

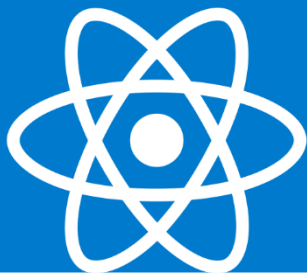


React JS

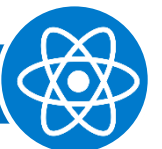
Conditional Rendering

#React Notes

| Conditional Rendering

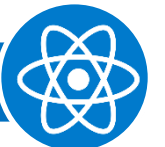


- Process of executing logic or rendering UI elements basis certain conditions is called conditional rendering.
- Conditional Rendering is used to show or hide elements based on certain conditions



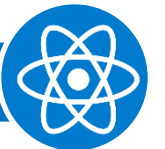
Methods for conditional rendering

- if/else
- Ternary operation
- Inline IF with Logical && operator
- Switch case operator
- Conditional Rendering with enums



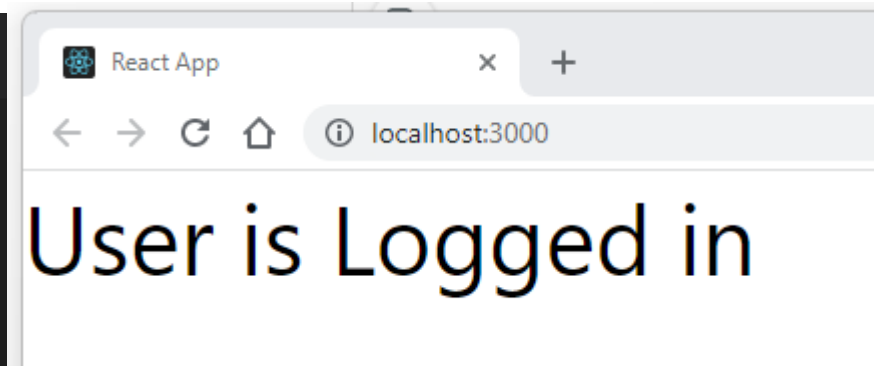
if/else

- Conditional rendering in React works the same way conditions work in JavaScript.
- Use JavaScript operators like if, and let React update the UI to match them. We use an if with our condition and return the element to be rendered

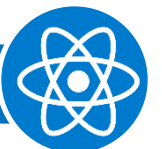


If Demo

```
JS App.js
src > JS App.js > ...
1  import React from "react";
2
3  function App() {
4    var isLoggedIn = true;
5
6    if (isLoggedIn) {
7      return <>User is Logged in</>;
8    }
9    return <>User is Not Logged in</>;
10 }
11 export default App;
12
```



```
import React from "react";
function App() {
  var isLoggedIn = true;
  if (isLoggedIn) {
    return <>User is Logged in</>;
  }
  return <>User is Not Logged in</>;
}
export default App;
```

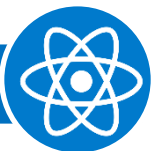


If Demo Using Props and Return Component

```
JS App.js x
src > JS App.js > ...
1  import React from "react";
2
3  function Loggedin(props) {
4    return (<>User is Logged In</>)
5  }
6  function Loggedout(props) {
7    return (<>User is Logout</>)
8  }
9
10 function Home(props) {
11   var isLoggedIn = props.isLoggedIn;
12
13   if (isLoggedIn) {
14     return <Loggedin />;
15   }
16   return <Loggedout />;
17 }
18
19 function App() {
20   return (<>
21     <Home isLoggedIn={true} />
22   </>)
23 }
24 export default App;
25
```

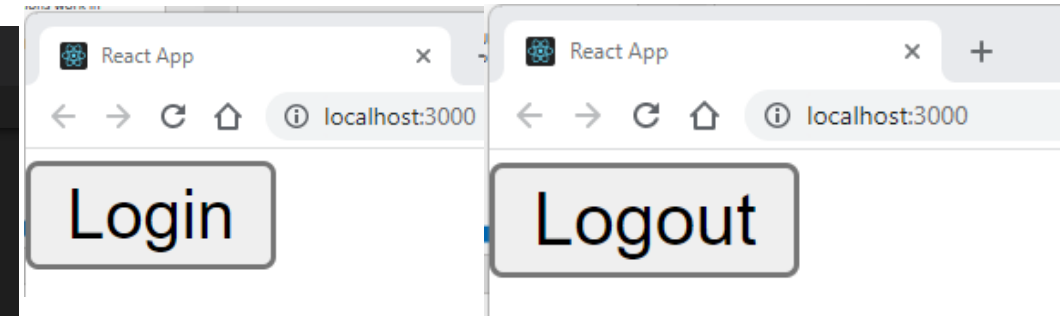
User is Logged In

```
import React from "react";
function Loggedin(props) {
  return (<>User is Logged In</>)
}
function Loggedout(props) {
  return (<>User is Logout</>)
}
function Home(props) {
  var isLoggedIn = props.isLoggedIn;
  if (isLoggedIn) {
    return <Loggedin />;
  }
  return <Loggedout />;
}
function App() {
  return (<>
    <Home isLoggedIn={true} />
  </>)
}
export default App;
```



