

HTML

- HTML Structure
- HTML Elements
 - Block-level Elements
 - Inline Elements
 - HTML Lists
 - Void Elements
 - HTML Forms
- HTML Attributes
- Multiple class names as an HTML attribute value
- Navigation within the webpage
- Getting URLs for our own Images
- Linking HTML and CSS Files

CSS

- CSS Properties
 - Text Properties
 - Background Properties
 - Font Properties
- CSS Box Model
- CSS Flexbox
- CSS Media Queries
- CSS Selectors
 - class selector
 - id selector
 - universal selector
- CSS Inheritance
- CSS Colors
- CSS Specificity & Cascade
- CSS Units - percentage, px, vh/vw
- CSS Gradients
- Inline Styles
- Reusability of CSS Rulesets
- Responsiveness

Bootstrap

- How to use Bootstrap
- Predefined Styles in Bootstrap
- Flexbox Properties
- Bootstrap Components
 - Carousel
 - Navbar
 - Modal
 - Spinner
 - Bootstrap Containers
- Bootstrap Utilities
 - Embed
 - Bootstrap Flex Utilities
 - Bootstrap Display Utilities
 - Bootstrap Position Utilities
- Bootstrap Grid System
 - Container
 - Row
 - Column
 - Creating Multiple Column Layouts
 - Column Wrapping
 - The Layout at different Breakpoints
- Bootstrap Spacing Utilities
- Bootstrap Background Color Utilities
- Bootstrap Sizing Utilities
- Bootstrap Icons

SQLite

- Data
- Database
- Database Management System(DBMS)
 - Advantages
- Types of Databases
 - Relational Database
 - Non-Relational Database
- SQL Querying
 - Create
 - Insert
 - Select
 - Update
 - Delete Rows
 - Drop Table
 - Alter

- Comparison Operators - =, >, <, >=, <=
- String Operations - LIKE
- Logical Operators - AND, OR, NOT
- IN, BETWEEN Operator
- ORDER BY, DISTINCT
- Pagination - LIMIT, OFFSET clauses
- Aggregations - COUNT, SUM, MIN, MAX, AVG
- Alias
- Group By, Group By with Having
- Expressions
- SQL Functions
- CASE Clause
- Set Operators
- Modelling Databases
 - Entity, Attributes, Attributes of an Entity, Key Attribute, Entity Type,
 - Relationships
 - One-to-One Relationship
 - One-to-Many or Many-to-One Relationship
 - Many-to-Many Relationship
 - Cardinality Ratio
 - Creating a Relational Database
 - Primary Key
 - Foreign Key
- JOINS
 - Natural JOIN, Inner JOIN, Left JOIN, Right JOIN, Full Join, Cross Join, Self Join
- Views
- Subqueries
- Transactions - ACID Properties
- Indexes
- SQL Syntax difference between SQLite, MySQL and Oracle

Python

- Variables and Data Types
- Sequence of Instructions
- Inputs and Outputs Basics
- Type Conversion
- String Slicing
- Operators
 - Relational
 - Logical
 - Arithmetic
- Conditional Statements
- Nested Conditional Statements
- Loops

- Nested Loops
- Loop Control Statements
- Patterns
- String Methods
- Comparing Strings & Naming Variables
- Rounding Numbers
- Floating Point Approximation
- Floating Point Errors
- Comments
- Lists, Tuple, Set, Dictionaries
- Tuples and Sequences
- List Slicing
- Nested Lists & String Formatting
- Functions
- Extended Slicing
- Arbitrary Function Arguments
- Built-in Functions
 - abs()
 - all()
 - any()
 - reversed()
 - enumerate()
- List Methods
 - copy()
 - reverse()
- Introduction to OOP
 - Objects
 - Classes
 - Attributes & Methods
 - Inheritance
 - Super Class & Sub Class
 - Composition
 - Overriding Methods
- Python Standard Library
- Built-in Functions
- Modules
- Map, Filter and Reduce
- Scope & Namespaces
- Errors & Exceptions
- Working With Dates & Times
- Problem Solving

JavaScript

- DOM and Event Fundamentals
 - Variables
 - Document Object Model (DOM)
 - Methods
 - Properties
 - Events
- Primitive Types & Conditionals
- Conversions
- String Manipulations
- Conditional Statements
- Comparison Operator
- Input Element and Math Functions
 - Math.random()
 - Math.ceil()
 - HTML Input Element
- DOM Properties
- DOM Manipulations
- More DOM Manipulations
 - Creating an HTML Element
 - Accessing HTML element using id attribute
 - Appending to an HTML Element
 - Adding Event Listeners Dynamically
 - Providing Class Names Dynamically
 - Removing Class Names Dynamically
- Data Structures
- Arrays
- Functions
- Objects
 - Object Destructuring
 - Object Methods
- Todos Application
- JavaScript Built-in Functions
- Storage Mechanisms
- JavaScript Object Notation (JSON)
 - JS Object vs JSON Object
 - JSON Methods
- Array Methods
- Callbacks & Schedulers
- Event Listeners
- JavaScript Advanced
 - Execution Context & Event Loop
 - Event Propagation & Bubbling
 - Hoisting
 - Event Delegation
 - Debouncing & Throttling
- Hypertext Transfer Protocol (HTTP)
 - HTTP Methods
 - HTTP Version

