**React basics**

**1. How to write your first React component**

**Task1**

**Question**

Create a simple functional component named Greeting that returns “Hello,

World!” within a <h1> tag.

**App.js**

function Greeting() {

return (

<h1>

Hello World

</h1>

);

}

export default function MyApp() {

return (

<Greeting />

);

}

**Task2**

**Question**

Modify the Greeting component to display “Hello, React!”.

**App.jsS**

function Display() {

return (

<h1>

Hello React!

</h1>

);

}

export default function MyApp() {

return (

<Display />

);

}

**Task3**

**Question**

Create a Gallery functional component to display an image.

**App.js**

const user = {

name: 'DINESH',

imageUrl: '<https://avatars.githubusercontent.com/u/140265283?v=4>',

imageSize: 90,

};

export default function Gallery() {

return (

<>

<h1>{user.name}</h1>

<img

className="avatar"

src={user.imageUrl}

alt={'Photo of ' + user.name}

style={{

width: user.imageSize,

height: user.imageSize

}}

/>

</>

);

}

**Task4**

**Question**

Add Greeting to the Gallery component and display the image and greeting.

**App.js**

import logo from './logo.svg';

import './App.css';

function Greeting()

{

return (

<h1>

Hello World

</h1>

);

}

const user = {

name: 'DINESH',

imageUrl: '<https://avatars.githubusercontent.com/u/140265283?v=4>',

imageSize: 90,

};

export default function Gallery() {

return (

<>

<Greeting />

<h1>{user.name}</h1>

<img

className="avatar"

src={user.imageUrl}

alt={'Photo of ' + user.name}

style={{

width: user.imageSize,

height: user.imageSize

}}

/>

</>

);

}

**Task5**

**Question**

Write a component called Profile which displays a hardcoded user’s name and age.

**App.js**

import logo from './logo.svg';

import './App.css';

export default function MyApp() {

const name = 'Dinesh';

const age = '20';

return (

<h4>{name} {age}</h4>

);

}

**2)When and how to create multi-component files**

**Task1**

**Question**

Create a file named UserComponents.js and inside it, define two

components: UserName and UserAge that display hardcoded names and ages respectively

**App.js**

import logo from './logo.svg';

import './App.css';

import {Username,Userage} from './UserCom.js';

export default function App()

{

return(

<>

<Username />

<Userage />

</>

);

}

**Usercom.js**

import logo from './logo.svg';

import './App.css';

const person={

name:"Dinesh",

age:"19"

};

export default function Username()

{

return(

<h1>{person.name}</h1>

);

}

export default function Userage()

{

return(

<h2>{person.age}</h2>

);

}

**Task2**

**Question**

Export both UserName and UserAge from UserComponents.js.

**App.js**

import logo from './logo.svg';

import './App.css';

import {Username,Userage} from './UserCom.js';

export default function App()

{

return(

<>

<Username />

<Userage />

</>

);

}

**Usercom.js**

import logo from './logo.svg';

import './App.css';

const person={

name:"Dinesh",

age:"19"

};

export default function Username()

{

return(

<h1>{person.name}</h1>

);

}

export default function Userage()

{

return(

<h2>{person.age}</h2>

);

}

**Task3**

**Question**

In a separate file, import and use both UserName and UserAge components

using named imports.

**UserName.js**

import logo from './logo.svg';

import './App.css';

const person={

name:"Dinesh"

};

export default function User1()

{

return (

<h4>{person.name}</h4>

);

}

**UserAge.js**

import logo from './logo.svg';

import './App.css';

const person={

age:"20"

};

export default function User()

{

return (

<h4>{person.age}</h4>

);

}

**App.js**

import './App.css';

import User1 from './UserName.js';

import User from './UserAge.js';

export default function App()

{

return(

<>

<User1 />

<User />

</>

);

}

**Task4**

**Question**

Convert UserAge into a default export and modify the importing file to

accommodate the change.

