Frontend Development + DBA powered by Mongo DB

SmartBridge Educational Services Pvt Ltd.

TOTAL DURATION :	60 HRS
MODE OF DELIVERY	Virtual Instructor-Led Training + Self-Paced Learning
TRAINER TO STUDENT RATIO:	1:1000
TOTAL MARKS:	75

Virtual Instructor-Led Training Duration	20 HRS
Self-Paced Learning Duration	20 HRS
Project Development Duration	20 HRS
Total Duration	60 HRS

TABLE 1	
OVERALL COURSE OBJECTIVE:	● Equip participants with comprehensive expertise in MongoDB administration, encompassing foundational knowledge, practical skills, and advanced techniques ● required to proficiently manage and optimize MongoDB databases across various deployment environments. ● Enable participants to understand the MongoDB's architecture, data modeling, administration tools, and security measures, fostering the ability to perform essential operations competently. ● Empower individuals with advanced skills in areas like replication, sharding, ● performance optimization, monitoring, troubleshooting, and deploying MongoDB in scalable and high-availability setups. ● Cultivate proficiency in designing, implementing, and maintaining MongoDB ● databases, ensuring optimal performance, security, and scalability while effectively handling real-world scenarios and challenges. ● Gain knowledge of how frontend and backend components interact & Comprehension of the client-server architecture. ● Develop a strong foundation in HTML and CSS for creating well-structured and visually appealing web pages. ● JavaScript Proficiency: Master JavaScript fundamentals, including variables, functions, control structures, and asynchronous programming. ● Responsive Design: Learn to design and implement responsive web pages that adapt seamlessly to various screen sizes and devices. ● Frontend Development with React.js: Proficiency in building user interfaces using React.js.

Frontend Development + DBA powered by Mongo DB

LEARNING OUTCOME:	● Gain foundational skills in frontend development by mastering HTML/CSS basics, advanced techniques like flexbox and grid layouts, and DOM manipulation in this introductory Full Stack course. ● Gain essential JavaScript skills, mastering variables, control flow statements, ES6 features, and asynchronous programming. Learn to manipulate the DOM, handle events, and create dynamic web experiences. ● Gain proficiency in setting up a development environment, mastering JSX syntax, creating reusable components, and efficiently passing data using props. You'll learn to implement conditional rendering, handle events, manage state, and work with forms effectively. Additionally, you'll optimize performance using lifecycle methods and explore React hooks for efficient state management. ● Database Administration Expertise: Gain a comprehensive understanding of MongoDB architecture, installation, configuration, and deployment to manage and optimize database performance effectively. ● Data Management Proficiency: Develop skills in designing schemas, managing collections, and handling large datasets, ensuring data integrity and scalability. ● Advanced Troubleshooting and Security: Learn techniques for diagnosing performance bottlenecks, implementing backup and recovery strategies, and applying robust security measures to protect data. ● Operational Automation: Master the use of MongoDB tools for monitoring, automation, and replication to streamline database operations and ensure high availability.

TABLE 2: MODULE-WISE COURSE CONTENT AND OUTCOME

SL.NO	MODULE NAME	MODULE CONTENT	MODULE LEARNING OUTCOME	DURATION (HRS)
1	**Getting Started with MongoDB (Basics & Indexes)**	**MongoDB and the Document Model** Introduction to MongoDB The MongoDB Document Model **MongoDB Data Modeling Intro** Introduction to Data Modeling Types of Data Relationships Modeling Data Relationships Embedding Data in Documents Referencing Data in Documents Scaling a Data Model **The MongoDB Shell** Installing and Connecting to the MongoDB Shell Configuring the MongoDB Shell MongoDB Shell Tips and Tricks **Connecting to a MongoDB Database** Using MongoDB Connection Strings	Gain a comprehensive understanding of MongoDB's core concepts, including its document-based model, data modeling strategies, and CRUD operations. You will learn to manage databases, collections, and documents effectively using tools like Atlas Data Explorer and MongoDB Shell, as well as establish secure connections to MongoDB clusters. Additionally, you will master advanced features such as indexing strategies, including compound, wildcard, and time-series	12