- HTTP Responses
 - Making HTTP Requests using Fetch
 - HTTP Response Object Properties and Methods
- Modern JS Concepts
 - Spread Operator
 - Rest Parameter
 - Functions
 - Template Literal
 - Ternary Operator
 - Switch Statement
 - Arrow Functions
- Factory Function
- Constructor Property
- Built-in Constructor Function
- Instance Methods
- this
- Immutable and Mutable Values
- Variables
- Prototypal Inheritance
 - Built-in Constructor Functions
 - Built-in Array Constructor Function
 - Prototype Property
 - Built-in Function Constructor Function
 - Instance Specific and Prototype Properties
- JS Classes
 - Class
 - Inheritance in JS Classes
 - this in classes
- JS Promises
 - Synchronous and Asynchronous Execution
 - JS Promises
 - Async/Await

React JS

1. **Components & Props**
 - **Component**
 - **Properties (Props)**
 - **Component is Reusable**
 - **Component is Composable**
 - **Third-party Packages**
 - **create-react-app**
 - **Pre-Configured tools**
2. **Lists & Keys**
 - **Keys**

- Keys as Props
 - Users List Application
- 3. **Class Component and State**
 - Components
 - Functional Components
 - Class Components
 - React Events
 - State
 - Updating State
 - State Updates are Merged
 - Functional Components vs Class Components
- 4. **Conditional Rendering**
 - Conditional Rendering
 - Using an If...Else Statement
 - Using Element Variables
 - Using Ternary Operators
 - Using Logical && Operator
 - Default Props
- 5. **Class Component and State**
 - `setState()` Object Syntax
 - Sending Function as Callback
 - Input Element
- 6. **Third-Party Packages**
 - Installing UUID
 - Importing of UUID
- 7. **Component Life Cycle**
 - Component Life Cycle Phases
 - Mounting Phase
 - Updating Phase
 - Unmounting Phase
- 8. **Routing using React Router**
 - Web Apps
 - React Routes
 - API Calls
 - Fetch
 - Route Props
 - Match
 - Introduction to React JS
 - React JS
 - Why React JS?
 - Advantages of React JS
 - Running JavaScript in HTML
 - Creating Elements using React JS
 - React CDN
 - `React.createElement()`
 - `ReactDOM.render()`
 - JSX
 - Babel

- Embedding Variables and Expressions in JSX
- Nesting JSX elements

9. Authentication & Authorization

- Client-Server Communication
 - Authentication
 - Authorization
- Authentication Flow
- Route Parameters
 - History

10. Authentication & Authorization - Part 2

- JWT Token
- Storage Mechanisms
- Cookies
 - Why Cookies?
 - Local Storage
 - Cookies vs Local Storage
 - Third Party Package js-cookie
- Redirect Component
- withRouter

11. Authentication & Authorization - Part 3

- Router Switch
- Wrapper Component
 - Protected Route

12. Authentication & Authorisation - Part 4

- Integrating APIs
- API Call Possible Views
 - Success View
 - Failure View
 - Loading View
- Best Practices
 - State Variable-isLoading
 - State Variable-apiStatus
 - Adding Initial State

13. React Concepts

- Reconciliation
- React Batch Updating
- setState() Syntax
- Children Prop
 - Passing Text as Children
 - Accessing Children

- Controlled vs Uncontrolled Input
- Props vs State
- State should be minimal
- User defined functions should be in the arrow function syntax
- Using Redirect Component in Callbacks or Event Listeners
- Using `history.replace()` in `render()` instead of Redirect Component
- Missing `withRouter()` to have history prop
- When request object keys are incorrect
- Not Updating the State when required
- `setState()` Callback Function
- React Icons
 - Installing React Icons
 - Searching React Icons
 - Importing React Icons

14. Third-Party Packages

- Third-Party Packages
 - Advantages
 - Selecting a Third-Party Package
- Third-Party Package
 - React Player
 - Recharts
 - React Chrono
 - React Slick
 - React Popup

15. React Context

- Prop Drilling
- React Context
 - Creating Context
 - Context Properties
 - How to Access the Value?
- Consumer Component
 - Understanding Syntax
- Provider
 - Syntax
- Context Flow For Windows App
- Best Practice
- Context
- Using Context in methods

16. Styled Components

- Styling React Components
 - Using CSS
 - CSS-in-JS
- Styled Components
- Advantages of Styled Components

