

**T.C DOĞUŞ ÜNİVERSİTESİ**  
**ENGINEERING FACULTY**  
**COMPUTER ENGINEERING**

# **MUNCH**

**COME 491 PROJECT**

**PREPARED BY**

**LEVENT EGE MUTLU**

**201631029**

**ADVISOR**

**Dr. Öğr. Üyesi Arif Murat Yağcı**

**İSTANBUL, FEBRUARY 2021**

## **PREFACE**

I would like to mention to be grateful to all developers who share their knowledge on internet for free and I also thank to my advisor.

İstanbul, February 2021

LEVENT EGE MUTLU

## **SUMMARY**

In summary, Munch is an application that customers and employees can use jointly for restaurants. The biggest reason for my motivation to do this application is tracking all daily routines online and taking forward it. At the same time, to provide an easier experience with online orders and payment for customers. Nowadays, almost everything is done from the applications on our phones, and most restaurants manage the order and stock process with the traditional method. For this purpose, this application enables online transactions such as stock tracking, order tracking, menu creation. With this application, you can order online inside or outside the restaurant. You can create a reservation on the date you specify. For the restaurant, this application, which provides a connection between the kitchen and the employee who takes the order, instantly displays the order on the admin screen, thus saving time. Order can also be viewed on this application when the order is ready. You can instantly change restaurant menus online and track stock after the order is served. The general idea that I aimed at trying to do this application was to make the overall process in restaurants done with less manpower.

Keywords: restaurant, customer, order, online,

## ÖZET

Özet olarak Munch, restoranlar için müşteri ve çalışanların ortak kullanabileceği bir uygulamadır. Bu uygulamayı yapma motivasyonumun en büyük nedeni restoranlardaki günlük rutinleri sanal ortamda ilerletmek ve takip etmek. Aynı zamanda müşteriler içinde online sipariş ve ödeme olacağından müşterilere de daha kolay bir deneyim sağlamak. Günümüzde hemen hemen her şey telefonumuzdaki uygulamalardan yapılıyor, çoğu restoran geleneksel yöntem ile sipariş ve stok süreci yönetiyor. Bunun için bu uygulama stok takibi, sipariş takibi, menü oluşturma gibi işlemleri online olarak yapmayı sağlıyor. Bu uygulama ile restoran içerisinde ya da dışarısında online sipariş verebilirsiniz. Belirlediğiniz tarihte rezervasyon oluşturabilirsiniz. Restoran içinse mutfak ve sipariş alan çalışan arasında bir bağlantı sağlayan bu uygulama, siparişi anında yönetici ekranında yönetici ekranında gösterir ve böylelikle zamandan kazanılmış olunur. Aynı zamanda sipariş hazır olunca bu uygulama üstünden görülebilir. Restoran menülerini anında online olarak değiştirebilir ve sipariş servis edildikten sonra stok takibi yapılabilir. Bu uygulamayı yapmaya çalışırken amaçladığım genel fikir, restoranlardaki genel sürecin daha az insan gücüyle yapılmasını sağlamaktır.

Anahtar Kelimeler: restoran, müşteri, sipariş, online, uygulama

## CONTENTS

PREFACE.....	i
SUMMARY.....	ii
ÖZET .....	iii
CONTENTS.....	iv
LIST OF FIGURES .....	ix
LIST OF TABLES.....	x
SYMBOLS.....	xi
ABBREVIATIONS .....	xii
REQUIREMENTS ANALYSIS DOCUMENT .....	1
1. INTRODUCTION .....	1
1.1. Purpose of the system.....	1
1.2. Scope of the system.....	1
1.3. Objectives and success criteria of the project .....	1
1.4. Definitions, acronyms, and abbreviations .....	1
1.5. References .....	1
1.6. Overview .....	1
2. CURRENT SYSTEM.....	2
3. PROPOSED SYSTEM .....	3
3.1. Overview .....	3
3.2. Functional requirements.....	3
3.3. Nonfunctional requirements.....	3
3.3.1. Usability.....	3
3.3.2. Reliability.....	3

