

**Yalova Üniversitesi**

**Mühendislik Fakültesi**

**Bilgisayar Mühendisliği Bölümü**

**Web Programlama Final Raporu**

**(okuoku.com)**

**Hazırlayan**

**Gökhan ŞİŞMAN**

**M. Bektaş ÇİMEN**

**Aralık-2019**

İçindekiler

[Önsöz-Giriş 2](#_Toc27632842)

[Kullanılan Teknolojiler 3](#_Toc27632843)

[Server-Side kısmı ( Express , MongoDB,Mongoose) 3](#_Toc27632844)

[Server Side ile birlikte güncellenen değişiklikler, API istekleri vb. 5](#_Toc27632845)

[Mobil Kısım ( Cordova ) 8](#_Toc27632846)

[PhoneGap 9](#_Toc27632847)

[Yararlanılan Kaynaklar 9](#_Toc27632848)

[Kullanılan Araçlar 9](#_Toc27632849)

[Dipnot 9](#_Toc27632850)

# Önsöz-Giriş

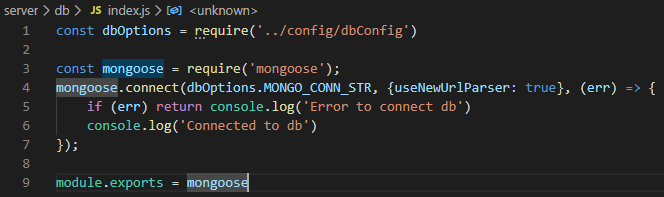
Öncelikle bu projede vueJS, vueCli , vueRouter vb. güncel teknolojileri ve ayrıca MongoDB, Cordova ve Express vb gibi teknolojileri kullanarak fullstack bir proje geliştirmenin mutluluğunu yaşıyoruz. Yaşanılan zorluklar , harcanılan büyük bir zamanla birlikte bu projeyi vaktinde projeyi teslim etmekteyiz. Teşekkürler.

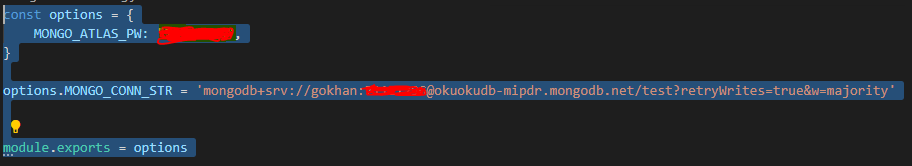
# Kullanılan Teknolojiler

VueJS , VueJSCLİ , MongoDB, Express, Mongoose, Cordova, JSON Server

# Server-Side kısmı ( Express , MongoDB,Mongoose)

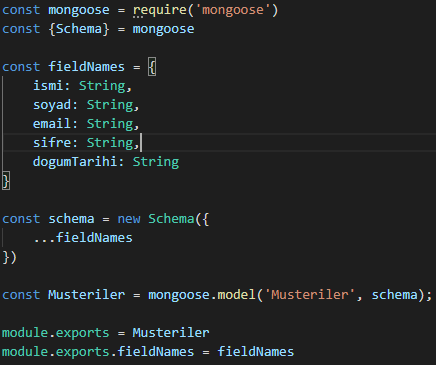
İlk önce <https://www.mongodb.com/cloud/atlas> adresinden kendimize bir hesap açtık ve buradan database oluşturduk. Daha sonra connection bağlantımızı aşağıdaki gibi gerçekleştirdik.



