

WEB APP DEVELOPMENT WITH REACTJS (INT252)

Lecture 5: React Tooling & Project Setup (Vite)

Unit I – JavaScript Refresher & React Foundations

Syllabus Mapping:

- React development environment
- Tooling (Node, npm)
- Project setup and structure

Target Learner: Beginner (HTML, CSS, JavaScript knowledge)

Why Tooling Matters in React

React is not used via `<script>` tags (in real projects).

Modern React requires:

- Module system
- Bundler
- Development server

This lecture shows **how React actually runs**.

What is Node.js?

Node.js:

- Runs JavaScript **outside the browser**
- Required for modern web tools

 React does NOT run on Node

Node is used only for **development & build**

What is npm?

npm = Node Package Manager

Used to:

- Install libraries (React, Router, etc.)
- Manage project dependencies
- Run scripts

Check Installation

```
node -v  
npm -v
```

If version shows → installed correctly

What is a Bundler?

A bundler:

- Combines multiple JS files
- Optimizes code
- Converts modern JS for browser

Examples:

- Webpack
- Vite

Why We Use Vite

Vite is:

- Extremely fast
- Simple configuration
- Recommended by React community

Perfect for beginners

Creating a React App Using Vite

```
npm create vite@latest react-app  
cd react-app  
npm install  
npm run dev
```

What Happens Internally

Vite Server Starts



Loads index.html



Loads main.jsx



React renders App component

Project Folder Structure

```
react-app/
  └── index.html
  └── src/
    ├── main.jsx
    ├── App.jsx
    └── assets/
  └── package.json
```

index.html (Entry Point)

```
<div id="root"></div>
```

React mounts UI here

main.jsx (React Entry)

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

ReactDOM.createRoot(document.getElementById('root'))
  .render(<App />);
```

App.jsx (Root Component)

```
function App() {  
  return <h1>Hello React</h1>;  
}  
  
export default App;
```

How React App Runs (Visual)

```
index.html  
  ↓  
main.jsx  
  ↓  
<App />  
  ↓  
Browser UI
```

package.json Explained

```
{  
  "dependencies": {  
    "react": "...",  
    "react-dom": "..."  
  }  
}
```

Lists project libraries

Development vs Production

Development:

- Fast reload
- Debugging

Production:

- Optimized build
- Smaller bundle

Common Beginner Mistakes

- ✗ Editing node_modules
- ✗ Deleting main.jsx
- ✗ Running React without dev server

Practice Exercises

1. Create a Vite React app
2. Change text in App.jsx
3. Run dev server

Answers – Practice Exercises

1. Create App

```
npm create vite@latest my-app  
cd my-app  
npm install  
npm run dev
```



Answers – Continued

2. Modify App.jsx

```
function App() {  
  return <h1>My First React App</h1>;  
}
```

Key Takeaways

- React requires tooling
- Node & npm manage dependencies
- Vite simplifies setup
- App.jsx is root UI component

Next Lecture

Lecture 6: JSX & Rendering Basics

Unit I – JavaScript Refresher & React Foundations