INTRODUCING REACT

FULL STACK SKILLS BOOTCAMP

AN INTRODUCTION TO BUILDING USER INTERFACES WITH REACT
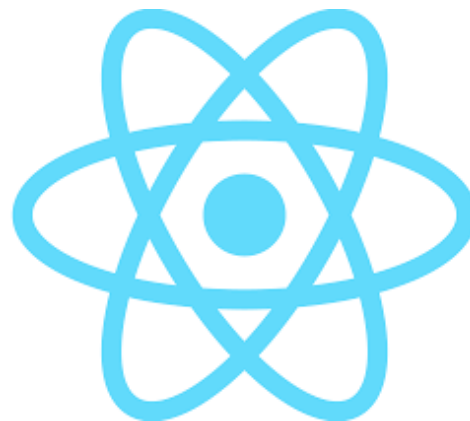
- **Lesson Overview:**
- In this lesson, we will be introduced to:

1. The history of React
2. Installation
3. JSX Expressions
4. Components and Props

WHO DEVELOPED REACT? A BRIEF HISTORY

- **Created by**:
  Facebook (now Meta) in 2011 for internal use.

- **Open-Sourced**:
  Released to the public in 2013.

- **Widespread Use**:
  Adopted by many companies worldwide due to its efficiency and flexibility.

WHAT PROBLEMS DOES REACT SOLVE?

- **Component Reusability**:
  Build reusable components to reduce
  redundancy and improve maintenance.

- **Efficient DOM Manipulation**:
  Uses a virtual DOM to optimize UI updates.

- **Declarative Syntax**:
  Simplifies the process of managing UI states
  across complex applications.

- **Component-Based Architecture**:
  Allows developers to build complex UIs by
  combining simple components.

REAL-WORLD USE CASES

- **Social Media**:
  Facebook and Instagram use React for real-time updates.

- **E-commerce**:
  Sites like Amazon and Shopify for a dynamic, responsive user experience.

- **Media and Streaming**:
  Netflix, Airbnb, and other platforms use React for a smooth, user-friendly interface.

- **Business Tools**:
  Used by companies like Slack and Salesforce to build responsive dashboards and data visualization tools.

PROS AND CONS OF REACT

■ **Pros**:

- Component Reusability

- Virtual DOM for better performance

- Strong ecosystem and community

- Easy integration with other libraries

■ **Cons**:

- Learning Curve: JSX and component architecture take time to learn

- JSX Syntax: Can feel non-standard for HTML/ CSS developers

- Frequent Updates: Requires developers to keep up with changes

VITE VS. CREATE REACT APP (CRA)

- **Create React App (CRA)**:
  - Traditional React setup tool.
  - Easy to use but can be slow due to bundling.
- **Vite**:
  - Modern development tool for fast builds and hot module replacement.
  - Better suited for larger, complex applications.

- **Why Vite?**
  Faster initial load times and builds, supports modern JavaScript features.

CREATING A NEW APP USING VITE

■ **Step 1**:
Install using Vite:

```
npm create vite@latest my-react-app
```

■ **Step 2**:
Navigate to the new project:

```
cd my-react-app
```

■ **Step 3**:
Install dependencies and start the server

demo…

```
npm install
npm run dev
```

BASIC JSX EXPRESSIONS

- **What is JSX?**:
  JavaScript syntax extension that looks similar to HTML.

- **Embedding JavaScript**:
  Use {} to embed JavaScript expressions within JSX.

- **Key Concept**:
  JSX makes code more readable and allows embedding logic directly in UI definitions.

  Demo…

```javascript
const name = "React";
const element = <h1>Hello, {name}!</h1>;
```

COMPONENTS AND PROPS

- **Components**:
  Building blocks of a React app. Can be functional or class-based.

- **Props**:
  Short for "properties," used to pass data from one component to another.

- **Usage:**
  Here, name="Alice" is a prop passed to the Greeting component.

```
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}
```

```
<Greeting name="Alice" />
```

CONCLUSION

- **React**:
  A powerful library for building user interfaces.

- **Benefits**:
  Reusable components, fast rendering with Virtual DOM, and declarative programming.

- **Development Tools**:
  Use Vite for faster development and modern features.

- **Core Concepts**:
  JSX, components, and props are the foundational elements of React.

QUESTIONS?