Mobil Teknolojileri

MMOT222 Bitirme Projesi



**Danışman**Nilgün İNCEREİS

NİSAN 2020

**FODDS CALCULATOR UYGULAMASI PROJE İLERLEME RAPORU**

GELİŞTİRME ORTAMI: React- Native

DİL: React- Native

**Hazırlayanlar:** Kerim Eraslan 18MY90317

**ÖZET**

**FOOD CALCULATOR**

**Danışman**Nilgün İNCEREİS

NİSAN 2020

**Bu proje ufak veya büyük çaplı şirketlerin yemekhanelerinde satmakta olduğu kahvaltı ürünlerinin kolayca hesaplanmasını sağlamaktadır.**

**Bu proje sayesinde çalışanlar daha basit bir şekilde hesap yapabilir ve yanlış hesap olasılığını %0 a kadar düşürebilmektedir.**

**Proje reat-native dilinde yazılmıştır her iki platformda da çalışmaktadır.**(İOS-ANDROİD)

**İÇİNDEKİLER**

**1.Genel Bilgiler ……………………………………………………………**

1.1 React-Native Nedir………………………………………………….... 4

1.2 Expo Nedir …………………………………………………………… 5

1.3 Google Play Store Uygulama Paylaşımı……………………………… 6

1.4 İOS Uygulama Paylaşımı …………………………………………….. 6 - 7

**2.Proje Bilgileri…………………………………………………………….**

2.1 Amaç …………………………………………………………………. 8

2.2 Proje Ekran Görüntüleri………………………………………………. 9

2.3 Proje Kod Satırları…………………………………………………….. 10-20

**React -Native Nedir?**

React Native için kısaca Facebook tarafından üretilen, cross-platform mobil uygulama geliştirme olanağı sağlayan bir framework diyebiliriz.

**“Cross-platform mobil uygulama geliştirme framework’ü”** ne demek derseniz; **“Ben tek bir dil ile mobil uygulama geliştireyim; ama hem Android, hem iOS, hem Windows … vs ortamlarında çalışsın. Kalkıp da Android için Java, iOS için Swift öğrenmekle uğraşmayayım!”** dediğiniz de sizin imdadınıza koşan diller, framework’lerdir.

React Native’de bunlardan biri. Kendine özgü bir mobil uygulama uzantısı yok, uygulamayı geliştiriyorsunuz ve Android için **.apk**, iOS için **.ipa**uzantılarını alıyorsunuz. Android ortamına uygulama geliştirirken karşılaşacağınız hatalar Java hataları, iOS ortamına uygulama geliştirirken karşılaşacağınız hatalar Objective-C hataları. Ama kodlarken kullanacağınız dil React.

Kısa, öz ve açık bir giriş oldu ise asıl konumuza geçebiliriz.

Kendine has bir dil formatı (JSX) olan React Native bizlere tek bir dil üzerinden kodlama yapabilme ve geliştirilen uygulamanın bir çok platformda çalışma olanağını sunuyor dedik. Geliştiricilere bu desteği sağlayan React Native, cihaz ile arayüz arasında bir köprü görevi görerek geliştirilen mobil uygulamaların sorunsuzca çalışmasını sağlıyor. Siz geliştirme ortamında bir bileşen tanımlıyorsunuz (örneğin;<Text>) ve React Native ilgili ortamda o bileşenin karşılığı ne ise (Android’deTextView, iOS’daUIView) sizin yerinize oluşturuyor. Bunun için de yazılan React (yani Javascript)kodlarını native dile (Java, Objective-C .. vs) çevirirken “babel” adı verilen bir yapı devreye giriyor.



**EXPO Nedir ?**

[Exponent](https://expo.io/) ekibinin yaptığı react-native ile online kod yazmayı mümkün kılan [snack.expo projesi](https://snack.expo.io/), react-native yazmak için en hızlı yol.

* <https://snack.expo.io/> adresine girip hemen kodlamaya başlayabilirsiniz. Online emulator'de kodunuzun çıktısını görebilirsiniz.
* Proje oluşturup, lokal makinenize indirip, normal react-native projesi gibi projenize devam edebilirsiniz.
* Yazdığınız kodu telefonunuzda derlemek için;
  + IOS kullanıyorsanız, [AppStore](https://www.google.com.tr/search?q=itunes+expo+client&rls=com.microsoft:en-GB:%7breferrer:source?%7d&ie=UTF-8&oe=UTF-8&sourceid=ie7&rlz=1I7TEUA_trTR487TR488)'dan expo client'ı telefonunuza indirmelisiniz.
  + Android kullanıyorsanız, [GooglePlay'](https://play.google.com/store/apps/details?id=host.exp.exponent)den expo client'ı telefonunuza indirmelisiniz.
  + <https://snack.expo.io/> adresindeki projenizin QR code'unu expo client uygulamasında okutup, projenizi telefonunuzda derleyebilirsiniz.

