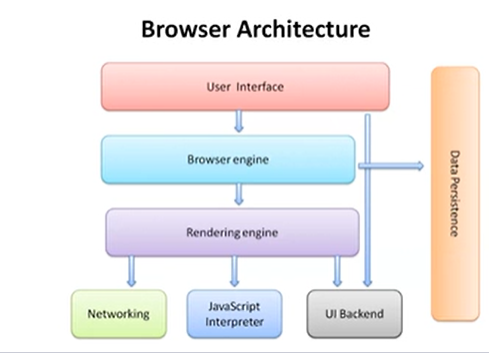
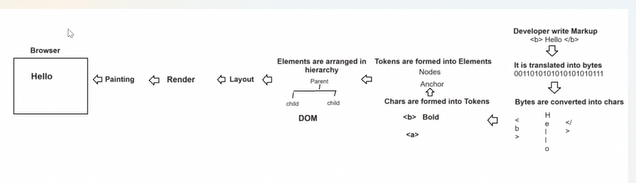
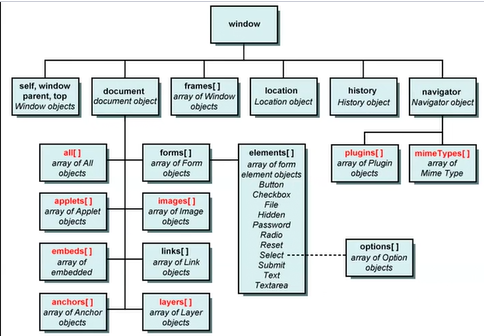
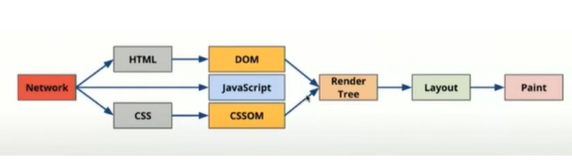
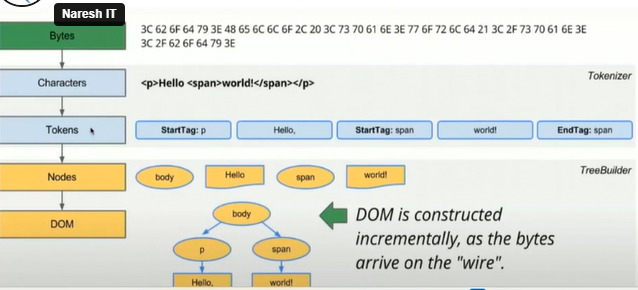
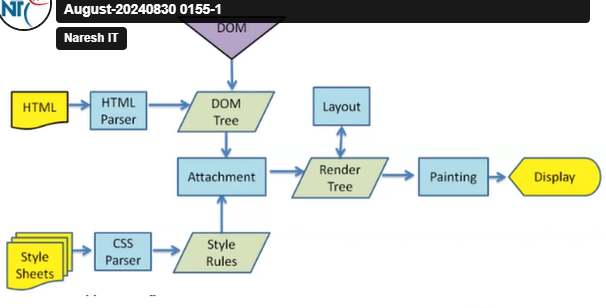
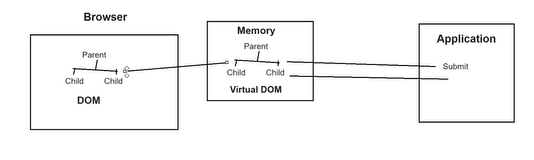
Day 1

