Name badge

```
import { StatusBar } from ’expo-status-bar’;
import { SafeAreaView, StyleSheet, Text, View } from ’react-native’;
import * as ScreenOrientation from ’expo-screen-orientation’;

export default function App() {
  ScreenOrientation.lockAsync(ScreenOrientation.OrientationLock.LANDSCAPE_LEFT);
  return (
    <SafeAreaView style={styles.safeArea}>
      <View style={styles.container}>
        <Text style={styles.welcomeText}>Hello</Text>
        <Text style={styles.subtitleText}>my name is</Text>
        <View style={styles.whiteBox}>
          <Text style={styles.whiteBoxText}>Nick    (he/him)</Text>
        </View>
        <StatusBar style="auto" />
      </View>
    </SafeAreaView>
  );
}

const styles = StyleSheet.create({
  safeArea: {
    flex: 1,
    width: "100%",
    height: "100%",
    backgroundColor: ’light gray’,
  },
  container: {
    backgroundColor: ’red’,
    alignItems: ’center’,
    justifyContent: ’center’,
    paddingLeft: 50,
    paddingRight: 50,
    paddingBottom: 50,
  },
  welcomeText: {
    fontSize: 90,
    textTransform: ’uppercase’,
    fontWeight: ’bold’,
    color: ’white’,
    textAlign: ’center’
  },
  subtitleText: {
    fontSize: 30,
    textTransform: ’uppercase’,
    fontWeight: ’bold’,
    color: ’white’,
    marginBottom: 20,
    textAlign: ’center’
  },
  whiteBox: {
    width: ’100%’,
    height: ’55%’,
    backgroundColor: ’white’,
    borderRadius: 5,
    justifyContent: ’center’,
    marginBottom: 10,
    borderColor: ’#ddd’,
    borderWidth: 10,
  },
  whiteBoxText: {
    fontSize: 60,
    textAlign: ’center’,
    fontWeight: ’bold’
  }
});
```

****

Naughts and Crosses UI

```
import React from ’react’;
import { StyleSheet, View, Text, Dimensions, SafeAreaView } from ’react-native’;

const { width } = Dimensions.get(’window’);

const Box = ({ value, borders }) => {
  return (
    <View style={[styles.box, borders]}>
      <Text style={styles.boxText}>{value}</Text>
    </View>
  );
};

const App = () => {
  return (
    <SafeAreaView style={styles.container}>
      <View style={styles.container}>
        <Text style={styles.title}>Naughts and Crosses</Text>
        <View style={styles.row}>
          <Box value="O" borders={styles.topLeftBox} />
          <Box value="X" borders={styles.topBox} />
          <Box value="O" borders={styles.topRightBox} />
        </View>
        <View style={styles.row}>
          <Box value="X" borders={styles.leftBox} />
          <Box value="O" borders={styles.centerBox} />
          <Box value="X" borders={styles.rightBox} />
        </View>
        <View style={styles.row}>
          <Box value="O" borders={styles.bottomLeftBox} />
          <Box value="X" borders={styles.bottomBox} />
          <Box value="O" borders={styles.bottomRightBox} />
        </View>
      </View>
    </SafeAreaView>
  );
};

const boxSize = width / 3 - 20;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: ’center’,
    alignItems: ’center’,
    backgroundColor: ’white’,
  },
  title: {
    fontSize: width * 0.08,
    fontWeight: ’bold’,
    marginBottom: 50,
  },
  row: {
    flexDirection: ’row’,
  },
  box: {
    width: boxSize,
    height: boxSize,
    justifyContent: ’center’,
    alignItems: ’center’,
    backgroundColor: ’#fff’,
  },
  boxText: {
    fontSize: boxSize / 2,
    fontWeight: ’bold’,
  },
  topLeftBox: {
    borderBottomWidth: 2,
    borderRightWidth: 2,
    borderColor: ’#000’,
  },
  topBox: {
    borderBottomWidth: 2,
    borderLeftWidth: 2,
    borderRightWidth: 2,
    borderColor: ’#000’,
  },
  topRightBox: {
    borderBottomWidth: 2,
    borderLeftWidth: 2,
    borderColor: ’#000’,
  },
  leftBox: {
    borderRightWidth: 2,
    borderBottomWidth: 2,
    borderTopWidth: 2,
    borderColor: ’#000’,
  },
  centerBox: {
    borderWidth: 2,
    borderColor: ’#000’,
  },
  rightBox: {
    borderTopWidth: 2,
    borderLeftWidth: 2,
    borderBottomWidth: 2,
    borderColor: ’#000’,
  },
  bottomLeftBox: {
    borderTopWidth: 2,
    borderRightWidth: 2,
    borderColor: ’#000’,
  },
  bottomBox: {
    borderTopWidth: 2,
    borderRightWidth: 2,
    borderLeftWidth: 2,
    borderColor: ’#000’,
  },
  bottomRightBox: {
    borderTopWidth: 2,
    borderLeftWidth: 2,
    borderColor: ’#000’,
  },
});

export default App;


