

---

# DOCUMENTATION

---

December 14, 2021

Tolga İnkaya

# Contents

0.1	Intro . . . . .	2
0.2	Tech Stack . . . . .	2
0.3	Data Model . . . . .	2
0.3.1	Database Relation . . . . .	3
0.4	System Design . . . . .	4
0.5	API - Postman . . . . .	5
0.5.1	Yeni bir sipariş Verilmesi (Create order) . . . . .	5
0.5.2	Siparişlerin listelenmesi (List order) . . . . .	6
0.5.3	Kullanıcının siparişin listelenmesi (List order <id>) . . . . .	6
0.5.4	Kuryenin Siparişi Alması (Pick Order) . . . . .	7
0.5.5	Siparişin Teslim Edilmesi (Order Delivered) . . . . .	7
0.6	. . . . .	7
0.7	Restoran sipariş istatistiği . . . . .	7

## 0.1 INTRO

Projede backend geliřtirmelerinin yanında kullanıcı ve restoranlar tarafından uygulamanın kullanılabilmesi için kullanıcı arayüzleri de bazı frontend componentleri kullanılarak gerçekleştirildi. Buradaki amaç uygulamanın ileride geliřtirilmeye devam edilmek istenmesi ve bu nedenle gerçek hayat senaryosu düşünülerek müşteri ve satıcı odaklı olarak kullanıcı dostu bir arayüz sağlanması. Bu arayüze arka tarafta API ile gerçekleşen HTTP sorguları sayesinde veri akışı sağlanmaktadır.

## 0.2 TECH STACK

- Python 3.9
- Django
- Django REST framework
- Postgresql
- SQLite
- Docker
- Redis
- Postman

## 0.3 DATA MODEL

Ařağıdaki kod bloğunda models.Model sınıfında inherit edilmiş ve uygulama design edilirken kullanılmış tüm modeller bulunmaktadır. Ayrıca model ilişkileri de *Database Model* figüründe gösterilmektedir.

```
1 class Restaurant(models.Model):
2     """Abstract Model"""
3
4 class Customer(models.Model):
5     """Abstract Model"""
6
7 class Driver(models.Model):
8     """Abstract Model"""
9
10 class Category(models.Model):
11     """Abstract Model"""
```

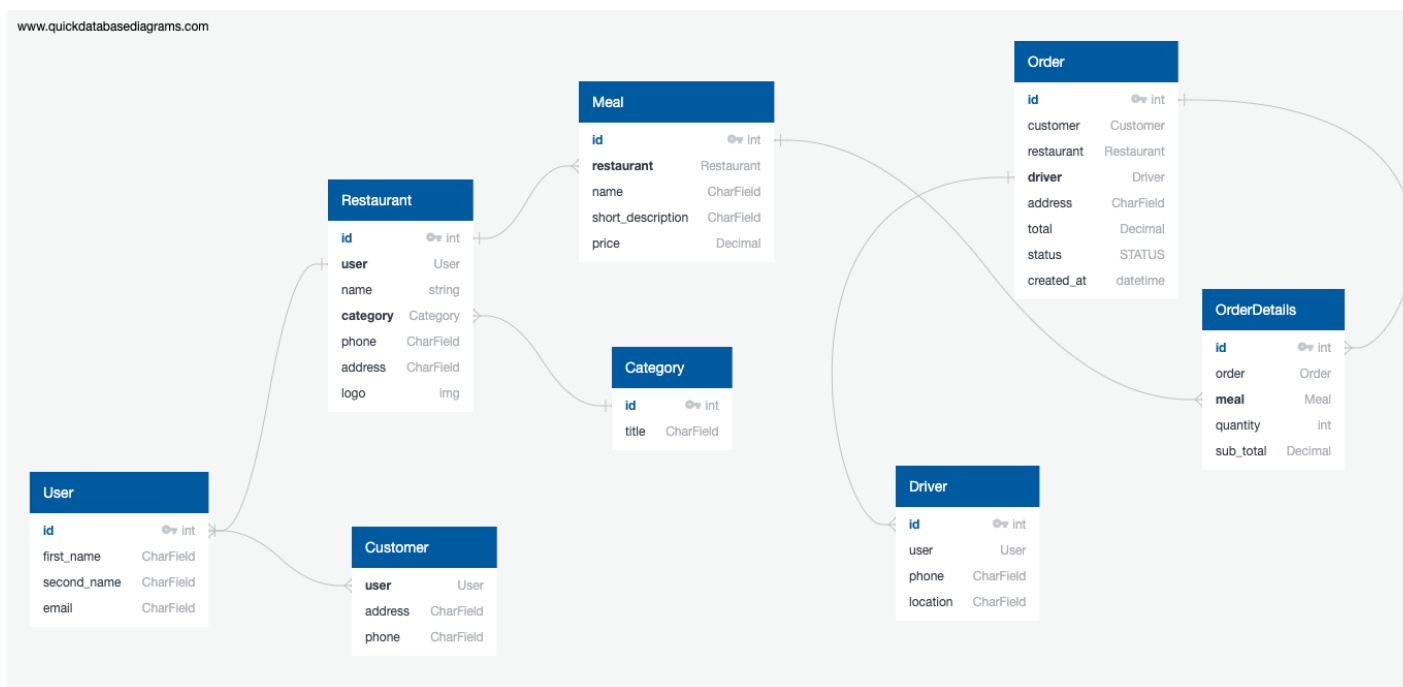
```

12
13
14 class Meal(models.Model):
15     """Abstract Model"""
16
17
18 class Order(models.Model):
19     """Abstract Model"""
20
21
22 class OrderDetails(models.Model):
23     """Abstract Model"""