Expo Projesi, ilk başlayanların react-native'e göz atması, hızlı bir şey denemek adına ya da kod paylaşmak için çok kullanışlı.



**Google Play Store ve App Store Platformları Uygulama Paylaşımı**

# Google Play Store

# Uygulama yükleme

Google Play geliştirici hesabına kaydolduktan sonra, Play Console hesabınızı kullanarak uygulamaları Google Play'e yükleyebilirsiniz.

**İpucu**: Uygulamanızı oluştururken Google Play'in [Android App Bundle](https://support.google.com/googleplay/android-developer/answer/9006925) adlı önerilen yayınlama biçimini kullanın.

## Uygulama oluşturma

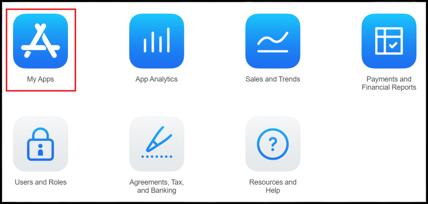
1. [Play Console](https://play.google.com/apps/publish/) hesabınıza gidin.
2. **Tüm uygulamalar** Tüm uygulamalar > Uygulama oluştur'u seçin.
3. Varsayılan dili seçin, uygulama için bir başlık ekleyin. Uygulamanızın adını Google Play'de görünmesini istediğiniz şekilde yazın.
4. Uygulamanızın [mağaza girişini](https://support.google.com/googleplay/android-developer/answer/113469#store_listing) oluşturun, [içerik derecelendirme anketini](https://support.google.com/googleplay/android-developer/answer/188189) tamamlayın, ardından [fiyatlandırma ve dağıtımı ayarlayın](https://support.google.com/googleplay/android-developer/answer/6334373).

**App Store**

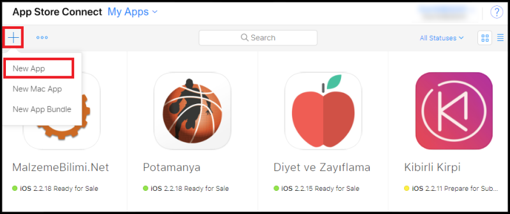
Bundle ID oluşturma işlemini tamamladıysanız yayınlama adımına çok yakınsınız demektir. Şimdi, gelin iOS uygulama Apple App Store’a nasıl yüklenir birlikte görelim…

Uygulamanızı Apple App Store’a yüklemek için [**https://developer.apple.com/account/**](https://developer.apple.com/account/) adresinden Apple ID’niz ile Geliştirici Hesabınıza giriş yapın.

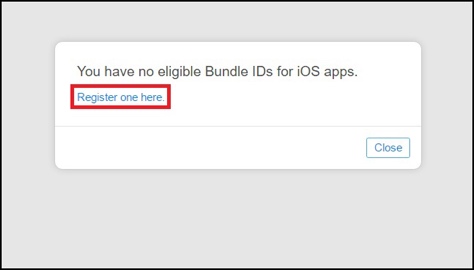
Şimdi karşınıza çıkan ekranda **My Apps** butonuna tıklayın.

[](https://support.mobiroller.com/wp-content/uploads/2016/09/App-Store-Connect-My-Apps-framed-1.png)

4. Yeni bir uygulama eklemek için ekranın sol üst tarafta bulunan **(+)** butonuna basın ve **New App**‘e tıklayın.

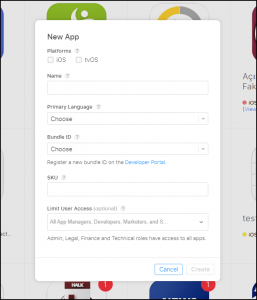
[](https://support.mobiroller.com/wp-content/uploads/2016/09/App-Store-My-New-App-framed-1-1.png)

5. Eğer Bundle ID’niz yok ise Bundle ID’niz olmadığına dair bir uyarı ekranıyla karşılaşacaksınız.

[](https://support.mobiroller.com/wp-content/uploads/2016/09/bundle1-framed-1.jpg)

Bu ekranda**Register one here** alanına tıklayarak yeni bir Bundle ID oluşturabilirsiniz.

