

# WEB APP DEVELOPMENT WITH REACTJS (INT252)

## Lecture 16: Programmatic Navigation & useNavigate

Unit IV – Routing & Advanced Concepts

Syllabus Mapping:

- Programmatic navigation
- useNavigate hook
- Navigation based on logic

Target Learner: Beginner → Intermediate

# Why Programmatic Navigation Is Needed

So far we navigated using:

- `<Link>` clicks

But real apps need:

- Redirect after login
- Navigation after form submit
- Conditional redirects

# What is Programmatic Navigation?

Navigation controlled by:

- JavaScript logic
- Not user clicking links

## useNavigate Hook

`useNavigate` allows React to:

- Change routes using code

## Importing useNavigate

```
import { useNavigate } from 'react-router-dom';
```

## Basic useNavigate Syntax

```
const navigate = useNavigate();
```

## Simple Navigation Example

```
function Home() {  
  const navigate = useNavigate();  
  
  return (  
    <button onClick={() => navigate('/about')}>  
      Go to About  
    </button>  
  );  
}
```

## Navigation Flow (Visual)

Button Click



`navigate('/path')`



URL changes



New component renders



## Redirect After Login Example

```
function Login() {  
  const navigate = useNavigate();  
  
  function handleLogin() {  
    // authentication logic  
    navigate('/dashboard');  
  }  
  
  return <button onClick={handleLogin}>Login</button>;  
}
```

## Navigate with Conditions

```
if (isAuthenticated) {  
  navigate('/home');  
}
```

## Navigate Back & Forward

```
navigate(-1); // back  
navigate(1);  // forward
```

## Replace Navigation (No History)

```
navigate('/login', { replace: true });
```

Used for logout

## useNavigate vs Link

Link	useNavigate
User clicks	Code logic
Static	Dynamic

## Common Beginner Mistakes

- ✗ Calling navigate outside component
- ✗ Using useNavigate without Router
- ✗ Infinite redirects

## Practice Exercises

1. Navigate on button click
2. Redirect after condition
3. Navigate back using button

## ✓ Answers – Practice Exercises

```
<button onClick={() => navigate('/home')}>Home</button>
```

```
isLoggedIn && navigate('/dashboard');
```

```
<button onClick={() => navigate(-1)}>Back</button>
```



## Key Takeaways

- `useNavigate` enables code-based routing
- Used for redirects and logic
- Essential for authentication flows

## Next Lecture

**Lecture 17: Context API – Global State Management**

**Unit V – State Management & Advanced React**