* React is library used to build UI for web and native(mobile) applications.
* React current 18 version
* React 16.0 was released in September 2017, while React 17.0 was released in August 2020. On March 29, 2022, React 18 was released, which introduced new features.
* Upto 71x reactjs, after 18x react
* Angular(70%frontend, 30%backend) is framework and react(30%frontend, 70%backend) is library
* If project handles majority of architectural flow backend then we can use react in frontend.
* If project have to handles majority of architectural flow in frontend then it requires angular framework.
* Unified UX (perfectly fit on pc, mobile etc.), same behaviour across any device.
* Fluid UX – (there is no page number, when we scroll down next content is keep on loading) user stays on single page and gets access from other pages,new details can added to pages without refreshing/ reloading.
* Loosely coupled and extensible – in bank application there are 2 modules like 1) personal banking 2) NRI banking, when we want to build new features where users are using that app, pushing new features without stopping application, without reinstalling applications.
* app open in browser – progressive web applications.
* SPA(single page application) – without refreshing page loading the content.

Day 2

* Javascript & jquery – lot of DOM interaction (getElementbyid etc.), lot of queries makes application heavy & legacy(loading unneccesorry function), lot of ajax calls, application will become slow.
* Features of react –
* It is modular (part by part, it loads what is required, it uses less memory, it is faster, it is light weight)
* Component based – reusability, extensibility, easy to test, easy to maintain.
* Input group – textbox, search button, dropdown (bootstrap)
* <input type=”date”> ---- input is element & date is component
* It uses virtual DOM (browser work, DOM, shadow DOM)
* 
* Browser architecture – UI, UI backend, browser engine, rendering engine, network, javascript interpreter, data persistence
* Browser engines – spider monkey, gecko, v8, Chakra, chromium
* HTML parsing(translater) – developer writes markup (<b></b> all tags) 🡪 it is translated into bytes 🡪 bytes converted into characters 🡪 characters are formed into tokens(<b> means bold) 🡪 tokens are formed into elements (nodes, anchor) 🡪 elements are arranged in hierarchical order(parent->child) this is called as DOM 🡪 layout (preparing to show) 🡪 render 🡪 painting after that we can see the page on browser.
* 
* 
* Above is DOM hierarchy.
* HTML presents DOM
* Javascripts manipulates DOM
* Css makes DOM more interactive & responsive
* Css used cssom (css object model) { color : red } 🡪 this is key and value pair
* 
* Above is critical rendering path
* 
* Above is HTML parsing
* 
* Shadow DOM –
* It is hierarchy of elements in components
* DOM is about page and shadow DOM is about component
* Virtual DOM – react uses –
* Maintaining DOM in memory that is duplicate DOM, it is faster than actual DOM, it renders output to users but later updates into actual DOM, it is copy of DOM in memory, below is virtual DOM, it uses one way binding.



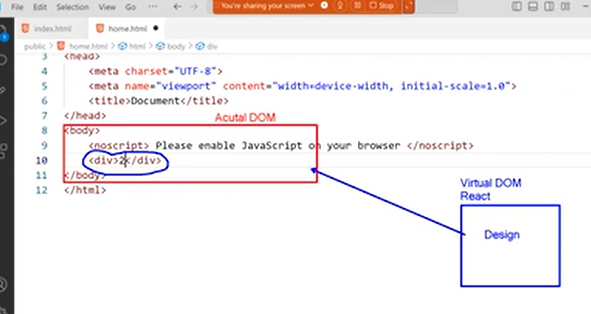


* We can only display value, but we can not change the value again this is one way binding.

Day 3

* Issue with react --- 1) not designed for what you are using. 2) Lot of GAP’s. 3) every technology is not capable of doing every tasks we need third party in this case. 4) pace of development(so much versions). 5) poor documentation( no clear solutions).
* Emmet – templating tool – auto generating – if we open .html file and enter ! mark and enter then automatically html code will appear

Day4

* CDN – content delivery network
* <noscript></noscript> - to verify wheather javascript id enabled or not, we can give message inside tag.
* If javascript is not enabled then this message will be display
* Run application by URL – 127.0.0.1:5500/web-app/public/home.html or right click on page in vscode and click on open with live server
* 
* Javascript is light weigth interpreted and JIT(just in time)(it is compiled after hitting on browser) compiled programming language
* ECMA/ MDN (Mozilla development network)
* Babel – javacript compiler – using AOT ( ahead of time)
* <script type=”text/babel> or < script type=”text/jsx>(javascript extension language)
* Module system – CJS, umd – here we are working with umd