6. Bundle ID’niz var ise aşağıdaki ekran üzerinden uygulama bilgilerini girmeniz istenecek.

[](https://support.mobiroller.com/wp-content/uploads/2016/09/New-App-framed.png)

Buradaki sihirbaz aracılığıyla uygulamanız hakkında şu bilgileri girmelisiniz:

* **Name:** Uygulama adı
* **Platforms:** Uygulamanın bulunacağı platform (MobiRoller’da tvOS desteği olmadığı için **sadece iOS seçimi** yapmalısınız.)
* **Primary Language :** Uygulama dili
* **Bundle ID:** Paket adı
* **SKU Number:** Uygulamanın sizin için oluşturacağı kod **(Tarih + Uygulama adı)**

İstenilen alanları doldurduktan sonra **Create** diyerek devam edin.

7. Yeni gelen **App Information** ekranında, sadece **Category** bölümünü doldurun ve sağ üstte bulunan **Save** butonuna tıklayın. Böylece bir sonraki **Pricing and Availability** ekranına geçiş yapabilirsiniz.

8. Pricing ve Availability ekranında **Price Schedule** bölümüne gidin ve **Choose** butonuna tıklayın. Bu bölümden uygulamanızın ücretini belirleyebiliyorsunuz.

Eğer uygulamanızı ücretsiz yayınlamak istiyorsanız ücreti “0” olarak seçmeniz gerekiyor. İstenilen verileri işaretledikten sonra **Save**‘e tıklayıp **Prepare for Submission**ekranına geçiş yapabilirsiniz.

9. Bu bölümde uygulamanın ekran görüntülerini 5,5 inç (1242×2208 piksel) ve 6,5 inç (1242×2688 piksel) JPEG formatında ekleyin. Bu iki alanada  ekleme yapmanız zorunludur.

Eklediğiniz resimlerden en az 2 tanesi, uygulamanızın içeriğini yansıtmalı.

10. Ekranı biraz aşağı doğru indirdiğinizde, uygulamanın açıklama ve anahtar kelimelerini girdiğiniz bölüme ulaşıyorsunuz.

* **Description:** Uygulamanın Apple App Store’daki açıklama bölümüdür.
* **Keywords:** Uygulamanın anahtar kelimelerini girdiğiniz bölümdür.
* **Support URL:** Uygulama ile ilgili web sitesi, destek veya iletişim bölümünü yazdığınız bölümdür.

Bu alanlar boş bırakılmamalı.

11. Aynı ekranın devamında uygulamanın simgesi, lisans anlaşması ve ticari hakların temsilinin kimde olduğuyla ilgili bilgiler giriliyor.

* **App Icon:** Uygulamanın markette gözükecek olan ikonunu girdiğiniz bölümdür. Uygulama ikonunun, 1024×1024 piksel ve JPEG formatında olması gerekli.
* **Copyright:** Uygulamanın telif hakkını gösterdiğiniz bölümdür.

Sağ tarafta bulunan iletişim bilgileri, boş bırakılmamalı.

12. Bu sayfanın son kısmında Apple’ın sizinle iletişime geçerken kullanacağı kişisel bilgileri girmeniz gerekiyor. Ayrıca uygulamanızda veya sitenizde üyelik olan bir bölüm var ise Apple’ın uygulamayı inceleyebilmesi için **Demo Account** alanına bir demo hesabı tanımlamanız gerekli.

İstenilen tüm bu bilgileri doldurduktan sonra MobiRoller Ekibi, uygulamanın paket işlemlerini yapmaya başlar. Ancak paket işlemlerinin yapılabilmesi için logonuzda herhangi bir şeffaf bölüm olmadığını kontrol etmeniz gerekir. Aksi halde uygulamanızın IPA’sı üretilemez.

