

## Day 11 – [3rd July 2025]

### TOPICS COVERED

#### try...catch Blocks – Error Handling in JavaScript:

try...catch blocks are used to handle errors in JavaScript gracefully without crashing the program.

Syntax:

```
try {  
    // code that may throw an error  
} catch (error) {  
    // code to run if there is an error  
}
```

Example:

```
try {  
    let result = 10 / 0;  
    console.log(result);  
    unknownFunction(); // This will throw an error  
} catch (err) {  
    console.error("An error occurred:", err.message);  
}
```

This prevents JavaScript from stopping completely when an error occurs and helps provide fallback behavior or messages to the user.

#### Introduction to React.js:

React.js is a JavaScript library developed by Facebook for building user interfaces, especially single-page applications.

It lets you create reusable components, manage data (state), and update the UI efficiently.

#### Key Features:

- ✓ Component-based architecture
- ✓ Virtual DOM for better performance
- ✓ JSX – a syntax extension to write HTML inside JavaScript
- ✓ One-way data flow

## Setting Up a React Project – Using Vite:

I used Vite, a modern build tool that is faster and more lightweight. It quickly scaffolds a React project with minimal setup.

### Steps I Followed:

- ✓ Installed the React template using Vite:  
npm create vite@latest
- ✓ Entered the project name and selected:
  - React
  - JavaScript
- ✓ Navigated into the project folder:  
cd my-vite-app
- ✓ Installed the dependencies:  
npm install
- ✓ Started the development server:  
npm run dev

The app ran on <http://localhost:5173/> and displayed the default React + Vite template.

## JSX – JavaScript XML

JSX is a syntax extension for JavaScript used in React to write HTML-like code inside JS.

It makes React components easier to write and read.

Example:

```
const element = <h1>Hello, React!</h1>;
```

Behind the scenes, this gets converted into:

```
React.createElement("h1", null, "Hello, React!");
```

Note: JSX must return only one parent element, and expressions must be wrapped in { }.

### Component Example with JSX:

```
function Welcome() {
  return (
    <div>
      <h2>Welcome to React</h2>
      <p>This is a JSX paragraph.</p>
    </div>
  );
}
```

**TOOLS USED:**

Visual Studio Code

Node.js & npm

Chrome Browser

React Developer Tools Extension