we

give for reducer actions we get from userSlice.js.

useSelectionHook:

to access the data from Route.

Bootstrap:

=> is a front-end open source framework

card:

=> is a container

=> building a webpage quickly

24/10/25

Backend :

\* Backend is called middleware.

Node.js, express.js.

\* Backend is use the module like in Python

import math

node is Backend lib

in node we can use express.js as middleware.

Run command:

node filename

Code:

```
const http = require('http'); // Built-in module.  
const express = require('express'); // Third Party module  
const sayHello = require('./greet'); // custom module.
```

The above code ~~require~~ is a keyword.

The sayHello part inside the require the filename.

Request.

Response.

Start the server then start the react.

Bootstrap : Data  
=> is a front-end opensource framework

Card :

=> its a container

=> the building a webpage quickly

NoSQL : process unstructured data

Example :

JSON is a Java Script object

JSON looks like a Java Script object  
(Java Script object notation).

Compass :

=> Compass helps to fetch data from mongodb server.

=> help us to reach the mongodb server.

which means using compass like a client.

Mongosh :

mongo shell was replace with mongosh

This gives interactive environment where can run queries manage database and perform administrative tasks.

## Data modeling

Q. Sharran

Data modeling:

nothing but the structure of your data  
Planning the structure.

Sharran:

obj actually blueprint of db which is created  
by fixing the format with data modelling

Example:

EMPLOYEES      id - int  
                  name - string  
                  salary - float

(collections are stored in DB.

Mongo will have

NOSQL:

SQL - Recordbased

MONGODB SHELL COMMANDS:

1. Open the mongo and press the enter  
button.

2. type > use admin

admin database is created

3. command: db.create => we employes.

1. Create db name computers
2. one collection name laptops
3. name : model, color, Status, Price  
One object vendor { vendorName, vendorPrice }
4. Status Have two things only available and not available.

1. List out particular model laptops
2. Change its status to unavailable.

mongosh: Commands:

1. Open mongosh command  
Press enter.
2. db.dropDatabase(); delete data base:
3. \$group by.
4. with the filter all the records from.

Computers dB 3200 to 50000 range and Go with dell. Update add one more filed shipping yes.

25/01/25  
1. data modelling

id, name, age, city (3 records)

data base name is computers (create a  
another collection name details)

1. id : 1

TWO records city name should be same.

Query:

```
db.customers.find({hobbies: h$in: ["Cooking"]})
```

This query will return all documents where customers hobbies does not

Query:

```
db.Products.find({$and: [{"price": h$gt: 8000}],  
{"brand": "apple"})
```

The above query The and operator and the price should be greater than 8000 and the brand name should be apple. The above condition should find

Query:

```
db.customers.find({$and: [{"hobbies": h$exists: false},  
{"age": 40}]})
```

all documents in the customers collection where the hobbies field does not exists and the age field is

greater than 10.

Aggregation is Group by:

In mongoDB is aggregation. In my SQL it

is called groupby.

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

The above query local field belongs to employees  
and the foreign field belongs to companies.  
New matched document will be added as a  
array in a companies and the array name  
is employees.

Create a database called Bank.

Two collection under that

1. customer
2. customer account

Personal.

Data model:

1. Customer Record fields:  
name, address (array), phoneno (object):

[1. no, 2. [2. no], , age, id]

2. data model and schema. customer account.  
id, account no, branch (string) , Bankname (string),  
(int)  
IFSC code (string) , accBalance, account type (saving or  
(float) (percent)

Overdraft : yes or no.

1. insert 5 records
2. filter only od category one is yes.
3. display only the customers address where the  
branch names are same.
4. accBal 10000 - 20000 filter only their  
Phone no.
5. filter only the saving account ~~or~~ peoples.
6. add the field called status : same.  
for same IFSC code peoples.

client

client.js

server

Node.js

27/01/25

Node:

in the popular library from the same  
own time environment which allows us

to run the

net

maintain split terminals in VS code.

In order to use Client and Server

run command:

start the server first - command

node server.js

Command to run Client:

**npm start**

1. (into Terminal)

2. install express:

npm i express (i - install)

require:

require is a keyword from the JS.

const express = require('express');

express the activate the express.

express is a function

=> server is json object

listen:

listen is a keyword to activate the port.

package.json

We can delete package.json. to get it back

node

npm init -y

server: run script

note:

Browser is Client inside the Browser is console.

But server under Terminal is console.

## Axios:

⇒ It is a popular JS library.

⇒ Used to make HTTP requests from the browser or node.js or client

⇒ Axios is known for easy and clean syntax also flexible.

⇒ Especially works well with API and REST API

⇒ When we write a API for exclusive purpose it is called as REST API

## CORS:

⇒ Cross origin resource sharing

⇒ You are asking something

⇒ When a web page request information from resource. (from <sup>any</sup> other side)

- Whether to accept the request process  
Object will be defined in a JSON file.

For this purpose we use CORS.

Command:

NPM i axios

NPM i CORS

<sup>example</sup>  
In this we are requesting data from sever

→ Server : Hello this is from sever

→ In DataComponent.js : Using DataComponent.js as client we  
HTTP get method via API /data

→ Server response as JSON.

→ From the JSON file I want to filter only

the message so we are using from user response

data.message

28/01/25  
Mongo crud API:

Create a three files inside src.

1. USERS.JS

2. CreateUser.JS

3. UpdateUser.JS

MongoDB:

Data model:

name, age, email

Dependencies:

1. express, 2. axios, 3. express - 4. cors

APP.JS :

⇒ In APP.JS add the routing for the three models.

⇒ User, Create

Create a source folder . NPM init

inside the folder npm init command.

Enter => yes.

inside the server folder

in index.js is inside in the server folder

Backend Code.

command

npm i

Server:

inside the server create a folder

models. inside the models users.js.

for MongoDB.

Collection:

Create a another collection. Under that same  
that database (and).

the database name Emp1,

data model schema.

1. emp\_id, Emp\_name, Salary, contact (array)