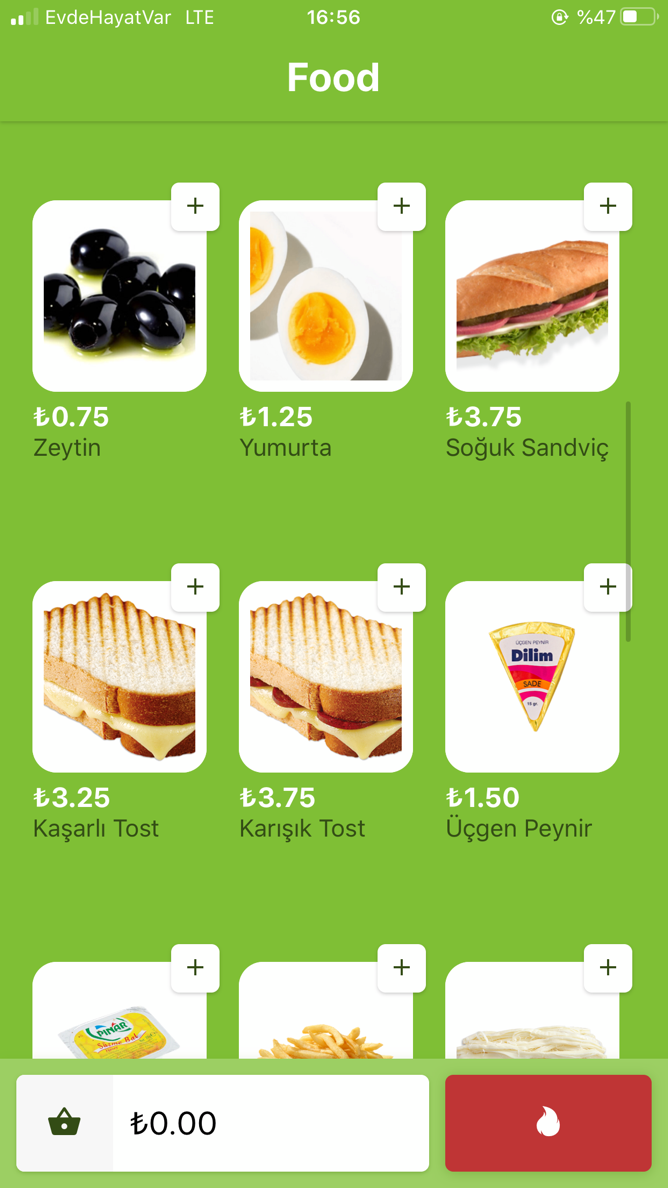
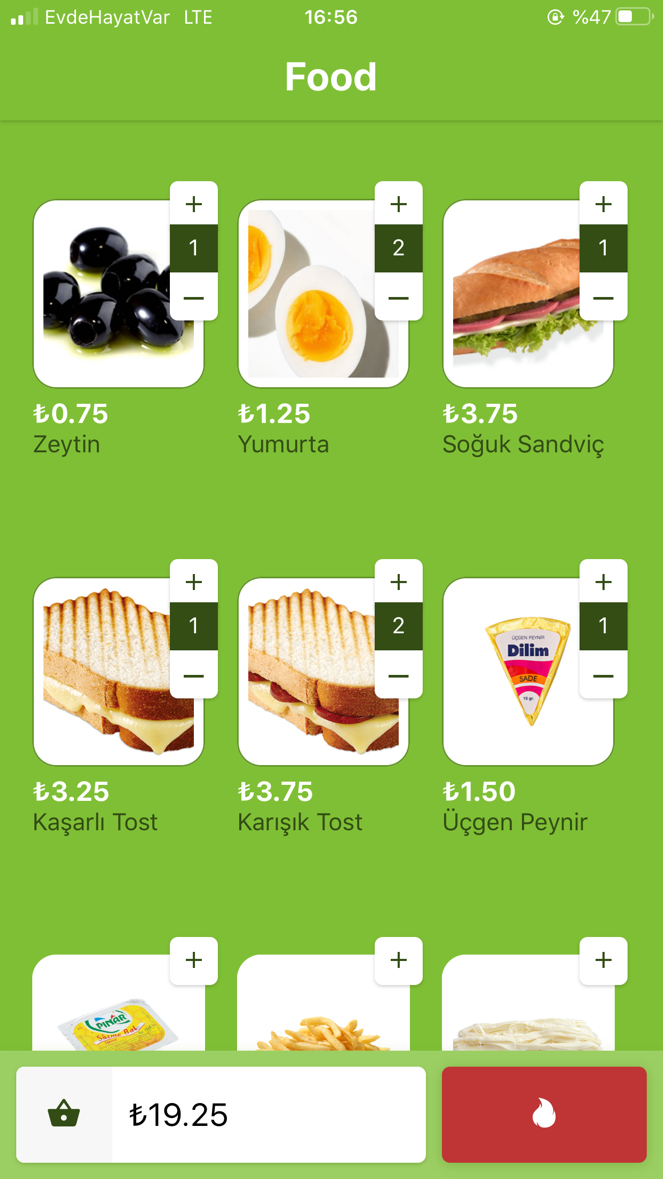
**PROJE AMACI**

