uconnect

making a difference

Course Outline: ReactJS and Advanced JavaScript

Month 1: Foundations and Intermediate JavaScript

Week 1: JavaScript Basics

- **Introduction to JavaScript:**
    - Variables
        - Let
        - Var
        - const
    - Data types
    - Operators
- **Control Flow:**
    - If statements
    - Loops

Week 2: Functions, Scope, and Data Structures

- **Functions and Scope:**
    - Function declaration and expression
    - Function parameters and arguments
    - Return statements and the concept of undefined
    - Function hoisting and best practices
    - Scope (global, function, block)
- **Arrays and Objects:**

  - Creating and manipulating arrays and objects
  - Common array
    - length() —> properties
    - indexOf()
    - concat()
    - push()
    - pop()
    - shift()
    - unshift()
    - slice()
    - splice()
    - find()
    - filter()

Week 3: Intermediate JavaScript and DOM Manipulation

- **DOM Manipulation Basics:**
  - Selecting and manipulating DOM elements
  - Handling basic events
- **Introduction to Git & GitHub:**
  - Understanding Git & GitHub
  - Setting up a GitHub account
  - Setting up Git configuration
- **Prototypes and Inheritance:**
  - Object prototypes
  - Using Object.create() for inheritance
  - Class-based inheritance with ES6 classes

Week 4: Advanced Git and ES6+ Features

- **Working with Repositories:**
  - Creating a new repository

    - Cloning a repository
- **Basic Git Commands:**
    - `git init`
    - `git clone` and `git status`
    - `git add`
    - `git commit`
    - `git log`
    - `git branch`
    - `git push`
    - `git pull`
    - `git merge`

Collaboration, Version Control, and Advanced Git

- **Collaboration and Version Control:**
    - Fork a repository on GitHub
    - Create pull requests
    - Review pull requests
- **Advanced Git Commands:**
    - `git rebase`
    - `git cherry-pick`
    - Undoing Changes:
        - Undo last commit
        - Discard changes in the working directory
- **ES6+ Features:**
    - Arrow functions
    - Classes and inheritance
    - Destructuring assignments
    - Template literals
    - Rest and spread operators

  - Enhanced object literals
  - Modules

Month 2: Advanced JavaScript and React Fundamentals

Week 1: Advanced JavaScript Concepts

- **Closures:**
  - Understanding lexical scope
  - Use cases and practical examples
- **Promises and Asynchronous Programming:**
  - Callback functions
  - Promises and the Promise object
  - Async/await syntax
  - Error handling in asynchronous code
- **Event Handling:**
  - Deeper exploration of event handling
  - Event delegation and best practices
- **Error Handling:**
  - Strategies for effective error handling
  - Handling errors in asynchronous code

Week 2: Introduction to ReactJS

- **What is React and Why Use It?**
  - Virtual DOM and component-based architecture
- **Setting Up a Development Environment:**
  - Node.js, npm, or yarn installation
  - Basic project setup
- **React Components, JSX, and Rendering:**
  - JSX syntax and its benefits
  - Creating and rendering React components
  - Stateless functional components

Week 3: State, Props

- **State and Props:**
  - Understanding and managing component state
  - Passing data via props
- **Component Lifecycle:**
  - Overview of key lifecycle methods

Week 4: React Hooks

- **React Hooks:**
  - useState
  - useEffect
  - useReducer
  - useCallback
  - useMemo
  - useRef
  - customHooks
  - useId

Month 3: Advanced React, Next.js, and Deployment

Week 1-2: Advanced React Concepts

- **React Router (Navigation):**
  - Implementing basic navigation
  - Handling nested routes
- **Redux or Context API (State Management):**
  - State management using Redux or Context API
  - Understanding the Flux architecture
- **Higher-Order Components (HOCs):**
  - Creating and using HOCs
  - Composing components with HOCs

  - Converting class components to functional components
- **Error Boundaries:**
  - Implementing error boundaries in React
- **Building and Real-World Practice:**
  - Applying learned concepts to real projects
  - Using any UI framework
  - Gaining hands-on experience

Week 3: Project Deployment on Vercel and Final Projects

- **Preparing the Project for Deployment:**
  - Optimizing the application
  - Environment variables
- **Deploying on Vercel:**
  - Setting up a Vercel account
  - Connecting the GitHub repository to Vercel
  - Deploying the project
  - Managing deployments and environment variables

Week 4: Introduction to Next.js

- **What is Next.js and Why Use It?**
  - Server-side rendering (SSR) vs. static site generation (SSG)
  - Benefits of using Next.js
- **Setting Up a Next.js Project:**
  - Installing Next.js and project structure
  - Creating pages and navigating between them
- **Data Fetching in Next.js:**
  - `getStaticProps` for static generation
  - `getServerSideProps` for server-side rendering
  - `getStaticPaths` for dynamic routes
- **Advanced Next.js and API Routes**

  - **API Routes in Next.js:**
    - Creating and using API routes within a Next.js project
  - **Advanced Next.js Features:**
    - Image optimization
    - Dynamic imports

This condensed outline ensures all key topics are covered within the 3-month timeframe, with a focus on practical application and deployment

Prepared by: Saima Akhtar