		Connecting to a MongoDB Atlas Cluster with the Shell Connecting to a MongoDB Atlas Cluster with Compass Connecting to a MongoDB Atlas Cluster from an Application Troubleshooting MongoDB Atlas Connection Errors **MongoDB CRUD Operations** Inserting Documents in a MongoDB Collection Finding Documents in a MongoDB Collection Replacing a Document in MongoDB Updating MongoDB Documents Deleting Documents in MongoDB Sorting and Limiting Query Results in MongoDB Returning Specific Data from a Query in MongoDB Counting Documents in a MongoDB Collection **MongoDB Indexes** Using MongoDB Indexes in Collections Creating a Single Field Index in MongoDB Creating a Multikey Index in MongoDB Working with Compound Indexes in MongoDB Deleting MongoDB Indexes How Indexes Work Index Usage Details via Explain Optimized Compound Indexes Wildcard Indexes Partial Indexes Sparse Indexes Clustered Indexes Time Series Collections How to Monitor Indexes **MongoDB Logging Basics** MongoDB Logs in Atlas MongoDB Logs on Self-Managed Instances MongoDB Log Events MongoDB Server Log Customizations MongoDB Server Log Rotation and Retention	indexes, and learn to optimize queries for performance. Finally, you will acquire knowledge of MongoDB's logging mechanisms, enabling you to monitor, customize, and troubleshoot database performance across managed and self-hosted environments.	

2	**MongoDB Database Administration & Data Security**	**MongoDB Database Administrator Tools** Get Started with DBA Tools Backup Tools Restore Tools Data Export Tools Data Import Tools Diagnostic Tools: mongostat Diagnostic Tools: mongotop **Self-Managed Server Administration** Managing MongoDB Servers Configuring MongoDB Servers Connecting to MongoDB Servers Logging Basics for MongoDB Servers **MongoDB Database Metrics & Monitoring** Core Metrics More Metrics Monitoring M10+ Configure Alerts Respond to Alerts Integrations Self-Managed Monitoring Command Line Metrics **Self-Managed Backup & Recovery** Backup Plans on a MongoDB Server Filesystem Snapshots on a MongoDB Server Filesystem Snapshot Volumes on a MongoDB Server Filesystem Archives on a MongoDB Server Backing Up a MongoDB Deployment Restoring a MongoDB Deployment **Replication in MongoDB** Introduction to Replication Replication in MongoDB Automatic Failover and Elections with MongoDB Deployments The MongoDB Operation Log Read and Write Concerns with MongoDB Deployments Deploying a Replica Set in a MongoDB Deployment Configuring a Replica Set in a MongoDB Deployment **Self-Managed Database**	Develop essential skills for administering MongoDB databases, both in self-managed and managed environments. You will learn to use key database administrator tools for backup, restoration, data import/export, and diagnostics with utilities like mongostat and mongotop. The module will enhance your understanding of server management, monitoring, and alert configuration, enabling you to track core metrics and respond effectively to performance issues. You will also gain expertise in backup and recovery strategies, replication, and high availability through replica sets, ensuring robust data protection and system reliability. Additionally, you will acquire critical knowledge of MongoDB security practices, including authentication, authorization, and auditing, to safeguard your deployments.	8

Frontend Development + DBA powered by Mongo DB

		Security Introduction to Security Enabling Authentication for a Self-Managed MongoDB Deployment Establishing Authorization for a Self-Managed MongoDB Deployment Security Auditing in MongoDB		
3	**Introduction to Frontend & Basic Web Technologies(HTML, CSS,Javascript)**	What is Frontend ? Roles & Responsibility of Frontend Developer Intro to Web Development & Basics Web Technologies Introduction to HTML Basic HTML structure Introduction to CSS Basic CSS syntax Basic elements, DOM-create/delete elements. Selectors. Advanced CSS techniques like flexbox and grid Best practices for HTML and CSS development Introduction to Javascript Variables, datatypes, and operators Control flow statements (if-else, for, while, switch) Introduction to ES6 (let, const, template strings) Arrow function, Spread operator, destructing,Callback, Promise. JavaScript fundamentals: functions, objects, arrays Manipulating the DOM with JavaScript Handling events and user interactions with JavaScript	Gain a strong foundation in front-end development, mastering essential web technologies such as HTML, CSS, and JavaScript. You will learn to create and structure web pages using HTML, style them effectively with CSS, including advanced techniques like Flexbox and Grid, and follow industry best practices. Through JavaScript, you will understand core programming concepts, ES6 features, and techniques for DOM manipulation, event handling, and user interaction. This module will equip you with the skills to build responsive, dynamic, and user-friendly web interfaces while understanding the roles and responsibilities of a front-end developer in modern web development.	7
4	**Getting Started with React**	**Introduction to React** - Basics, component based Architecture, virtual DOM **Setting up the development environment -** Node & npm installation, code editor installation & configuartion, Development environment **Create React App** (using npm CLI) - Understand Project structure and file organization **Components:** Functional and class components. **Props:** Passing data to	Gain a solid foundation in React, including understanding its component-based architecture and virtual DOM. Gain proficiency in setting up a React development environment and creating projects using Create React App. Learners will master key concepts such as functional and class components, props, state management, lifecycle	10

