

1.	<p><b>Which of the following is used in React.js to increase performance?</b></p> <ul style="list-style-type: none"> <li>A. Virtual DOM</li> <li>B. Original DOM</li> <li>C. Both Virtual DOM and Original DOM</li> <li>D. Container</li> </ul> <p>Answer: A</p>
2.	<p><b>What is ReactJS?</b></p> <ul style="list-style-type: none"> <li>A. Server-side framework</li> <li>B. user interface framework</li> <li>C. program language</li> <li>D. controller</li> </ul> <p>Answer: B</p>
3.	<p><b>Identify the one which is used to pass data to components from outside</b></p> <ul style="list-style-type: none"> <li>A. Render with arguments</li> <li>B. setState</li> <li>C. PropTypes</li> <li>D. props</li> </ul> <p>Answer: D</p>
4.	<p><b>Which of the following port is the default where webpack-dev-server runs?</b></p> <ul style="list-style-type: none"> <li>A. 3000</li> <li>B. 3306</li> <li>C. 3030</li> <li>D. 8080</li> </ul> <p>Answer : D</p>
5.	<p><b>A state in React.js is also known as?</b></p> <ul style="list-style-type: none"> <li>A. The internal storage of the component</li> <li>B. External storage of the component</li> <li>C. Permanent storage</li> <li>D. Temporary storage</li> </ul> <p>Answer: A</p>
6.	<p><b>Identify the smallest building block of React.JS.</b></p> <ul style="list-style-type: none"> <li>A. Props</li> <li>B. Elements</li> <li>C. Components</li> <li>D. class</li> </ul> <p>Answer: C</p>
7.	<p><b>What is ReactJS mainly used for building?</b></p> <ul style="list-style-type: none"> <li>A. Database</li> <li>B. Connectivity</li> <li>C. User interface</li> </ul>

	<p>D. Design platform</p> <p>Answer: C</p>
8.	<p><b>Among the following which acts as the input of class-based component?</b></p> <p>A. Factory B. Render C. Class D. Props</p> <p>Answer : D</p>
9.	<p><b>Among the following which is used to create a class inheritance?</b></p> <p>A. Inherits B. Extends C. Create D. this</p> <p>Answer: B</p>
10.	<p><b>Total ways of defining variables in ES6 is?</b></p> <p>A. 1 B. 2 C. 3 D. 4</p> <p>Answer: C</p>
11.	<p><b>In which language is React.js written?</b></p> <p>A. Python B. JavaScript C. Java D. PHP</p> <p>Answer: B</p>
12.	<p><b>Among the corner which is a must-have for every ReactJS component?</b></p> <p>A. renderComponent B. SetinitialComponent C. Initrender D. CreateDOM</p> <p>Answer : A</p>
13.	<p><b>Choose the method with refers to the parent class in ReactJS?</b></p> <p>A. this() B. super() C. iniherits() D. self()</p> <p>Answer: B</p>

14.	<p><b>The function which is called to render HTML to a web page in react?</b></p> <ul style="list-style-type: none"> <li>A. ReactDOM_render()</li> <li>B. render()</li> <li>C. render_DOM()</li> <li>D. DOM_HTML()</li> </ul> <p>Answer : B</p>
15.	<p><b>A class is a type of function, but instead of using the keyword function to initiate it, which keyword do we use?</b></p> <ul style="list-style-type: none"> <li>a. Constructor</li> <li>b. Class</li> <li>c. Object</li> <li>d. DataObject</li> </ul> <p><b>Answer: B</b></p>
16.	<p><b>What is a state in React?</b></p> <ul style="list-style-type: none"> <li>a. A permanent storage.</li> <li>b. Internal storage of the component.</li> <li>c. External storage of the component.</li> <li>d. A temporary storage</li> </ul> <p><b>Answer: B</b></p>
17.	<p><b>Which of the following are two ways to handle data in react?</b></p> <ul style="list-style-type: none"> <li>A. services and components</li> <li>B. State and props</li> <li>C. state and services</li> <li>D. state and component</li> </ul> <p>Answer: B</p>
18.	<p><b>Identify the command used to create a react app.</b></p> <ul style="list-style-type: none"> <li>A. npm install create-react-app</li> <li>B. npm install -g create-react-app</li> <li>C. install -g create-react-app</li> <li>D. install create-react-app</li> </ul> <p>Answer: B</p>
19.	<p><b>Which of the following method is used to access the state of a component from inside of a member function?</b></p> <ul style="list-style-type: none"> <li>A. this.prototype.stateValue</li> <li>B. this.getState()</li> <li>C. this.values</li> <li>D. this.state</li> </ul> <p>Answer: C</p>

