**Batch: A-3 Roll No.: 16010122104**

**Experiment / assignment / tutorial No. 5**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| Title: Demonstrate the Use of node.js |

**AIM:** To implement the node js Concepts based on the following topics.

**Problem Definition:**

**Problem statement:**

Consider the basic concepts, which are useful in the creation of an application.

Considering the following points, demonstrate the functionality of each with a simple script

**1) Basic Routing:**

          1. Build First server application using http module

           2. Basic routing: Demonstrate it using simple HTML/Json file

          3. Demonstrate the callback in node.js

 2)  **File operation**

          - Check Permissions of a File or Directory.

          - Checking if a file or a directory exists.

          - Determining the line count of a text file.

          - Reading a file line by line.

          - See the file content through browser.

**3) Building your custom modules**

          -To demonstrate this use some mathematics function to create custom module.

**$) Blocking and Non Blocking**

**Resources used:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO 2:**. Illustrate the concepts of various front-end, back-end web application development technologies & frameworks using different web development tools.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Shelly Powers Learning Node O’ Reilly 2 nd Edition, 2016.

**Pre Lab/ Prior Concepts:**

**Node.js**

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine, enabling developers to run JavaScript code server-side. Its non-blocking, event-driven architecture makes it well-suited for building scalable, real-time applications such as chat applications, online games, and streaming services. Using Node.js, developers can create fast and efficient server-side applications that handle a large number of simultaneous connections with low resource consumption. It also allows for a unified development experience as developers can use JavaScript for both client-side and server-side code.

**Basic Routing**

In web development, routing refers to determining how an application responds to a client request for a particular endpoint, which could be a URL path and a specific HTTP method. Basic routing in Node.js, often handled by frameworks like Express, allows developers to create distinct routes for different URLs or HTTP methods such as GET, POST, PUT, and DELETE. By defining various routes, developers can control how a server responds to user requests, enabling different content or actions to be served depending on the requested URL. This is essential for building interactive and multi-page applications.

**Custom Module**

Custom modules in Node.js are user-defined, reusable blocks of code that can be imported into other parts of an application, allowing for modularization and improved code organization. By breaking down functionality into smaller, custom modules, developers can create and manage code more efficiently and reduce redundancy. Custom modules can be as simple as utility functions or more complex pieces of an application, such as a database handler or a custom API wrapper. Node.js makes it easy to define and export these modules, supporting better code maintainability and reuse.

**Asynchronous Programming**

Asynchronous programming is a paradigm that allows a program to perform tasks, such as making network requests or file operations, without blocking the execution of other parts of the program. In Node.js, asynchronous programming is crucial because of its single-threaded, event-driven architecture. It utilizes callbacks, promises, and async/await functions to handle asynchronous operations, enabling applications to manage multiple I/O operations efficiently. This approach is essential for developing scalable applications that handle a large number of concurrent tasks without impacting performance.

**Methodology:**

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

import { ShopContext } from '../context/ShopContext'

import Title from './Title';

import ProductItem from './ProductItem';

const BestSeller = () => {

    const {products} = useContext(ShopContext);

    const [bestSeller,setBestSeller] = useState([]);

    useEffect(()=>{

        const bestProduct = products.filter((item)=>(item.bestseller));

        setBestSeller(bestProduct.slice(0,5))

    },[products])

  return (

    <div className='my-10'>

      <div className='text-center text-3xl py-8'>

        <Title text1={'BEST'} text2={'SELLERS'}/>

      </div>

      <div className='grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 lg:grid-cols-5 gap-4 gap-y-6'>

        {

            bestSeller.map((item,index)=>(

                <ProductItem key={index} id={item.\_id} name={item.name} image={item.image} price={item.price} />

            ))

        }

      </div>

    </div>

  )

}

export default BestSeller

**Implementation Details:**

**1. Importing React and Required Hooks**

* Import necessary libraries and components.
* useContext, useEffect, and useState are essential React hooks.
* ShopContext, Title, and ProductItem are imported components, assumed to be defined elsewhere in your project.

javascript

Copy code

import React, { useContext, useEffect, useState } from 'react';

import { ShopContext } from '../context/ShopContext';

import Title from './Title';

import ProductItem from './ProductItem';