3.3.3.	Performance .....	3
3.3.4.	Supportability.....	4
3.3.5.	Implementation .....	4
3.3.6.	Interface .....	4
3.3.7.	Packaging.....	4
3.3.8.	Legal .....	4
3.4.	System Models .....	4
3.4.1.	Scenarios .....	5
3.4.1.1.	S-01: Customer Register.....	5
3.4.1.2.	S -02: Customer Login .....	5
3.4.1.3.	S -03: Edit Customer Profile .....	6
3.4.1.4.	S -04: Change Customer Password .....	6
3.4.1.5.	S -05: Employer or Admin Login.....	7
3.4.1.6.	S -06: Employee Register .....	7
3.4.1.7.	S -07: Employee Login.....	8
3.4.1.8.	S -08: Admin Change User Role .....	8
3.4.1.9.	S -09: Add Menu .....	8
3.4.1.10.	S -10: Edit Menu .....	9
3.4.1.11.	S -11: Delete Menu.....	9
3.4.1.12.	S -12: Customer Add Items to Basket .....	9
3.4.1.13.	S -13: Customer Edit Items to Basket .....	10
3.4.1.14.	S -14: Customer Remove Items to Basket.....	10
3.4.1.15.	S -15: Complete Order.....	10
3.4.1.16.	S -16: Get Orders of Restaurant .....	10

3.4.1.17.	S -17: Change Order Status .....	11
3.4.1.18.	S -18: Change Order Payment Status .....	11
3.4.1.19.	S -19: Get Orders .....	11
3.4.1.20.	S -20: Get all orders (Current User) .....	12
3.4.1.21.	S -21: Online Payment.....	12
3.4.1.22.	S -22: Pre-order/Reservation .....	12
3.4.1.23.	S -23: Manage Inventory .....	13
3.4.1.24.	S -24: Change Inventory with Order .....	13
3.4.1.25.	S -25: Review for customers .....	13
3.4.1.26.	S -26: Review for restaurant.....	14
3.4.1.27.	S -27: Suggestion for Cart Items .....	14
3.4.1.28.	S -28: Top 5 Foods .....	14
3.4.1.29.	S -29: Get Food Statistics .....	15
3.4.1.30.	S -30: Search with text in Food .....	15
3.4.2.	Use Case Model .....	16
3.4.3.	Object Model .....	19
3.4.4.	Dynamic Model .....	20
3.4.5.	Database model — ER diagram.....	20
3.4.6.	User interface—navigational paths and screen mock-ups .....	21
4.	GLOSSARY .....	25
	PROJECT BUDGET AND TIME PLAN .....	26
5.	INTRODUCTION .....	26
5.1.	Gantt Chart .....	26
5.1.	Project Budget.....	27

5.2.	Project Phase Distribution.....	27
5.3.	Other Tables .....	27
	PROJECT SPECIFIC CONTENT.....	28
6.	INTRODUCTION .....	28
6.1.	ACCESSIBILITY .....	28
6.2.	USER MANUAL .....	28
6.2.1.	Landing Page .....	28
6.2.2.	Menu Page .....	28
6.2.3.	Reservation Page.....	30
6.2.4.	Contact Page .....	31
6.2.5.	Search Page.....	31
6.2.6.	Login Page .....	32
6.2.7.	Register Page .....	32
6.2.8.	Cart Page.....	33
6.2.9.	Order Page .....	33
6.2.10.	Admin Page.....	34
6.2.11.	Admin Customer Page .....	34
6.2.12.	Admin Profile Page .....	35
6.2.13.	Admin Category Page .....	36
6.2.14.	Admin Inventory Page .....	36
6.2.15.	Admin Contact Page .....	37
6.2.16.	Admin Reservation Page.....	38
6.2.17.	Admin Order Page.....	38
6.2.18.	Admin Foods Page .....	39



6.3. TEST PLAN.....	40
6.4. MAINTENANCE PLAN .....	40
CONCLUSION AND DISCUSSION .....	41
REFERENCES .....	42
EKLER.....	43
BIOGRAPHY .....	44

## LIST OF FIGURES

Figure 5.1 Gantt chart sample 1 .....	26
Figure 5.2 Gantt chart sample 2 made with excel.....	<b>Error! Bookmark not defined.</b>

## **LIST OF TABLES**

Table 5.1 Project budget table .....	27
Table 5.2 Project phase distribution .....	27

## **SYMBOLS**

$\beta$       Sample Test Phase

## **ABBREVIATIONS**

VS	Visual Studio
API	Application Programming Interface

## **REQUIREMENTS ANALYSIS DOCUMENT**

### **1. INTRODUCTION**

#### **1.1. Purpose of the system**

This project addresses the problem of restaurants. The restaurant owner will be able to control inventory, order using this application. Customers also will be able to order and pay online even if they are in the restaurant. The application's main purpose is to use fewer human resources to handle daily restaurant routines and customers no longer need to make an effort for the waiters to see them.

#### **1.2. Scope of the system**

A web application will be done for restaurants and customers. Restaurants shall be able to do track their orders, add-update-remove their menu, the inventory shall be able to update according to order status, also shall be able to add their contacts in the admin panel of this web page. Users shall be able to register and create orders, make an online payment, and make table reservations from this web page.