20.	<p><b>Which of the following function is used to change the state of react component?</b></p> <ul style="list-style-type: none"> <li>A. this.setState()</li> <li>B. this.state()</li> <li>C. this.setChangeState()</li> <li>D. this.prob,state()</li> </ul> <p>Answer: A</p>
21.	<p><b>In react, the key should be?</b></p> <ul style="list-style-type: none"> <li>A. Unique among his siblings only</li> <li>B. unique in the DOM</li> <li>C. Does not requires to be unique</li> <li>D. unique in the class</li> </ul> <p>Answer: A)The key should be unique among his siblings only.</p>
22.	<p>What is the declarative way to render a dynamic list of components based on values in an array?</p> <ul style="list-style-type: none"> <li>a. Using the reduce array method</li> <li>b. Using the &lt;Each /&gt; component</li> <li>c. Using the Array.map() method</li> <li>d. With a for/while loop</li> </ul> <p>Answer: C</p>
23.	<p>Which of the following are the advantages of React.js?</p> <ul style="list-style-type: none"> <li>a. React.js cannot increase the application's performance with Virtual DOM.</li> <li>b. React.js is not easy to integrate with other frameworks such as Angular, BackboneJS since it is only a view library.</li> <li>c. React.js can render both on client and server side.</li> <li>d. React.js cannot render both on client and server side.</li> </ul> <p>Answer: C</p>
24.	<p><b>Choose the functionality of setState?</b></p> <ul style="list-style-type: none"> <li>A. Access the previous state before the setState operation.</li> <li>B. invoke code after the setState operation is done</li> <li>C. replace this date completely instead of the default merge action</li> <li>D. invoke when called</li> </ul> <p>Answer: B</p>

## Unit 2

1.	<p>Field is a name-value pair in a ____.</p> <ul style="list-style-type: none"><li>A. Attribute</li><li>B. Table</li><li>C. Document</li><li>D. Data</li></ul> <p>Answer: C</p>
2.	<p>MongoDB stores documents in ____.</p> <ul style="list-style-type: none"><li>A. Store</li><li>B. Collections</li><li>C. Indexes</li><li>D. Array</li></ul> <p>Answer: B</p>
3.	<p>Aggregation operations in MongoDB ____ values from multiple documents.</p> <ul style="list-style-type: none"><li>A. Data set</li><li>B. Set</li><li>C. Group</li><li>D. record</li></ul> <p>Answer: C</p>
4.	<p>Amongst which of the following is the data types of MongoDB?</p> <ul style="list-style-type: none"><li>A. Time</li><li>B. Object</li><li>C. Symbolic</li><li>D. int</li></ul> <p>Answer: B</p>
5.	<p>A record in MongoDB is a ____.</p> <ul style="list-style-type: none"><li>A. Document</li><li>B. Table</li><li>C. Application</li><li>D. Database</li></ul> <p>Answer: A</p>
6.	<p>Cursor is a pointer to the result set of a ____.</p> <ul style="list-style-type: none"><li>A. Query</li><li>B. Table</li></ul>