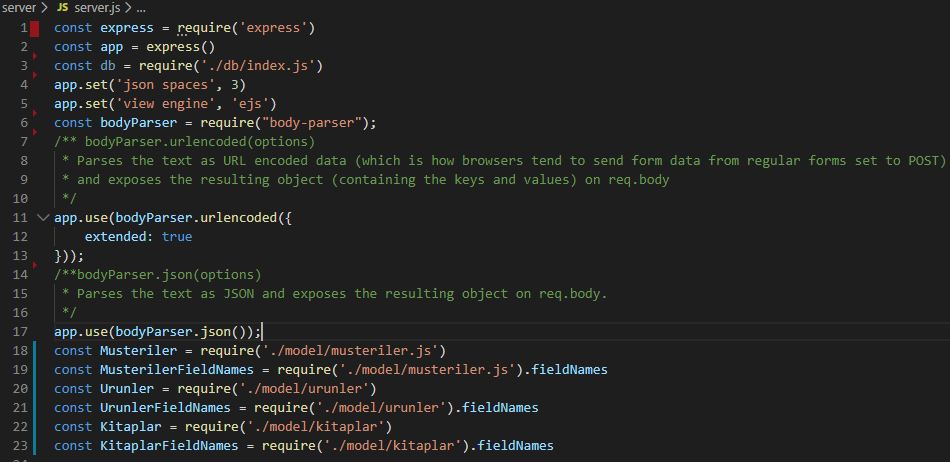


Bağlantımızı bu şekilde gerçekleştiriyoruz. İkinci resimde dbConfig dosyamız gözüküyor ve connection string görülüyor. Artık cloud üzerindeki veritabanımıza bağlandık.

Şimdi , Mongoose da kullanılan şema yapısını oluşturuyoruz. Şema aslında bir mongoose model olarak tanımlanıyor. FieldNames kısımları ise veritabanındaki sütun isimlerimiz. Şemayı , veritabanında ki bir tablo gibi düşünebilirsiniz.

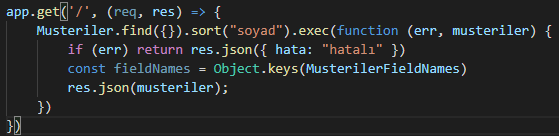


Şimdi sıra geldi , server.js dosyasını yazmaya. Artık yavaş yavaş sunucu tarafını oluşturmaya başlıyoruz.



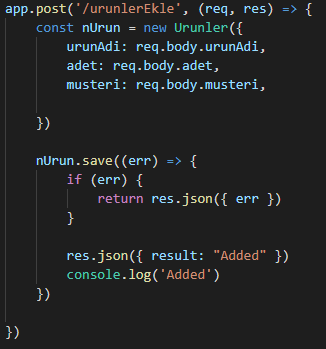
Gerekli olan, kullanılacak kütüphaneleri server.js içine ekliyoruz. Şekilde görüldüğü üzere oluşturduğumuz Müsteriler modelini ve diğer modelleri de buraya ekliyoruz. Artık istek yazmaya başlayabiliriz.

İlk örnek basitçe bir ‘get’ işlemi ;



Burada, Mongoose’un sağlamış olduğu find komutu ile müsteriler modelimizden yani tablomuzdan soyada göre sıralı bir şekilde json veri tipinde döndürüyoruz. Req – Request(istek) parametresi post işlemlerinde lazım olacak , res-Response (cevap) ise serverın döndürdüğü yanıt. Biz şuan veritabanından müsterileri çekmiş olduk.

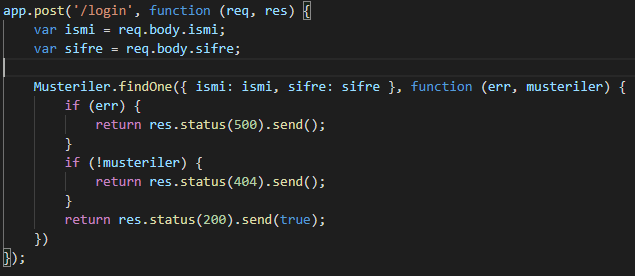
Bir de post işlemine bakalım ;



Yukarıda görüldüğü üzere bir Ürün modelimiz(tablomuz) var ve biz buna yeni bir ürün oluşturarak kayıt işlemi yapıyoruz. Görüldüğü gibi , req.body. kısmı bizim frontend tarafından gelen veya herhangi bir cihazdan gelen bir istek ve bunu tablo sütunlarımıza eşitliyoruz. Daha sonra save fonksiyonuyla Ürünü kaydediyoruz ve geriye ‘Added’ yanıtını döndürüyoruz. Eğer herhangi bir hata alırsakta err parametresi bunun ne olduğunu bize gösteriyor. Genel olarak server kısmımızda işlemlerimiz böyle.