```

**Listing 1: Models**

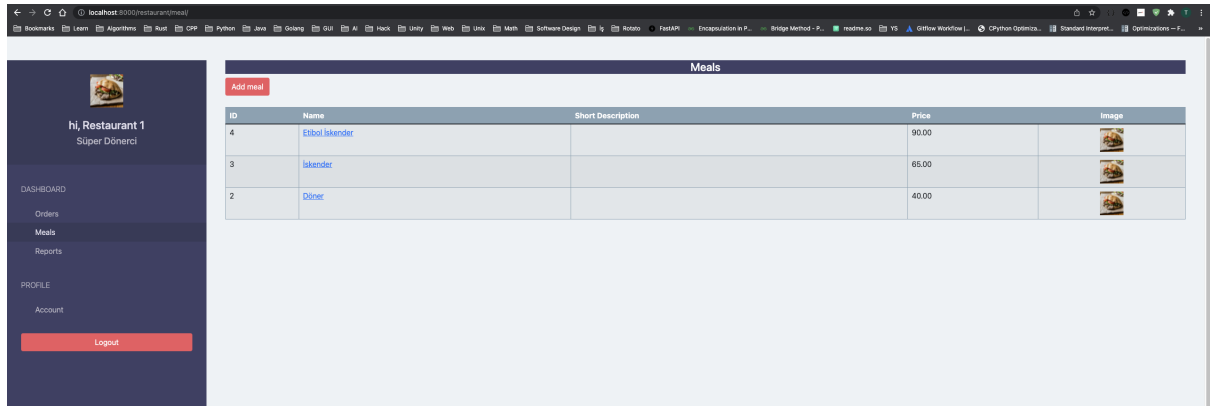
### 0.3.1 Database Relation



**Figure 1: Database Model**

## 0.4 SYSTEM DESIGN

- Customer ve Restaurant sahibi öncelikle bir User sınıfına bağlı olmalı dolayısıyla sign-up olmayan ve sign-in yapmayan kullanıcı sistemi kullanamaz.
- Bu işlemi admin paneli dışında UI üzerinden yapmak için geliştirme yapıldı.
- Sign-up olmak için endpoint *localhost:8000-up* ve Sign-in olmak için endpoint *localhost:8000-in*, aşağıda endpointlerin render olmuş halleri gösterilmektedir.
- Sign-in olmuş bir restoran sahibi, kendisi için tasarlanmış dashboard üzerinden sipariş takibi yapabilir, accountunu güncelleyebilir ve yiyeceklerini editleyebilir. Ayrıca report sekmesinden detaylı bir ürün satış istatistiğine ulaşabilir. Bu bahsedilen dashboard endpointleri ve render alınmış halleri aşağıdadır.
- Restoranın almış olduğu siparişler : *http://localhost:8000/restaurant/order/*
- Restoranın ürünleri : *http://localhost:8000/restaurant/meal/*
- Restoranın satış raporu : *http://localhost:8000/restaurant/report/*
- Restoranın hesap ayarları: *http://localhost:8000/restaurant/account/*
- Örnek dashboard ve restorana ait ürünler aşağıda gösterilmektedir.