**2. Define the BestSeller Component and Initialize State**

* Define the BestSeller component as a functional component.
* Inside the component, use useContext to access the products array from ShopContext.
* Initialize bestSeller state with an empty array using useState, where the filtered bestseller products will be stored.

javascript

Copy code

const BestSeller = () => {

const { products } = useContext(ShopContext);

const [bestSeller, setBestSeller] = useState([]);

**3. Filter Best-Seller Products Using useEffect**

* Use useEffect to filter products every time the products array changes.
* In the useEffect callback, filter products with bestseller: true.
* Slice the resulting array to get only the top 5 bestseller products.
* Update bestSeller state with this array.

javascript

Copy code

useEffect(() => {

const bestProduct = products.filter((item) => item.bestseller);

setBestSeller(bestProduct.slice(0, 5));

}, [products]);

**4. Rendering the Component**

* Create the component's main layout:
  + A div with a top margin (my-10) wraps the entire component.
  + Inside, a centered title is displayed using the Title component, passing text1 as "BEST" and text2 as "SELLERS".
* Define a responsive grid container for displaying bestseller products.
* Map through the bestSeller array, rendering each ProductItem component and passing properties such as id, name, image, and price.

javascript

Copy code

return (

<div className='my-10'>

<div className='text-center text-3xl py-8'>

<Title text1={'BEST'} text2={'SELLERS'} />

</div>

<div className='grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 lg:grid-cols-5 gap-4 gap-y-6'>

{bestSeller.map((item, index) => (

<ProductItem

key={index}

id={item.\_id}

name={item.name}

image={item.image}

price={item.price}

/>

))}

</div>

</div>

);

**5. Export the BestSeller Component**

* Export BestSeller as the default export for use in other parts of the application.

javascript

Copy code

export default BestSeller;

**Final Complete Code**

Combining all the steps, the complete BestSeller component code is:

javascript

Copy code

import React, { useContext, useEffect, useState } from 'react';

import { ShopContext } from '../context/ShopContext';

import Title from './Title';

import ProductItem from './ProductItem';

const BestSeller = () => {

const { products } = useContext(ShopContext);

const [bestSeller, setBestSeller] = useState([]);

useEffect(() => {

const bestProduct = products.filter((item) => item.bestseller);

setBestSeller(bestProduct.slice(0, 5));

}, [products]);

return (

<div className='my-10'>

<div className='text-center text-3xl py-8'>

<Title text1={'BEST'} text2={'SELLERS'} />

</div>

<div className='grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 lg:grid-cols-5 gap-4 gap-y-6'>

{bestSeller.map((item, index) => (

<ProductItem

key={index}

id={item.\_id}

name={item.name}

image={item.image}

price={item.price}

/>

))}

</div>

</div>

);

}

export default BestSeller;

**Steps for execution:**

 **Import Libraries and Components**  
Import React, hooks (useContext, useEffect, useState), ShopContext, Title, and ProductItem in BestSeller.js.

javascript

Copy code

import React, { useContext, useEffect, useState } from 'react';

import { ShopContext } from '../context/ShopContext';

import Title from './Title';

import ProductItem from './ProductItem';

 **Define Component and Initialize State**  
Use useContext to access products from ShopContext, and initialize bestSeller with useState.

javascript

Copy code

const BestSeller = () => {

const { products } = useContext(ShopContext);

const [bestSeller, setBestSeller] = useState([]);

 **Filter Bestsellers with useEffect**  
In useEffect, filter products with bestseller: true, slice to get the top 5, and update bestSeller.

javascript

Copy code

useEffect(() => {

setBestSeller(products.filter(item => item.bestseller).slice(0, 5));

}, [products]);

 **Render the Component**  
Display a title and a responsive grid of ProductItem components mapped from bestSeller.

javascript

Copy code

return (

<div className='my-10'>

<div className='text-center text-3xl py-8'>

<Title text1='BEST' text2='SELLERS' />

</div>

<div className='grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 lg:grid-cols-5 gap-4 gap-y-6'>

{bestSeller.map((item, index) => (

<ProductItem key={index} {...item} />

))}

</div>

</div>

);

 **Export and Use the Component**  
Export BestSeller for use in other parts of the app.

javascript

Copy code

export default BestSeller;

**Conclusion: We demonstrated the use of node.js.**