----------------------------------------------------------
**********************************************************
```

Calculator

```
----------------------------------------------------------
```

```
import { StatusBar } from ’expo-status-bar’;
import { useState } from ’react’;
import { StyleSheet, Text, View, Dimensions, SafeAreaView, TouchableOpacity } from ’react-native’;

const { width } = Dimensions.get(’window’);

export default function App() {

  const [answerValue, setAnswerValue] = useState(0);
  const [readyToReplace, setReadyToReplace] = useState(true);
  const [memoryValue, setMemoryValue] = useState(0);
  const [operatorValue, setOperatorValue] = useState(’’);

  const handleNumber = (number) => {
    if (readyToReplace) {
```

```
    setReadyToReplace(false);
    return number;
  } else {
    return parseFloat(‘${answerValue}${number}‘);
  }
};

const calculateEquals = () => {
  const previous = parseFloat(memoryValue);
  const current = parseFloat(answerValue);
  let result;

  switch (operatorValue) {
    case ’+’:
      result = previous + current;
      break;
    case ’-’:
      result = previous - current;
      break;
    case ’x’:
      result = previous * current;
      break;
    case ’/’:
      result = previous / current;
      break;
    default:
      return current;
  }

  return parseFloat(result.toFixed(4));
};

const buttonPressed = (value) => {
  console.log(‘Button pressed: ${value}‘);
  if(!isNaN(value)){
    const number = parseFloat(value);
    setAnswerValue(handleNumber(number));
  }
  else if (value === ’C’) {
    setAnswerValue(0);
    setMemoryValue(0);
    setOperatorValue(’’);
    setReadyToReplace(true);
  }
  else if (value === ’+/-’) {
    setAnswerValue((currentValue) => currentValue * -1);
  }
  else if ([’+’, ’-’, ’x’, ’/’].includes(value)) {
    let newMemoryValue = answerValue;
    if (operatorValue !== ’’) {
      newMemoryValue = calculateEquals();
    }
    setMemoryValue(newMemoryValue);
    setReadyToReplace(true);
    setOperatorValue(value);
  }
  else if (value === ’=’) {
    const result = calculateEquals();
    setMemoryValue(0);
    setReadyToReplace(true);
    setAnswerValue(result);
  }
  else {
    setAnswerValue(answerValue * 0.01);
  }
};

return (
  <SafeAreaView style={styles.container}>
    <View style={styles.container}>
      <View style={styles.resultBox}>
        <Text style={styles.result}>{answerValue}</Text>
      </View>

      <View style={styles.row}>
        <TouchableOpacity style={styles.greyButton} onPress={() => buttonPressed(’C’)}>
          <Text style={styles.blackButtonText}>C</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.greyButton} onPress={() => buttonPressed(’+/-’)}>
          <Text style={styles.blackButtonText}>+/-</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.greyButton} onPress={() => buttonPressed(’%’)}>
          <Text style={styles.blackButtonText}>%</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.blueButton} onPress={() => buttonPressed(’/’)}>
          <Text style={styles.whiteButtonText}>/</Text>
        </TouchableOpacity>
      </View>
      <View style={styles.row}>
        <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’7’)}>
          <Text style={styles.whiteButtonText}>7</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’8’)}>
          <Text style={styles.whiteButtonText}>8</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’9’)}>
          <Text style={styles.whiteButtonText}>9</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.blueButton} onPress={() => buttonPressed(’x’)}>
          <Text style={styles.whiteButtonText}>x</Text>
        </TouchableOpacity>
      </View>
      <View style={styles.row}>
        <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’4’)}>
          <Text style={styles.whiteButtonText}>4</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’5’)}>
          <Text style={styles.whiteButtonText}>5</Text>
        </TouchableOpacity>
        <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’6’)}>
```

```
            <Text style={styles.whiteButtonText}>6</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.blueButton} onPress={() => buttonPressed(’-’)}>
            <Text style={styles.whiteButtonText}>-</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.row}>