**Figure 2:** Ürünler sayfası ve dashboard

- Eğer kayıtlı bir müşteri yeni bir sipariş oluşturursa bu restorana notifikasyon uyarısıyla dashboarda düşüyor ve restoran o ürün için statüyü güncelliyor. Eğer ürün hazırlanırsa bu sefer ürünün statüsünü READY olarak güncelledikten sonra kuryenin(Driver Model) gelmesini bekliyor. Kurye ürünü aldığı anda siparişin statüsü yine güncellenerek (On The Way) statüsüne geçiyor. Son olarak kurye ürünü teslim ettiğinde orderin statüsü delivered olmakta ve bu şekilde kurye de müsait oluyor ve queue data structure yapısında müsaitlik sırasına ekleniyor. Sistem bu şekilde çalışıyor. Bir sonraki kısımda API'den istekler gönderilecek ve bu aşamalar UI üzerinden de görülebilecek.

**RESTAURANT - SIGN UP**

Username

Password

First name

Last name

Email

Name

Phone

Address

Choose File No file chosen

**Sign Up**

[Already have an account?](#)

**Figure 3: Sign-up**

**RESTAURANT - SIGN IN**

Username

Password

**Sign In**

[Become a Restaurant](#)

**Figure 4: Sign-in**

## 0.5 API - POSTMAN

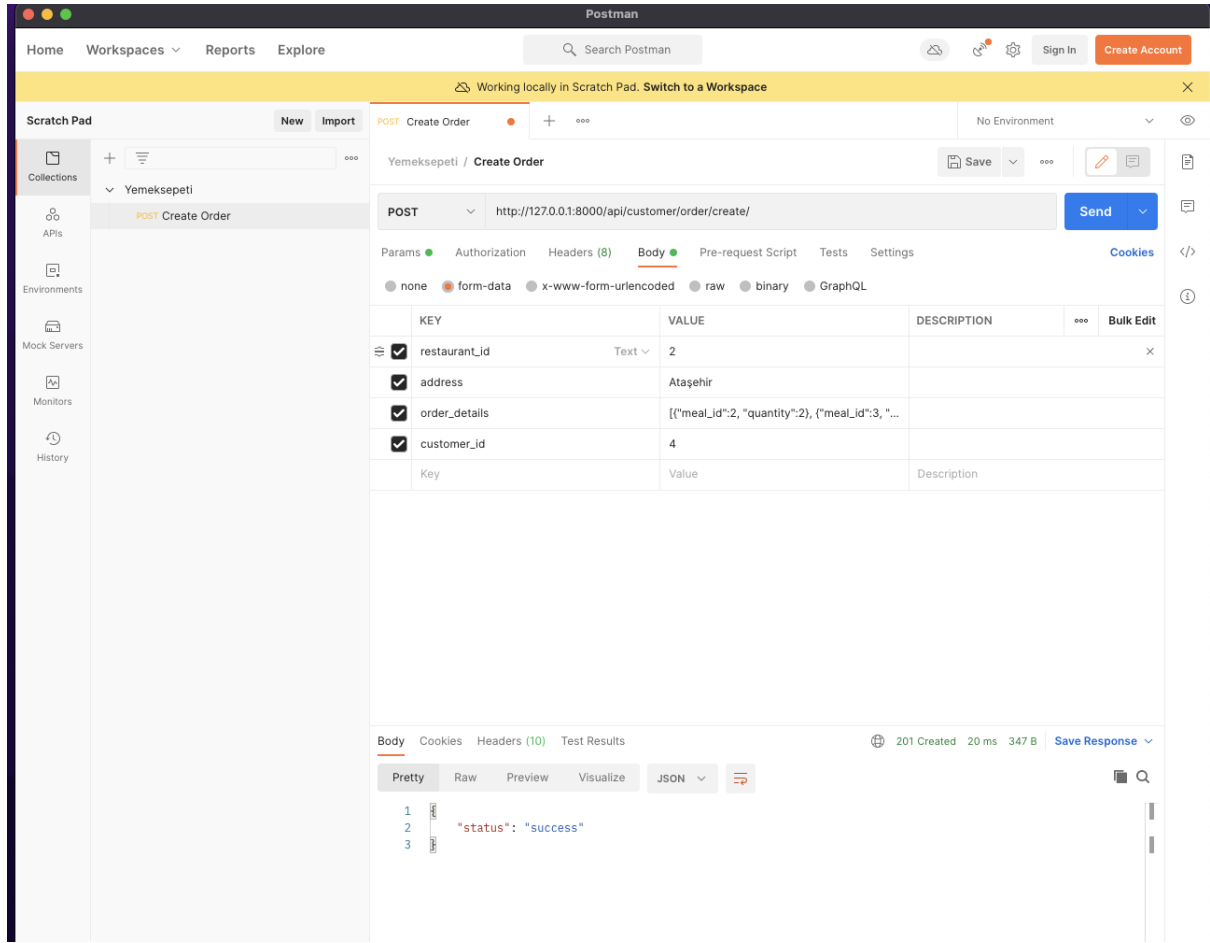
Örnek requestler ve API dökümantasyonu için Postman collection oluşturuldu (*Yemeksepeti.postman\_collection*).