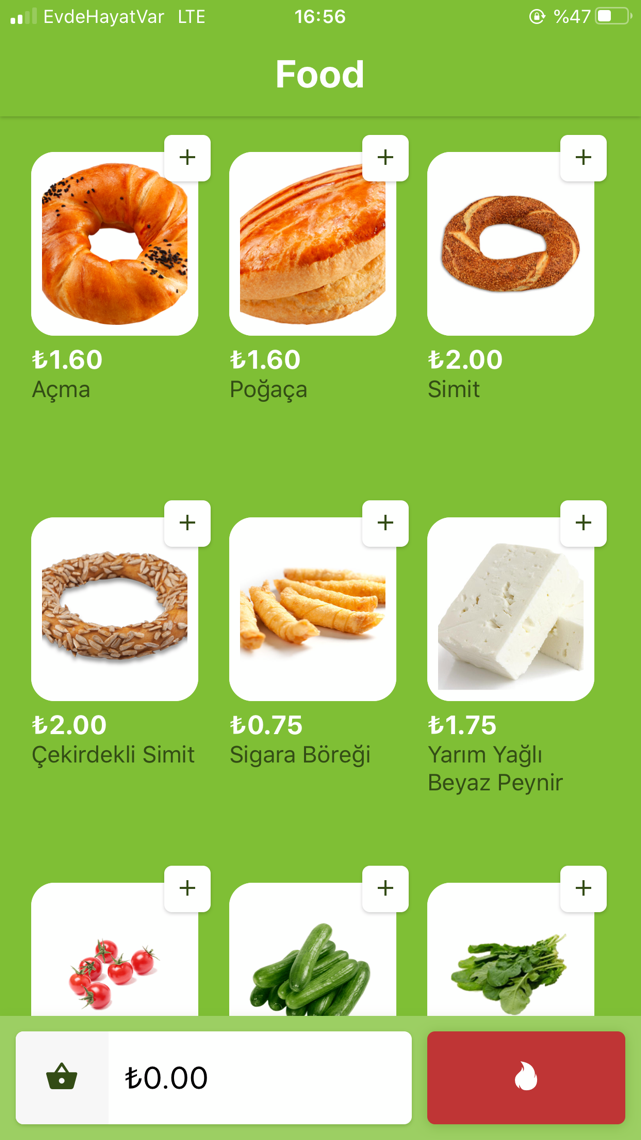
Bu proje ufak veya büyük çaplı şirketlerin yemekhanelerinde satmakta olduğu kahvaltı ürünlerinin kolayca hesaplanmasını sağlamaktadır.

Bu proje sayesinde çalışanlar daha basit bir şekilde hesap yapabilir ve yanlış hesap olasılığını %0 a kadar düşürebilmektedir.

Proje reat-native dilinde yazılmıştır her iki platformda da çalışmaktadır.(İOS-ANDROİD)

**PROJE EKRAN GÖRÜNTÜLERİ**

****

****

**PROJE KOD**

**APP.JS**

import React, { useState, Component } from 'react';

import { View, StyleSheet } from 'react-native'

import store from './src/store';

import { Provider } from 'react-redux';

import Constants from 'expo-constants';

import Header from './src/components/header';

import List from './src/components/list';

import Calculator from './src/components/calculator';

import {data} from './src/components/products';

// const DATA = [

// {

// id: 1,

// title: "Simit",

// price: 2,

// img: require("./src/images/simit.jpg")

// },

// {

// id: 2,

// title: "Ezine Peyniri",

// price: 1.75,

// img: { uri: "https://reimg-carrefour.mncdn.com/mnresize/600/600/productimage/30084518/30084518\_0\_MC/8796845670450\_1528879596665.jpg" }

// }

// ]

export default class App extends Component {

render() {

return (

<Provider store={store}>

<View style={styles.container} >

<Header />

<List

data={data}

/>

<Calculator/>

</View>

</Provider>

)

};

}

const styles = StyleSheet.create({

container: {

flex: 1,

backgroundColor: "#7fbf35" })

**CALCULATOR.JS**

import React from 'react';

import { StyleSheet, Text, View, Dimensions, Touchab leOpacity } from 'react-native';

import { Ionicons } from '@expo/vector-icons';

import { connect } from 'react-redux'

import \* as Actions from '../store/actions/counterActions';

const WIDTH = Dimensions.get("window").width;

function Calculator(props) {

return (

<View style={styles.container}>

<View style={styles.price}>

<View style={styles.iconView}>

<Text style={styles.iconText}>

<Ionicons name="md-basket" size={23} color={"#334c15"} />

</Text>

</View>

<Text style={styles.priceText}>

₺{props.price.toFixed(2)}

</Text>

</View>

<TouchableOpacity style={styles.button}

onPress={() => {

props.reset\_price()

props.remove\_cart\_lenght()

