React Native Intro Workshop

Evan L Daley

Agenda

Activity	Time
Introduction to React Native	5 minutes
Creating an App with the Expo CLI	5 minutes
Creating Our First Component	10 minutes
Component Styling	5 minutes
Creating Our Second Component	5 minutes
Creating a FlatList	10 minutes
Form Elements	10 minutes
Passing Props	20 minutes
Wrapping up	10 inutes

Hola!

- My name is Evan L. Daley

- Instructor Lead for Spokane nucamp community.

- Full Stack Dev for etailz inc.

- Fun fact: I have two cats (Charlie and Abby)

Welcome to React Native!

Here are some of the key concepts
That I'll be going over this evening:

• Expo CLI to scaffold the app
• Custom Components
• React Native Components:
    • <View>, <Text>, <Icon>
    • <ScrollView>
	• react-native-elements
	    • <Card>
	• react-navigation
	    • createStackNavigator
	    • createDrawerNavigator

Next slides will review these concepts,
along with the Exercises you did this week.

Web App vs. Native Mobile OS App

- **React Native** is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android.

- It has a React-like component structure. Many things are the same, some different.

- We will use Built-in React Native components/elements as well as a separate library called React UI Elements.

- React Native gives you a standalone mobile app (not a web page).

- Can access native features (camera, push notifications, ARKit, device storage, location features, etc)
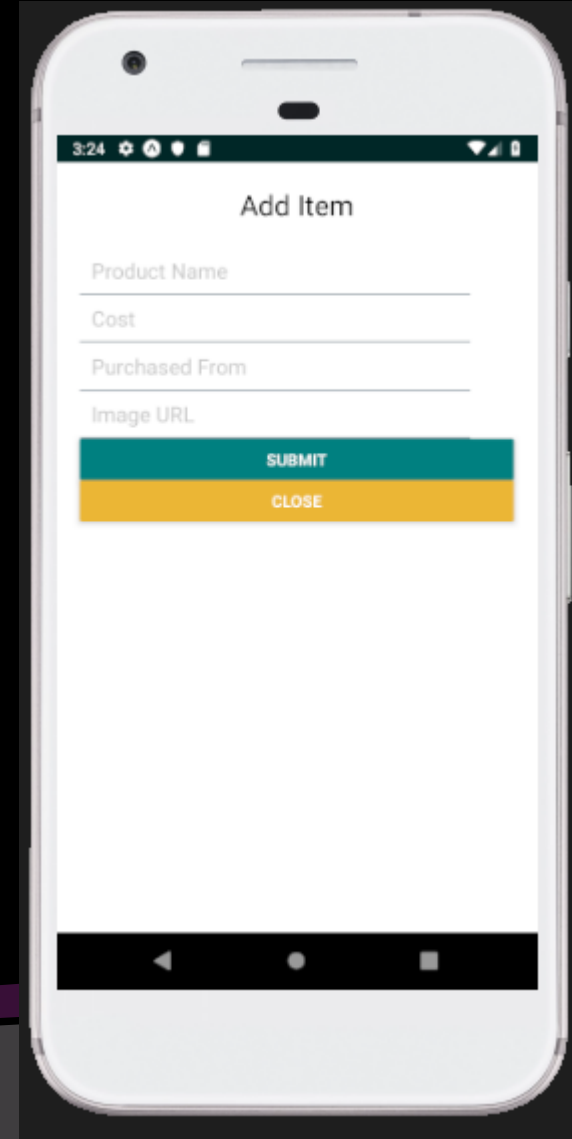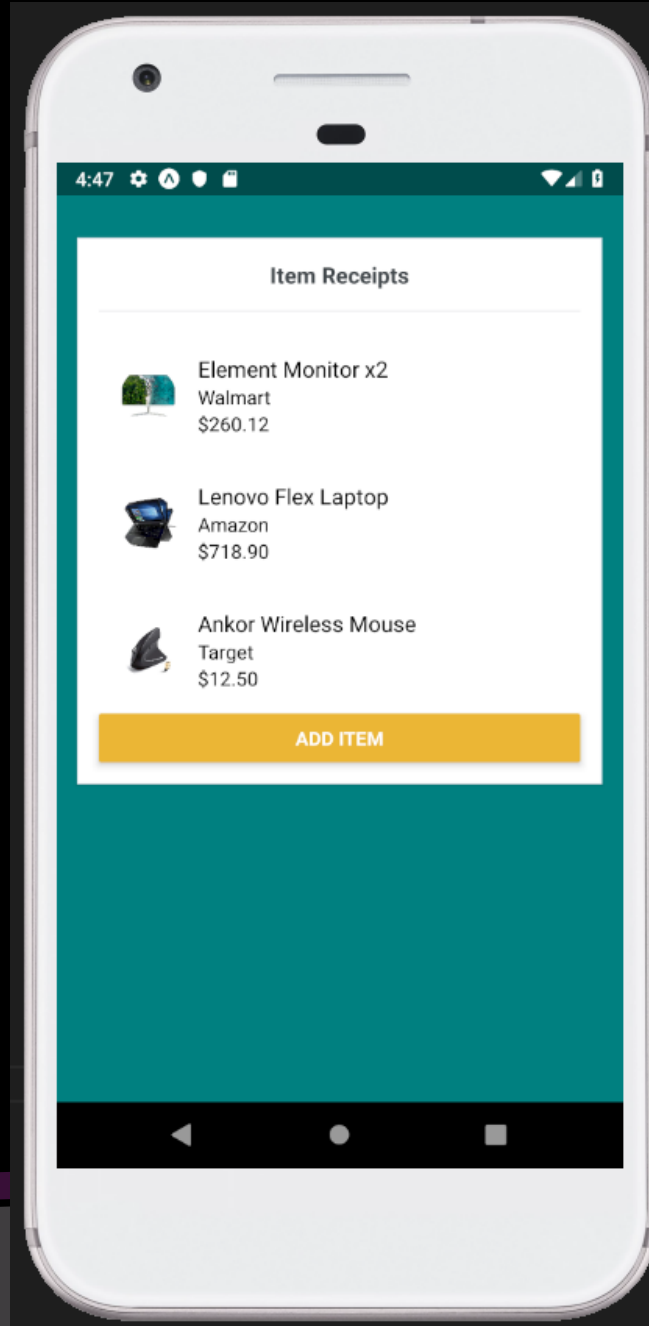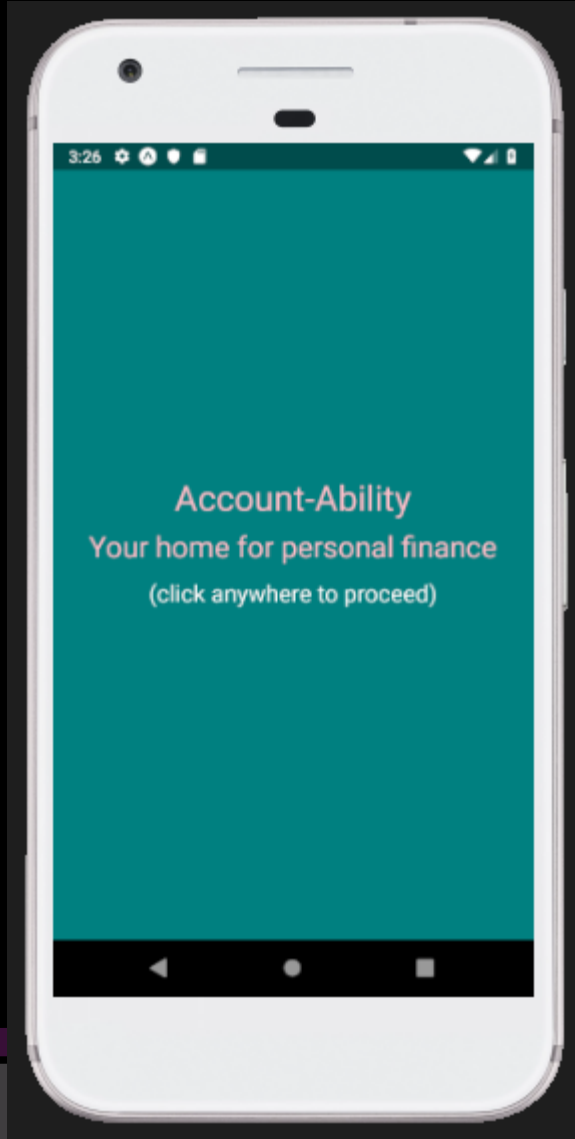
Expo Toolchain