# Server Side ile birlikte güncellenen değişiklikler, API istekleri vb.

Projemize eklediğimiz sunucu tarafı ile artık login işlemi yapılabiliyor. Basitçe kullanıcının adı ve sifresiyle giriş yapılabiliyor. Herhangi bir validation yok.

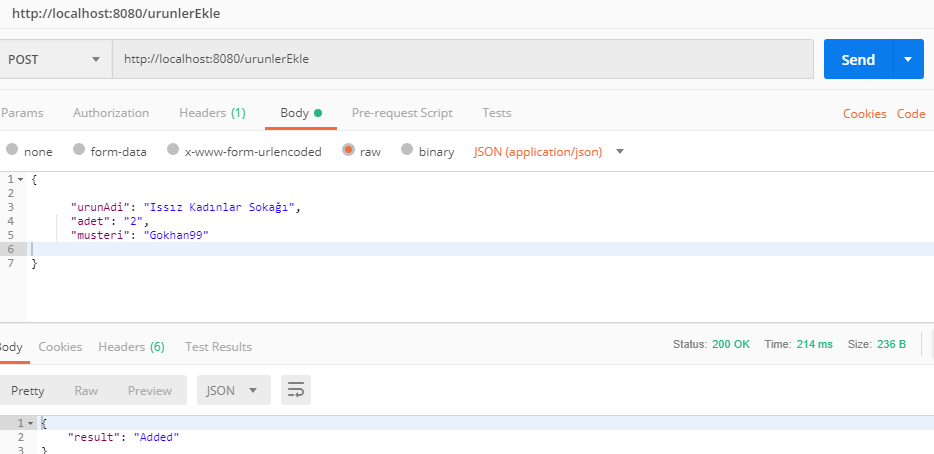


Sunucu tarafında, bize siteden gelen ismi ve sifreyi kontrol ediyor , burada findOne fonksiyonu kullanılıyor, böyle bir kullanıcı ve şifre var ise sunucu bize true değeri döndürüyor. LoginPage sayfasındaki gerekli inputlar v-model ile müsteri ismine ve sifresine bind ediliyor. Daha sonra bir method yazarak sunucu tarafına ön yüzden fetch komutu ile istek gerçekleştiriliyor.

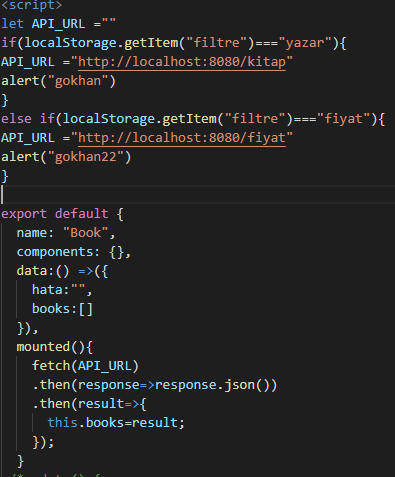


Burada bazı bilgiler localStorage kısmına kayıt ediliyor çünkü giriş yapıldıktan sonra bazı yerlerde kullanacağız. Ayrıca burada giris basarili olduktan sonra yönlendirme yapmak için iki farklı yöntem bulunuyor. Birincisi yorum satırı olan kısım , sizi istediğiniz yere yönlendiriyor diğeri ise klasik bir fonksiyon olan window.history.back() bir önceki sayfaya geri atıyor.