	<p>C. Dataset D. Commands</p> <p>Answer: A</p>
7.	<p>The data model available within MongoDB allows us to represent ____.</p> <p>A. Hierarchical relationships B. Able to handle complex structures C. Both Hierarchical relationships and Able to handle complex structures D. Structural relationships</p> <p>Answer: C</p>
8.	<p>A document is a set of ____.</p> <p>A. Key-value pairs B. Application pairs C. Activity pair set D. record</p> <p>Answer: A</p>
9.	<p>MongoDB supports searching by ____.</p> <p>A. Field B. Record C. Database D. Groups</p> <p>Answer: A</p>
10.	<p>Indexes can be created to improve the performance of searches within ____.</p> <p>A. MongoDB B. NoSQL C. Both MongoDB and NoSQL D. SQL</p> <p>Answer: C</p>
11.	<p>Amongst which of the following is / are the client and user authentication mechanisms in MongoDB.</p> <p>A. SCRAM-SHA-1 B. MongoDB Challenge and Response C. Both SCRAM-SHA-1 and MongoDB Challenge and Response D. SHA-1</p> <p>Answer: C</p>

12.	<p>Point out the correct statement.</p> <ul style="list-style-type: none"> <li>a) MongoDB is classified as a NoSQL database</li> <li>b) MongoDB favours XML format more than JSON</li> <li>c) MongoDB is column oriented database store</li> <li>d) All of the mentioned</li> </ul> <p>Answer: a</p>
13.	<p>MongoDB Queries can return specific fields of documents which also include user-defined _____ functions.</p> <ul style="list-style-type: none"> <li>a) Javascript</li> <li>b) C</li> <li>c) C++</li> <li>d) All of the mentioned</li> </ul> <p>Answer: a</p>
14.	<p>Point out the wrong statement.</p> <ul style="list-style-type: none"> <li>a) Secondary indices are not available in MongoDB</li> <li>b) MongoDB supports search by field, range queries, regular expression searches</li> <li>c) MongoDB can store the business subject in the minimal number of documents</li> <li>d) All of the mentioned</li> </ul> <p>Answer: a</p>
15.	<p>MongoDB has been adopted as _____ software by a number of major websites and services.</p> <ul style="list-style-type: none"> <li>a) frontend</li> <li>b) backend</li> <li>c) proprietary</li> <li>d) all of the mentioned</li> </ul> <p>Answer: b</p>
16.	<p>MongoDB is a _____ database that provides high performance, high availability, and easy scalability.</p> <ul style="list-style-type: none"> <li>a) graph</li> <li>b) key value</li> <li>c) document</li> <li>d) all of the mentioned</li> </ul> <p>Answer: c</p>
17.	<p>Which of the following is the primary key by default in MongoDB?</p> <ul style="list-style-type: none"> <li>A) _key</li> <li>B) _id</li> <li>C) id</li> <li>D) key</li> </ul>

	Answer: B)
18.	<p>Which MongoDB command is used to display all the databases?</p> <p>A) show databases B) show dbs C) display databases D) list databases</p> <p>Answer: B</p>
19.	<p>In MongoDB, what is a collection?</p> <p>A) A group of related documents B) A single document C) A table of related rows D) A set of SQL queries</p> <p>Answer: A</p>
20.	<p>How can you insert a document into a collection in MongoDB?</p> <p>A) db.collection.insertOne(document) B) db.collection.insert(document) C) db.collection.add(document) D) db.collection.addOne(document)</p> <p>Answer: A</p>
21.	<p>Which of the following is used to update a document in MongoDB?</p> <p>A) db.collection.modify(query, update) B) db.collection.update(query, update) C) db.collection.change(query, update) D) db.collection.alter(query, update)</p> <p>Answer: B</p>
22.	<p>What is the command to delete a collection in MongoDB?</p> <p>A) db.collection.delete() B) db.collection.drop() C) db.collection.remove() D) db.collection.erase()</p>



	<p>Answer: B</p>
23.	<p>Which of the following methods is used to retrieve all documents in a collection?</p> <p>A) db.collection.findAll()  B) db.collection.get()  C) db.collection.find()  D) db.collection.retrieve()</p> <p>Answer: C</p>
24.	<p>In MongoDB, how do you perform a query to find documents where the "age" field is greater than 25?</p> <p>A) db.collection.find({age: {\$gt: 25}})  B) db.collection.find({age: {\$greater: 25}})  C) db.collection.find({age: "&gt; 25"})  D) db.collection.find({age: "&gt;25"})</p> <p>Answer: A</p>
25.	<p>What is the role of a replica set in MongoDB?</p> <p>A) It provides high availability and redundancy.  B) It is used to partition data across multiple servers.  C) It is used for schema validation.  D) It is used for indexing data.</p> <p>Answer: A</p>

