



Lecture 4: Myntra Project in React :

👉 React + Vite + Parcel ka ultimate combo!



Vite Bundler Kaise Install Karein?

```
npm create vite@latest
```



Step by Step:

- 🍷 Project Name: `Vite-project`
 - ⚙️ Framework: `React`
 - 💻 Language: `JavaScript`
-



Node Modules Kaise Install Karte Hai?

```
npm install
```



Ye command **packaging me jitni bhi dependencies hongi**, un sabko download kar legi.



Note: Vite ek lightweight aur fast bundler hai — Isme sab kuch fast hota hai, isliye setup me jyada time nahi lagta.



Development Server Start Kaise Karein?

```
npm run dev
```

```
# ya
```

```
npm run start
```



Ye command development server ko start karti hai




App run hoti hai:


```
http://localhost:5000
```

Scripts Ko Kaise Samjhein?

Kisi bhi company ka code mila project ke form me?

 `package.json` me jao

 `scripts` section me dekh lo kya-kya command likhi gayi hai


 Example:


```
"scripts": {  
  
  "dev": "vite",  
  
  "build": "vite build",  
  
  "start": "vite"  
  
}
```

 Ye bataata hai ki kis command se kya hoga.


Code Ko Production Me Kaise Le Jayein?

`npm run build`

 Ye command optimized production code create karti hai.

 Script me likha hota hai:

```
"build": "vite build"
```

 *Is command ko chaaho to apne hisaab se change bhi kar sakte ho.*

Parcel vs Vite – Kya Difference Hai?

 Parcel:

- React setup ke liye **readymade config deta hai**
- File manually banana nahi padti
- Thoda slow ho sakta hai bade projects me

⚡ Vite:

- Ek modern bundler hai
- Saare packages ek saath install hote hain
- Super fast hota hai — HMR (Hot Module Replacement) ke saath

🔍 Short Comparison:

Feature	Parcel	Vite
Setup	Ready-made	Manual step-by-step
Speed	Medium	Very Fast (HMR Support)
Packages	Alag-alag install hote hain	Ek saath aa jaate hain
Learning Curve	Easy for Beginners	Slightly advanced (but fast)

📚 Dependencies & Dev Dependencies

✅ Normal Dependencies (Production Me Chahiye):

1. `react`
2. `react-dom`

📁 Example:

```
"dependencies": {  
  
  "react": "^19.1.0",  
  
  "react-dom": "^19.1.0"  
}
```

🔧 Dev Dependencies (Sirf Code Likhte Time Chahiye):

1. `parcel`
2. `vite`
3. `babel`

📄 Example:

```
"devDependencies": {  
  
  "parcel": "^2.14.3"  
  
}
```

💡 **Babel:** JSX ko React components me convert karta hai

👉 JSX = JavaScript + XML (e.g. `<div>Hello</div>`)

⚠️ Dev dependencies ko production me deploy nahi karte.

🔧 Parcel Me React Project Initialize Karna

```
npm init
```

📄 Ye command ek `package.json` file banata hai

🔧 React Ko Install Kaise Karein (Manually – Parcel Me)?

```
npm install react
```

```
npm install react-dom
```

📌 Isse react aur react-dom ki libraries manually download ho jaati hain.

🔧 Parcel Ko Install Karna

As a Normal Dependency:

```
npm install parcel
```

As a Dev Dependency:

```
npm install -D parcel
```

🧠 `-D` ka matlab hota hai: "Dev only dependency"

▶ Parcel Se Project Run Kaise Karein?

```
npx parcel index.html
```

✅ Ye command project ko run kar deta hai

🔑 React ko parcel ke through run karne ke liye `index.html` file ka use hota hai

🧠 Extra Concepts (Bonus)

🔄 HMR (Hot Module Replacement):

- Vite aur Parcel dono me hota hai
 - Code change karte hi browser automatic refresh hota hai
 - Developer experience fast aur smooth hota hai
-

🔒 Why Use Vite in 2025?

- Super Fast Start-up time
 - Modern build tools ka support
 - React, Vue, Svelte, sab ke saath compatible
 - Easy configuration
-

💡 Tips (Jarur Yaad Rakho):

- ◆ Hamesha `package.json` check karo kisi bhi naye project me
- ◆ Dev dependencies ko production me deploy mat karo

- ◆ Vite me `public` folder ke andar static files daalo
 - ◆ Build ke baad `/dist` folder generate hota hai — isko hi deploy karte hain
 - ◆ Babel use hota hai JSX ko normal JavaScript banane ke liye
 - ◆ `npm run dev` ka matlab hota hai "Development mode me chalao"
-



FINAL SUMMARY



Hamne dekha kaise:

- React project ko Vite aur Parcel ke through setup karein
- Scripts aur commands ka proper use karein
- Dependencies aur Dev Dependencies ka farak samjhein
- Parcel aur Vite me kya difference hai
- Production me code kaise le jayein
- Aur kaafi saare chhote-chhote important points 🧠

JSX, Props, Rendering (Detailed Notes)



Desi Devs ke liye simple bhasha mein full React magic!