#### **1.3. Objectives and success criteria of the project**

The application should be completed by 45 days and it should be costless for the restaurant and customers. Both user sides should satisfy when using this app. Restaurants should not confuse with the admin panel and should get everything they need. My start-up objective is to reach real enterprises that can implement this project for their business area.

#### **1.4. Definitions, acronyms, and abbreviations**

Definitions, acronyms, and abbreviations are located at page ix.

#### **1.5. References**

References are located at section "REFERENCES" on page 9.

#### **1.6. Overview**

In the general definition, the app will use to do online most of the restaurant routine.

## **2. CURRENT SYSTEM**

Most part of the project is done and working properly. Online payment and some additional front-end issues can be added.

### **3. PROPOSED SYSTEM**

#### **3.1. Overview**

This system must be easy to use for customers and restaurants. The restaurant should use business panel features with no problem and the customer should order with no problem. There should be no system vulnerability and a security bug.

#### **3.2. Functional requirements**

This app support three types of users: Employer should be able to get-create-update-delete food, inventory, category, get users, get-update-delete orders, get reviews, get a statistic, get access panel. Employees should be able to get-update orders, get foods, users, category, get-create-update-delete inventory, get reviews, get a statistic, access to the panel. Customers should be able to get foods, category, get (only own order)-create order, get-create-update-delete review, register, login, delete an account.

#### **3.3. Nonfunctional requirements**

##### **3.3.1. Usability**

The application can be easy to use by people which is 15 from 80 per restaurant. This app shall be able to be used by members without training. People who want to use this app shall be able to be login fast if enter a few personal information. Members can simply proceed by making their own choices on each simple screen. There can be a tutorial page for a user after signing up. On other hand, restaurants shall have a manual API document and instruction manual for developing or changing code.

##### **3.3.2. Reliability**

This application will be tested, and, in this application, nothing should be going wrong or fail in case of a web server is working.

##### **3.3.3. Performance**

This system can be handling multiple users' requests at the same time. This system should load in under 10 seconds when the number of simultaneous users is greater than 100. When the user ordered food, it can be delivered to the restaurant immediately. The system can operate reliably over a period.



#### **3.3.4. Supportability**

There is no public API usage inside the app but there are small packages in-app. The restaurant will be able to download these with the node package manager these packages.

#### **3.3.5. Implementation**

In the Server-side application, NodeJS and for database MongoDB will be used. For creating an API, the Express library to create a server on NodeJS will be used. Since I decided to use JavaScript for the server-side, the front-end will be implemented with React.

#### **3.3.6. Interface**

The web application has a simple user interface. It is designed for everyone to use comfortably. Bootstrap and Material UI used in-app for the UI framework. These frameworks are providing fine-looking and user-friendly features.

#### **3.3.7. Packaging**

Users can use our application without any download.

#### **3.3.8. Legal**

There will be a license agreement between users and restaurant.

### **3.4. System Models**

### 3.4.1. Scenarios

#### 3.4.1.1. S-01: Customer Register

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-00
Description:	<p>The registers to the system. They must provide the following information:</p> <ul style="list-style-type: none"> <li>● Email</li> <li>● Password</li> <li>● Confirm Password</li> </ul>

#### 3.4.1.2. S -02: Customer Login

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-01
Description:	<p>The restaurant logins to the system. They must provide the following information:</p> <ul style="list-style-type: none"> <li>● Email</li> <li>● Password</li> </ul> <p>On successful login, redirect to main page as logged in restaurant</p>

#### 3.4.1.3. S -03: Edit Customer Profile

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-02
Description:	<p>The user can edit their attributes such as:</p> <ul style="list-style-type: none"> <li>• Name</li> <li>• Surname</li> <li>• Phone</li> <li>• Email</li> <li>• Image</li> <li>• Address</li> </ul>

#### 3.4.1.4. S -04: Change Customer Password

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-03
Description:	Change passport of user account if forgotten or for security reasons.

### 3.4.1.5. S -05: Employer or Admin Login

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Administration
Use Case(s):	S-05
Description:	<p>The admin logs in to the system. They must provide the following information:</p> <ul style="list-style-type: none"> <li>● Email</li> <li>● Password</li> </ul> <p>On successful login, redirect to main page as logged in user</p>

### 3.4.1.6. S -06: Employee Register

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Employee
Use Case(s):	S-06
Description:	<p>The restaurant can edit their attributes such as:</p> <ul style="list-style-type: none"> <li>● Username</li> <li>● User Email</li> <li>● Password</li> </ul>

**3.4.1.7. S -07: Employee Login**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Employee
Use Case(s):	S-07
Description:	<p>When admin sign user role as employee they can login as employee with these:</p> <ul style="list-style-type: none"> <li>• Email</li> <li>• Password</li> </ul>

**3.4.1.8. S -08: Admin Change User Role**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Administration
Use Case(s):	S-08
Description:	When admin login to business panel, admin can change role of specific user.