### 0.5.1 Yeni bir sipariş Verilmesi (Create order)

— *ENDPOINT* —> *localhost:8000/api/customer/order/create/*

- *Postman'de orderın oluşturulması ve ardından UI'de orderın gösterilmesi aşağıdadır.*

— *Bu aşamada Restoran siparişini hazırlamaya başlar ve hazır olduğunda dashboardun en sağındaki statü kolonundaki READY butonuna tıklayarak kuryenin(DRIVER) gelmesini bekler(Figure 5-6).*



**Figure 5:** Postman Create Order

### 0.5.2 Siparişlerin listelenmesi (List order)

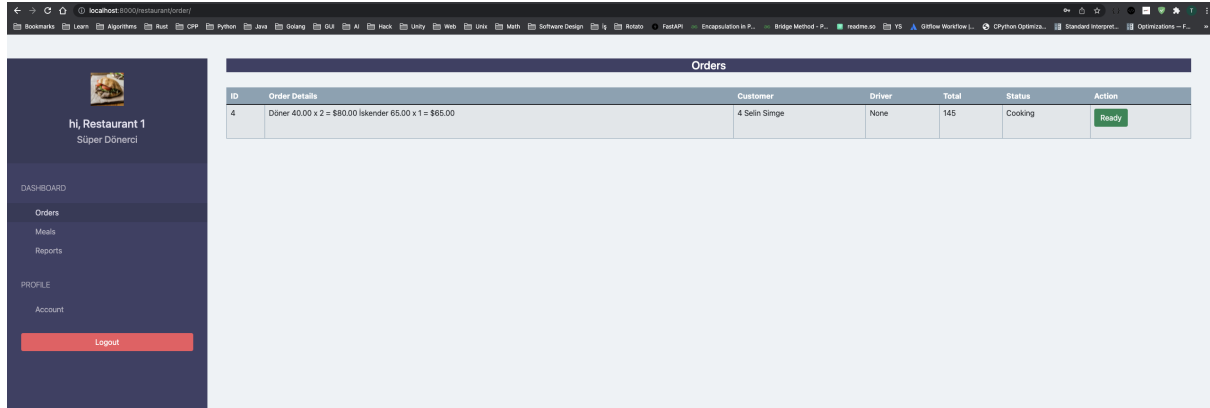
—- *ENDPOINT* —-> `localhost:8000/api/customer/order/list/`

Bu endpointten tüm verilmiş olan siparişlere ulaşabiliyoruz (Figure 7).