**App.js**

import logo from './logo.svg';

import './App.css';

import {Userage} from './UserCom.js';

export default function App()

{

return(

<>

<Userage />

</>

);

}

**Usercom.js**

import logo from './logo.svg';

import './App.css';

const person={

age:"19"

};

export default function Userage()

{

return(

<h2>{person.age}</h2>

);

}

**Task5**

**Question**

Split UserName and UserAge into separate files and adjust your imports

**UserName.js**

import logo from './logo.svg';

import './App.css';

const person={

name:"Dinesh"

};

export default function User1()

{

return (

<h4>{person.name}</h4>

);

}

**UserAge.js**

import logo from './logo.svg';

import './App.css';

const person={

age:"20"

};

export default function User()

{

return (

<h4>{person.age}</h4>

);

}

**App.js**

import './App.css';

import User1 from './UserName.js';

import User from './UserAge.js';

export default function App()

{

return(

<>

<User1 />

<User />

</>

);

}

**3. How to add markup to JavaScript with JSX**

**Task1**

**Question**

Create a component that displays an unordered list (<ul>) of 3 favorite fruits.

**fruit.js**

import logo from './logo.svg';

import './App.css';

export default function Apple()

{

return(

<ul>

<li>Orange</li>

<li>Mango</li>

<li>Graphes</li>

</ul>

)

}

**App.js**

import './App.css';

import Apple from './fruit.js';

export default function App()

{

return(

<>

<Apple />

</>

);

}

**Task2**

**Question**

Update the above component to display a picture (<img>) of each fruit next to its name. (Use hardcoded image URLs for now.)

**fruit.js**

import logo from './logo.svg';

import './App.css';

const image={

width:70,

height:50,

imageurl:'orange.jfif',

imageurl1:'Mango.jfif',

imageurl2:'grapes.jpg',

}

export default function Apple()

{

return(

<ul>

<li>Orange</li>

<img src={image.imageurl}></img>

<li>Mango</li>

<img src={image.imageurl1}></img>

<li>Grapes</li>

<img src={image.imageurl2}></img>

</ul>

)

}

**Apple.js**

import './App.css';

import Apple from './fruit.js';

export default function App()

{

return(

<>

<Apple />

</>

);

}

**Task3**

**Question**

Create a component WebsiteLink that displays a hardcoded URL in an anchor (<a>) tag.

**link.js**

import logo from './logo.svg';

import './App.css';

export default function Url(){

return(

<ul>

<li><a href="https://www.google.com/gmail/about/">Gmail</a></li>

<li><a href="https://www.codechef.com/login">Codechef</a></li>

<li><a href="https://www.hackerrank.com/">HackerRank</a></li>

</ul>

);

}

**App.js**

import './App.css';

import Url from './link.js';

export default function App()

{

return(

<>

<Url />

</>

);

}

**Task4**

**Question**

Make a JSX component that mimics a simple blog post with a title, content,

and author. (All hardcoded.)

**Blog.js**

import './App.css';

import Url from './link.js';

const book={

title:'Java Script',

content:'how to make flexible webpage',

author:'Brendan Eich',

}

export default function Foot(){

return(

<>

<h1>{book.title}</h1>

<h1>{book.content}</h1>

<h1>{book.author}</h1>

</>

);

}

**App.js**

import './App.css';

import Url from './blog.js';

import Foot from './blog.js';

export default function App()

{

return(

<Foot />

);

}

**4. JavaScript in JSX with Curly Braces**

**Task1**

**Question**

Display today’s date in a component using the JavaScript Date object

**Date.js**

import logo from './logo.svg';

import './App.css';

export function GetCurrentDate(separator=''){

let newDate = new Date()

let date = newDate.getDate();

let month = newDate.getMonth() + 1;

let year = newDate.getFullYear();

return `${year}${separator}${month<10?`0${month}`:`${month}`}${separator}${date}`

}

**App.js**

import logo from './logo.svg';

import './App.css';

import './Date.js'

function App() {

return (

<GetCurrentDate />

);

}

export default App;

**Task2**

**Question**

Create a component that displays a random quote from a hardcoded list of quotes.

**quotes.js**

import logo from './logo.svg';

import './App.css';

const quotes={

a1:"The purpose of our lives is to be happy.",

a2:"Life is what happens when you're busy making other plans",

a3:"Never let the fear of striking out keep you from playing the game."

}

export default function App1()

{

return(

<>

<h2>{quotes.a1}</h2>

<h2>{quotes.a2}</h2>

<h2>{quotes.a3}</h2>

</>

)

}

**App.js**

import logo from './logo.svg';

import './App.css';

import './quote.js'

function App1() {

return (

<App1 />

);

}

export default App1;

**Task3**

**Question**

Write a component called MathResult that displays the result of a simple arithmetic operation (e.g., addition) of two hardcoded numbers.