Day5

* React component – in SPA we need to combine components.
* Components are building blocks of react application
* Component has 3 elements – 1) design(HTML) 2) style(CSS) 3) functionality(JSX – javascript extension language, TSX - typescript)
* **Components are created by 1) javascript classes 2) javascript functions**
* Component using JS functions –
* Function topics – function declaration, function expression, function signature, function parameter, rest parameter, spread operator, function with return, function with recursion, function callback, function promise, function generator, function closure, arrow function, anonymous function, IIFE(immediently invoking function expression)
* **Basic rule - A function component must return jsx element, cannot be void.**
* **A function component name must start with uppercase letter**.
  + function ComponentName(){

return(

<div>

</div>

);

}

* function must return only 1 fragment that only 1 container.
  + Function ComponentName(){

return(

<h1></h1>

<p></p>

)

}

Above code returning 2 fragment is invalid, below is main div container and in div container we have <h1> and <p> tag is valid.

* + Function ComponentName(){

return(

<div>

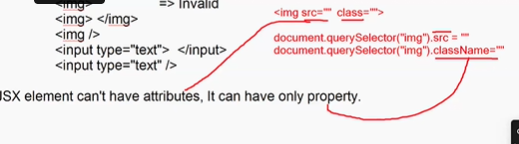
<h1></h1>

<p></p>

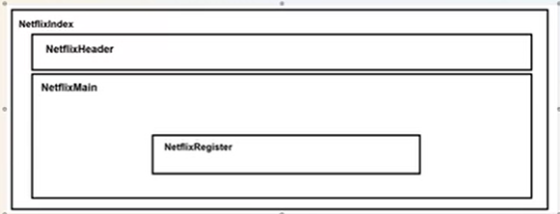
</div>

)

}

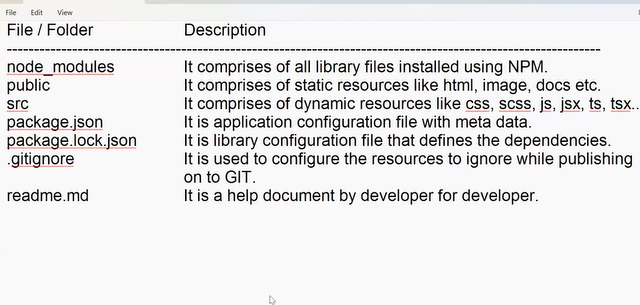
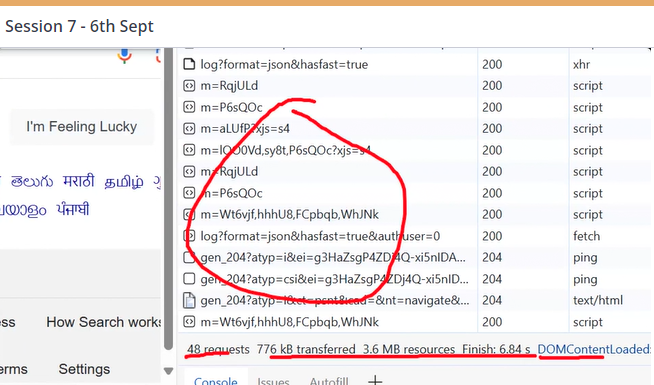
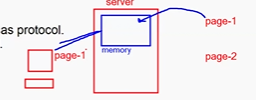
* **JSX can not have void elements, every element must have end token.**
* HTML elements category – normal(required to have end tag, example <b></b>), void (don’t have end tag), RC, raw text, foreign
* Inline element, block level elements are behaviour of elements
* <https://whatwg.org/> - for HTML
* **JSX elemtns can not have attributes it can have only property.**
* 
* <img class=””> ---invalid
* <img className=””> --- valid
* What is key feature of component – component enables many features like reusability, extensibility, testability, maintainability
* Key primary feature is reusability