### 0.5.3 Kullanıcının siparişin listelenmesi (List order <id>)

—- *ENDPOINT* —-> `localhost:8000/api/customer/order/list/<int:id>`

Bu endpointten belirli bir id numarasına sahip kullanıcının siparişleri listelenir.



ID	Order Details	Customer	Driver	Total	Status	Action
4	Döner 40.00 x 2 = \$80.00 Iskender 65.00 x 1 = \$65.00	4 Selin Simge	None	145	Cooking	Ready

**Figure 6:** Restoran dashboardunda sipariş durumu

### 0.5.4 Kuryenin Siparişi Alması (Pick Order)

— *ENDPOINT* —> `localhost:8000/api/driver/order/pick/`

Bu endpointten kurye belirli bir id numarasına sahip siparişi alır ve sipariş statüsünü "ON THE WAY" yapar. (Figure 8).

### 0.5.5 Siparişin Teslim Edilmesi (Order Delivered)

— *ENDPOINT* —> `localhost:8000/api/driver/order/complete/`

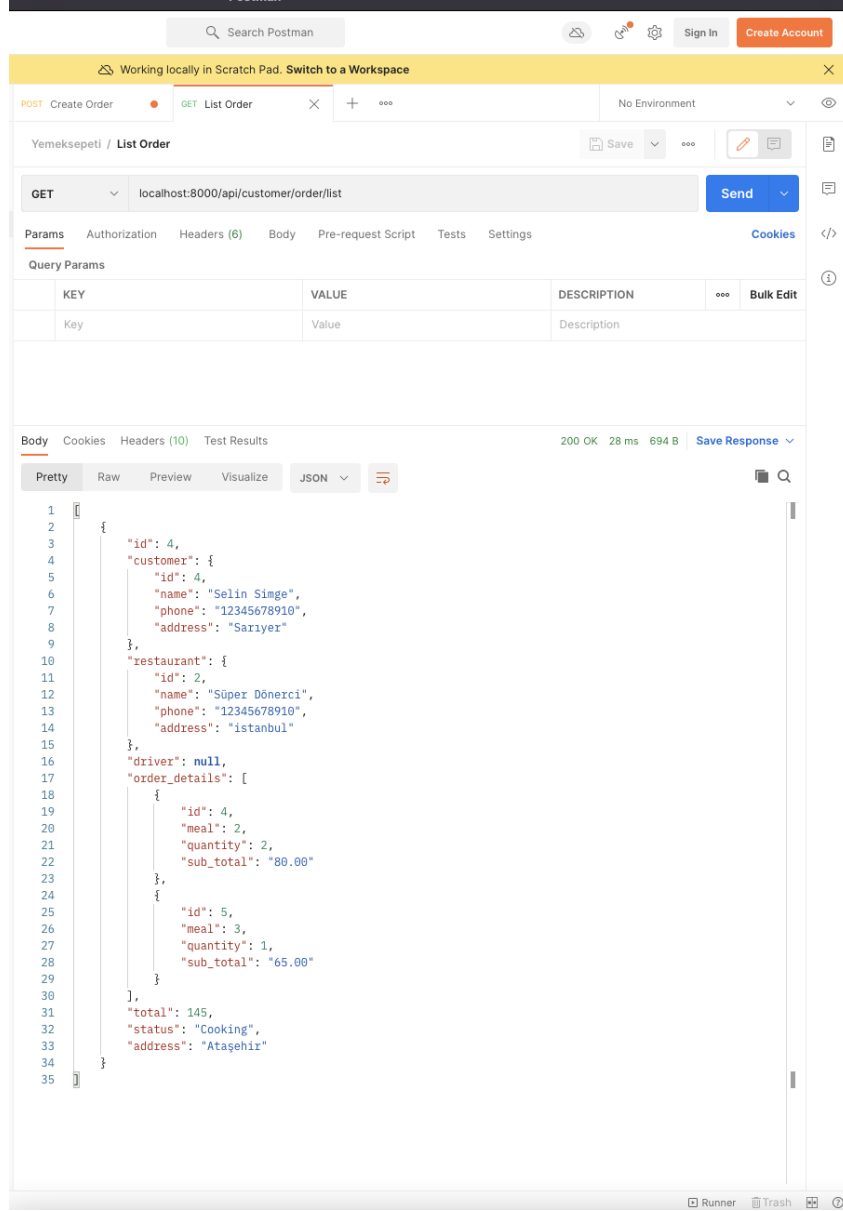
Bu endpointten belirli bir siparişin teslim edildiğine dair bilgi gönderilir ve restoranın dashboardunda sipariş statüsü Delivered olarak güncellenir. (Figure 9).