**Sum.js**

import logo from './logo.svg';

import './App.css';

const sum={

a:10,

b:50

}

export default function Result()

{

return(

<h2>{sum.a+sum.b}</h2>

)

}

**App.js**

import logo from './logo.svg';

import './App.css';

import './Sum'

import Result from './Sum';

function App1() {

return (

<Result />

);

}

export default App1;

**Task4**

**Question**

Create a component that displays the word count of a hardcoded paragraph.

**Para.js**

import logo from './logo.svg';

import './App.css';

let name='I am a good boy';

var a = name.split(" ");

var b = a.length;

export default function Para1()

{

return(

<h2>{b}</h2>

)

}

**App.js**

import logo from './logo.svg';

import './App.css';

import './Para.js'

import Result from './Sum';

import Para1 from './Para.js';

function App1() {

return (

<Para1 />

);

}

export default App1;

**Task5**

**Question**

Create a component that calculates and displays the product of two

hardcoded numbers

**sum.js**

import logo from './logo.svg';

import './App.css';

const sum={

a:10,

b:50

}

export default function Result()

{

return(

<h2>{sum.a\*sum.b}</h2>

)

}

**App.js**

import logo from './logo.svg';

import './App.css';

import './Sum'

import Result from './Sum';

function App1() {

return (

<Result />

);

}

export default App1;

**5. Passing Props to a Component**

**Question**

Create a Movie component that displays the title, year, and rating of a movie using props

**Task1**

import Movie from './Movie';

const App = () => {

  const movie = {

    title: 'Thunivu',

    year: '2023',

    rating: '8.3',

  };

  return (

    <div>

      <Movie *title*={movie.title} *year*={movie.year} *rating*={movie.rating} />

    </div>

  );

};

export default App;

const Movie = (*props*) => {

  const { title, year, rating } = *props*;

  return (

    <div>

      <h4>{title}</h4>

      <p>Year: {year}</p>

      <p>Rating: {rating}</p>

    </div>

  );

};

export default Movie;

**Question**

Update the Movie component to have a default prop for rating as “Not Rated”.

**Task2**

import Movie from './Movie';

const App = () => {

  const movie = {

    title: 'Thunivu',

    year: '2023'

  };

  return (

    <div>

      <Movie *title*={movie.title} *year*={movie.year} *rating*={movie.rating} />

    </div>

  );

};

export default App;

const Movie = (*props*) => {

  const { title, year, rating } = *props*;

  return (

    <div>

      <h2>{title}</h2>

      <p>Year: {year}</p>

      <p>Rating: {rating}</p>

    </div>

  );

};

Movie.defaultProps = {

  rating: 'Not Rated',

};

export default Movie;

**Question**

Design a Button component that takes in a label prop and displays the label on the button.

**Task3**

const Button = ({ *label* }) => {

  return (

    <button>{*label*}</button>

  );

};

export default Button;

import Button from './Button';

const App = () => {

  return (

    <div>

      <Button *label*="Click me!" />

    </div>

  );

};

export default App;

**Question**

Make a UserProfile component and pass an object containing user details as props and display them.

**Task4**

const UserProfile = (*props*) => {

  const { user } = *props*;

  return (

    <div *className*="user-profile">

      <h2>User Profile</h2>

      <p>Name: {user.name}</p>

      <p>Email: {user.email}</p>

      <p>Age: {user.age}</p>

    </div>

  );

};

export default UserProfile;

import UserProfile from './UserProfile';

const App = () => {

  const user = {

    name: "John Doe",

    email: "johndoe@example.com",

    age: 30

  };

  return (

    <div *className*="App">

      <h1>User Details</h1>

      <UserProfile *user*={user} />

    </div>

  );

};

export default App;

**Question**

Develop a Modal component that accepts and displays a title and some content passed as props.

**Task5**

function Modal({ *title*, *content* }){

  return (

    <div>

      <h2>{*title*}</h2>

      <p>{*content*}</p>

    </div>

  );

};

export default Modal;

import Modal from './Modal';

function App(){

  return (

    <div>

      <Modal *title*="Tell about contact" *content*="This is the modal content." />

    </div>

  );

};

export default App;

**6.Conditional Rendering**

**Question**

Design a UserStatus component that displays “Online” or “Offline” based on a isOnline prop.