- Extending Styles
- The "as" prop
- Boolean Attribute
- 17. Common Mistakes
 - Missing Export Statement
 - Missing Import Statement
 - Missing Extending the React Component Class
 - class vs className
 - onclick vs onClick
 - Event Listeners inside Class Component
 - Passing Event Handler
 - Modifying the State Directly
 - Calling setState() from render()
 - Invoking Event Handler
 - setState() is Asynchronous
 - React Component should return a single JSX element
 - Fragments
 - Props are Read-only
 - Rendering Promise Object
 - Using Link and Route Components without BrowserRouter
 - Providing Wrong Route Path
 - Missing exact keyword
 - Missing Switch
 - Placing Common component inside Switch
 - Providing the same PATH for multiple Routes
 - Missing Colon(:) while providing Path Parameters
 - Accessing Promise Object
 - Ordering of Routes inside Switch
- 18. Best Practice
 - Updating a Property of an Item inside List
- 19. Debugging using Developer tools
 - Debugging
 - Browser Developer Tools
 - React Developer Tools
- 20. Hooks
 - useState Hook
 - useEffect Hook
 - Making API calls with Hooks
 - Schedulers with Hooks and Race condition

Node JS

- Introduction to Node JS

- MERN Stack
- Node JS
- Running JavaScript Using Node JS
 - Node REPL (Read-Eval-Print-Loop)
 - Node CLI
- Module
 - Common JS Module Exports
 - Default Exports
 - Named Exports
 - Modern JS Module Exports
 - Default Exports
 - Named Exports
- Core Modules
 - Path
- Package
 - Node Package Manager (NPM)
- Steps to create a Node JS Project
- Third-Party Packages
 - date-fns
- Introduction to Express JS
 - HTTP Server
 - Server-side Web Frameworks
 - Express JS
 - Network Call using Express JS
 - Handling HTTP Request
 - Testing Network calls
 - Network Call to get a Today's Date
 - Network Call to get HTML content as an HTTP Response
 - Sending file as an HTTP Response
 - Application Programming Interface (API)
 - Database
 - SQLite
 - SQLite CLI
 - SQLite Methods
 - Open
 - Executing SQL Queries
 - SQL Third-party packages
 - Connecting SQLite Database from Node JS to get the books from Goodreads Website
 - SQLite Database Initialization
 - Goodreads Get Books API
 - SQLite Methods
 - get()
 - run()

- Node JS Third-party packages
 - Nodemon
- GoodReads API
 - Get Book
 - Add Book
 - Update Book
 - Delete Book
 - Get Author Books
- REST APIs
 - Get Books API
 - Filtering Books
 - REST APIs
 - Why Rest Principles?
 - REST API Principles
- Debugging Common Errors
 - Importing Unknown Modules
 - Starting Server in Multiple Terminals
 - Starting Server outside the directory
 - Accessing Wrong URL
 - Missing Function Call
 - Importing Unknown File
 - Request Methods
 - Missing colon(:)
 - Accessing Path Parameters
 - Query Formatting
 - Replacing SQLite Methods
 - HTTP Request URL
 - Missing '?'
 - Missing '&' between Query Parameters
 - Replacing '&' with ',' in Query Parameters
 - Accessing Unknown Database
 - Accessing Wrong Port Number
- Authentication
 - Installing Third-party package bcrypt
 - Goodreads APIs for Specified Users
 - Register User API
 - Login User API
 - Authentication Mechanisms
 - Token Authentication mechanism
 - Access Token
 - How Token Authentication works?
 - JWT
 - How JWT works?
 - JWT Package

- **Login User API by generating the JWT Token**
- **How to pass JWT Token?**
- **Get Books API with Token Authentication**
- **Middleware functions**
 - **Multiple Middleware functions**
- **Logger Middleware Implementation**
 - **Defining a Middleware Function**
 - **Logger Middleware Function**
 - **Get Books API with Logger Middleware**
- **Authenticate Token Middleware**
- **Get Books API with Authenticate Token Middleware**
- **Passing data from Authenticate Token Middleware**
- **Get User Profile API with Authenticate Token Middleware**

MongoDB

Non Relational Databases(MongoDB)

- Introduction to MongoDB
- CRUD operations
- Querying Documents
- Conditional Operators
- Ordering Results
- Sorting on Multiple Fields
- Pagination
- Logical Operators
- Grouping & Aggregations
- Nested Documents

ProductionReadyPractices

- Intro to Real World Projects
- Approach To Complete A Real-World Project
- Understanding Instructions
- Requirement Document Template
- Exploring & Selecting Design Files
- Exploring APIs
- Selecting APIs
- Updating API Details in the Requirement Document
- Exploring & Selecting Third-Party Packages
- Setting Up a Real-World Project
- Development & Testing a Real-World Project
- Deploying a Real-World Project on Vercel
- Deploying a Real-World Project on Netlify

- GitHub Best Practices
 - Create a GitHub Account
 - Generate a Personal Access Token
 - Create a Repository on GitHub
 - Set up Git
 - Initialize a Local Git Repository
 - Link Local & Remote Git Repositories
 - Stage and Commit Changes
 - Push Changes to Remote Git Repository
- Clean Code Guidelines

GitHub

- **Setting Up a README.md with setup and run instructions:** Error Handling and Debugging
- Hosting Platforms : render, Netlify, or Vercel.
- Code Cleanliness & Best Practices
- Staging Area and Commits
 - Creating a Commit
 - Working with Remote Repository