

WEB APP DEVELOPMENT WITH REACTJS (INT252)

Lecture 22: Testing Basics in React

Unit VI – Debugging, Testing & Deployment

Syllabus Mapping:

- Testing concepts
- Introduction to React Testing Library
- Component testing basics

Target Learner: Beginner → Intermediate

Why Testing Matters

Testing ensures:

- Code works as expected
- Changes don't break features
- Confidence in deployment

Professionals always test 

What is Testing?

Testing means:

- Writing code to verify code
- Checking expected behavior

Types of Testing

1. Unit Testing
2. Integration Testing
3. End-to-End (E2E)

Unit Testing (Focus of This Lecture)

- Tests a single component
- Fast and reliable

Tools for React Testing

We use:

- Jest (test runner)
- React Testing Library (RTL)

Install Testing Libraries

```
npm install --save-dev @testing-library/react @testing-library/jest-dom
```

First Test File

Naming convention:

```
Component.test.jsx
```

Simple Component to Test

```
function Greeting({ name }) {  
  return <h1>Hello, {name}</h1>;  
}
```

Writing First Test

```
import { render, screen } from '@testing-library/react';
import Greeting from './Greeting';

test('renders greeting text', () => {
  render(<Greeting name="React" />);
  expect(screen.getByText('Hello, React')).toBeInTheDocument();
});
```

What This Test Does

1. Renders component
2. Searches text
3. Verifies output

Common RTL Queries

- getByText
- getByRole
- getByLabelText

Testing User Interaction

```
import userEvent from '@testing-library/user-event';
```

Button Click Test

```
userEvent.click(screen.getByRole('button'));
```

Testing Best Practices

- Test behavior, not implementation
- Avoid testing internal state
- Keep tests simple

Common Beginner Mistakes

- ✗ Testing styles
- ✗ Over-mocking
- ✗ Skipping tests

Practice Exercises

1. Test a button render
2. Test click event
3. Test conditional rendering

Answers – Practice

1. Use getByRole
2. `userEvent.click`
3. Conditional expect

Key Takeaways

- Testing improves reliability
- RTL focuses on user behavior
- Unit tests are essential

Next Lecture

Lecture 23: Deployment & Production Build

Unit VI – Debugging, Testing & Deployment