		components. **State:** Managing state within components. **Lifecycle Methods:** Component lifecycle in class components (componentDidMount, componentDidUpdate, etc.). **Event Handling:** Handling user inputs and events. **Conditional Rendering:** Rendering elements based on conditions. **Lists and Keys:** Rendering lists and understanding the importance of keys. **Forms:** Controlled vs. uncontrolled components. **Hooks:** useState, useEffect, Custom Hooks, useContext **React Router** Setting up React Router Route and Link components Nested Routes	methods, event handling, and conditional rendering. Additionally, they will explore advanced features like hooks (e.g., useState, useEffect, and custom hooks) and learn to build dynamic, interactive web applications using React Router for navigation and routing.	
5	**Integrating APIs and Backend Communication & Implementation**	Fetching data with Fetch API and Axios CRUD operations Handling API responses and errors Using Async/Await in React Authentication and Authorization Implementing a use-case	Acquire the ability to fetch and manipulate data in React applications using Fetch API and Axios. They will become proficient in performing CRUD operations, effectively handling API responses and errors, and leveraging Async/Await for asynchronous programming. The module also covers essential concepts of authentication and authorization, enabling learners to build secure applications, culminating in the implementation of a practical use case.	8

TABLE 3: OVERALL COURSE LEARNING OUTCOME ASSESSMENT CRITERIA AND USE-CASES		
LEARNING OUTCOME	ASSESSMENT CRITERIA	USE-CASES

Frontend Development + DBA powered by Mongo DB

Gain a comprehensive understanding of MongoDB's core concepts, including its document-based model, data modeling strategies, and CRUD operations. You will learn to manage databases, collections, and documents effectively using tools like Atlas Data Explorer and MongoDB Shell, as well as establish secure connections to MongoDB clusters. Additionally, you will master advanced features such as indexing strategies, including compound, wildcard, and time-series indexes, and learn to optimize queries for performance. Finally, you will acquire knowledge of MongoDB's logging mechanisms, enabling you to monitor, customize, and troubleshoot database performance across managed and self-hosted environments.	● Demonstrate understanding of MongoDB's document-based model, CRUD operations, and effective management of databases, collections, and documents using tools like Atlas Data Explorer and MongoDB Shell. ● Showcase proficiency in implementing advanced indexing strategies (e.g., compound, wildcard, and time-series indexes) and optimizing query performance. ● Apply knowledge of MongoDB's logging mechanisms to monitor, customize, and troubleshoot database performance in both managed and self-hosted environments.	**Labs:** ● Inserting Documents in a MongoDB Collection ● Finding Documents in a MongoDB Collection ● Finding Documents by Using Comparison Operators ● Querying on Array Elements in MongoDB ● Finding Documents by Using Logical Operators ● Replacing a Document in MongoDB ● Updating MongoDB Documents by Using updateOne() ● Updating MongoDB Documents by Using findAndModify() ● Updating MongoDB Documents by Using updateMany() ● Deleting Documents in MongoDB ● Sorting and Limiting Query Results in MongoDB ● Returning Specific Data from a Query in MongoDB
Develop essential skills for administering MongoDB databases, both in self-managed and managed environments. You will learn to use key database administrator tools for backup, restoration, data import/export, and diagnostics with utilities like mongostat and mongotop. The module will enhance your understanding of server management, monitoring, and alert configuration, enabling you to track core metrics and respond effectively to performance issues. You will also gain expertise in backup and recovery strategies, replication, and high availability through replica sets, ensuring robust data	● Demonstrate proficiency in administering MongoDB databases, including backup, restoration, data import/export, and diagnostics using tools like mongostat and mongotop. ● Showcase the ability to configure and manage replica sets for high availability, implement effective backup and recovery strategies, and monitor key metrics to address performance issues proactively. ● Apply MongoDB security best practices, including configuring authentication, authorization, and auditing, to ensure secure and reliable database deployments.	**Labs:** ● Backup Tools ● Restore Tools ● Data Export Tools ● Data Import Tools ● Diagnostic Tools: mongostat ● Diagnostic Tools: mongotop ● Diagnostic Tools: bsondump ● MongoDB as a Filesystem ● Managing MongoDB Servers ● Configuring MongoDB Servers ● Connecting to MongoDB Servers ● Logging Basics for MongoDB Servers