React Code Basics

```
import React from "react";
```

```
// React import karna mandatory hai JSX use karne ke liye
```

```
import ReactDOM from "react-dom/client";
```

```
// ReactDOM use hota hai React app ko actual DOM me render karne ke liye
```



JSX (JavaScript + XML)

JSX ke andar **expression** allowed hai, **statement** nahi ❌

✅ Expressions Allowed:

```
<h1>{2 + 2}</h1>           // Valid
```

```
<h1>{name}</h1>           // Valid
```

❌ Statements Not Allowed:

```
<h1>{let x = 2}</h1>      // ❌ Invalid
```

🧠 JSX ke andar jo cheez render ho, wo ek result deni chahiye (jaise string, number, array)

🚫 JSX me Object Direct Render Nahi Kar Sakte:

```
let obj = {  
  name: "Harshal Chauhan"  
};
```

```
// ❌ Error
```

```
<h1>{obj}</h1>
```

🧠 JSX expects value that can be converted into visible content

✅ Valid JSX Values:

```
const arr = ["Harshal", 10, 20];
```

```
<h1>{arr}</h1>    // ✅ Valid
```

🚫 Lekin array ke andar agar koi object hai to wo bhi render nahi hoga

⚠ JSX Rule: Functions Capital Letter Se Shuru Hona Chahiye

```
function Greet() {  
  return <h1>Ram Ram Bhai</h1>;  
}
```

❓ Capital Kyu?

- ✅ Capital letter se pata chalta hai ki ye **custom component** hai, HTML tag nahi
 - ✅ React optimization technique hai — fast rendering ke liye
 - 🧠 React internally capital se start hone wale elements ko component treat karta hai.
-

📌 Props Kya Hota Hai? (React ka Real Power)

Props matlab hota hai — **data ko component ke andar bhejna (as arguments)**

📌 Argument Pass Karna (in JSX):

```
const element = <Greet name="Harshal Chauhan" age={21} />
```

- 🧠 Ye ek tarike se `Greet()` function ko **call** kar raha hai aur **props** pass kar raha hai
-

📌 Function Me Props Ko Catch Karna:

```
function Greet(props) {  
  console.log(props);  // Props ek object hota hai  
  
  return (  
    <h1>  
      Ram Ram ji {props.name} ({props.age})  
    </h1>  
  );  
}
```


Props As Object:

```
props = {  
  name: "Harshal Chauhan",  
  age: 21  
}
```


JSX Example vs HTML Example

```
// React Component Call  
  
const element = <Greet name="Harshal" age={21} />  
  
//⊕ HTML type JSX  
  
const htmlLike = <h1 id="first" className="first">Hello</h1>
```

 `class` ki jagah `className` use hota hai React me

Root Setup in React

```
const root = ReactDOM.createRoot(document.getElementById("root"));  
  
root.render(element);
```

 Yahan React DOM ke andar `element` ko mount kar diya gaya hai, jo ki HTML ke root div me dikhai dega.

☐ React can render **components, elements, strings, arrays** — bas wo visible output de

Extra Knowledge (Bhai ke liye Bonus!)

Fragment Use Karna (Multiple Tags Without Wrapper):

```
return (  
  <>  
    <h1>Hello</h1>
```

```
<p>World</p>

</>

);
```

❑ Destructuring Props (Clean Code Technique):

```
function Greet({ name, age }) {

  return <h1>Namaste {name}, age: {age}</h1>

}
```

🧠 Ye bhi wahi props hai, bas shortcut way me likha gaya hai

💬 Function vs Component Confusion?

- Agar Capital me likha hai → Component hai
- Small letter me likha hai → HTML tag samjha jayega

```
<Greet />      // ✅ Component
```

```
<greet />      // ❌ Invalid (React will search it as HTML tag)
```

📝 Summary (Ek Nazar Me)

- ✅ React and ReactDOM import karna padta hai JSX use karne ke liye
- ✅ JSX ke andar sirf **expression** likh sakte ho, statement nahi
- ❌ Object ko directly render nahi kar sakte
- ✅ Props use hote hai data pass karne ke liye (function ke arguments ki tarah)
- ✅ JSX me Capital letter se component banate hai (React optimization ke liye)
- ✅ ReactDOM ke `render()` function se component ko page me mount karte hai