

fullstack

Date: / /

6/12/25
Monday

Why HTML 5

- 1) It Enhances the structure of Code.
- 2) It has multimedia tags.

Exclusive tags under HTML 5.

nav, footer, header, footer, aside, audio, video, bold, time, source.

CSS

flex, grid, position

6/11/25
Monday

- 1) What is console and why we are using console.
- * It is a powerful tool for developers.
- * Part of web development comes with most browsers like chrome and Firefox.
- * When we write console.log() in javascript code it sends info to console which is used for debugging process.

Uses:

- i) debugging
- ii) testing code
- iii) monitoring Data
- iv) Improving Performance.

- 2) What is tailwind CSS?

- * Utility first CSS framework
- * Easy customization of styles in ~~css~~ & it is default.
- * It works well with modern frameworks or libraries such as React, Vue, Angular



★ Responsive design for different screens using utility classes!

- CSS
- * provides custom approach from scratch
- L) Approach

* Syntax: uses custom styling Selectors

* Speed less

- Tailwind CSS
- * Provides utility classes.
- ↳ Approach

* Relies on predefined utility classes in HTML

* Speed: increase efficiency by reducing need to write own custom style

7/11/25
Tuesday

- 1) On Git daywise we need to create a folder

↳ `Jan2025` → folder
 ↳ Content (notes)
 ↳ hands-on (all progs)

11/12/25
Tuesday

Date:

1) assignment -1 Tailwind CSS

- ① Like we can use script inside body outside it even in head but the correct way to use js is using it inside script.

②

Es-6

- 1) advanced version of js
- 2) called as ECMAScript
- 3) Like using arrow function instead of using function.
- 4) We are using in code because it will be easier to use and reduce the time too.

convention

js object → json

document.getElementById

↓
Object

innerHTML

↳ property.

html → everything will be called as element

js → "object"
(HTML) Arrow function [Es-6)
i) for efficiency
<body> this id="response" </body>
<body> this id="result" </body> increase readability
<script>

const howareyou = () => iii) we can create functions without name

{
 return 100;
}; iv) it is called as Arrow function.

var add =

(a, b) =>

{ return a + b } ↗ js object ↗ html element
document.getElementById("response").innerHTML = ↗ property
howareyou ();

do . getElementById

("response").innerHTML =

add(100, 200);

</script>

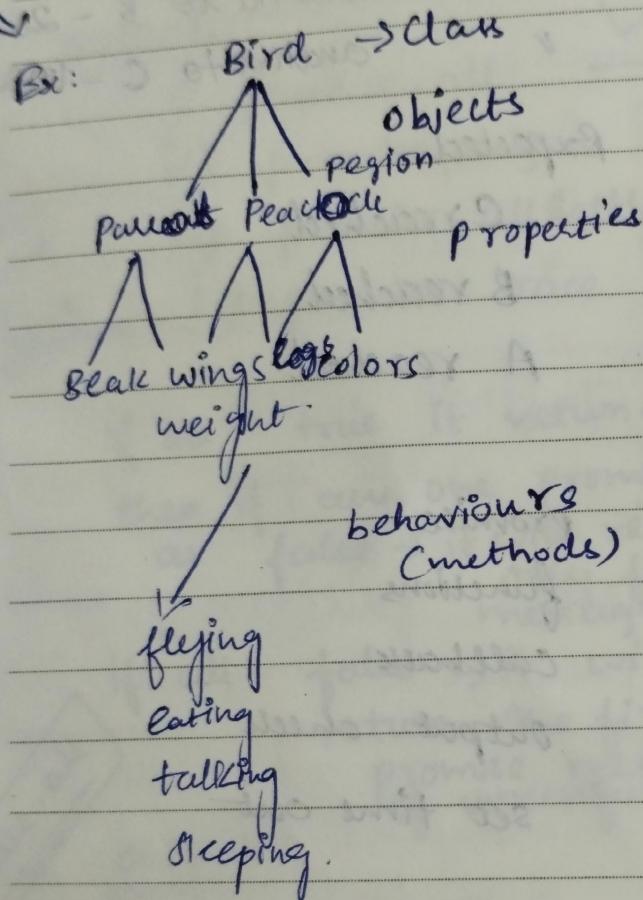
<html>

array
splice
slice

12 34 56
12 99 456
0 (DE)

Object Oriented programming
Structure (oops)

(Chatgpt → Studying Platforms)



Class → Objects
House (Blueprint) (Need to build rooms)

• Class in js not need to create it is inbuilt we just can use that

have keys Date: and values

Bx:

<html>

<script>

var student = {

"name": "Eswar",
"Age": 19}

alert(student);

student.name

alert(key);

* in is called as membership operator

for (i in student)
alert(i)

</script>

</html>

Object Inbuilt method

- 1) Keys
- 2) values
- 3) entries

Extract value using
Keys have 2 ways.

