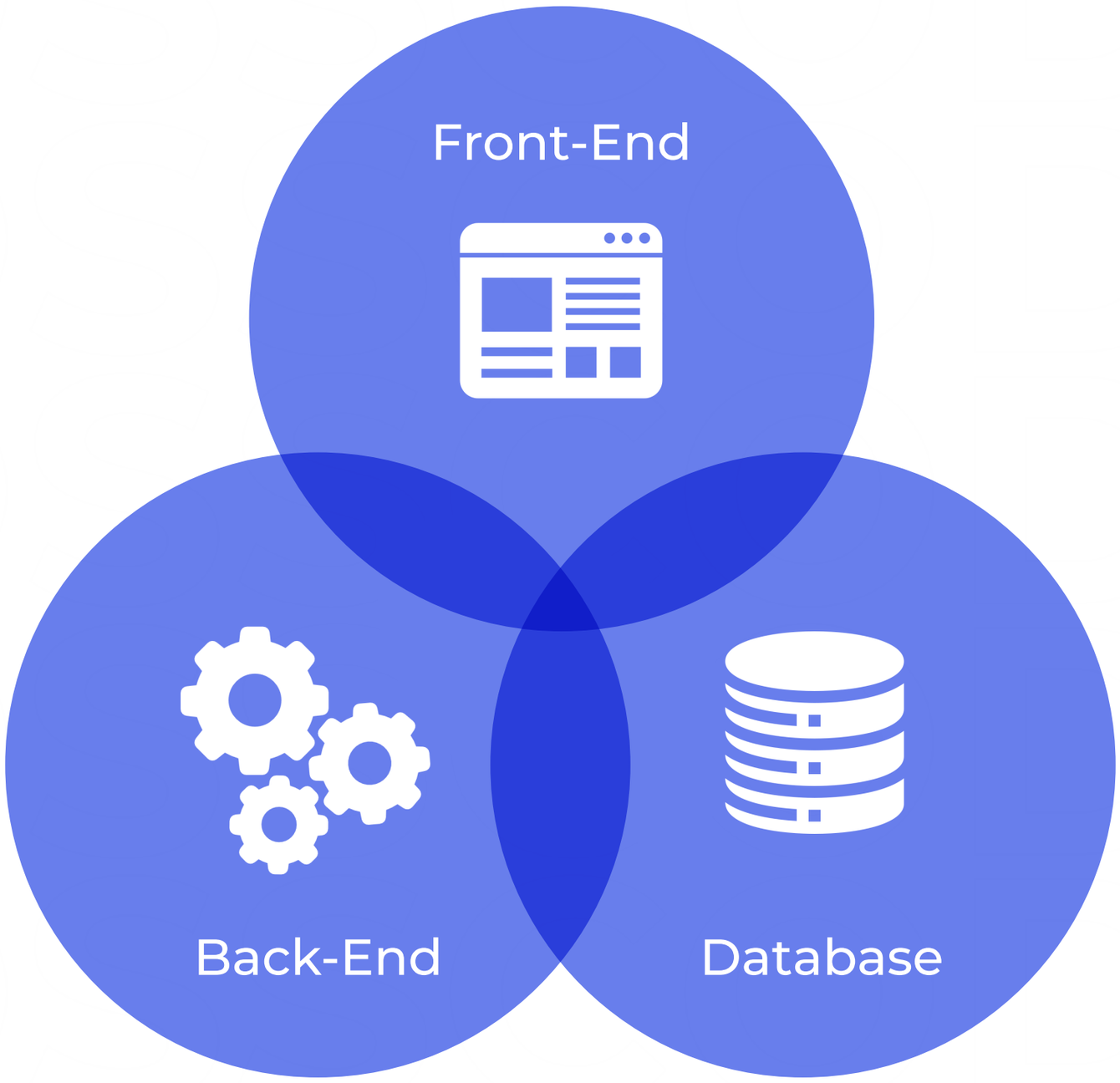
BOSSCODER ACADEMY

LEARN

Full Stack Development



IN 60 DAYS

Day 1-3

Understand How The Web Works

📃 Topics to Cover

- Client-server architecture
- Basics of HTTP
- IP Addressing, DNS
- Request Response Cycle

