

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.

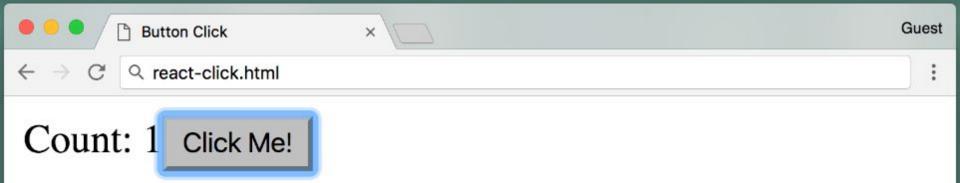
Steps:

- 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)



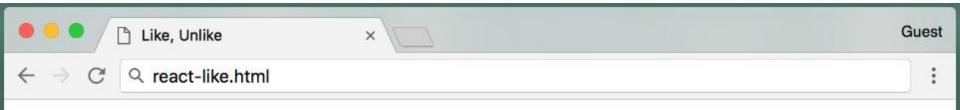
Count: 0 Click Me!



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

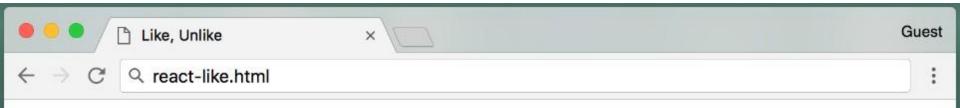
App 2. Like and Dislike (10 Marks)



Java 👍 Like



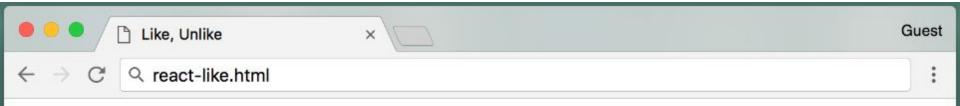
Java 👍 Like



Java 🖕 Unlike

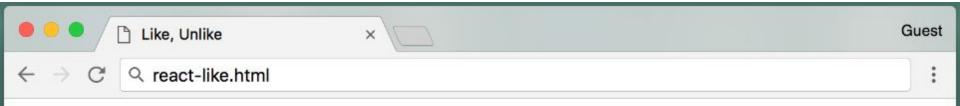


Java 🖕 Unlike



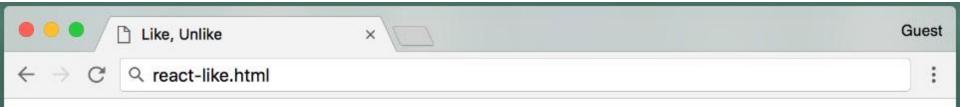
Java 🖕 Unlike

JavaScript 🖕 Unlike



Java 🖕 Unlike 🖈

JavaScript dunlike



k

Java 👍 Like

JavaScript dunlike

App 3. React Mouse (15 Marks)

•••/	Mouse Over, Mouse Out	Guest
$\leftrightarrow \ \Rightarrow \ G$	Q react-mouse.html	:

•••	Mouse Over, Mouse Out	Guest
$\ \ \leftarrow \ \ \Rightarrow \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	Q react-mouse.html	:

Mouse over

•••/	Mouse Over, Mouse Out	Guest
$\leftrightarrow \ \Rightarrow \ C$	Q react-mouse.html	

K

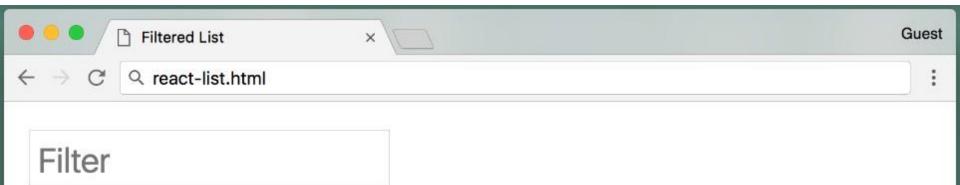
•••	Mouse Over, Mouse Out	Guest
$\ \ \leftarrow \ \ \Rightarrow \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	Q react-mouse.html	:

Mouse over

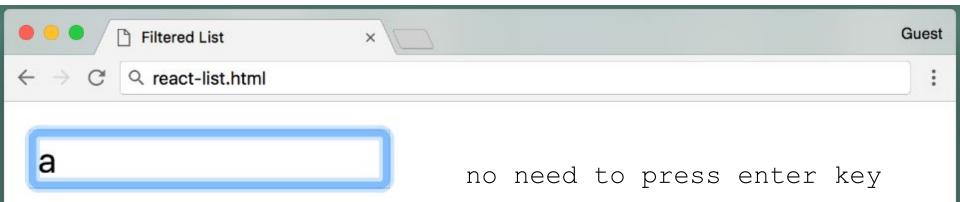
•••/	Mouse Over, Mouse Out	Guest
$\leftrightarrow \ \Rightarrow \ G$	React-mouse.html	:

Click

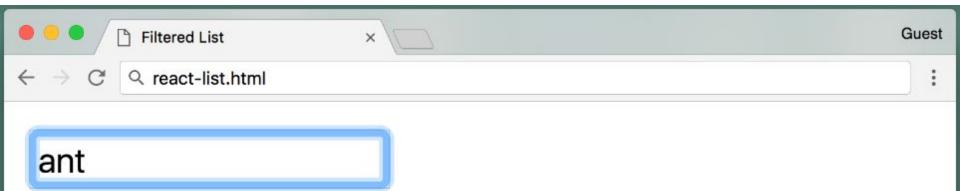
App 4. List Filtering (15 Marks)



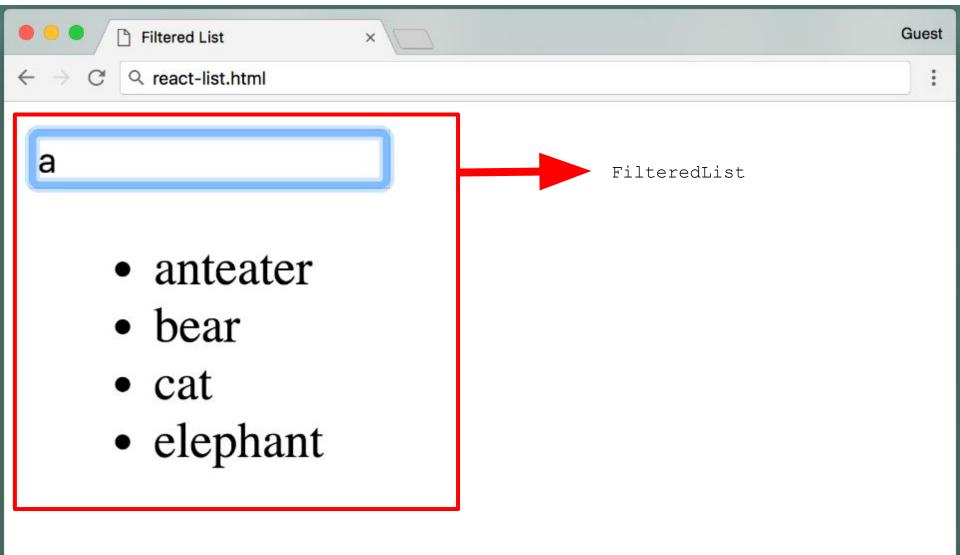
- anteater
- bear
- cat
- dog
- elephant
- fox