}}

>

<Text style={styles.buttonText}>

<Ionicons name="md-flame" size={23} color={"#fff"} />

</Text>

</TouchableOpacity>

</View>

);

}

const styles = StyleSheet.create({

container: {

flexDirection: "row",

height: WIDTH >= 400 ? 80 : 65,

backgroundColor: "#9ccf63",

shadowColor: "#64962a",

shadowOffset: {

width: 0,

height: 1,

},

shadowOpacity: 0.15,

shadowRadius: 3.84,

elevation: 5,

position: "absolute",

bottom: 0,

width: WIDTH

},

button: {

flex: 1,

margin: 10,

backgroundColor: "#bf3535",

borderRadius: 5,

flexDirection: "row",

alignItems: "center",

marginLeft: 0,

shadowColor: "#64962a",

shadowOffset: {

width: 0,

height: 1,

},

shadowOpacity: 0.5,

shadowRadius: 3.84,

elevation: 5,

},

buttonText: {

flex: 1,

textAlign: "center",

},

price: {

flex: 2,

margin: 10,

backgroundColor: "white",

borderRadius: 5,

flexDirection: "row",

alignItems: "center",

shadowColor: "#64962a",

shadowOffset: {

width: 0,

height: 1,

},

shadowOpacity: 0.5,

shadowRadius: 1.2,

elevation: 5,

},

priceText: {

flex: 1,

fontSize: 20,

textAlign: "left",

marginLeft: 10

},

iconView: {

borderTopLeftRadius: 5,

borderBottomLeftRadius: 5,

backgroundColor: "#f7f7f7",

height: WIDTH >= 400 ? 60 : 40,

width: WIDTH >= 400 ? 60 : 40,

alignItems: "center",

flexDirection: "row",

// textAlign: "center"

},

iconText: {

textAlign: "center",

flex: 1

}

})

const mapStateToProps = (state) => {

return {

price: state.counter.price

}

}

const mapDispatchToProps = (dispatch) => {

return {

reset\_price: () => dispatch(Actions.reset\_price()),

remove\_cart\_lenght: () => dispatch(Actions.remove\_cart\_length())

}

}

export default connect(mapStateToProps, mapDispatchToProps)(Calculator);

**HEADER.JS**

import React from 'react';

import { StyleSheet, Text, View } from 'react-native';

import Constants from 'expo-constants';

const TOPHEIGHT = Constants.statusBarHeight;

export default function Header() {

return (

<View style={styles.container}>

<Text style={styles.logo}>Food</Text>

</View>

);

}

const styles = StyleSheet.create({

container: {

backgroundColor: '#7fbf35',

alignItems: 'center',

justifyContent: 'center',

height: 75,

paddingTop: TOPHEIGHT,

shadowColor: "#000",

shadowOffset: {

width: 0,

height: 1,

},

shadowOpacity: 0.18,

shadowRadius: 1.00,

zIndex: 2,

elevation: 1,

},

logo: {

color: "#fff",

fontSize: 25,

fontWeight: "bold"

}

});

**İTEM.JS**

import React, { useState, useEffect, Component } from 'react';

import { View, Text, StyleSheet, Dimensions, Image, TouchableOpacity } from 'react-native';

import { Ionicons } from '@expo/vector-icons';

import { connect } from 'react-redux';

import \* as Actions from '../store/actions/counterActions';

const WIDTH = Dimensions.get("window").width;

const isSmallDevice = WIDTH <= 414;

const noPadding = WIDTH - (isSmallDevice ? 90 : 100);

const numColumns = isSmallDevice ? 3 : 4;

const boxWidth = noPadding / numColumns

let itemId = null;

class Item extends Component {

constructor(props) {

super(props);

this.state = {

cart: this.props.cart

}

}

state = {

isSelected: false,

isClicked: false,

}

add\_cart = (product) => {

this.props.add\_cart(product);

this.props.add\_price(product.price);

}

remove\_cart = (product) => {

this.props.remove\_cart(product);

this.props.remove\_price(product.price);

}

render() {

const { item, adet } = this.props;

return (

<View style={styles.box}>

<TouchableOpacity style={[styles.addButton, { borderBottomLeftRadius: this.props.cart[adet].length >= 1 ? 0 : 5, borderBottomEndRadius: this.props.cart[adet].length >= 1 ? 0 : 5 }]}

onPress={() => this.add\_cart(item)}

>

<Text style={[styles.addIcon]}>

<Ionicons name="md-add" size={16} color={"#334c15"} />

</Text>

</TouchableOpacity>

{

this.props.cart[adet].length >= 1 ?