State Value Change Conditional Render

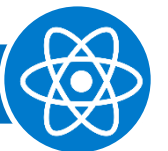
```
App.js
src > App.js > default
1 import React from 'react';
2 class App extends React.Component {
3   constructor(props) {
4     super(props);
5     this.state = { isLoggedIn: false };
6   }
7
8   handleLogout(){
9     this.setState({ isLoggedIn: false });
10  }
11  handleLogin (){
12    this.setState({ isLoggedIn: true });
13  }
14
15  render() {
16    let myBtn = null;
17    if (this.state.isLoggedIn) {
18      myBtn = <button onClick={this.handleLogout.bind(this)}>Logout</button>
19    } else {
20      myBtn = <button onClick={this.handleLogin.bind(this)}>Login</button>
21    }
22    return (
23      <div>
24        {myBtn}
25      </div>
26    );
27  }
28 }
29 export default App;
```



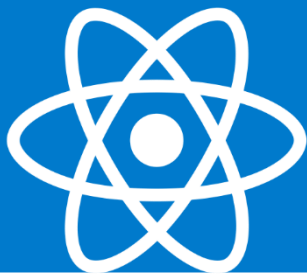
```
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = { isLoggedIn: false };
  }

  handleLogout(){
    this.setState({ isLoggedIn: false });
  }
  handleLogin (){
    this.setState({ isLoggedIn: true });
  }

  render() {
    let myBtn = null;
    if (this.state.isLoggedIn) {
      myBtn = <button onClick={this.handleLogout.bind(this)}>Logout</button>
    } else {
      myBtn = <button onClick={this.handleLogin.bind(this)}>Login</button>
    }
    return (
      <div>
        {myBtn}
      </div>
    );
  }
}
export default App;
```

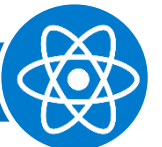


| Ternary operation



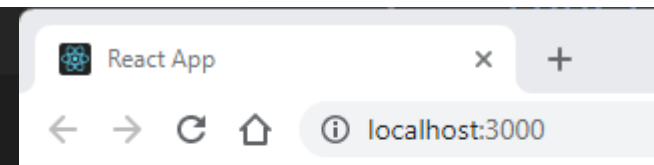
Ternary operation

- The conditional (ternary) operator is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the if statement.
- condition ? “This is True” : “This is False”
- When condition evaluates to true, the operator returns “This is True”; otherwise (when condition is falsy) it returns “This is False”.



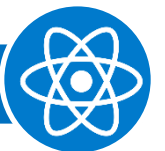
Ternary Operator Using Element Variables

```
App.js
src > App.js > ...
1 import React from "react";
2 function App() {
3
4   var isLoggedIn = true;
5
6   var ans = isLoggedIn ? "logged in" : "Logged out";
7
8   return(
9     <>
10    {ans}
11    </>
12  )
13 }
14 export default App;
15
```



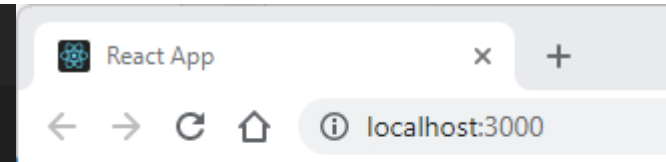
logged in

```
import React from "react";
function App() {
  var isLoggedIn = true;
  var ans = isLoggedIn ? "logged in" : "Logged out";
  return(
    <>
    {ans}
    </>
  )
}
export default App;
```