🗂️ Resources for Learning

1. How the Web Works
2. Basics of HTTP
3. Client Server Architecture
4. What is DNS? Domain Name System, DNS Server, and IP Address Concepts Explained

Questions

1. Imagine you're building a chat application. Is the chat server considered the client or the server in the client-server architecture? Explain your choice.
2. Perform a command in your terminal (e.g., nslookup) to retrieve the IP address of a specific domain name (e.g., www.example.com). Explain the output.
3. Describe the difference between HTTP and HTTPS.
4. Explain the difference between IPv4 and IPv6.
5. Open your browser's developer tools and inspect the network tab. Load a website and analyze the requests and responses exchanged during the page load.
6. In a client-server model, where does the processing of data primarily occur, on the client or the server?

Day 4-7

HTML and CSS

📃 Topics to Cover

- Introduction to HTML tags and elements
- HTML document structure
- CSS syntax and selectors
- Styling elements and layouts
- CSS Block Model
- Responsive design principles (Media Queries)

🗂️ Resources for Learning

1. https://www.freecodecamp.org/news/html-css-handbook-for-beginners/
2. https://www.w3schools.com/html/
3. https://www.w3schools.com/css/
4. https://developer.mozilla.org/en-US/docs/Learn/CSS
5. https://www.youtube.com/watch?v=G3e-cpL7ofc

Questions

1. Create an HTML structure for a simple blog post, including headings, paragraphs, images, and links.
2. Write a CSS selector to target all paragraphs within a specific class, and provide an example of a style you could apply to those paragraphs.
3. How can you style a specific HTML element using an inline style? Provide an example.
4. Create a CSS rule that sets the font color to red for all headings of a specific class.
5. Describe the purpose of media queries in responsive design. Write a media query that targets screens with a maximum width of 600px and adjusts the font size of a heading to 18px.

Day 8-11

Javascript

📃 Topics to Cover

- Data Types, Variables, Scope
- Conditionals, Loops
- Functions and Objects
- DOM Manipulation
- Event Handling, Call backs
- ES6 Javascript Syntax

🗂️ Resources for Learning

1. https://www.w3schools.com/js/
2. ttps://www.youtube.com/watch?v=ec8vSKJuZTk&t=82s
3. https://developer.mozilla.org/en-US/docs/Learn/JavaScript
4. https://www.youtube.com/watch?v=PkZNo7MFNFg

Questions

1. Explain the difference between null and undefined in JavaScript.
2. What is the difference between let, const, and var when declaring variables in JavaScript? Provide an example for each.
3. Explain the concept of variable scope in JavaScript. What is the difference between global and local scope?
4. Define a JavaScript function that takes two parameters and returns their sum. Call the function with different arguments.
5. Create an object called person with properties for name, age, and email. Access and log the name property.
6. Write JavaScript code to select a specific HTML element by its ID and change its text content.
7. Attach an event listener to an HTML element that listens for the mouseover event and changes the element's text color.

Day 12-14

Build a Basic Website with HTML CSS Javascript

1. **Structure and Content:**
   - Have a navigation bar with various tabs
   - Include sections like About Me, Projects and Contact
   - Use various tags and components for each of these sections
2. **Styling and Design:**
   - Use CSS to apply a color theme throughout the site.
   - Set a modern font for headings and paragraphs.
   - Add padding and margin for spacing.

3. **Interactivity with JavaScript:**
    - Smooth scroll to sections when clicking on navigation links.
    - Change background color on button clicks.
    - Validate contact form before submission.
4. **Responsive and Deployment:**
    - Make the website adjust its layout for mobile devices using media queries.
    - Deploy the website using GitHub Pages for easy online access.

Day 15-17

Version Control

📃 Topics to Cover

- Basic Git commands (init, add, commit, push, pull)
- Collaborating with GitHub repositories
- Branching and merging
- Troubleshooting with Git

📁 Resources for Learning

1. https://www.youtube.com/watch?v=RGOj5yH7evk
2. https://www.atlassian.com/git/tutorials/what-is-version-control
3. https://www.tutorialspoint.com/git/

Questions

1. Demonstrate the process of adding changes to the staging area using git add for a specific file.
2. Describe the process of pushing local commits to a remote repository using git push.
3. Explain the steps to clone a remote repository to your local machine using git clone.
4. Explain the process of merging a branch into the main/master branch using git merge.
5. Describe a scenario where you accidentally committed sensitive information. How can you remove the commit from history using git reset?
6. Demonstrate the process of forking a GitHub repository, making a change, and submitting a pull request.