Objt.name

obj["name"]

promise inbuilt methods

promise.all

↳ when there is more than one promise in order to review them we can use promise inbuilt methods according to the requirement

methods:

- i) promise.all
- ii) promise.any
- iii) promise.allSettled
- iv) promise.race

second part along with success future

if all true it return all true message

then if any one promise as false → it will return only false message

If all false → it will return the first promise message of unsuccessful.

promise.all

Output

false

false

false

"Not Reached"

True

True

True



A Reached, B Reached

C Reached.

any two true

1 false

Date:

it will return the message

* ones it sees all promise false it will stop

gives shortest time promise → any provided status should be true.

false

false

false

(AggregateError)

true

true

true

true = False

true ② B reached True

false) True

* If two true means it will return the second true ans

promise.allSettled

will display one among these three

- i) fulfilled
- ii) rejected
- iii) pending.

- * Read is
- * Important files \Rightarrow
- index.html
- index.js
- index.css

* Note:
i) As of now don't touch index files

* Note:
Initial do or write your code in APP.js.

DOM

• React follows:

V - DOM (Virtual DOM)
 \downarrow
 [Document Oriented Model].

i) They are unlike HTML, once dom gets created the changes or manipulation what we do gets coupled and only that part will be re-rendered.

Whereas in HTML we make change entire DOM will be rendered

Web Application (Amazon)

i) In web application created by React.js each and everything is called as component

Coding

Date:

function App() {
 finally we are exporting default app.

in index.js.

App \rightarrow Component

/ App \rightarrow file name

Exporting it from app.js and Importing it in index.js.

~~APP.js~~
~~index.js~~

rendering into ReactDOM.

called as JSX

i) Javascript XML.

ii) Writing HTML XML inside javascript-

XML -

• Used for data transfer

• Used for data trans-

• Used for data st

iii) It makes easier to understand.

Structure of UI Component.

i) Instead of using

React.createElement(),

JSX allows you to write

When we use effect render in App.js it will use call back function & it will display the output.

points
useeffect → accepts two arguments

- i) Callback function
- ii) dependency array that is optional.

Note:

Callback function is like constructor in Java.

Use effect function runs the callback function will get triggered.

[]
after rendering once only you want to run na we are using dependency array.

forms

11/125 Saturday Date: Timer
program

* use effect - 1 → file name

• component name: timer

Definition:
useeffect

use effect hook

i) upon the condition or action we apply in our functional components monitoring the impact or side effects can be done using use effect hook.

Combination of use state and use effects with dependency array

action perform panna Condition enna kudalere based upon that

use:

usestate → initial & monitor changes if useEffect need to give condition.

↳ useeffect take two all usestate take one all return() => clear interval;

- ① call back function calls another using argu
- ② [] → dependency array it is opt

usecontext
saves

• usestate
• usecontext
• useeffect

→ takes two arguments
callback & dependency
array.

copy
into
notes

Whatever actions
given it will get
triggered

```
import React, {  
  useState, useEffect  
} from 'react'
```

```
function UseEffectOne () {
```

```
  const [ state,  
    updates ] = useState(0);
```

```
  useEffect(() =>
```

```
    console.log('The  
    state value has  
    changed!', state);  
,
```

```
[ state ]);
```

```
return
```

```
<div>
```

```
<h1>
```

```
{ state } { state } { state }
```

```
</h1>
```

```
<div>
```

```
  onClick = { () =>
```

```
    updates( + 1 )
```

```
  } { state + 1 }
```

```
</div>
```

```
</div>
```

VNC
GROUP

Rising with values

Counter code

+ - → two buttons

Date & value /

Copy paste

useeffect

Clo

① button creation.