Diğer değişikliğe uğrayan kısım ise , vue-stepper ile oluşturulan sepet kısmı. Artık burada müşteri seçtiği ürünü satın alarak veritabanına kaydediliyor. Burada yine local storage kullanıyoruz , verileri veritabanına kaydetmeden önce geçiçi olarak tutuyoruz. Testler sunum sırasında yapılacaktır. Ama yine de aşağıdaki resimde postman ile yapılan bir satın alma işlemini görebilirsiniz.



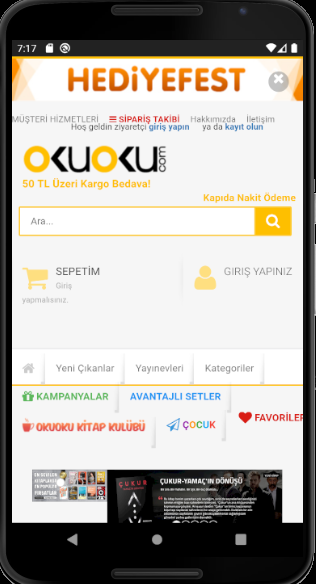
Diğer bir değişiklik ise , YeniCikanlar sayfasındaki kitapların veritabanından gelmesi ve bunlara birkaç filtre uygulanması olmuştur.



Öncelikle burası çok basit bir şekilde düşündüğümüz bir sistem. Yeni çıkanlar sayfasındaki filtreler local storage’ı triggerlayacak ve Book componenti ise serverdan istekleri ona göre değişecek. Burada /kitap isteği zaten yazarlara göre sıralanmıştır o yüzden öyle bıraktık. Daha birçok filtre eklenebilir. Burada mounted() fonksiyonu özel bir fonksiyondur. Component oluştuktan mount olduktan sonra çalışıyor ve bizde sunucudan veri çekiyoruz burada. Sunucu kısmı sunumda gösterilecektir ama basitçe Mongoose un verdiği kolaylıkla findOne fonksiyonu kullanıldı. Genel hatlarıyla sunucu kısmı bu şekilde gerekli olan bağlantılar , yapılar oluşturuldu daha geniş bir zamanla daha karmaşık yapılar kurulabilir ve istenilen her sonuca ulaşılabilir.

# Mobil Kısım ( Cordova )

Öncelikle , responsive olmadığı için mobil görünümde büyük bir problem yaşadık. Emülatör de bir nebze daha iyi sonuç aldık fakat PhoneGap ile yaptığımız uygulamanın kullanıcı dostu olmadığını biliyoruz. Cordovayı kendi sitesinden bakarak adım adım takip ederek kurduk.



Cordovayı npm run cordova-serve-android ile çalıştırabiliyoruz. npm run cordova-build-android dersek ise bize production buildini yapıyor yani apksını oluşturuyor. Açıkcası bu görüntüyü görene kadar bir çok hatayla karşılaştık. Cordova dosyalarını src-cordova adı altında www klasörü altında production kısmını tutuyor. Sunucu kısmıyla bağlantı kuramadık. Lakin bütün router fonksiyonları , aksiyonlar vb şeyler çalışıyor. Sunucu kısmının olmadığı sayfalarda problem var sadece.

# PhoneGap

Cordova’nın aslında önceki ismi. PhoneGap bize production’ı olan bir projenin html css mobil görünümü veriyor. Sunumda telefon üzerinden gösterilecektir.

# Yararlanılan Kaynaklar

<https://mongoosejs.com/>

<https://cordova.apache.org/>

<https://www.mongodb.com/cloud/atlas>

<https://build.phonegap.com/>

<https://vegibit.com/vue-js-express-tutorial/>

# Kullanılan Araçlar

* Visual Studio Code
* Github
* Android Studio
* AVD Manager
* Postman
* MongoDB-ATLAS-Cloud

# Dipnot

Karşılaşılan en büyük zorluk , sunucu tarafının mobil üzerinde çalışmamasıydı. Localhost ‘ u mobil cihaz hiçbir şekilde görmedi. Projeyi büyütmek istersek , bir sunucu satın alıp oradan API istekleri atılabilir.

Saygı ve Sevgilerle ,

Teşekkürler ☺