## 0.6

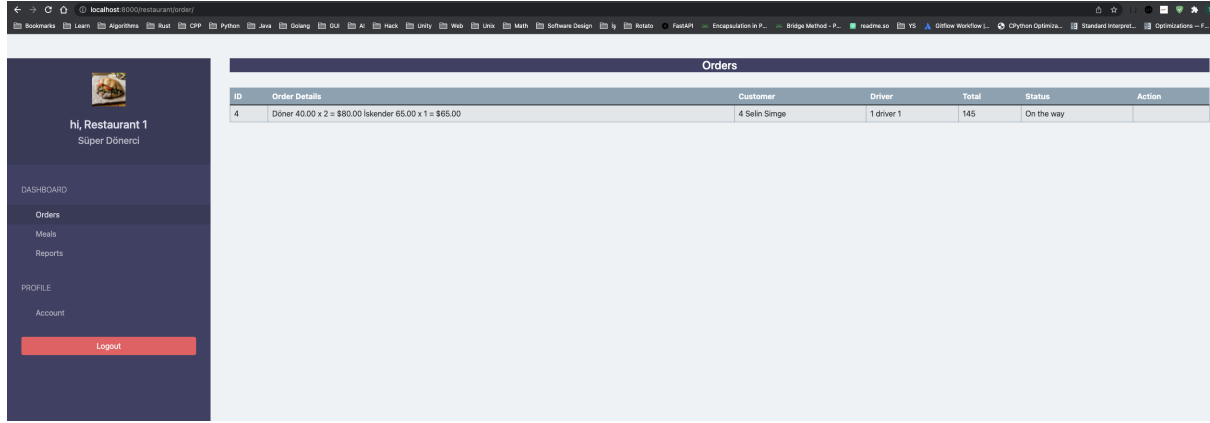
## 0.7 RESTORAN SİPARİŞ İSTATİSTİĞİ

— Restoran admini `localhost:8000/restaurant/report/` adresinden raporlara ulaşabilir. Örnek restoran raporu Figure 10'da gösterilmektedir.