Day6

* Npm install bootstrap bootstrap-icons –save
* Link bootstrap files to web page
* 
* Day7
* React 18 environment
* It recommends function components
* Better error handling (strict mode)
* New hooks to handle various services
* Concurrency is not a feature, it is a behind the scene mechanism that enable react to prepare multiple versions of UI at a one time
* Behind the scene concept that improved react performance
  + Automatic batching
  + Strict mode
  + Concurrency
  + Transitions
  + Server side rendering
* In react upto 17 virtual dom created using ReactDom.render()
* In react 18 it uses

const root = createRoot(document.getElementbyId(“root”))

root.render(<component/ >)

* Creating application
* By using bundling tools like 1)webpack 2)parcel 3)vite 4)node etc. (creating servers by using this tools)
* Apln using webpack
* Run command – **npx create-react-app app-name**
* Npx uses webpack bundler to create react appn
* It installs latest libraries for react 18
* It configures react appln with file system and pre-defined templates
* 
* Strictly type – programming rules
* Strongly typed –data types
* 
* Day8
* Data binding in react
* 
* Client requested for page 1, server responds back and clear memory for page 1, again same as for page 2 and so on this is stateless protocol.
* So we need to maintain state to available information across the application.

Day9

* We are not storing data in variable because it is immutable, and component have mutable data.
* All web application uses http protocol, http is stateless thereof we need state.
* useState() – ir create state and store data in it.
* 
* We can initialize by using below
* const [userName, setUserName] = useState('john');
* but we can reinitialize by using mount or any event like below we have used useEffect
* useEffect(() => {
* setUserName('David');
* },[])
* Every component will only mount once, after creation component will load, after it will not be updated, if we want to mount again then we have to give dependencies in useEffect. You have to give blank array if there is not dependencies.
* What are dependencies in mount phase 🡪 dependencies are used to remount the component, that is reload.
* Bindings various data types
* 2 types of data type : Primitive & non-primitive types
* 
* {(isNaN(price))?"Price must be a number":price} – in this stmt if price is number then print a number if not then display msg as Price must be a number

Day10

* a> Double Quote " "
* b> Single Quote ' '
* c> Backtick ` `
* Backtick allows embedded expression by using "${}"
* Dynamic expreesion is allowed only with backtick
* For symbol type :

const hiddenProperty = Symbol("hidden"); // create a unique symbol

const obj = {

visibleProperty: "This is visible",

[hiddenProperty]: "This is hidden"

};

console.log(obj.visibleProperty); // Output: "This is visible"

console.log(obj[hiddenProperty]); // Output: "This is hidden"

// Attempting to iterate over properties of the object

for (let key in obj) {

console.log(key); // Output: "visibleProperty" only

}

console.log(Object.keys(obj)); // Output: ["visibleProperty"]

console.log(Object.getOwnPropertyNames(obj)); // Output: ["visibleProperty"]

// Accessing symbol properties requires specific methods

console.log(Object.getOwnPropertySymbols(obj)); // Output: [ Symbol(hidden) ]

* var collection = []; Static memory – memory allocated once, used for any number of request, it is still remain in memory after use
* var collection = new Array(); dynamic memory - it destroy the memory after use

Day11

* **object type –**
* 1. How to access all keys from an object?
* A. By using 2 techniques
* a> for..in
* b> object.keys()
* In jsx we can use only object.keys
* object.keys will return an array
* 2. How to delete a key?
* By delete operator
* Syntax:
* delete object.key;

delete product.price;

* **Array of Objects –**
* Day12
* Javascript uses XMLHTTPRequest and fetch() to access json file