protection and system reliability. Additionally, you will acquire critical knowledge of MongoDB security practices, including authentication, authorization, and auditing, to safeguard your deployments.		
● Gain a strong foundation in front-end development, mastering essential web technologies such as HTML, CSS, and JavaScript. You will learn to create and structure web pages using HTML, style them effectively with CSS, including advanced techniques like Flexbox and Grid, and follow industry best practices. Through JavaScript, you will understand core programming concepts, ES6 features, and techniques for DOM manipulation, event handling, and user interaction. This module will equip you with the skills to build responsive, dynamic, and user-friendly web interfaces while understanding the roles and responsibilities of a front-end developer in modern web development.	● Proficiency in HTML/CSS and DOM Manipulation: Evaluating proficiency in HTML/CSS fundamentals, including creating/deleting DOM elements, and applying advanced CSS techniques like flexbox and grid layouts. ● Demonstrate proficiency in acquiring essential knowledge for ES6 like arrow functions, spread operator, rest operator,etc. ● Acquire knowledge of promises which will help in understanding asynchronous programming. And also get to know about Document Object Model (DOM).	**Use Case: 1** Form Validation with JS **Scenario:** Eva, an experienced web developer, is tasked with designing a user-friendly registration form for a new website. The form includes fields like username, password, email, address, and more, essential for account creation. To ensure data accuracy and improve user experience, Eva integrates JavaScript-based client-side validation, providing real-time feedback to minimize errors. Her goal is to create a robust, user-centric form that enhances the registration process and sets a positive tone for user interactions. **Tasks :** Creating a functional and visually appealing form by developing three interconnected files. First, design and structure the form elements, such as input fields, labels, buttons, and other necessary components, in an HTML file. Next, craft a CSS file to enhance the visual appearance and layout of the form, ensuring a user-friendly interface. Finally, implement a JavaScript file to provide client-side validation, writing functions to check user input against predefined criteria, such as required fields, valid email addresses, and password strength. Utilize event listeners to trigger validation during user interactions, ensuring the form meets the specified rules.

Frontend Development + DBA powered by Mongo DB

		Use Case:2 Weather App with Javascript **Scenario:** Emma, a traveler, relies on the Weather Information Application to plan her outdoor activities during her vacation. She inputs her destination city to check the weather forecast, ensuring a pleasant and enjoyable trip without unexpected weather disruptions. **Tasks:** To create a weather application, start by obtaining an API key from OpenWeatherMap to access weather data. Design the HTML structure, including input fields for user queries and display areas for weather information. Write JavaScript code to fetch data from the OpenWeatherMap API based on user input, handle API responses, and dynamically update the UI with the retrieved information. Implement error handling to manage cases where the entered city is not found, ensuring a seamless user experience. Finally, style the application using CSS to enhance its visual appeal and usability.
Gain a solid foundation in React, including understanding its component-based architecture and virtual DOM. Gain proficiency in setting up a React development environment and creating projects using Create React App. Learners will master key concepts such as functional and class components, props, state management, lifecycle methods, event handling, and conditional rendering. Additionally, they will explore advanced features like hooks (e.g., useState, useEffect, and custom hooks) and learn to build	● Demonstration of React Concepts: Assess the implementation of JSX syntax, component creation, data handling with props, state management, form usage, and understanding of the React lifecycle. ● Proficiency with React Hooks: Evaluate the adeptness in utilizing React hooks, encompassing both built-in and custom hooks, for effective state management and performance optimization in React applications.	**Use Case 1:** Digital timer **Scenario:**You are tasked with developing a productivity application that incorporates the Pomodoro Technique—a time management method that uses a timer to break work into intervals, traditionally 25 minutes in length, separated by short breaks. The application should allow users to track their work sessions, take breaks, and customize timer intervals. **Task**: The task involves implementing a digital timer component with essential functionalities. Users can start, pause, and reset the timer, as well as set custom time limits according to their

dynamic, interactive web applications using React Router for navigation and routing.		needs. The timer will be displayed in a visually appealing format, ensuring a user-friendly experience. Additionally, the timer will accurately track time intervals, maintaining precision throughout its operation. **Use Case 2:** Color picker **Scenario:** Sophia encounters challenges during her interaction with the Color Picker application, hindering the seamless selection and preview of colors. Sophia experiences difficulties in properly passing color data from the ColorPicker component to the parent component (App) via props, resulting in inconsistencies or delays in updating the color preview. Sophia observes discrepancies between the color displayed in the color picker component and the color preview area, indicating potential errors in color representation or data transmission. **Task:** Sophia needs to address two key tasks to improve the functionality of the ColorPicker component. First, she will enhance prop data handling by optimizing the communication between the ColorPicker component and the parent component (App), ensuring smooth and efficient data transmission. Second, Sophia will investigate and resolve any issues with color representation, ensuring that the color displayed in the preview area accurately reflects the user's selection in the color picker.