- A free and open source toolchain built around **React Native** to help you build native iOS and Android apps using JavaScript and React.

- Wirelessly deploy to your phone or an Android/iOS emulator in seconds.

- Over the air updates to published projects.
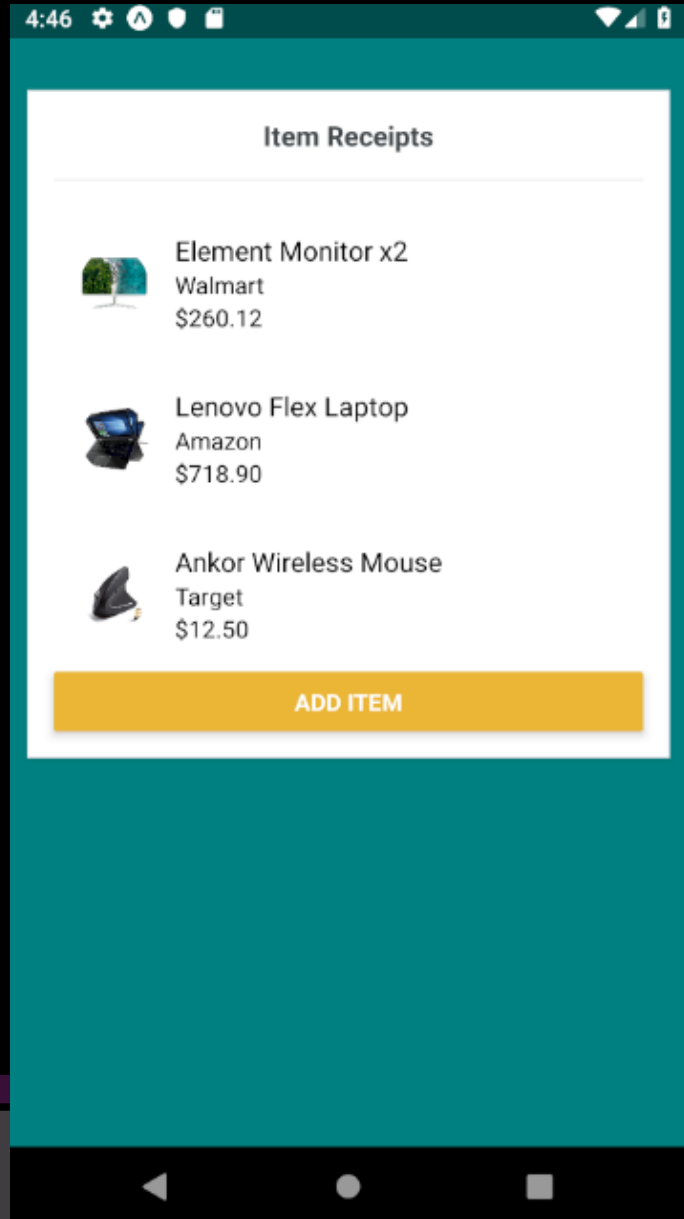
Today's Goal

Today's Goal

Account-Ability
Your home for personal finance

(click anywhere to proceed)