Day 18-20

Introduction to Front-end Frameworks - React

📑 Topics to Cover

- Setting Up React.js
- React JSX
- React Components
- State and Props in React
- List and Keys in React

🗂️ Resources for Learning

1. https://www.freecodecamp.org/news/react-beginner-handbook/
2. https://reactnative.dev/docs/environment-setup
3. https://react.dev/learn
4. https://www.youtube.com/watch?v=bMknfKXIFA8

Questions

1. Write JSX code to render a <h1> element with the text "Hello, React!".
2. Build a class component named Counter that displays a count that increments on each button click.
3. Implement a functional component Person that receives a name prop and displays "Hello, [name]".
4. Create a class component Scoreboard with a state variable score. Display the score and update it when a button is clicked.
5. Construct a component ToDoList that takes an array of tasks as a prop and renders an unordered list of tasks.
6. Render a list of <li> elements with unique keys using the map() function.

Day 21-26

Advanced React Concepts

📃 Topics to Cover

- Form Handling
- Data Handling
- Virtual DOM
- React Hooks and Custom Hooks
- State Management
- Routing

🗂️ Resources for Learning

1. https://www.freecodecamp.org/news/react-beginner-handbook/
2. https://react.dev/learn
3. https://www.youtube.com/watch?v=bMknfKXIFA8
4. https://www.geeksforgeeks.org/learn-reactjs/

Questions

1. Explain controlled components and uncontrolled components in React.
2. Create a form with input fields for name and email. Handle form submission to log the input values to the console.
3. Implement form validation for an email input using the pattern attribute and display an error message if the input is invalid.
4. What is the role of props in passing data between components in React?
5. Implement data fetching using the fetch API or a library like Axios, and display the fetched data in a component.
6. Use the React DevTools to inspect the Virtual DOM and see how it differs from the actual DOM.
7. Convert a class component that uses state into a functional component using the useState hook.

8. Create a custom hook that manages local storage and can be used to store and retrieve data.
9. Explain the concept of "lifting state up" in React.
10. Build a counter application using local state and demonstrate how state management works in a single component.
11. Describe the purpose of React Router and its key components.

Day 27-31

Introduction to Back-end Frameworks - Node.js

📑 Topics to Cover

1. Setting up Node.js environment
2. Modules in Node.js
3. Asynchronous Concepts
4. Event loops, Callbacks
5. npm and Packages

🗂️ Resources for Learning

1. https://nodejs.dev/en/learn/
2. https://www.youtube.com/watch?v=TlB_eWDSMt4
3. https://www.w3schools.com/nodejs/

Questions

1. Why is it important to have a package manager like npm when working with Node.js?
2. Differentiate between built-in modules and external modules in Node.js.
3. Create a JavaScript file with a function. Export the function using module.exports and import it in another file.
4. Explain the purpose of callbacks in Node.js and how they prevent blocking behavior.
5. Create a simple program that reads a file asynchronously using the fs module. Use a callback to handle the data.
6. How do you find, install, and manage packages using npm?

Day 32-34

Basics of Databases

📃 Topics to Cover

- Relational databases
- Non-relational databases

🗂️ Resources for Learning

1. System Design — SQL vs NoSQL. Concepts and considerations for SQL and… | by Larry | Peng Yang | Computer Science Fundamentals | Medium
2. How To Choose The Right Database?
3. System Design Interview Prep: SQL vs NoSQL Databases - Exponent
4. Databases: system design interview concepts (2 of 9)

Questions

1. Explain the concept of normalization in the context of relational databases. What are the benefits and drawbacks of normalization?
2. Compare the differences between INNER JOIN, LEFT JOIN, and RIGHT JOIN operations in SQL.
3. Explain the trade-offs between using a relational database and a NoSQL database for a specific use case.
4. How does MongoDB differ from traditional relational databases in terms of data storage and querying?
5. Discuss the trade-offs between using a document-oriented database like MongoDB and a column-family database like Cassandra.
6. What are the benefits of denormalization in a document-oriented database like MongoDB?