Frontend Development + DBA powered by Mongo DB

Acquire the ability to fetch and manipulate data in React applications using Fetch API and Axios. They will become proficient in performing CRUD operations, effectively handling API responses and errors, and leveraging Async/Await for asynchronous programming. The module also covers essential concepts of authentication and authorization, enabling learners to build secure applications, culminating in the implementation of a practical use case.	● Data Handling and CRUD Operations: Demonstrate proficiency in fetching data using Fetch API and Axios, performing CRUD operations, and managing state dynamically in a React application. ● Error Management and Async/Await: Effectively handle API responses and errors while utilizing Async/Await for seamless asynchronous programming. ● Authentication and Use-Case Implementation: Successfully implement authentication, authorization, and a practical use-case project integrating secure data handling and dynamic functionality.	**Use Case :1** Random Quote Generator **Scenario:** Emma, an enthusiast of motivational quotes, discovers a Random Quote Display project online but finds its lack of interactive features frustrating. Despite her interest, she's unable to easily refresh the page or generate new quotes, hindering her browsing experience. Emma desires seamless interaction, envisioning clickable buttons or swipe gestures for effortless navigation. Feeling dissatisfied, she considers providing feedback and ultimately seeks alternatives that prioritize user engagement. **Task:** The tasks involve improving the functionality and user experience of an application by focusing on four key areas. First, pagination logic is implemented to display a limited number of users per page, ensuring optimal performance. Next, user interaction controls are developed, allowing users to easily refresh the page or generate a new random quote by interacting with buttons or other controls in the interface. The integration with the external API is enhanced to fetch random quotes more efficiently, with error handling mechanisms in place to manage any API request failures and provide appropriate feedback to users. Finally, the user interface is improved by enhancing the visual presentation of quotes, using frameworks like Bootstrap to optimize the layout and design for better readability and aesthetics. **Use Case :2** Dog Breed Fetching **Scenario:** Ethan encounters challenges during the usage of the React project, hindering his ability to explore and interact with dog breeds effectively. Ethan experiences difficulties in integrating the API to fetch data

Frontend Development + DBA powered by Mongo DB

		about dog breeds, leading to errors or inconsistencies in displaying breed information and photos. Ethan encounters issues with the functionality of React components such as useState and useEffect, affecting the proper rendering and updating of UI elements. **Task:** Ethan needs to address two main tasks to resolve the issues. First, he will troubleshoot API integration problems by reviewing the implementation code and ensuring that API requests and responses are handled properly. Second, he will debug the functionality of React components, specifically focusing on useState and useEffect, to identify and fix any errors or inconsistencies in state management and component rendering.

TABLE 4: LIST OF FINAL PROJECTS (10 PROJECTS THAT COMPREHENSIVELY COVER ALL THE LEARNING OUTCOME)	
SL.NO	**FINAL PROJECT**
1	Cryptoverse: A Cryptocurrency Dashboard
2	CodeCraft: A Dynamic Code Editor
3	RhythmicTunes: Your Melodic Companion
4	InsightStream: Navigate the News Landscape
5	FitFlex: Your Personal Fitness Companion
6	CookBook: Your Virtual Kitchen Assistant
7	GrocerEase: Simplifying Your Shopping Experience
8	TriviaTrek: Journey through a World of Questions
9	SpendSmart: Your Personal Expense Tracker
10	StreamSavvy: Your Ultimate Entertainment Destination

TABLE 5: COURSE ASSESSMENT RUBRICS (TOTAL MARKS: 75)				
ASSESSMENT CRITERIA	**DESCRIBE THE CRITERIA OF THE BELOW CATEGORY PERFORMANCE**			**TOTAL MARKS**
	FAIR	**GOOD**	**EXCELLENT**	
Problem Definition & Design Thinking	**3**	**5**	**8**	**10**

Frontend Development + DBA powered by Mongo DB

Innovation & Problem Solving	**1**	**2**	**4**	**5**
Implementation of Project	**6**	**12**	**18**	**20**
Performance of the Project	**1**	**2**	**4**	**5**
Project Demonstration & Documentation	**3**	**5**	**8**	**10**
MCQ-based assessment 25 Questions				**25**

References

1. https://learn.mongodb.com/learn/learning-path/mongodb-database-admin-path-self-managed-for-smartbridge
2. https://learn.mongodb.com/learning-paths/mongodb-nodejs-developer-path-for-smartbridge
3. https://react.dev/learn

JOB ROLES

1. Frontend Developer (React.js)
2. Software Developer
3. UI/UX Developer
4. Database Administrator