**Task1**

function UserStatus({ isOnline }) {

return (

<div className={`user-status ${isOnline ? 'online' : 'offline'}`}>

{isOnline ? 'Online' : 'Offline'}

</div>

);

}

export default UserStatus;\2

**Question**

Write a component AgeCheck that displays “Adult” or “Minor” based on

an age prop

**Task2**

import React from 'react';

function AgeCheck({ age }) {

  const isAdult = age >= 14;

  return (

    <div className={`age-check ${isAdult ? 'adult' : 'minor'}`}>

      {isAdult ? 'Adult' : 'Minor'}

    </div>

  );

}

export default AgeCheck;

**Question**

Create a Loading component that either displays “Loading…” or content based on a isLoading prop.

**Task3**

import React from 'react';

const Loading = ({ *isLoading*, *content* }) => {

  return (

    <div>

      {*isLoading* ? <p>Loading...</p> : *content*}

    </div>

  );

};

export default Loading;

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

import Loading from './Loading';

const App = () => {

  const [isLoading, setIsLoading] = useState(true);

  const [data, setData] = useState(null);

  useEffect(() => {

    setTimeout(() => {

      setData(‘windows);

      setIsLoading(false);

    }, 2000);

  }, []);

  return (

    <div>

      <Loading *isLoading*={isLoading} *content*={<p>{data}</p>} />

    </div>

  );

};

export default App;

**Question**

Design a Feedback component that displays feedback in either green (positive) or red (negative) based on a type prop.

**Task4**

import React from 'react';

const Feedback = ({ *message*, *isPositive* }) => {

  const feedbackStyle = {

    color: *isPositive* ? 'green' : 'red',

  };

  return (

    <div>

      <p *style*={feedbackStyle}>{*message*}</p>

    </div>

  );

};

export default Feedback;

import React from 'react';

import Feedback from './Feedback';

const App = () => {

  return (

    <div>

      <Feedback *message*="Your work is good ." *isPositive*={true} />

      <Feedback *message*="Your work is bad." *isPositive*={false} />

    </div>

  );

};

export default App;

**7.Rendering Lists**

**Question**

Write a component that takes an array of names as a prop and displays them

in a list.

**Task1**

import React from 'react';

function NameList({ names }) {

  return (

    <div>

      <h2>List of Names</h2>

      <ul>

        {names.map((name, index) => (

          <li key={index}>{name}</li>

        ))}

      </ul>

    </div>

  );

}

export default NameList;

import React from 'react';

import NameList from './NameList';

function App() {

  const names = ['Alice', 'Thomas', 'Charlie', 'David'];

  return (

    <div>

      <h1>Names App</h1>

      <NameList names={names} />

    </div>

  );

}

export default App;

**Question**

Create a TodoList component that displays a list of tasks and marks the

completed ones.

**Task2**

import React, { useState } from 'react';

function TodoList() {

  const [tasks, setTasks] = useState([

    { id: 1, text: 'Task 1', completed: false },

    { id: 2, text: 'KCE', completed: true },

    { id: 3, text: 'CBE', completed: false },

  ]);

  const toggleCompleted = (taskId) => {

    const updatedTasks = tasks.map((task) =>

      task.id === taskId ? { ...task, completed: !task.completed } : task

    );

    setTasks(updatedTasks);

  };

  return (

    <div>

      <h2>Todo List</h2>

      <ul>

        {tasks.map((task) => (

          <li

            key={task.id}

            style={{ textDecoration: task.completed ? 'line-through' : 'none' }}

            onClick={() => toggleCompleted(task.id)}

          >

            {task.text}

          </li>

        ))}

      </ul>

    </div>

  );

}

export default TodoList;

**Question**

Design a ProductList component that only displays products with a price less than $10 using the filter() method.

**Task3**

const products=[

    {name:'Apple',price:20},

    {name:'Banana',price:5},

    {name:'Mango',price:8},

    {name:'Grapes',price:15},

    {name:'Papaya',price:10},

    {name:'Orange',price:10},

    {name:'Pineapple',price:13},

    {name:'Watermelon',price:7}

];

function ProductList() {

  const affordableProducts = products.filter((product) => product.price <= 10);

  return (

    <div>

      <h2>Products Under $10</h2>

      <ul>

        {affordableProducts.map((product, index) => (

          <li key={index}>

            {product.name} - ${product.price.toFixed(2)}

          </li>

        ))}

      </ul>

    </div>

  );

}

export default ProductList;

**Question**

Make a UserList component that takes an array of user objects and displays their names and emails.