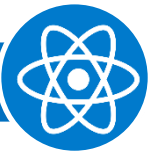
Ternary Operator (Inline if-else)

```
JS App.js ×
src > JS App.js > ...
1 import React from "react";
2 function Loggedin(props) {
3   return (<>User is Logged In</>)
4 }
5 function Loggedout(props) {
6   return (<>User is Logout</>)
7 }
8
9 function App() {
10   var isLoggedIn = true;
11   return (<>
12     {
13       isLoggedIn ? (<Loggedin />) : (<Loggedout />)
14     }
15   </>)
16 }
17 export default App;
```



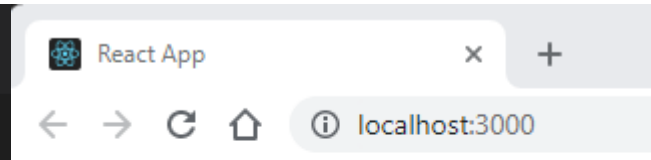
User is Logged In

```
import React from "react";
function Loggedin(props) {
  return (<>User is Logged In</>)
}
function Loggedout(props) {
  return (<>User is Logout</>)
}
function App() {
  var isLoggedIn = true;
  return (<>
    {
      isLoggedIn ? (<Loggedin />) : (<Loggedout />)
    }
  </>)
}
export default App;
```



Nested Ternary Operator

```
Js App.js x
src > Js App.js > [🔍] default
1  import React from "react";
2  function App() {
3
4      var isRegistered = false;
5      var isLoggedIn = false;
6
7      return (
8          <>
9              {isLoggedIn
10                 ? <p>Welcome to Website</p>
11                 : (isLoggedIn
12                    ? <p>Do Login</p>
13                    : <div>Do Registration</div>
14                 )
15             }
16          </>
17      )
18  }
19  export default App;
20
```

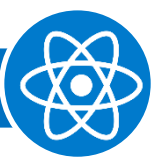


Do Registration

```
import React from "react";
function App() {

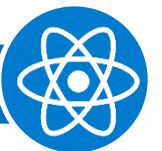
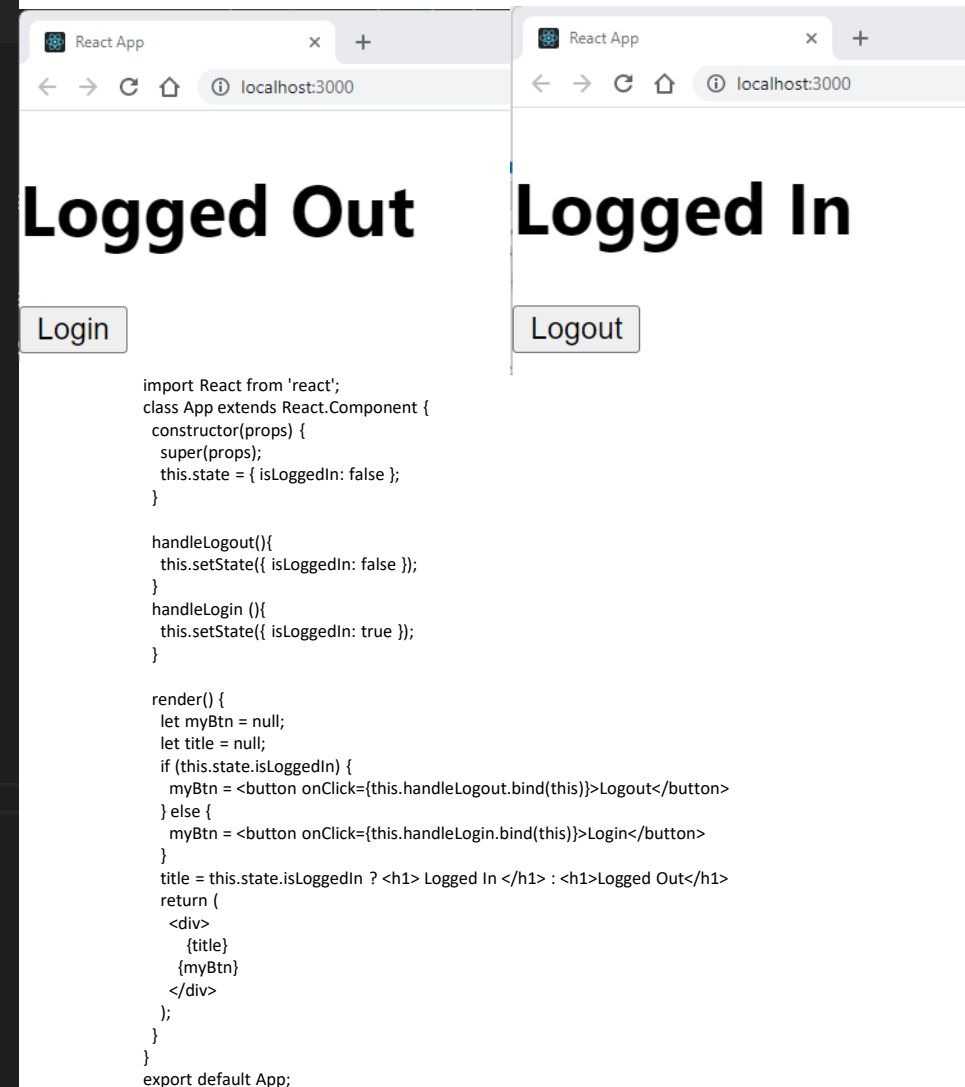
    var isRegistered = false;
    var isLoggedIn = false;

    return (
        <>
        {isLoggedIn
        ? <p>Welcome to Website</p>
        : (isLoggedIn
        ? <p>Do Login</p>
        : <div>Do Registration</div>
        )
        }
        </>
    )
}
export default App;
```

