⭐ React – Complete Course Syllabus (Beginner to Advanced)

📘 Module 1: Introduction to React

- What is React?
- History & Evolution (React 16 → 18)
- React Fiber Architecture
- Why React? Use cases & features
- React vs Angular vs Vue
- Understanding SPA (Single Page Applications)
- Setting up the React development environment
    - Node.js, npm, yarn
    - create-react-app
    - Vite (modern setup)
- Understanding JSX (JavaScript XML)
- Rendering elements in React

📘 Module 2: Core Concepts of React

- Function vs Class Components
- Props
- State & setState()
- Event Handling
- Conditional Rendering
- Lists and Keys
- Class Component Lifecycle Methods (legacy but important)
- Component Reusability & Composition

📘 Module 3: Styling in React

- Inline CSS in JSX
- External CSS, CSS Modules
- SASS / SCSS
- Styled Components
- Using UI libraries
    - Bootstrap
    - Tailwind CSS
    - Material UI (optional)

📘 Module 4: TypeScript for React Developers

(Mandatory because React in industry uses TS)

- Why TypeScript in React?
- Setting up React + TypeScript project
- Typing Props & Children
- Typing State & Events
- Typing Hooks (useState, useReducer, useRef, useEffect)
- Typing Custom Hooks
- TypeScript with Axios (API calls)
- TypeScript with Redux Toolkit
- TypeScript with React Query
- TypeScript with Next.js
- Folder structure & best practices with TS

📘 Module 5: React Hooks (Modern React)

- Introduction to Hooks
- useState() Hook
- useEffect() Hook
- useRef() Hook (DOM manipulation & value storage)
- useContext() Hook (Light state management)
- useReducer() Hook (Complex state)
- useMemo() & useCallback() (Performance optimization)
- useLayoutEffect()
- Creating Custom Hooks (very important)

📘 Module 6: React Forms & User Input

- Controlled vs Uncontrolled components
- Handling Inputs: text, radio, checkbox, dropdown
- Form validation techniques
- Working with form libraries:
    - Formik
    - React Hook Form (industry standard)
- File upload handling
- Error handling in forms

📘 Module 7: React Router v6 (Client-Side Routing)

- Installing & configuring React Router
- Routes, Route, BrowserRouter
- Link & NavLink
- useParams, useNavigate, useLocation
- Nested Routing
- Dynamic Routing
- Redirects
- Private Routes (Authentication-based routing)
- Protected layouts
- Lazy loading routes

📘 Module 8: State Management in React

- Why state management?
- Context API with useReducer
- Redux (concepts)
    - Flux Architecture
    - Actions, Reducers, Store
- **Redux Toolkit (RTK)** – modern Redux
    - createSlice()
    - createAsyncThunk()
    - Immer.js
    - RTK Query for caching data
- Modern alternatives:
    - Zustand
    - Recoil
    - Jotai

📘 Module 9: Working with APIs & Data Fetching

- Fetch API
- Axios
- Async/Await in API calls
- Global API handling
- Loading, Success, Error states
- Abort Controllers (Cancel API requests)
- **React Query (TanStack Query)**
  - Query mutations
  - Optimistic updates
  - Caching & background refetching
- WebSockets & real-time data

📘 Module 10: Performance Optimization

- Code splitting
- Lazy loading components
- Suspense
- React.memo()
- PureComponent
- useMemo & useCallback (real-world usage)
- Windowing / Virtualization (React Window, React Virtualized)
- Debouncing & Throttling

📘 Module 11: React + Backend Integration

- Connecting React with Node.js / Express.js
- Working with MongoDB, Firebase, MySQL APIs
- JWT Authentication (Login/Signup/Refresh Token)
- Firebase Authentication
- OAuth (Google/Facebook Login)
- Uploading files to backend
- Role-based UI rendering

📘 Module 12: React and Next.js (Modern Full-Stack)

- What is Next.js? Why use it?
- Next.js Routing system
- Server-Side Rendering (SSR)
- Static Site Generation (SSG)
- Client Components & Server Components
- API Routes in Next.js
- Image optimization
- Authentication in Next.js
- Deploying Next.js applications

📘 Module 13: Advanced React Concepts

- Higher Order Components (HOC)
- Render Props Pattern
- Controlled vs Uncontrolled patterns
- React Portals (modals, tooltips)
- Error Boundaries
- Fragments
- Suspense & Concurrent Mode (React 18)

📘 Module 14: Accessibility (A11y) in React

- Why accessibility matters
- Semantic HTML
- ARIA roles & labels
- Focus management
- Screen reader compatibility
- Keyboard navigation
- Accessibility testing tools

📘 Module 15: React Architecture & Project Structure

- Proper folder structure for scalable React apps
- Atomic design & feature-based architecture
- Organizing hooks/services/components
- Environment variables
- Clean code in React
- Reusable components & UI patterns
- Error handling architecture

📘 Module 16: Testing in React (Optional)

- Why testing?
- Unit testing with Jest
- Component testing with React Testing Library
- Mocking API calls
- Snapshot testing
- Integration tests
- End-to-End (E2E) testing with Cypress

📘 Module 17: Deployment & Best Practices

- Building React apps for production
- Deployment on:
  - Netlify
  - Vercel
  - Firebase
  - AWS S3 + CloudFront
- Dockerizing a React application
- Setting up CI/CD pipelines (GitHub Actions)
- Security best practices in React

📘 Module 18: React Native (Optional Track)

- Introduction to React Native vs React
- Setting up development environment
- Core Components (View, Text, Button, Image)
- Navigation in React Native
- API integration
- Building & deploying mobile apps (Android/iOS)

📘 Module 19: Capstone Project (Real-World MERN Stack Project)

Build a full production-grade application:

- React Frontend
- Node.js + Express Backend
- MongoDB Database
- Authentication (JWT)
- Role-based Access
- Protected Routes
- Form validations
- File Uploads
- Real-time features (WebSockets)
- Payment Gateway Integration (Razorpay/Stripe)
- Performance optimization
- Deployment to cloud (Optional)