Prerequisites section

* Full Stack Overview
* Frontend Deep Dive (HTML, CSS, JS)
* Backend & Databases
* Key Tech Differences (Library vs Framework vs API)

**🌐 Prerequisites – What You Should Know Before React**

Before diving into React, it's good to have a basic understanding of **Full Stack Web Development**.

**🔁 Full Stack Development Overview**

**Full Stack** means working with both:

* **Frontend** (what users see)
* **Backend** (what powers the logic, and server)
* **Database** (what manipulate the data)

**🖼 Frontend Technologies**

|  |  |
| --- | --- |
| Tech | Description |
| HTML | Content and Structure of the webpage. Think of it like the **skeleton**. |
| CSS | Styling and layout. Adds **colors, spacing, and responsiveness**. |
| JavaScript | Makes the page **interactive** — for animations, validations, UI logic, etc. |

**💡 Example:**

<button onclick="alert('Hello!')">Click Me</button>

**🔧 Backend Technologies**

Backend runs **on the server**, handles **logic, processing, APIs**, and connects to the database.

Common backend languages:

* **Java** (Spring Boot)
* **Python** (Django, Flask)
* **Node.js** (JavaScript backend)
* **PHP**, **.NET (C#)**, etc.

**🗃️ Databases**

Where we store data permanently.

| **Type** | **Examples** |
| --- | --- |
| **Relational (SQL)** | MySQL, PostgreSQL, Oracle |
| **Non-relational (NoSQL)** | MongoDB, Firebase |

**🎨 Deep Dive into Frontend**

**HTML – Structure**

* Defines headings, paragraphs, images, buttons, forms, etc.
* Tags like <h1>, <div>, <p>, <img>, <a>, <form>

**CSS – Styling & Layout**

* Adds design: colors, fonts, spacing
* Handles responsiveness using **media queries** and **flex/grid**

@media (max-width: 600px) {

.box {

flex-direction: column;

}

}

**JavaScript – Interaction**

* Handles logic and events (onClick, onSubmit)
* Can validate forms, change elements, interact with APIs

**🧠 Library vs Framework vs API – Key Differences**

|  |  |  |
| --- | --- | --- |
| Concept | Definition | Example |
| Library | Pre-written code you call in your app. You stay in control. | React, Lodash |
| Framework | Provides a full structure and calls your code. It controls the flow.  Framework calls the Program | Angular, Django, Spring |
| API | A set of functions and rules for interacting with software.  Program calls the API | REST API, Web API, OpenWeather API |

**📌 Simple Analogy:**

💼 **Library** is like ordering food from a menu – you choose what you want.  
🏗 **Framework** is like a set menu – you follow their structure.  
🧾 **API** is like a waiter – you send a request, get a response.

Day 1: React Fundamentals! 💻✨

**🌟 Day 1: React Fundamentals**

**What You’ll Learn:**

* What is React and why it’s popular
* Setting up your development environment
* Understanding JSX
* Creating components
* Using props and basic state

**💡 1. What is React?**

React is a **JavaScript library** for building user interfaces (Single page web application).  
It helps you create **interactive, reusable components** that manage their own state.

✅ Made by Facebook  
✅ Used in modern apps like Instagram, Netflix, WhatsApp

**🛠 2. Setup Environment**

**Option A: Vite (faster & recommended)**

npm create vite@latest

cd my-app

npm install

npm run dev

**Option B: CRA (Create React App)**

npx create-react-app my-app

cd my-app

npm start

You'll need **Node.js** installed before running the above commands.

**💡 3. JSX - JavaScript + XML**

JSX lets you write HTML inside JavaScript:

const element = <h1>Hello, world!</h1>;

✅ Use **camelCase** for HTML attributes (className, onClick)  
✅ Return **only one root element** from a component

**💡 4. Components**

React apps are built using components.

**Functional Component:**

function Welcome() {

return <h2>Welcome to React!</h2>;

}

**Props Example:**

function Greet(props) {

return <h3>Hello, {props.name}!</h3>;

}

<Greet name="Darwin" />

**🔄 5. Basic State with useState**

import { useState } from 'react';

function Counter() {

const [count, setCount] = useState(0);

return (

<div>

<p>Count: {count}</p>

<button onClick={() => setCount(count + 1)}>Add</button>

</div>

);

}

**💡 Practice Task**

✅ Create a small **Profile Card App** with:

* A functional component for a profile
* Props for name, title, and image
* A button to like the profile using useState

**📚 Tools Used**

* **Vite or CRA** for project setup
* **VS Code** as the code editor
* **Chrome DevTools** for debugging