<button onClick = { increment } >

+ </button>

Use Effect

i) Synchronizing a component with
external system.

ii) After our action monitoring
the side effects (seeing) the
functional component is
possible using use effect hook

(Two function + + function
no function, number)

21/1/25
Tuesday eventbinder

- onclick
- on

→ for clicking

→ on mouse-hover

Components type

- functional / class component
- naming variables
functions, objects
are very important

User reducer

if Takes two arguments
Reducer,

points

- User reducer takes two arguments

first is reducer function which says that what we want to do (increment or decrement) & second is initial value of state. (state, action)

- If returns two values

initial count & dispatch function

we call it as state & dispatch

we can use more than one reduce function

User reducer

(a tha

increment

decrement

truleem

atha

call pannuron

ana

User reducer

la

dispatch function

aatha call

pannuron

other we

reduce la

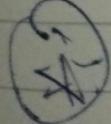
call pannuron

*

State Ritta initial
value truleem

\Rightarrow

& updated once
you call
dispatch
function.



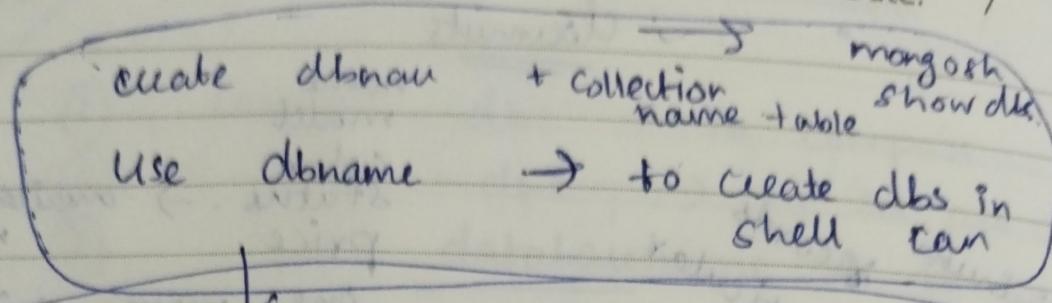
button increment decrement \Rightarrow calls

function of that if has dispatch
dispatch calls user reducer function
actions, performs.

ref your old note.

Mongo db queries

Date: / /



- 1) Create db. → in compass
- 2) use dbs
- 3) Show dbs.
- 4) db. collectionname. (table) → to see collection that created in dbs.
- 5) db. collectionname. insertOne ({}³)
- 6) " insertMany ([]{}³)
- 7) db. collectionname. find ({}³)
- 8) db. collectionname. find () ↗ e.g. name: "ab".
- 9) cls → screen will get cleared
- 10) nested object [Embedded document] db. collection name.
- 11) db. collection.name. updateOne ({"name": "John"}, {\$set: {age: 20}})
- 12) Array updateMany.
- 13) drop
- 14) replaceOne

Mongodb Commands

Date:

- ① Create db → in compass
- ② Show dbs → to see all database
- ③ use databaseName → to switch to particular database
- ④ Show collections →
It will show all collections

database: Computer

details → ~~details~~ configuration

* "id": 1, "name": "xyz", "age": 25, "city": "Colombatore"

data modeling.

schema : id int
 string
 string
 string

Create & add three records.

Project

Database: Bank

two collections: Customer, Personal

Customer Account

Model	Schema
name	string
address	array
phonenumbers	object
age	1, 2

Data Model:

Savings
 Current
 int
 resNo
 >
 S

VNC
 GROUP
 Rising with values

Accounts	int
Branches	string

aggregation is group by

db. companies. aggregate [{ \$ lookup:
Date: / /

{ from: 'employee', localField: 'id',
foreignField: 'company_id', as: 'Employee' }])

queries

1) Filter only off category yes

2) Display only the customers address where the branch names are same.

3) current balance 1000.00 to 2000.00
filter only the phone no.

4) filter only the savings account people collection, complete details

5) add updated field called status (seen) those who are having same issue code

X Backend also called as middle ware

Backend we use Node.js also called Library called React.js.

which is used to connect Frontend with database.

Mongodb.

Module

Date: / /

math → module. It's a package that
consists of functions

function

modules

math → custom module

3rd party → numpy/pandas
module

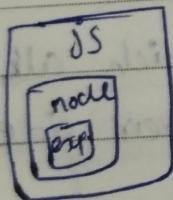
Node.js

Backend

Run command: node

Backend library from
js

In node we can
use express js as middleware



Backend
we will export as
modules

file name: greet.js

require → keyword from node.js.

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

const express = require('express');

// Third-party module

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

// Custom module