**Module 1**

**Topic 1: Course Introduction**

**Lecture 1: Introduction to the course: React Basics**

✅ **Introduction to React Basics**

* React is a JavaScript library used to build user interfaces for websites.
* This course introduces the basic structure and usage of React.js.

✅ **Understanding Components**

* User interfaces are made up of small, independent parts called components.
* Components can be reused and combined to create more complex structures.
* React helps organize and manage these components efficiently.

✅ **Why React is Popular**

* React makes it easy to build and manage UI components.
* It performs well and does not use too many browser resources.

✅ **Working with State**

* State refers to all the values that the app is using at a specific time.
* Managing state allows apps to update and respond to user actions.

✅ **Styling in React**

* You will learn how to add styles to your app and reuse them across components.

✅ **Handling Events**

* The app can respond to events like clicks or data entered by the user.

✅ **Navigation and Assets**

* You will learn how to create navigation between pages.
* You will also use assets like images and icons in your app.

✅ **Course Outcome: Portfolio Projects**

* You will complete projects to apply what you’ve learned.
* These projects will help you build a personal portfolio.

**Lecture 2: How is React used in the real world?**

✅ **React at Meta (Facebook.com)**

* Meta uses many experimental and new React features.
* Facebook.com served as a testing ground for unreleased features from the React team.
* Rewriting Facebook.com involved building on a modern and faster tech stack using React.

✅ **Role of the React Apps Team**

* The React Apps team at Meta builds new features for Facebook.com.
* The team experiments with advanced and upcoming features of React.

✅ **Popular Apps Built with React**

* React is used by many companies including Facebook, Instagram, Netflix, Airbnb, and The New York Times.
* Interactive websites often use React for building dynamic UIs.

✅ **Why Rewrite Facebook.com in React**

* The older site was outdated and had performance limitations.
* React offered better speed, responsiveness, and ease of development.
* The redesign aimed to create a modern look and improved user experience.

✅ **Scale and Collaboration**

* Around 40 engineers worked together on the Facebook.com rewrite.
* Collaboration with various product teams helped transition products to the new stack.
* It was a large, high-risk effort requiring strong proof of performance and reliability.

✅ **Proving the Value of React**

* Teams needed to be convinced that the new version was not only modern-looking but also faster and more efficient.
* The goal was to support product teams in delivering better web experiences.

✅ **React as an Open Source Project**

* React is open source and welcomes contributions from developers worldwide.
* Engineers inside and outside of Meta contribute to its development.
* This fosters a strong, active community that continuously maintains and improves React.

✅ **Community and Contribution**

* Developers can ask questions, suggest features, or contribute code.
* There's active engagement through documentation, forums, and updates.
* A React conference is held annually to share updates and connect developers.

✅ **Getting Involved with React**

* The React community offers many ways to participate and learn.
* Beginners are encouraged to explore official documentation.
* Connecting with the community enhances the development experience and opens contribution opportunities.

**Topic 2: React Components and Where They Live**

**Lecture 1:** **Why React?**

✅ **Introduction to React Usage in Tech Careers**

* React is widely used across the tech industry.
* Learning React is valuable for long-term career growth.

✅ **React vs Traditional Programming Concepts**

* most programming was object-oriented using inheritance.
* React uses **composition**, a different approach that may feel unfamiliar at first.
* With strong documentation and community support, the transition becomes easier.

✅ **Learning and Community Support**

* Numerous resources available: documentation, YouTube, and open-source projects.
* React has a large, active community that helps beginners and experienced developers alike.

✅ **Why Developers Choose React**

* Easy to learn and pick up.
* Offers high flexibility for creating custom UIs.
* Ideal for building modern, interactive web applications.

✅ **Integration with Other Libraries**

* React is just a **front-end library**, not a full-stack solution.
* It integrates well with tools like Redux and other third-party libraries.
* Perfect for developers who need flexibility in their tech stack.

✅ **Handling Complex UIs**

* React excels in building applications with many custom UI elements.
* Supports **component reusability** across different parts of an application.

✅ **Code Reusability and Efficiency at Meta**

* Meta uses a shared library of core UI components across Facebook, Instagram, and Messenger.
* Fixing a bug in one component improves the entire system.
* Promotes efficiency and high code quality.

✅ **React vs Angular**

* **Angular**: Full-fledged framework with built-in routing, HTTP, etc.
* **React**: Flexible and lightweight, but needs third-party tools for full app development.
* React is better for custom UI needs; Angular is easier for creating single-page apps quickly.

✅ **Advice for Learning React**

* Start with simple projects—don’t aim for complexity at the beginning.
* Always follow best practices and consult the documentation.
* Focus on learning key concepts like **hooks** properly.

✅ **Long-Term Value of React**

* React will likely appear in many roles throughout a developer's career.
* Its large support network and ecosystem make it a strong skill to master.

**Lecture 2: React.js overview**

✅ **Understanding Single-Page Applications (SPAs)**

* SPAs load content dynamically without full page reloads.
* Unlike traditional websites, they don't reload common elements like headers and footers.
* React is a popular tool for building SPAs, providing better speed and user experience.
* SPAs are ideal for businesses needing fast, scalable, and interactive web apps.

✅ **React and Its Growing Popularity**

* React is developed by Meta and powers many top websites.
* Often, React apps update content without changing the URL, a common SPA trait.
* React offers flexibility, speed, and rich user interfaces.

✅ **React Basics: Component-Based Architecture**

* React apps are built using reusable **components**.
* Each component handles its own functionality and UI structure.
* Components are modular, allowing easy integration and maintenance.

✅ **Benefits of Using Components in React**

* Components can be reused and moved around the app as needed.
* Multiple developers can work on different components independently.
* Promotes clean, maintainable, and scalable code.

✅ **Components in UI Design**

* All user interfaces are created using a mix of **simple and complex components**.
* Example: An e-commerce checkout page may have
  + **Header** (logo, nav, cart)
  + **Payment Section** (form, validation, submit button)
  + **Sidebar** (order summary)
* Each section is a self-contained component with its own HTML, CSS, and JavaScript.

✅ **Not Just React — Component Use is Common**

* Component-based UI design exists outside React too.
* React simplifies and streamlines this design approach.

✅ **Rendering Components in React**

* Components are rendered efficiently to the **DOM (Document Object Model)**.
* React minimizes performance impact when updating components.

✅ **Traditional DOM Manipulation Challenges**

* Before React, developers had to manually manipulate the DOM.
* This made code messy and hard to maintain — known as **spaghetti code**.

✅ **React’s Virtual DOM Advantage**

* React uses a **virtual DOM** — a lightweight in-memory copy of the real DOM.
* Updates are first made in the virtual DOM, then selectively applied to the real DOM.
* This results in **faster rendering**, improved performance, and fewer browser resource issues.