### Unit 3

1.	<p>Which of the following statements are true?</p> <p>A. Node.js is a server-side language.  B. Node.js is a client-side language.  C. Node.js is both a server and client-side language.  D. Node.js is a template</p>
----	---

	Ans. A
2.	<p>_____ is an interactive shell that processes Node.</p> <p>A. REPL B. REAL C. RESP D. MERN</p> <p>Ans. A</p>
3.	<p>Which of the following are examples of node modules?</p> <p>A. Express B. parser C. Socket.js D. socker</p> <p>Ans. A</p>
4.	<p>To include the HTTP server in the node module, what function do we use?</p> <p>A. get() B. require() C. createServer() D. create()</p> <p>Ans. B</p>
5.	<p>What method is used to return the current working directory of the process?</p> <p>A. cwd() B. pwd() C. cmd() D. ccd()</p> <p>Ans. A</p>
6.	<p>How many node objects methods are available?</p> <p>A. 21 B. 18 C. 19 D. 20</p> <p>Ans. B</p>
7.	<p>Which of the following are the core features of the Express framework?</p> <p>a. It not allows us to set up middleware to respond to HTTP Requests. b. It defines a routing table that can work as per HTTP Method and URL. c. It is not used to render the HTML pages dynamically. d. Complex and unorganized way to handle different HTTP requests</p>

	Answer: B is the correct option.
8.	<p>Which of the following function arguments are available to Express.js Route handlers?</p> <ul style="list-style-type: none"> <li>a. req - the request object</li> <li>b. rep - the response object</li> <li>c. new</li> <li>d. prev</li> </ul> <p>Answer: A</p>
9.	<p>What is the full form of REPL?</p> <ul style="list-style-type: none"> <li>A. Read Eval Print Loop</li> <li>B. Research Eval Program Learn</li> <li>C. Read Earn Point Learn</li> <li>D. Read Eval Point Loop</li> </ul> <p>Ans. A</p>
10.	<p>Default scope in Node.js application is?</p> <ul style="list-style-type: none"> <li>A. Global</li> <li>B. Local</li> <li>C. Local to Object</li> <li>D. Local to class</li> </ul> <p>Ans. B</p>
11.	<p>How can we initiate a NodeJS file called Hello in Node?</p> <ul style="list-style-type: none"> <li>A. Hello.js</li> <li>B. node Hello.js</li> <li>C. javascript Hello.js</li> <li>D. node Hello</li> </ul> <p>Ans. B</p>
12.	<p>Which of the following are built-in node modules?</p> <ul style="list-style-type: none"> <li>A. zlib</li> <li>B. httpsec</li> <li>C. datagram</li> <li>D. httplio</li> </ul> <p>Ans. A</p>
13.	Which of the following is a valid HTTP method?

	<p>A. gethttp B. put C. posthttp D. puthttp</p> <p>Ans. B</p>
14.	<p>Which of the following NodeJS modules is used for file uploads?</p> <p>A. formidable B. fs C. express D. files</p> <p>Ans. A</p>
15.	<p>Where are the captured values populated regarding the route parameters?</p> <p>a. req.data b. app.locals c. req.params d. app.global</p> <p>Answer: C</p>
16.	<p>How is it possible to create chainable route handlers for a route path in Express.js?</p> <p>a. Using app.route() b. Using app.routes() c. Using app.router() d. Using app.routing()</p> <p>Answer: A</p>
17.	<p>Where are the captured values populated regarding route parameters?</p> <p>a. app.locals object b. req.data object c. req.params object d. res.params object</p>