- Functional Component
- Custom Styles (color, font, padding)
- TouchableOpacity for handling touch events

Today's Goal

- Class Component
- Button
- Card
- Flatlist & RenderItem with Images
- Read/Save State

Today's Goal

- Modal
- Form Elements
- Buttons

Skill Check-In

- JavaScript ES6
  - Arrow functions
  - Class Inheritance
  - Object Destructuring
- React
  - Components
  - State
  - Props

```
// Arrow Functions
function func(param) {return param;}
const func = (param) => {param;}

// Object Destructuring
const key = this.object.key;
const { key } = this.object;
```

Exercise Creating a React Native App

- The Expo CLI – Objectives:

  - Use the Expo CLI to scaffold out a React Native app

    - `expo init demoApp`

  - Serve the project using:

    - `expo start`

- If you want to try it out you can install the Expo client app on your Android/iOS device scan the QR code on the screen.

React Native Components Part 1

Objectives:

- Create a home screen called "Home" as a functional component.

- Set up the App component to render Home as its child component.

Demo: Creating a splash screen

React Native Components Part 1 (cont)

```
export default (Home = props => {
  return (
    <View style={styles.homeContainer}>
      <Text style={{ color: "pink", fontSize: 30 }}>Account-Ability</Text>
      <Text
        style={{
          color: "pink",
          fontSize: 25,
          paddingTop: 5,
          paddingBottom: 10
        }}
      >
        Your home for personal finance
      </Text>
    </View>
  );
});

const styles = StyleSheet.create({
  homeContainer: {
    flex: 1,
    backgroundColor: "#fff",
    alignItems: "center",
    justifyContent: "center",
    backgroundColor: "teal"
  }
});
```

React Native Components Part 2

Objectives:

- Add a "click anywhere to proceed" message.

- Update the App.js component state with

  { isAwaitingClick: true }

- Handle the click with a TouchableOpacity component.

  { isAwaitingClick: false }

- Render components conditionally based on state.isAwaitingClick

Demo: User action triggers the next page to load

React Native Components Part 2

Objectives:

- Add Card component for the receipts

- Read data from shared items.js file

- Add FlatList component

- Create RenderItem component for elements in FlatList

```
return (
  <View style={styles.itemsContainer}>
    <Card title="Item Receipts">
      <FlatList
        data={this.state.items}
        renderItem={renderItem}
        keyExtractor={item => item.id.toString()}
      />
      <Button title="Add Item" color="#EBB634" />
    </Card>
  </View>
);
```

```jsx
const renderItem = ({ item, index }) => {
  const Sub = () => {
    const cost = item.cost ? "$" + parseFloat(item.cost).toFixed(2) : "na";


    return (
      <View>
        <Text style={{ textAlign: "left" }}>
          {item.purchased_from}
        </Text>
        <Text style={{ textAlign: "left" }}>
          {cost}
        </Text>
      </View>
    );
  };
  return (
    <ListItem
      key={index}
      title={item.name}
      subtitle={<Sub />}
      leftAvatar={{
        source: item.avatar_url && { uri: item.avatar_url },
        title: item.name
      }}
    />
  );
};
```

React Native Components Part 2

Objectives:

- Add the "Add Item" Button

- Add modalVisible to Item component state.

- Toggle modalVisible value when "Add Item is clicked"

Demo: User action triggers modal

React Native Components Part 2

Objectives:

- Add form fields to modal.

- Save new item to component state.

Going Forward

Each of these concepts is covered in the Nucamp React Native course:

- Camera support
  - Access permissions and storage

- Proper state management with Redux

- How to persist application state to device storage

- Push notifications

- Animations

- Complex app navigation (layers, tabs, menus)

Thanks for coming!

- Now you can stand up an app with the Expo cli. You can create components, and navigation elements. And you know how to handle props and state!

- If you want to go deeper, ask me about Nucamp. We have a 4 week React Native course.