          <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’1’)}>
            <Text style={styles.whiteButtonText}>1</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’2’)}>
            <Text style={styles.whiteButtonText}>2</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’3’)}>
            <Text style={styles.whiteButtonText}>3</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.blueButton} onPress={() => buttonPressed(’+’)}>
            <Text style={styles.whiteButtonText}>+</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.row}>
          <TouchableOpacity style={styles.longButton} onPress={() => buttonPressed(’0’)}>
            <Text style={styles.longButtonText}>0</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.darkGreyButton} onPress={() => buttonPressed(’.’)}>
            <Text style={styles.whiteButtonText}>.</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.blueButton} onPress={() => buttonPressed(’=’)}>
            <Text style={styles.whiteButtonText}>=</Text>
          </TouchableOpacity>
        </View>

        <StatusBar barStyle="light-content" />
      </View>
    </SafeAreaView>
  );
}

const buttonSize = width / 5;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: ’black’,
    alignItems: ’center’,
    justifyContent: ’flex-end’,
  },
  resultBox: {
    width: width,
    marginRight: width * 0.25
  },
  result: {
    color: ’white’,
    fontSize: width * 0.15,
    textAlign: ’right’,
  },
  row: {
    flexDirection: ’row’,
    justifyContent: ’space-between’,
  },
  greyButton: {
    backgroundColor: ’#a6a6a6’,
    margin: width * 0.01,
    height: buttonSize,
    width: buttonSize,
    borderRadius: width * 0.2,
    justifyContent: ’center’,
  },
  blackButtonText: {
    color: ’black’,
    fontSize: width * 0.08,
    textAlign: ’center’,
  },
  whiteButtonText: {
    color: ’white’,
    fontSize: width * 0.08,
    textAlign: ’center’,
  },
  darkGreyButton: {
    backgroundColor: ’#333333’,
    margin: width * 0.01,
    height: buttonSize,
    width: buttonSize,
    borderRadius: width * 0.2,
    justifyContent: ’center’,
  },
  blueButton: {
    backgroundColor: ’#0984e3’,
    margin: width * 0.01,
    height: buttonSize,
    width: buttonSize,
    borderRadius: width * 0.2,
    justifyContent: ’center’,
  },
  longButton: {
    backgroundColor: ’#333333’,
    margin: width * 0.01,
    height: buttonSize,
    width: 2*buttonSize,
    borderRadius: width * 0.2,
    justifyContent: ’center’,
    textAlign: ’left’
  },
  longButtonText: {
    marginLeft: width * 0.07,
    color: ’white’,
    fontSize: width * 0.08,
    textAlign: ’left’,
  },
});
```

****

Food app

```
import { StatusBar } from ’expo-status-bar’;
import { ScrollView, StyleSheet, Text, View, Image, Dimensions } from ’react-native’;
import { NavigationContainer } from ’@react-navigation/native’;
import { createStackNavigator } from ’@react-navigation/stack’;
import { Cell, Section, TableView } from ’react-native-tableview-simple’;

const { width } = Dimensions.get(’window’);

const HomescreenCell = (props) => (
  <Cell
    {...props}
    backgroundColor="transparent"
    highlightColor="#f0f0f0"
    onPress={props.action}
    cellContentView={
      <View style={styles.restaurantCellContent}>
        <Image source={props.imgUri} style={styles.headerImage} />
        <View style={styles.etaContainer}>
          <Text style={styles.etaText}>{‘${props.eta}\n mins‘}</Text>
        </View>
        <Text style={styles.title}>{props.title}</Text>
        <Text style={styles.subtitle}>{props.tagline}</Text>
      </View>
    }
  />
);

function Restaurants({navigation}) {
  return (
    <ScrollView style={styles.scroll}>
      <TableView style={styles.table}>
        <Section name="" hideSeparator="true" separatorTintColor="#ccc">
          <HomescreenCell
              title="French Montana"
              tagline="Desert, Ice cream, £££"
              eta="10-30"
              imgUri={require(’./images/french-restaurant.jpg’)}
              action={() => navigation.navigate(’Menu’, {
                items: [
                  {
                    title: "Gelato",
                    contents: [{ title: "Vanilla", quantity: 0 },
                               { title: "Chocolate", quantity: 10 },