	<p>Answer: C</p>
18.	<p>How can we expose node modules?</p> <ul style="list-style-type: none"> <li>A. exports</li> <li>B. expose</li> <li>C. requires</li> <li>D. show</li> </ul> <p>Ans. A.</p>
19.	<p>The advantages of using node modules are?</p> <ul style="list-style-type: none"> <li>A. Provides a means of not dividing up tasks</li> <li>B. Provides a means of reuse of program code</li> <li>C. Provides a means of executing</li> <li>D. Provides a means of installing</li> </ul> <p>Ans. B</p>
20.	<p>When does a stream fire a data event?</p> <ul style="list-style-type: none"> <li>A. When there is data to read.</li> <li>B. When there is data to write</li> <li>C. When there is no data available</li> <li>D. When there is data to execute</li> </ul> <p>Ans. A</p>
21.	<p>For path specific operations which of the following operations is required to be used?</p> <ul style="list-style-type: none"> <li>A. OS Module</li> <li>B. Path Module</li> <li>C. Fs Module</li> <li>D. Dir module</li> </ul> <p>Ans. B</p>
22.	<p>Which of the following operations can we do with the NodeJS events module?</p> <ul style="list-style-type: none"> <li>A. Create</li> <li>B. List</li> <li>C. First</li> <li>D. next</li> </ul> <p>Ans. A</p>
23.	<p>What is the way to store local variables that can be accessed within the application?</p> <ul style="list-style-type: none"> <li>a. Using Config file</li> <li>b. Using database</li> <li>c. Using app.locals</li> </ul>

	<p>d. Using app.storage</p> <p>Answer: C</p>
24.	<p>Which of the following is a middleware in Express.js?</p> <p>a. <code>function(req){ }</code>  b. <code>method(req){ }</code>  c. <code>function(req,res,next){ }</code>  d. <code>method(req,res,next){ }</code></p> <p>Answer: C</p>
25.	<p>Which of the following is a middleware that parses cookies attached to the client request object?</p> <p>a. cookie  b. cookies  c. cookie-parser  d. parser</p> <p>Answer: C</p>

- Write a React component that features a controlled input element for capturing a user's name. Implement the required state and event handlers to make sure the input value is managed by the component's state. Additionally, include a submit button that logs the entered name to the console when clicked.
- Create a React application that features a parent component (ListManager) and a stateless functional child component (ListItem). The ListManager component should maintain a list of items with each item having a name and a delete button. The ListItem component should be responsible for displaying an individual item and should communicate with the ListManager to remove itself from the list when its delete button is clicked.
  - Implement the ListItem component as a stateless functional component.
  - Use the ListManager component to manage an array of items, each represented by the ListItem component.
  - Allow each ListItem component to trigger the removal of its corresponding item from the list when its delete button is clicked.
- Create a React project with the following file structure: In the Navbar.js file, create a functional component called Navbar that renders a navigation bar with links to Home, About, and Contact. In the MainContent.js file, create a functional component called MainContent that

renders the main content of the website. Include a section with a heading and some dummy text. In the Sidebar.js file, create a functional component called Sidebar that renders a sidebar with links to different sections of the website. In the Footer.js file, create a functional component called Footer that renders a footer element with the text "© 2024 My Website". In the App.js file, import the Navbar, MainContent, Sidebar, and Footer components from their respective files. Render these components within a parent div element in the following order: Navbar, Sidebar, MainContent, and Footer.

4. Create a React form that captures a user's email and password. Implement form validation to ensure the email follows a valid format (e.g., contains an '@' symbol) and the password meets specific criteria (e.g., minimum length). Display appropriate error messages when validation fails, and enable form submission only when the input meets the validation criteria. Provide a detailed explanation of your validation approach and any libraries used.
5. Build a React application with three functional components: Title, Description, and Page. Start by defining the Title component in the Title.js file to render a static title, such as "Welcome to My React App." Next, create the Description component in the Description.js file to display a static description, such as "Explore the world of React components." Finally, compose both the Title and Description components within the Page component, implemented in the Page.js file. The Page component should render a cohesive webpage with a title and description without explicitly passing any data as props between the components. Ensure your code adheres to React best practices, maintains an organized project structure, and consider incorporating additional styling or content for an enhanced user experience.
6. Create a React functional component named UserInfo that accepts two properties, username and userRole, from its parent component. Inside the UserInfo component, display the provided username and userRole within a paragraph element. Ensure the component structure adheres to React best practices. This component is designed to be reusable and can dynamically showcase user information based on the data passed to it from the parent component.
7. Create a React component for a simple login form that includes fields for username and password. Implement client-side form validation to ensure both fields are non-empty before allowing form submission. Display appropriate error messages if the fields are empty and prevent form submission until they are filled.
8. Develop a React form component (StyledForm) that includes input fields for username, email, and password. Implement styling using both native CSS and a SaaS (e.g., SCSS) preprocessor. The form should include client-side validation for the email field, checking for a valid email

format. The form should display appropriate visual feedback for validation errors, such as changing the border color of the input field.