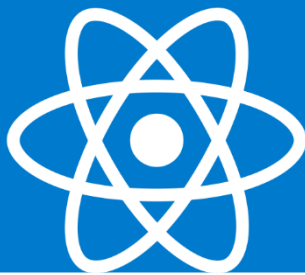


Conditional Render using ?

```
Js App.js
src > Js App.js > App > render
1 import React from 'react';
2 class App extends React.Component {
3   constructor(props) {
4     super(props);
5     this.state = { isLoggedIn: false };
6   }
7
8   handleLogout(){
9     this.setState({ isLoggedIn: false });
10  }
11  handleLogin (){
12    this.setState({ isLoggedIn: true });
13  }
14
15  render() {
16    let myBtn = null;
17    let title = null;
18    if (this.state.isLoggedIn) {
19      myBtn = <button onClick={this.handleLogout.bind(this)}>Logout</button>
20    } else {
21      myBtn = <button onClick={this.handleLogin.bind(this)}>Login</button>
22    }
23    title = this.state.isLoggedIn ? <h1> Logged In </h1> : <h1>Logged Out</h1>
24    return (
25      <div>
26        {title}
27        {myBtn}
28      </div>
29    );
30  }
31 }
32 export default App;
```

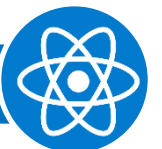


| Inline IF with | Logical && operator



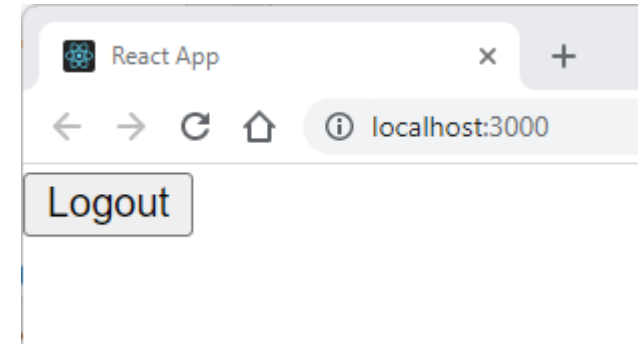
Inline IF with Logical && operator

- && is a boolean operator, which essentially means “and”.
- For the condition to evaluate to true, both of the statements must be individually true.



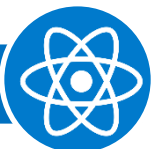
Inline IF with Logical && operator

```
JS App.js X
src > JS App.js > ...
1  import React from "react";
2  function App() {
3    var isLoggedIn = true;
4
5    return (
6      <>
7        {isLoggedIn && <button>Logout</button>}
8        {!isLoggedIn && <button>Login</button>}
9      </>
10   )
11 }
12 export default App;
13
```



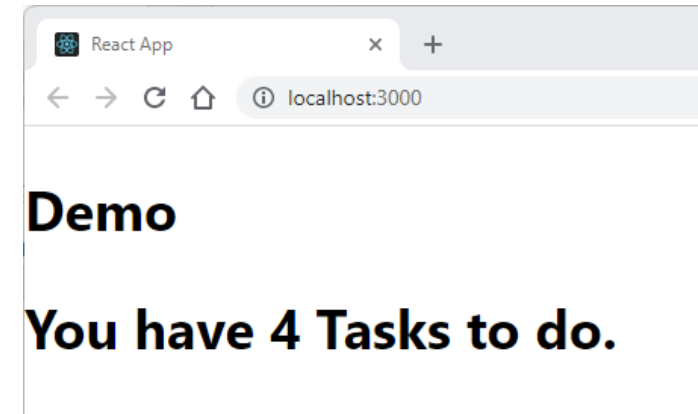
```
import React from "react";
function App() {
  var isLoggedIn = true;

  return (
    <>
      {isLoggedIn && <button>Logout</button>}
      {!isLoggedIn && <button>Login</button>}
    </>
  )
}
export default App;
```

