

# WEB APP DEVELOPMENT WITH REACTJS (INT252)

## Lecture 10: State & useState Hook

Unit III – State & Hooks

Syllabus Mapping:

- State in React
- useState Hook
- Dynamic UI updates

# Why State Is Needed

Normal variables:

- Change value
- ✗ Do NOT update UI

React State:

- Stores data
- ✓ Automatically updates UI

# What is State?

State is:

- Data owned by a component
- Can change over time
- Triggers re-render

## Problem with Normal Variable

```
let count = 0;  
count = count + 1; // UI will NOT update
```

## Solution: useState Hook

`useState` allows React to:

- Remember values
- Update UI when changed

## Importing useState

```
import { useState } from 'react';
```

## Basic useState Syntax

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

- `count` → state value
- `setCount` → update function

# Counter Example

```
function Counter() {  
  const [count, setCount] = useState(0);  
  
  return (  
    <>  
      <h1>{count}</h1>  
      <button onClick={() => setCount(count + 1)}>+</button>  
    </>  
  );  
}
```

# How State Updates Work

setState called



React re-renders component



UI updates

## Rules of useState

- Only call at top level
- Only inside components
- Never inside loops or conditions

# Multiple State Variables

```
const [name, setName] = useState('');  
const [age, setAge] = useState(0);
```

## Practice Exercises

1. Create counter with decrement
2. Toggle text show/hide

## Answers

```
const [show, setShow] = useState(true);
{show && <p>Hello</p>}
```

## Key Takeaways

- State controls UI
- useState triggers re-render
- Never modify state directly

## Next Lecture

Lecture 11: Handling Events & Forms