- a. Create the StyledForm component using both native CSS and a SaaS preprocessor for styling.
  - b. Implement input fields for username, email, and password within the form.
  - c. Add client-side validation specifically for the email field to ensure it follows a valid email format.
  - d. Apply styling changes to visually indicate validation errors, focusing on the email input field.
9. Create three React functional components: Title, Description, and Page. In Title.js, develop a component that renders a static title, like "Welcome to My React App." In Description.js, create a component that displays a static description, such as "Explore the world of React components." Then, in Page.js, compose both the Title and Description components. The Page component should render a page with a title and description without explicitly passing any data as props. Ensure the project structure is organized, adheres to React best practices, and displays the composed components on a webpage. Additional styling or content can be applied to enhance the page's appearance and user experience.
10. Explain the concept of indexing in MongoDB and how it can be used for query optimization. Provide an example scenario where indexing would significantly improve the performance of a MongoDB query.
11. Explain in simple terms what a MongoDB document is and how it differs from a traditional row in a relational database. Additionally, describe how you would query a MongoDB collection to find all documents where the "status" field equals "active."
12. Implement a Node.js script that utilizes the built-in events module. Create an EventEmitter object that emits a custom event, for example, "dataReceived," with some payload. Develop a listener function that logs the received payload when the custom event is emitted. Demonstrate the functionality by emitting the custom event and observing the log output.
13. Define the concept of middleware in the context of Express.js. Discuss the role of middleware functions and how they are used in an Express.js application. Provide examples of common use cases for middleware, such as logging, authentication, and error handling



14. Develop an Express.js application that exposes RESTful endpoints for a simple resource, such as "books." Create routes to handle GET, POST, PUT, and DELETE requests for the "books" resource. Use an in-memory array as a data store for simplicity, and implement appropriate responses for each HTTP method.
15. Explain the concept of the Node.js event loop and how it enables asynchronous operations in Node.js. Discuss the importance of non-blocking I/O operations in the event loop. Provide an example scenario where the event loop enhances the performance of a Node.js application.
16. Create a Node.js script that reads the content of a text file named "example.txt" using the File System module. Log the file's content to the console. Implement error handling to log an error message if the file reading operation fails.
17. Describe the purpose of module.exports in Node.js and how it aids in creating modular code. Explain the difference between exporting a single function or object versus exporting multiple functions or objects. Discuss how using modules enhances code maintainability and reusability in Node.js applications.

### **Task 1: To-Do List Application using React**

#### **Requirements:**

---

##### **1. Task List Component (5 marks):**

- Create a React component (**TaskList**) to display a list of tasks.
- Use state to manage an array of tasks, with each task having a title and a completion status (completed or not).
- Display each task in the list along with options to edit, delete, and mark as complete.

---

##### **2. Task Form Component (5 marks):**

- Implement a form component (**TaskForm**) to add new tasks to the list.
- Allow users to input a task title and add it to the task list.
- Ensure that the form updates the task list without requiring a page reload.

---

##### **3. Task Editing and Deletion (3 marks):**

- Enable editing functionality for each task. When a user clicks the "edit" option, allow them to modify the task title.
- Implement a delete option to remove a task from the list.