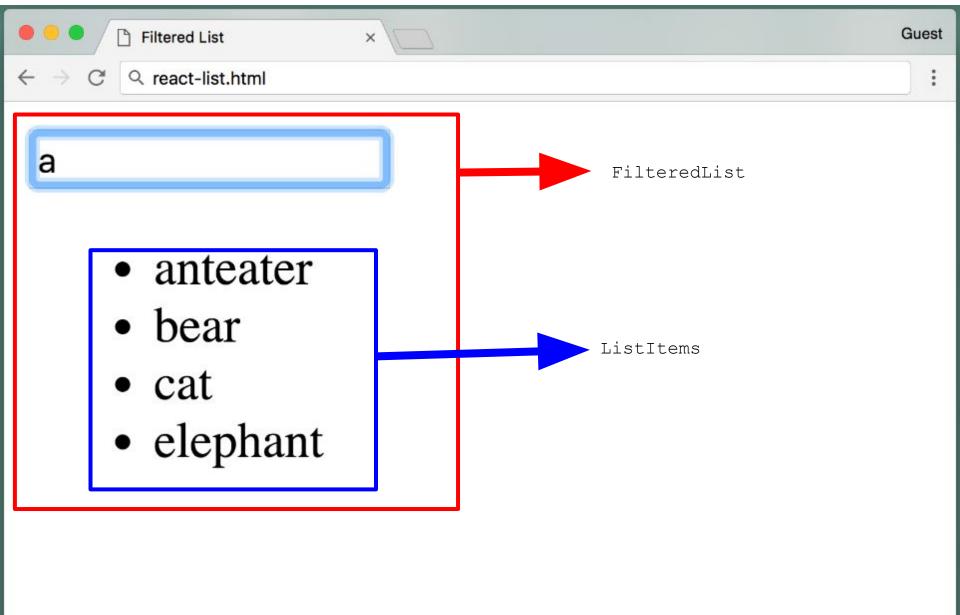


- anteater
- bear
- cat
- elephant



- anteater
- elephant





App 5. To-Do List (25 Marks)

Steps:

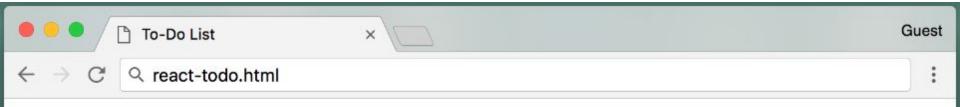
- Step 1: Break the UI into a component hierarchy
- Step 2: Build a static version in React
- Step 3: Find the minimal but complete representation of UI state
- Step 4: Identify where your state should live
- Step 5: Add inverse data flow

•••/	🗋 To-Do List	×	Guest
$\ \ \leftarrow \ \ \Rightarrow \ \ G$	Q react-todo.html		

Add

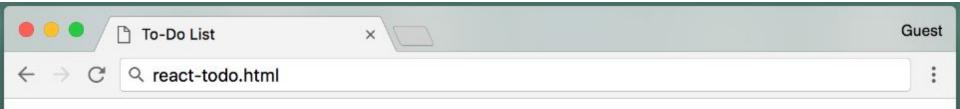
•••/	🗋 To-Do List	×	Guest
$\leftrightarrow \ \ni \ G$	♀ react-todo.html		:

learn JavaScript Add



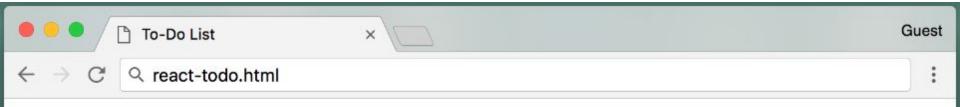
learn JavaScript





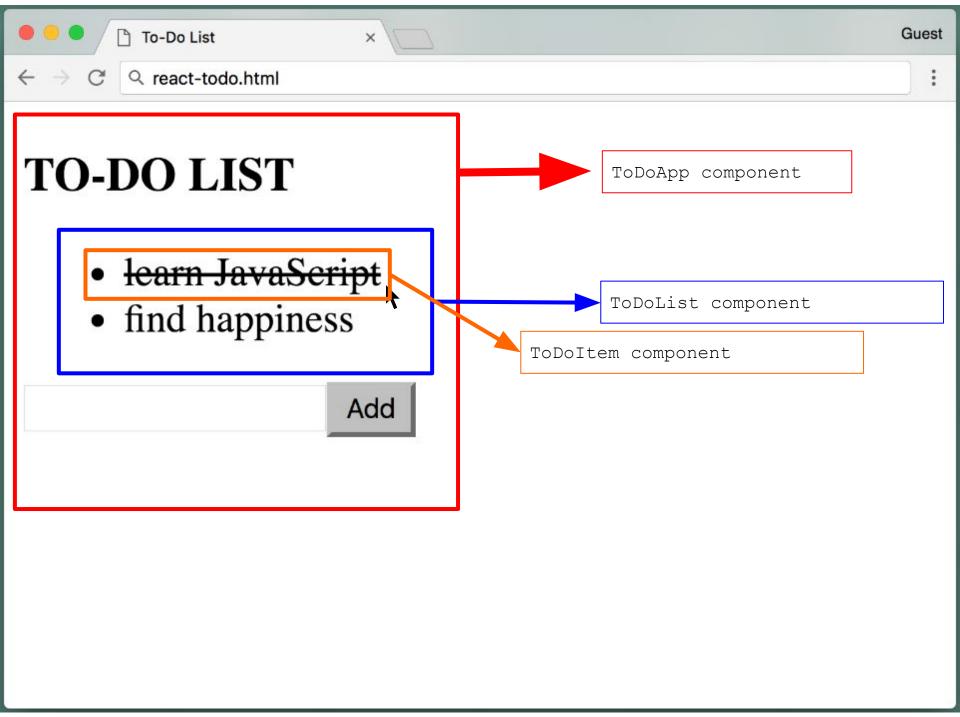
- learn JavaScript
- find happiness





- learn JavaScript
- find happiness

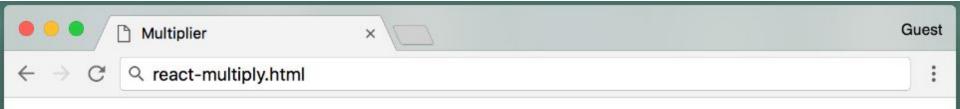
Add



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

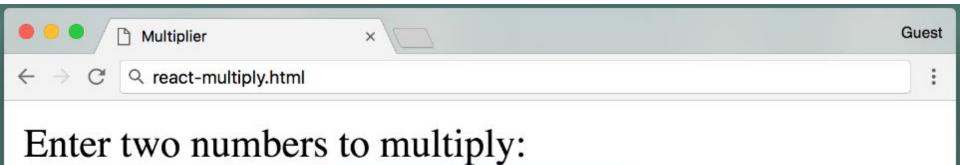


The product is 0.

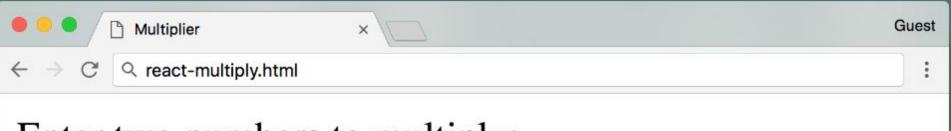


8

The product is 0.



The product is 72.



94

8

The product is 752.

•••/1	Multiplier	×	Guest
$\leftarrow \ \Rightarrow \ G$	Q react-multiply.html):
T			

94

83

The product is 7802.

•••/	D Multiplier	×	Guest
$\leftarrow \ \ni \ C$	Q react-multiply.html		:

94

83.6

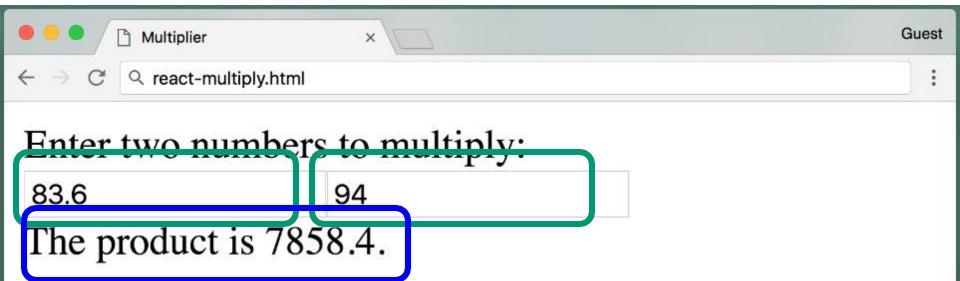
The product is 7858.4.

•••/	🗋 Multiplier	×	Guest
$\leftarrow \ \ni \ C$	Q react-multiply.html		

83.6 94

The product is 7858.4.

• • • Multiplier	×	Guest
\leftrightarrow \Rightarrow C \bigcirc react-mult	iply.html):
Enter two nu	mbers to multiply.	
83.6	94	
The product i		
The product i	.5 70.50.7.	



• • • Multiplier	×	Guest
\leftrightarrow \rightarrow C \bigcirc react-multiple	y.html	:
Enter two num	bers to multiply.	
83.6	94	
The product is	7858.4.	
- Product is		

• • • Multiplier	×	Guest
\leftrightarrow \rightarrow C \bigcirc react-multiple	y.html	:
Enter two num	bers to multiply.	
83.6	94	
The product is	7858.4.	
- Product is		

Zip all three apps and submit via Moodle