<View style={styles.productLength}>

<Text style={{ color: "#fff", fontSize: 14, textAlign: "center" }}>{this.props.cart[adet].length}</Text>

</View> : null}

{

this.props.cart[adet].length >= 1 ?

<TouchableOpacity style={styles.removeButton}

onPress={() => this.remove\_cart(item)}

>

<Text style={styles.addIcon}>

<Ionicons name="md-remove" size={20} color={"#334c15"} />

</Text>

</TouchableOpacity> : null

}

<View style={[styles.boxTop, { borderColor: this.props.cart[adet].length >= 1 ? "#64962a" : "#fff" }]}>

<Image

style={{ width: "100%", height: "100%", position: "relative" }}

source={item.img}

/>

</View>

<Text style={styles.boxPrice}>₺{item.price.toFixed(2)}</Text>

<Text style={styles.boxTitle}>{item.title}</Text>

</View >

);

}

}

const styles = StyleSheet.create({

box: {

width: boxWidth,

height: boxWidth \* 2,

marginRight: 20,

marginBottom: 10,

paddingTop: 13,

marginTop: 10,

},

boxTop: {

width: boxWidth,

height: boxWidth \* 1.1,

borderRadius: 15,

padding: 6,

backgroundColor: "#fff",

borderColor: "#64962a",

borderWidth: 1,

},

boxPrice: {

fontWeight: "bold",

color: "white",

fontSize: 17,

marginTop: 5

},

boxTitle: {

color: "#334d15",

fontSize: 15,

marginTop: 0

},

addButton: {

width: 30,

height: 30,

top: 2,

position: "absolute",

right: -8,

backgroundColor: "white",

paddingBottom: 6,

paddingTop: 6,

zIndex: 1,

borderRadius: 5,

shadowColor: "#000",

shadowOffset: {

width: 0,

height: 1,

},

shadowOpacity: 0.18,

shadowRadius: 1.00,

elevation: 1,

},

productLength: {

width: 30,

height: 30,

position: "absolute",

right: -8,

backgroundColor: "#334d15",

top: 29,

paddingBottom: 6,

paddingTop: 6,

zIndex: 1,

shadowColor: "#000",

shadowOffset: {

width: 0,

height: 1,

},

shadowOpacity: 0.18,

shadowRadius: 1.00,

elevation: 1,

},

addIcon: {

flex: 1,

textAlign: "center",

},

removeButton: {

width: 30,

height: 30,

top: 59,

position: "absolute",

right: -8,

backgroundColor: "white",

paddingBottom: 6,

paddingTop: 6,

zIndex: 1,

borderRadius: 5,

shadowColor: "#000",

shadowOffset: {

width: 0,

height: 1,

},

shadowOpacity: 0.18,

shadowRadius: 1.00,

elevation: 1,

borderTopLeftRadius: 0,

borderTopEndRadius: 0

},

})

const mapStateToProps = (state) => {

return {

price: state.counter.price,

cart: state.cart.cart,

selects: state.cart.selects

}

}

const mapDispatchToProps = (dispatch) => {

return {

add\_price: (price) => dispatch(Actions.add\_price(price)),

remove\_price: (price) => dispatch(Actions.remove\_price(price)),

add\_cart: (product) => dispatch(Actions.add\_cart(product)),

remove\_cart: (product) => dispatch(Actions.remove\_cart(product)),

}

}

export default connect(mapStateToProps, mapDispatchToProps)(Item);

**LİST.JS**

import React, { Component } from 'react';

import { StyleSheet, Text, View, FlatList, Dimensions, Button } from 'react-native';

import Item from '../components/item';

import { connect } from 'react-redux';

import \* as Actions from '../store/actions/counterActions';

// screen sizing

const { width, height } = Dimensions.get('window');

// orientation must fixed

const SCREEN\_WIDTH = width < height ? width : height;

const SCREEN\_HEIGHT = height;