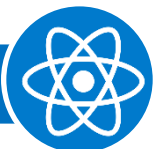


Inline IF with Logical && operator

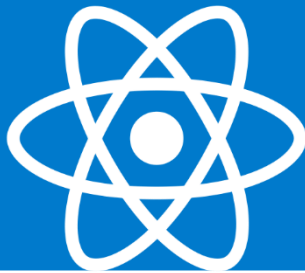
```
Js App.js ×
src > Js App.js > [🔗] default
1
2 function App() {
3   const todo = ['Eat', 'Sleep', 'Code', 'Repeat'];
4
5   return (<>
6     <h2>Demo</h2>
7
8     {todo.length > 0 &&
9       <h2>
10        You have {todo.length} Tasks to do.
11      </h2>
12    }
13   </>);
14 }
15 export default App;
```



```
function App() {
  const todo = ['Eat', 'Sleep', 'Code', 'Repeat'];
  return (<>
    <h2>Demo</h2>
    {todo.length > 0 &&
      <h2>
        You have {todo.length} Tasks to do.
      </h2>
    }
  </>);
}
export default App;
```

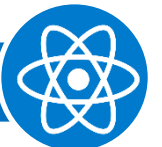


| Switch



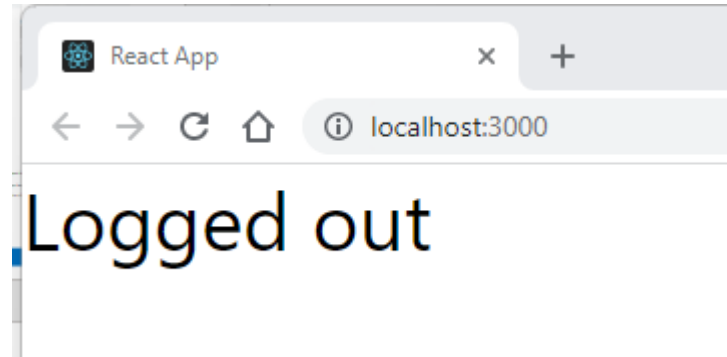
Switch

- If multiple conditions are present, then if-statement is not the best way to go as it may reduce performance.
- In such cases, we can use switch statements. To avoid re-rendering



Switch Example

```
Js App.js ×
src > Js App.js > ...
1  function App() {
2
3    var isLoggedIn = false;
4
5    switch (isLoggedIn) {
6      case true:
7        return ("Logged in");
8      case false:
9        return ("Logged out");
10     default:
11       return ("Not Set");
12   }
13 }
14 export default App;
15
```

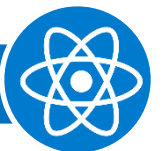


```
function App() {

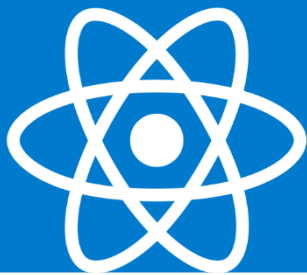
  var isLoggedIn = false;

  switch (isLoggedIn) {
    case true:
      return ("Logged in");
    case false:
      return ("Logged out");
    default:
      return ("Not Set");
  }
}

export default App;
```

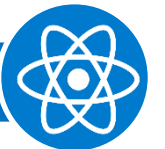


| Enum



Enum

- In JavaScript, an object can be used as an enum when it is used as a map of key-value pairs.