Day 35-37

Advanced Database Concepts

📃 Topics to Cover

- ACID Properties
- Database sharding and partitioning
- Database indexing
- Concurrency Control

🗂️ Resources for Learning

1. ACID Properties in DBMS - GeeksforGeeks
2. Database Sharding vs. Partitioning: What's the Difference?
3. System Design — Sharding / Data Partitioning | by Larry | Peng Yang | Computer Science Fundamentals | Medium
4. An in-depth look at Database Indexing
5. Concurrency Control in DBMS - GeeksforGeeks

Questions

1. Explain the ACID properties in the context of database transactions. How do they ensure data integrity?
2. Discuss the trade-offs between ensuring strong ACID properties and achieving high performance in a database system.
3. Define database sharding and partitioning. How do they contribute to scalability?
4. What are the key factors to consider when selecting a sharding key for a database table?
5. Compare B-tree indexes and hash indexes. When would you choose one over the other?
6. What is a deadlock? Provide an example and describe strategies to prevent or resolve deadlocks.
7. Explain the concept of a database lock. How can locks be used to manage concurrent access?

Day 38-41

REST API

📃 Topics to Cover

- HTTP Verbs
- Request and Response Format
- Statelessness
- Security and Authentication

🗂️ Resources for Learning

1. https://www.restapitutorial.com/
2. https://www.youtube.com/watch?v=lsMQRaeKNDk
3. https://www.geeksforgeeks.org/rest-api-introduction/
4. https://rapidapi.com/learn/rest

Questions

1. Explain the purpose of HTTP verbs (GET, POST, PUT, DELETE) in RESTful APIs.
2. Create a simple Express.js server with routes for handling POST, PUT, and DELETE requests.
3. Explain the structure of an HTTP response, including status codes and response bodies.
4. Design an API response format for a user registration endpoint that returns a success message and a user ID.
5. Explain how statelessness contributes to scalability and simplicity in API design.
6. Explore API key authentication: Generate an API key, send it as a header in a request, and validate it on the server.
7. Explain the difference between authentication and authorization.

Day 42-44

Deployment Basics

📃 Topics to Cover

- Deploying applications on platforms like Heroku or Netlify
- Setting up environment variables
- Connecting a domain name

📁 Resources for Learning

1. https://www.freecodecamp.org/news/develop-deploy-first-fullstack-web-app/
2. https://www.codecademy.com/article/deploying-a-full-stack-app-with-heroku
3. https://www.youtube.com/watch?v=MusIvEKjqsc

Day 45-46

Planning the Final Project

1. **Define Goals:**
   Clearly outline project goals and features.

2. **User Stories:**
   Create user stories to understand user interactions.

3. **Wireframes:**
   Design basic layouts for the project.

Day 47-50

Designing UI Elements

Create a consistent and visually appealing design for your project's components and layout.

1. **UI Design:**
   Create visually consistent components.
2. **Responsive UI:**
   Ensure UI works across devices.
3. **CSS Styling:**
   Apply styling to UI elements.
4. **Interactivity:**
   Add user interactions using JavaScript.
5. **Usability Testing:**
   Continuously test for a good user experience.

Day 51-56

Integrating front-end and back-end

Design and document the endpoints for communication between the front-end and back-end.

1. **Front-End Dev:**
   Develop UI, integrate API calls.
2. **Back-End Dev:**
   Handle requests, process data.
3. **API Testing:**
   Test endpoint responses.
4. **CORS:**
   Configure cross-domain communication.

Day 57-60

Deploying the final project

Choose a hosting platform that suits your project's needs (e.g., Heroku, Netlify, AWS).

1. **Database Setup:** Migrate and set up data.
2. **Environment Config:** Configure settings.
3. **CI/CD:** Automate deployment process.
4. **Monitoring:** Regularly monitor and maintain.

BOSSCODER ACADEMY

WHY BOSSCODER?

- **750+** Alumni placed at Top Product-based companies.
- More than **136% hike** for every **2 out of 3** working professional.
- Average package of **24LPA.**

The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**

Lavanya
Meta

Course is very well **structured** and **streamlined** to crack any **MAANG** company

Rahul
Google

EXPLORE MORE