Coding Activity 5.1

Refactoring The ATM (120:00)

#### **Learning Outcomes Addressed**

 1.Implement user-friendly UI and React components

 2.Manage state of various UI elements

 3.Validate form content in conjunction with state of other components

In this activity, you’ll walk through specific steps that guide you into one way of refactoring the ATM.

**ATM Improvements Task 1 - Select Refactor**

## Adding Features to ATM

Through this week's videos, you have seen the steps required to implement a basic ATM. Now, it's your turn to implement a few improvements to the same ATM shown in the browser window on the right.

Looking at this implementation, the following are a few short comings:

* **No input validation**: This means that you can make your account balance go negative by withdrawing more money than is actually available in the account.
* **Confusing UI**: The Deposit and Cash Back options are right next to each other and there is no difference in the UI other than a label above the number input box, it is easy for a user to do the opposite of what they intended.

For the user to toggle between a Deposit UI and a Cash Back UI, you need to change this UI, providing a more clearly defined difference between them