Enum

```
JS App.js x
src > JS App.js > [e] default
1 import React from 'react'
2
3 const Foo = () => {
4   return <button>FOO</button>;
5 };
6 const Default = () => {
7   return <button>DEFAULT</button>;
8 };
9 const ENUM_STATES = {
10   foo: <Foo />,
11   default: <Default />
12 };
13
14 function EnumState({ state }) {
15   return <div>{ENUM_STATES[state]}</div>;
16 }
17
18 class App extends React.Component {
19   render() {
20     return (
21       <div>
22         <h1>Conditional Rendering with enums</h1>
23         <EnumState state="default"></EnumState>
24         <EnumState state="foo"></EnumState>
25       </div>
26     );
27   }
28 }
29 export default App;
```

Conditional Rendering with enums

DEFAULT

FOO

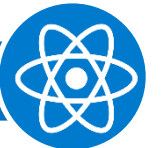
```
import React from 'react'

const Foo = () => {
  return <button>FOO</button>;
};
const Default = () => {
  return <button>DEFAULT</button>;
};
const ENUM_STATES = {
  foo: <Foo />,
  default: <Default />
};

function EnumState({ state }) {
  return <div>{ENUM_STATES[state]}</div>;
}

class App extends React.Component {
  render() {
    return (
      <div>
        <h1>Conditional Rendering with enums</h1>
        <EnumState state="default"></EnumState>
        <EnumState state="foo"></EnumState>
      </div>
    );
  }
}

export default App;
```

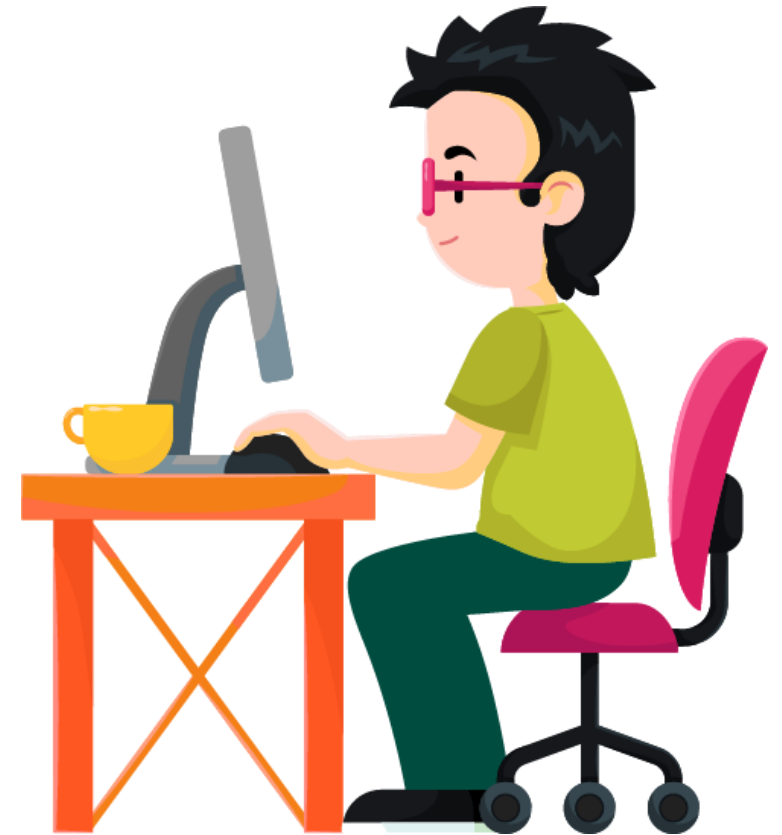


Get Exclusive Video Tutorials



www.apptutorials.com

<https://www.youtube.com/user/Akashtips>





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



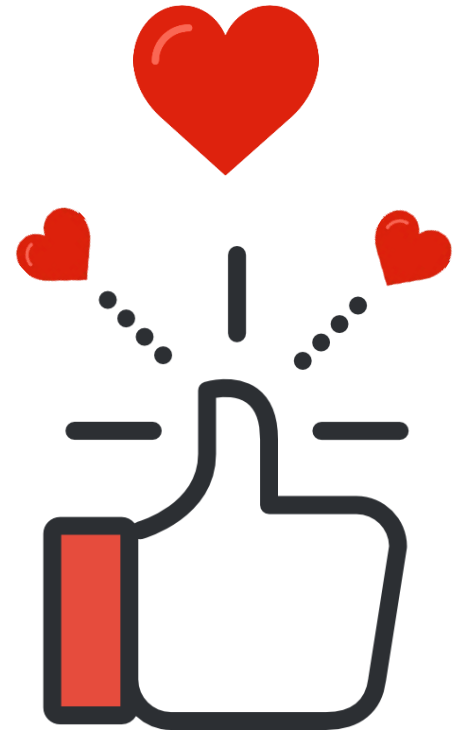
Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com

www.akashtechlabs.com

www.akashpadhiyar.com

www.apptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar

#apptutorials