// const SCREEN\_HEIGHT = width < height ? height : width;

const isSmallDevice = SCREEN\_WIDTH <= 414;

const numColumns = isSmallDevice ? 3 : 4;

class List extends Component {

\_renderItem(item, index) {

// console.log(this.props)

return (<Item

item={item}

adet={index}

/>

)

}

render() {

const { data } = this.props;

// console.klog(this.props.cart.cart[0])

return (

<View style={styles.container}>

<FlatList

data={data}

numColumns={numColumns}

renderItem={({ item, index }) => this.\_renderItem(item, index)}

keyExtractor={item => item.id}

/>

</View>

);

}

}

const styles = StyleSheet.create({

container: {

padding: 20,

paddingTop: 0,

paddingBottom: 30,

height: SCREEN\_HEIGHT - 110

},

list: {

alignItems: "baseline"

}

})

const mapStateToProps = (state) => {

return {

cart: state.cart.cart

}

}

export default connect(mapStateToProps)(List);

PRODUCT JS

export const data = [

{

id: 1,

title: "Açma",

price: 1.6,

img: require("../images/acma.png")

},

{

id: 2,

title: "Poğaça",

price: 1.6,

img: require("../images/pogaca.png")

},

{

id: 3,

title: "Simit",

price: 2,

img: require('../images/simit.jpg')

},

{

id: 4,

title: "Çekirdekli Simit",

price: 2,

img: require('../images/cekirdekliSimit.png')

},

{

id: 5,

title: "Sigara Böreği",

price: 0.75,

img: require('../images/sigaraBoregi.png')

},

{

id: 6,

title: 'Yarım Yağlı Beyaz Peynir',

price: 1.75,

img: require('../images/yarimYagli.jpg')

},

{

id: 7,

title: 'Çeri Domates',

price: 0.5,

img: require('../images/ceriDomates.png')

},

{

id: 8,

title: 'Salatalık',

price: 0.75,

img: require('../images/salatalik.png')

},

{

id: 9,

title: 'Roka',

price: 0.5,

img: require('../images/roka.png')

},

{

id: 10,

title: 'Zeytin',

price: 0.75,

img: require('../images/siyahZeytin.jpg')

},

{

id: 11,

title: 'Yumurta',

price: 1.25,

img: require('../images/yumurta.png')

},

{

id: 12,

title: 'Soğuk Sandviç',

price: 3.75,

img: require('../images/sogukSandvic.png')

},

{

id: 13,

title: 'Kaşarlı Tost',

price: 3.25,

img: require('../images/kasarliTost.png')

},

{

id: 14,

title: 'Karışık Tost',

price: 3.75,

img: require('../images/karisikTost.png')

},

{

id: 15,

title: 'Üçgen Peynir',

price: 1.5,

img: require('../images/ucgenPeynir.png')

},

{

id: 16,

title: 'Süzme Bal',

price: 1.1,

img: require('../images/suzmeBal.png')

},

{

id: 17,

title: 'Patates Kızartması',

price: 2,

img: require('../images/patates.png')

},

{

id: 18,

title: 'Çeçil Peyniri',

price: 1.75,

img: require('../images/cecilPeyniri.png')

},

{

id: 19,

title: 'Kavrulmuş Fındık',

price: 4,

img: require('../images/findik.png')

},

{

id: 20,

title: 'Gevrek',

price: 4,

img: require('../images/gevrek.png')

},

{

id: 21,

title: 'Menemen',

price: 4,

img: require('../images/menemen.jpg')

},

{

id: 22,

title: 'Örgü Peyniri',

price: 1.75,

img: require('../images/orguPeyniri.png')

},

{

id: 23,

title: 'Kaşar Peyniri',

price: 1.75,

img: require('../images/tazeKasar.jpg')

},

{

id: 24,

title: 'Pınar Beyaz',

price: 1.1,

img: require('../images/suzmePeynir.jpg')

},

{

id: 25,

title: 'Tam Yağlı Beyaz Peynir',

price: 2,

img: require('../images/tamYagliPeynir.png')

},

{

id: 26,

title: 'Tam Yağlı Süt',

price: 1.6,

img: require('../images/sut.png')

},

{

id: 27,

title: 'Yarım Yağlı Süt',

price: 1.6,

img: require('../images/sut.png')

},

{

id: 28,

title: 'Portakal Suyu',

price: 4.5,

img: require('../images/portakalSuyu.png')

},

{

id: 29,

title: 'Reçel',

price: 1.5,

img: require('../images/recel.jpg')

},

{

id: 30,

title: 'Çikolata',

price: 1.1,

img: require('../images/cikolata.png')

}

]