                               { title: "Strawberry", quantity: 20 }
                              ]
                  },
      {
        title: "Coffee",
        contents: [{ title: "Flat white", quantity: 23 },
                   { title: "Latte", quantity: 12 },
                   { title: "Caffe Americano", quantity: 56 }
                  ]
      },
      {
        title: "Tea",
        contents: [
          { title: "Green Tea", quantity: 32 },
          { title: "Black Tea", quantity: 0 },
          { title: "Chamomile", quantity: 12 },
          { title: "Earl Grey", quantity: 0 },
          { title: "Matcha", quantity: 0 }
        ]
      },
      {
        title: "Smoothies",
        contents: [
          { title: "Mango Smoothie", quantity: 0 },
          { title: "Berry Blast", quantity: 0 },
          { title: "Green Detox", quantity: 42 },
          { title: "Tropical Paradise", quantity: 21 },
          { title: "Peanut Butter Banana", quantity: 32 }
        ]
      },
    ]
  })}
/>
<HomescreenCell
  title="Sushi Palace"
  tagline="Sushi, Japanese, ££££"
  eta="20-40"
  imgUri={require(’./images/lantern.jpg’)}
  action={() => navigation.navigate(’Menu’, {
    items: [
      {
        title: "Sushi Rolls",
        contents: [
          { title: "California Roll", quantity: 0 },
          { title: "Spicy Tuna Roll", quantity: 5 },
          { title: "Dragon Roll", quantity: 32 }
        ]
      },
      {
        title: "Sashimi",
        contents: [
          { title: "Salmon", quantity: 54 },
          { title: "Tuna", quantity: 0 },
          { title: "Yellowtail", quantity: 12 }
        ]
      }
    ]
  })}
```

```
            />
            <HomescreenCell
              title="Pasta Paradise"
              tagline="Italian, Pasta, ££"
              eta="15-35"
              imgUri={require(’./images/outdoor-dining.jpg’)}
              action={() => navigation.navigate(’Menu’, {
                items: [
                  {
                    title: "Pasta",
                    contents: [
                      { title: "Spaghetti Carbonara" },
                      { title: "Fettuccine Alfredo" },
                      { title: "Penne Arrabbiata" }
                    ]
                  },
                  {
                    title: "Salads",
                    contents: [
                      { title: "Caesar Salad" },
                      { title: "Caprese Salad" },
                      { title: "Greek Salad" }
                    ]
                  }
                ]
              })}
            />
        </Section>
      </TableView>
    </ScrollView>
  )
}

function Menu({ route }) {

  const { items } = route.params;

  return (
    <ScrollView style={styles.scroll}>
      <TableView style={styles.table}>
        {items.map((item, index) => (
          <Section key={index} header={item.title}>
            {item.contents.map((content, idx) => (
              <Cell
                key={idx}
                title={content.title}
                isDisabled={content.quantity === 0}
              />
            ))}
          </Section>
        ))}
      </TableView>
    </ScrollView>
  )
}
```

```
const Stack = createStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
        <Stack.Navigator screenOptions={{ headerTitleAlign: ’center’ }}>
          <Stack.Screen name="Restaurants" component={Restaurants} />
          <Stack.Screen name="Menu" component={Menu} />
        </Stack.Navigator>
    </NavigationContainer>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: ’white’,
    alignItems: ’center’,
    justifyContent: ’center’,
  },
  scroll: {
    height: ’100%’,
  },
  restaurantCell: {
    height: 290,
  },
  restaurantCellContent: {
    flex: 1,
    paddingBottom: 25
  },
  headerImage: {
    width: ’100%’,
    height: 200,
    resizeMode: ’cover’,
    borderRadius: 8,
  },
  etaContainer: {
    position: ’absolute’,
    bottom: 50,
    right: 10,
    width: width / 4,
    backgroundColor: ’white’,
    padding: 8,
    borderRadius: width,
  },
  etaText: {
    color: ’black’,
    fontSize: 16,
    fontWeight: ’bold’,
    textAlign: ’center’,
  },
  title: {
    fontSize: 20,
    fontWeight: ’bold’,
    marginTop: 10,
  },
  subtitle: {
    fontSize: 14,
    color: ’#666’,
    marginTop: 5,
  },
});
```

****