



"PORTFOLIO WEBSITE"

Task - 3



Your Project Name

- Your Project Introduction

LMS Username	Name	Batch
2115a102	ANUROOPA PV	A10
2115a1011	NAJEETHA BANU.N.	A10
2115a1019	SWETHA S	A10
2115a1017	SATHIYAVARSHINI.S	A10



Task 3 :: Frontend Creation

React.js: A popular JavaScript library for creating dynamic user interfaces. Ideal for single-page applications like portfolio websites, enabling dynamic content updates without page refresh. React.js can be used to create components displaying portfolio items, with the React Router used for navigation.

1.CSS: Used for styling website layout, fonts, colors, and visual elements. Enables creation of responsive designs that adapt to different screen sizes, improving user experience on desktop and mobile devices.

2.HTML: The foundation of any website, used for structuring content such as headings, paragraphs, images, and links. Can also be used for creating forms for user input.

3.React.js: A popular JavaScript library for creating dynamic user interfaces. Ideal for single-page applications like portfolio websites, enabling dynamic content updates without page refresh. React.js can be used to create components displaying portfolio items, with the React Router used for navigation.

Learning outcome

- Developing complicated UI using HTML components
- Using props drilling and context to pass variables
- Getting familiar with different type of api calls
- Handling different input data

Step-Wise Description

1. Plan your design: Before you start coding, it's important to plan your design. Decide what layout you want to use, what colors you want to use, and what features you want to include in your portfolio.
2. Set up your development environment: Install Node.js and a code editor like Visual Studio Code. This will provide the tools necessary to create and run your ReactJS application.
3. Create a new ReactJS project: Use the create-react-app tool to create a new ReactJS project. This will create a basic structure for your website.
4. Install necessary packages: Use npm to install necessary packages like react-router-dom for routing, react-bootstrap for styling, and any other libraries or packages you might need.
5. Create components: Break down your portfolio into small, reusable components. This will make it easier to manage and update your portfolio in the future.
6. Write code: Write the code for each component, making sure to import necessary packages and to use appropriate ReactJS

Summary of your task

- syntax. Use CSS or a styling library like react-bootstrap to add styles to your components.
- Implement responsiveness: Make sure your portfolio looks good and functions properly on different devices by implementing responsive design.
- Test your portfolio: Use a local server to test your portfolio in a web browser.
 - Deploy your portfolio: Use a service like Netlify or GitHub Pages to deploy your portfolio.
 - Update your portfolio: Continue to update and maintain your portfolio by adding new projects, updating styles, and fixing bugs as they arise

Assessment Parameter





<https://github.com/swethasenthilnathan63>

Thank
you!