---

##### **4. Task Completion (2 marks):**

- Implement a feature to mark tasks as completed. Completed tasks should visually differentiate from incomplete tasks.
-

## Task 2: Shopping Cart Application using React

### Requirements:

---

#### 1. ProductList Component (5 marks):

---

- Create a **ProductList** component that displays a list of products, each with properties like id, name, price, and an "Add to Cart" button.
  - Implement state or a function in **ProductList** to handle the addition of products to the shopping cart.
- 

#### 2. ShoppingCart Component (5 marks):

---

- Develop a **ShoppingCart** component responsible for displaying the items added to the cart and calculating the total price.
  - Ensure seamless passing of data between components so that when a user clicks "Add to Cart" on a product in **ProductList**, the product details are passed to **ShoppingCart**, dynamically updating the displayed items.
- 

## Task 3: Recipe Explorer Application using React

### Requirements:

---

#### 1. Recipe List Component (5 marks):

---

- Create a **RecipeList** component to display a list of recipes.
  - Populate the list with mock recipe data, including properties like recipe name, category, and a brief description.
- 

#### 2. Category Filtering (4 marks):

---

- Implement filtering functionality to allow users to filter recipes based on categories (e.g., appetizers, main courses, desserts).
  - Provide UI controls, such as buttons or dropdowns, to select different recipe categories.
- 

#### 3. Recipe Details Component (4 marks):

---

- Create a **RecipeDetails** component to display detailed information about a selected recipe.
  - When a user clicks on a recipe in the list, show the details of that recipe, including ingredients and instructions.
- 

#### 4. Search Functionality (2 marks):

---

- Implement a search bar that allows users to search for recipes by name.
  - Ensure that the list dynamically updates as the user types in the search bar.
- 

## Task 4: React Application with MongoDB Interaction

### Requirements:

---

#### 1. Server-Side Setup (5 marks):

---

- Set up a Node.js server using Express to handle backend operations.
- Connect the server to a MongoDB database using a suitable MongoDB Node.js driver (e.g., Mongoose).

- Create a MongoDB schema for the data you'll be storing (e.g., a schema for blog posts).

---

#### 2. Read from MongoDB (5 marks):

---

- Develop a **DataDisplay** component that retrieves data from the MongoDB database using an API endpoint on the server.
- Display the retrieved data in a visually appealing manner within the **DataDisplay** component.

---

#### 3. Write to MongoDB (5 marks):

---

- Create a **DataEntryForm** component that allows users to input new data.
  - Implement functionality within **DataEntryForm** to send the entered data to the server, which, in turn, stores it in the MongoDB database.
  - Ensure proper validation and error handling during the data submission process.
- 

### Task 5: Interactive Quiz Application using React

#### Requirements:

---

##### 1. Quiz Component (6 marks):

---

- Create a **Quiz** component to display a set of questions.
- Fetch the quiz data from a mock API endpoint or store it locally as a JSON object.
- Display each question along with multiple-choice answers.
- Use React state to manage the user's selected answers.

---

##### 2. Answer Selection (4 marks):

---

- Allow users to select one answer per question.
- Highlight the selected answer visually to provide feedback to the user.

---

##### 3. Submission and Feedback (4 marks):

---

- Implement a submission mechanism that checks the selected answers against the correct answers.
- Provide immediate feedback to the user, indicating which answers were correct and incorrect.
- Display the user's overall score at the end of the quiz.

---

##### 4. Restart Quiz (1 mark):

---

- Include a button to restart the quiz, allowing users to retake it.
- 

### Task 6: React Application with MongoDB Integration

#### Requirements:

---

##### 1. Reading Data Component (5 marks):

---

- Create a **DataDisplay** component responsible for fetching and displaying data from MongoDB.
- Utilize a library like Axios or the built-in fetch API to make HTTP requests to the MongoDB database.
- Display the fetched data in the component's UI, ensuring proper error handling and loading indicators.

---

##### 2. Writing Data Component (5 marks):

---

- Develop a **DataForm** component that allows users to input new data entries.
- Implement form fields for relevant data attributes and a submit button to add the new entry.
- Upon submission, send a POST request to MongoDB to save the new data entry to the database.

---

### 3. Integration and Composition (5 marks):

---

- Create a parent component called **MongoDBIntegration** that composes both the **DataDisplay** and **DataForm** components.
  - Ensure that the **MongoDBIntegration** component manages the state of the application and passes necessary data and functions as props to child components.
  - Test the application to ensure seamless integration between reading and writing operations with MongoDB.
-