**Task4**

import React from 'react';

function UserList({ users }) {

  return (

    <div>

      <h2>User List</h2>

      <ul>

        {users.map((user) => (

          <li key={user.id}>

            <strong>Name:</strong> {user.name}, <strong>Email:</strong> {user.email}

          </li>

        ))}

      </ul>

    </div>

  );

}

export default UserList;

import React from 'react';

import UserList from './UserList';

function App() {

  const users = [

    { id: 1, name: 'M S Dhoni', email: 'msdhoni@gmail.com' },

    { id: 2, name: 'Suriya', email: 'suriya@gmail.com' },

    { id: 3, name: 'Rolex', email: 'rolex@gmail.com' },

  ];

  return (

    <div>

      <UserList users={users} />

    </div>

  );

}

export default App;

**Question**

Create a ShoppingCart component that displays a list of items and their prices. Ensure each item has a unique key.

**Task5**

import React from 'react';

function ShoppingCart({ items }) {

  return (

    <div>

      <h2>Shopping Cart</h2>

      <ul>

        {items.map((item) => (

          <li key={item.id}>

            <strong>Item:</strong> {item.name}, <strong>Price:</strong> ${item.price.toFixed(2)}

          </li>

        ))}

      </ul>

    </div>

  );

}

export default ShoppingCart;

import React from 'react';

import ShoppingCart from './ShoppingCart';

function App() {

  const items = [

    { id: 1, name: 'Rice', price: 10.99 },

    { id: 2, name: 'Snacks', price: 5.49 },

    { id: 3, name: 'Detergent', price: 7.99 },

  ];

  return (

    <div>

      <ShoppingCart items={items} />

    </div>

  );

}

export default App;

**8.Keeping Components Pure**

**Question**

Convert an impure component that uses Math.random() within the render

phase to a pure one

**Task1**

import React, { Component } from 'react';

class RandomNumberGenerator extends Component {

  render() {

    const randomNumber = Math.random();

    return (

      <div>

        <h2>Random Number:</h2>

        <p>{randomNumber}</p>

      </div>

    );

  }

}

export default RandomNumberGenerator;

**Task2**

**Question**

Create a pure component Clock that displays the current time and updates

every second without causing side-effects during the render phase.

**Component.js**

import React, { Component } from 'react';

class Clock extends Component {

constructor(props) {

super(props);

this.state = { time: new Date() };

}

componentDidMount() {

this.timerID = setInterval(() => {

this.setState({ time: new Date() });

}, 1000);

}

componentWillUnmount() {

clearInterval(this.timerID);

}

render() {

const { time } = this.state;

const formattedTime = time.toLocaleTimeString();

return (

<div>

<h2>Current Time:</h2>

<p>{formattedTime}</p>

</div>

);

}

}

export default Clock;

**App.js**

import React from 'react';

import Clock from './Clock';

function App() {

return (

<div>

<h1>Live Clock</h1>

<Clock />

</div>

);

}

export default App;

**Task3**

**Question**

Use Strict Mode in an existing application and identify any warnings in the console.

**App.js**

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

ReactDOM.render(

<React.StrictMode>

<App />

</React.StrictMode>,

document.getElementById('root')

);

**Task4**

**Question**

Convert a class-based component with side effects in its lifecycle methods

to a pure functional component using hooks.

**App.js**

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

function MyComponent() {

const [data, setData] = useState(null);

useEffect(() => {

fetch('https://api.example.com/data')

.then((response) => response.json())

.then((data) => {

setData(data);

});

}, []);

return (

<div>

{data ? (

<div>

<h1>Data Loaded</h1>

<p>{data}</p>

</div>

) : (

<p>Loading...</p>

)}

</div>

);

}

export default MyComponent;

**Task5**

**Question**

Make a pure ProfilePic component that takes a user ID as a prop and fetches the user’s profile picture URL from an array without side-effects during rendering.

**Component.js**

import React from 'react';

import mango from './mango.jpg'

function ProfilePic({ userId, users }) {

const user = users.find((user) => user.id === userId);

if (!user) {

return <div>User not found</div>;

}

const { profilePicUrl, name } = user;

return (

<div>

<h2>{name}'s Profile Picture</h2>

<img src={mango} alt={`${name}'favorit fruit`} style={{width:50}} />

</div>

);

}

export default ProfilePic;

**App.js**

import React from 'react';

import mango from './mango;

function App() {

const users = [

{ id: 1, name: 'My', profilePicUrl: 'url\_for\_john' },

];

return (

<div>

<ProfilePic userId={1} users={users} />

</div>

);

}

export default App;