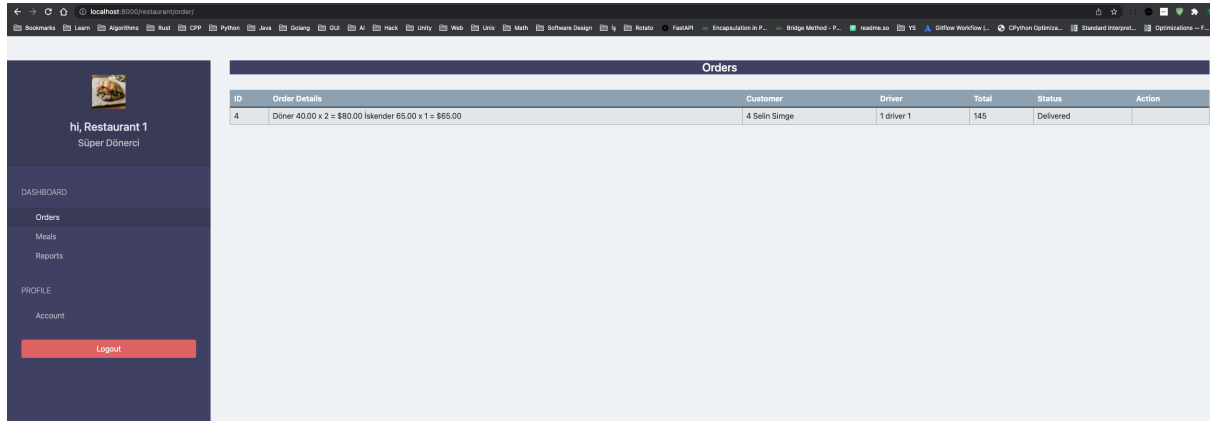
**Figure 7:** Tüm siparişler



The screenshot shows a web application interface for a restaurant named "hi, Restaurant 1 Süper Dönerci". The left sidebar contains a "DASHBOARD" section with links for "Orders", "Meals", and "Reports", and a "PROFILE" section with a link for "Account". A "Logout" button is at the bottom of the sidebar. The main content area is titled "Orders" and displays a table with one order. The order details are: ID 4, Order Details "Döner 40.00 x 2 = \$80.00 Iskender 65.00 x 1 = \$65.00", Customer "4 Selin Simge", Driver "1 driver 1", Total "145", and Status "On the way".

ID	Order Details	Customer	Driver	Total	Status	Action
4	Döner 40.00 x 2 = \$80.00 Iskender 65.00 x 1 = \$65.00	4 Selin Simge	1 driver 1	145	On the way	

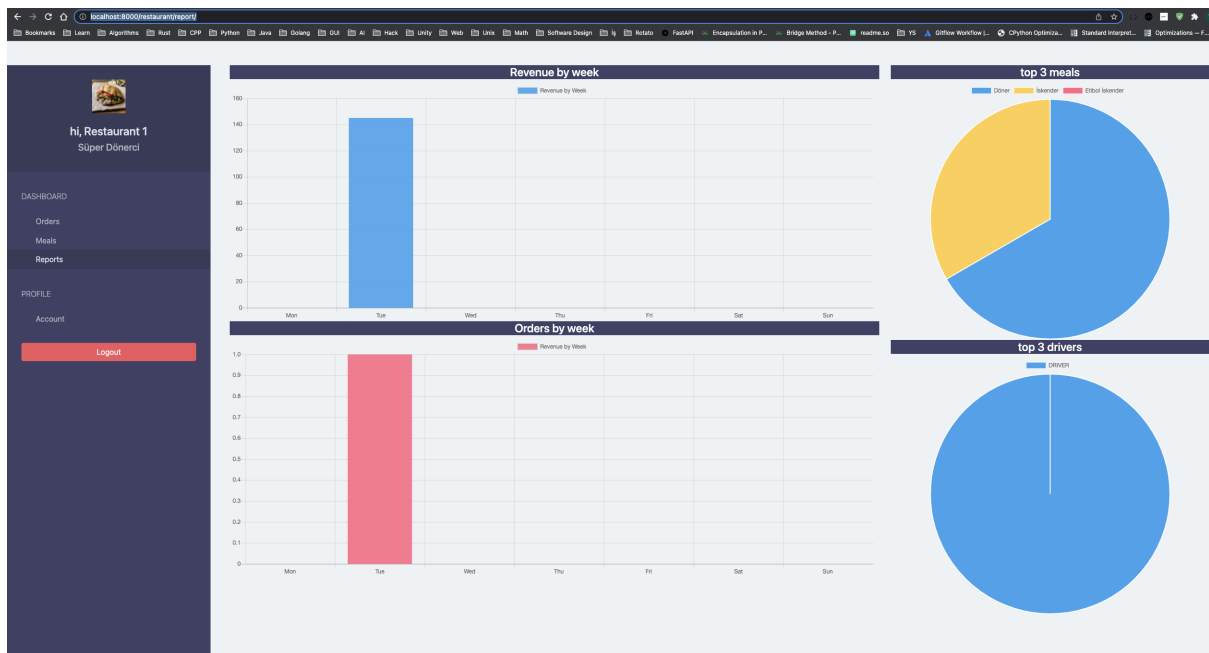
**Figure 8:** Sipariş yolda



The screenshot shows the same web application interface as Figure 8, but the status of the order has changed to "Delivered".

ID	Order Details	Customer	Driver	Total	Status	Action
4	Döner 40.00 x 2 = \$80.00 Iskender 65.00 x 1 = \$65.00	4 Selin Simge	1 driver 1	145	Delivered	

**Figure 9:** Sipariş teslim edildi



**Figure 10:** Restoran sipariş istatistiği