React Frontend Web Developer

Project1: Build a Personal Portfolio Website:

- **Content:** Populate your website with relevant content about yourself, your skills, and your projects.
- **Images:** Use high-quality images that complement your design.
- **Responsiveness:** Ensure your website looks good and functions properly on different devices.
- **Navigation:** Create a clear and intuitive navigation menu.
- **Accessibility:** Make your website accessible to users with disabilities.
- **Deployment:** Consider deploying your website to a hosting platform for public access.
- The delivery will be in 4 phases

Assignment1:

- Create the basic HTML structure for your portfolio website.
- Include essential elements like header, navigation, main content, and footer.
- Structure your content logically using HTML headings, paragraphs, and semantic elements.
- Create placeholder text for sections like about me, projects, skills, and contact.

Assignment2:

- Design the visual appearance of your website using CSS.
- Choose a color palette and typography that reflects your personal style.
- Create layouts for different screen sizes using CSS media queries.
- Style the header, navigation, main content, and footer elements.
- Add visual elements like images, icons, and background images.

Assignment3:

- Add interactive elements to your website using JavaScript.

- Create dynamic effects like animations, transitions, and hover effects.
- Implement responsive navigation menus.
- Incorporate user interactions (e.g., forms, buttons, click events).
- Consider adding features like a contact form or a portfolio slider.

Assignment4:

- Incorporate Bootstrap components to streamline development.
- Use Bootstrap's grid system for responsive layouts.
- Utilize pre-built components like buttons, forms, and navigation bars.
- Customize Bootstrap styles to match your website's design.
- Optimize your website for different screen sizes and devices.

Project2: Build a simple e-commerce product listing application using React.

- **Description:** The app will display a list of products, allowing users to filter, sort, and view product details.
- **Product Details:** Implement a product details page to display more information about a selected product.
- **Cart Functionality:** Add basic cart functionality for users to add products to a cart.

Assignment1:

- Create the fundamental structure of the application.
- Set up a React project using Create React App.
- Create basic components: ProductCard, ProductList, Filter, and Sorting components.
- Define the basic structure of the application using JSX.
- Implement a simple product data structure (e.g., an array of objects).

Assignment2:

- Implement data management and core component logic.
- Create a state management solution using useState or useReducer to manage product data and filter/sort criteria.

- Implement filtering and sorting functionality based on user interactions.
- Pass data between components using props.
- Handle user interactions (e.g., clicks, input changes) to update the product list.

Assignment3:

- Enhance the user interface and user experience.
- Style the components using CSS or a CSS-in-JS library.
- Create a visually appealing and user-friendly layout.
- Implement responsive design to adapt to different screen sizes.
- Consider accessibility guidelines (e.g., color contrast, keyboard navigation).
- Implement navigation to product details page.
- Display detailed product information on the product details page.
- Add shopping cart functionality (basic implementation).
- Implement user interactions like adding products to the cart, removing products, and viewing cart contents.
- Add loading indicators or error handling for improved user experience.

Assignment4:

- Set up a Node.js and Express server.
- Define API endpoints for product data.
- Implement database integration to store product information.
- Implement CRUD operations (Create, Read, Update, Delete) for products.
- Connect the React frontend to the backend API to fetch and display product data.
- Handle error handling and response formatting.
- Handle user interactions (adding to cart, etc.) by sending requests to the backend.
- Implement authentication and authorization (optional).

- Deploy the application to a hosting platform (optional).

Assignment5:

- Develop detailed user stories and acceptance criteria for the e-commerce application.
- Create clear and concise functional requirements documentation.
- Use a suitable documentation format
- Write unit tests for core components and functions.
- Achieve good test coverage.
- Initialize a Git repository for the project.
- Commit code changes regularly with descriptive commit messages.
- Push code to a GitHub repository.
- Create branches for different features or bug fixes.
- Use pull requests for code review and collaboration.

Project	Assignment	Due date
Project1	Assignment1	Week
	Assignment2	Week
	Assignment3	Week
	Assignment4	Week
Project2	Assignment1	Week
	Assignment2	Week
	Assignment3	Week
	Assignment4	Week
	Assignment5	Week