



# Lab 06 - React - Mini Apps

SENG 4640

Software Engineering for Web Apps

Sina Keshvadi

Thompson Rivers University

In this lab, you'll create six mini apps.

## Steps:

Step 1: Break the UI into a component hierarchy

Step 2: Build a static version in React

Step 3: Find the minimal but complete representation of UI state

Step 4: Identify where your state should live

Step 5: Add inverse data flow

## **App 1 - Click Counter (5 Marks)**



Button Click



Guest



react-click.html



Count: 0

Click Me!

Count: 1

Click Me!

## Steps:

Step 1: Break the UI into a component hierarchy

Step 2: Build a static version in React

Step 3: Find the minimal but complete representation of UI state

Step 4: Identify where your state should live

Step 5: Add inverse data flow

## **App 2. Like and Dislike (10 Marks)**

Java 👍 Like

JavaScript 👍 Like

Java 👍 Like



JavaScript 👍 Like

**Java** 👍 Unlike

JavaScript 👍 Like

**Java** 👍 Unlike

JavaScript 👍 Like

**Java** 👍 Unlike

**JavaScript** 👍 Unlike

**Java** 👍 Unlike

**JavaScript** 👍 Unlike

Java 👍 Like

JavaScript 👍 Unlike

## **App 3. React Mouse (15 Marks)**

Look at me!

Look at me!

Mouse over

Look at me!





# Look at me!



Mouse over

Look at me!

Click

## **App 4. List Filtering (15 Marks)**

   react-list.html

- anteater
- bear
- cat
- dog
- elephant
- fox

a

no need to press enter key

- anteater
- bear
- cat
- elephant

ant

- anteater
- elephant

a

- anteater
- bear
- cat
- elephant

FilteredList

a

- anteater
- bear
- cat
- elephant

FilteredList

ListItems

## **App 5. To-Do List (25 Marks)**

## Steps:

Step 1: Break the UI into a component hierarchy

Step 2: Build a static version in React

Step 3: Find the minimal but complete representation of UI state

Step 4: Identify where your state should live

Step 5: Add inverse data flow



react-todo.html




# TO-DO LIST

Add

# TO-DO LIST

Add

 react-todo.html

# TO-DO LIST

- learn JavaScript

# TO-DO LIST

- learn JavaScript
- find happiness

# TO-DO LIST

- ~~learn JavaScript~~
- find happiness

# TO-DO LIST

- ~~learn JavaScript~~
- find happiness

Add

ToDoApp component

ToDoList component

ToDoItem component

## **App 6. Multiply App (30 Marks)**

## Steps:




Step 1: Break the UI into a component hierarchy

Step 2: Build a static version in React

Step 3: Find the minimal but complete representation of UI state

Step 4: Identify where your state should live

Step 5: Add inverse data flow

    react-multiply.html 



Enter two numbers to multiply:

<input type="text"/>	<input type="text"/>
----------------------	----------------------

The product is 0.

Enter two numbers to multiply:

The product is 0.

   react-multiply.html 

Enter two numbers to multiply:

The product is 72.

 react-multiply.html



Enter two numbers to multiply:

The product is 752.

   react-multiply.html 

Enter two numbers to multiply:

The product is 7802.

   react-multiply.html 

Enter two numbers to multiply:

The product is 7858.4.

   react-multiply.html 

Enter two numbers to multiply:

The product is 7858.4.

react-multiply.html

Enter two numbers to multiply:

The product is 7858.4.

   react-multiply.html 

Enter two numbers to multiply:

83.6

94

The product is 7858.4.

react-multiply.html

Enter two numbers to multiply:

The product is 7858.4.

react-multiply.html

Enter two numbers to multiply:

The product is 7858.4.

Zip all three apps and submit via Moodle