**3.4.1.9. S -09: Add Menu**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Administration
Use Case(s):	S-09
Description:	Admin can add menu for restaurant at business panel.

**3.4.1.10. S -10: Edit Menu**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Administration
Use Case(s):	S-10
Description:	Admin can edit restaurant menu at business panel.

**3.4.1.11. S -11: Delete Menu**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Administration
Use Case(s):	S-11
Description:	Admin can delete restaurant menu at business panel.

**3.4.1.12. S -12: Customer Add Items to Basket**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-12
Description:	Customer can add items from menus to their basket.

**3.4.1.13. S -13: Customer Edit Items to Basket**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-13
Description:	Customer can edit items which is in their basket.

**3.4.1.14. S -14: Customer Remove Items to Basket**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-14
Description:	Customer can remove items from their baskets while browsing their basket.

**3.4.1.15. S -15: Complete Order**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-15
Description:	Confirm the items in your basket and send an order to the restaurant.

**3.4.1.16. S -16: Get Orders of Restaurant**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Employee or Admin
Use Case(s):	S-16
Description:	Get orders that users sent to your Restaurant.

**3.4.1.17. S -17: Change Order Status**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Employee or Admin
Use Case(s):	S-17
Description:	<p>Change order status.</p> <p>Order Status:</p> <ul style="list-style-type: none"> <li>● In queue</li> <li>● Preparing</li> <li>● Served</li> <li>● Canceled</li> </ul>

**3.4.1.18. S -18: Change Order Payment Status**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Employee or Admin
Use Case(s):	S-18
Description:	<p>Change payment status.</p> <p>Payment Status:</p> <ul style="list-style-type: none"> <li>● Waiting Payment</li> <li>● Paid</li> <li>● Canceled</li> </ul>

**3.4.1.19. S -19: Get Orders**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Restaurant Employee or Admin
Use Case(s):	S-19
Description:	Get all orders, include past orders, you must be login as employee or admin.



**3.4.1.20. S -20: Get all orders (Current User)**

Priority:	Essential
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-20
Description:	Customer can get all orders that own.

**3.4.1.21. S -21: Online Payment**

Priority:	Optional
Risk:	Safe
Functional Area(s):	Customer
Use Case(s):	S-21
Description:	Option to pay from an online application instead of paying at restaurant checkout. You must be logged in.

**3.4.1.22. S -22: Pre-order/Reservation**

Priority:	Optional
Risk:	1-Risk
Functional Area(s):	Customer
Use Case(s):	S-22
Description:	Tell a restaurant you will be there at a specific hour and want your order to be ready. Payment is done online while completing your order. You must be logged in.

**3.4.1.23. S -23: Manage Inventory**

Priority:	Essential
Risk:	1-Risk
Functional Area(s):	Admin
Use Case(s):	S-23
Description:	Admin can add, remove, update, get inventory from business panel.

**3.4.1.24. S -24: Change Inventory with Order**

Priority:	Optional
Risk:	1-Risk
Functional Area(s):	All
Use Case(s):	S-24
Description:	<ul style="list-style-type: none"> <li>• Set materials in restaurant inventory.</li> <li>• Specify the necessary materials when preparing an item. The application will remove the materials from restaurant inventory when an order status changes in the queue to preparing. Inventory can see in the admin panel.</li> </ul>

**3.4.1.25. S -25: Review for customers**

Priority:	Optional
Risk:	1-Risk
Functional Area(s):	Customer
Use Case(s):	S-25
Description:	<p>Customer can</p> <ul style="list-style-type: none"> <li>• Create Review</li> <li>• Edit Review</li> <li>• Delete Review</li> <li>• Get All Own Review</li> </ul> <p>Customer also can get all reviews of other users.</p>

**3.4.1.26. S -26: Review for restaurant**

Priority:	Optional
Risk:	1-Risk
Functional Area(s):	Admin
Use Case(s):	S-26
Description:	Admin can <ul style="list-style-type: none"> <li>• Delete Review</li> <li>• Get Review</li> </ul>

**3.4.1.27. S -27: Suggestion for Cart Items**

Priority:	Essential
Risk:	1-Risk
Functional Area(s):	Customer
Use Case(s):	S-27
Description:	Check the cart current user and if there is a category that is not taken category of item in the cart, the system will suggest an item in that category.

**3.4.1.28. S -28: Top 5 Foods**

Priority:	Essential
Risk:	1-Risk
Functional Area(s):	Customer
Use Case(s):	S-28
Description:	Most ordered foods are showing at the top of their category.

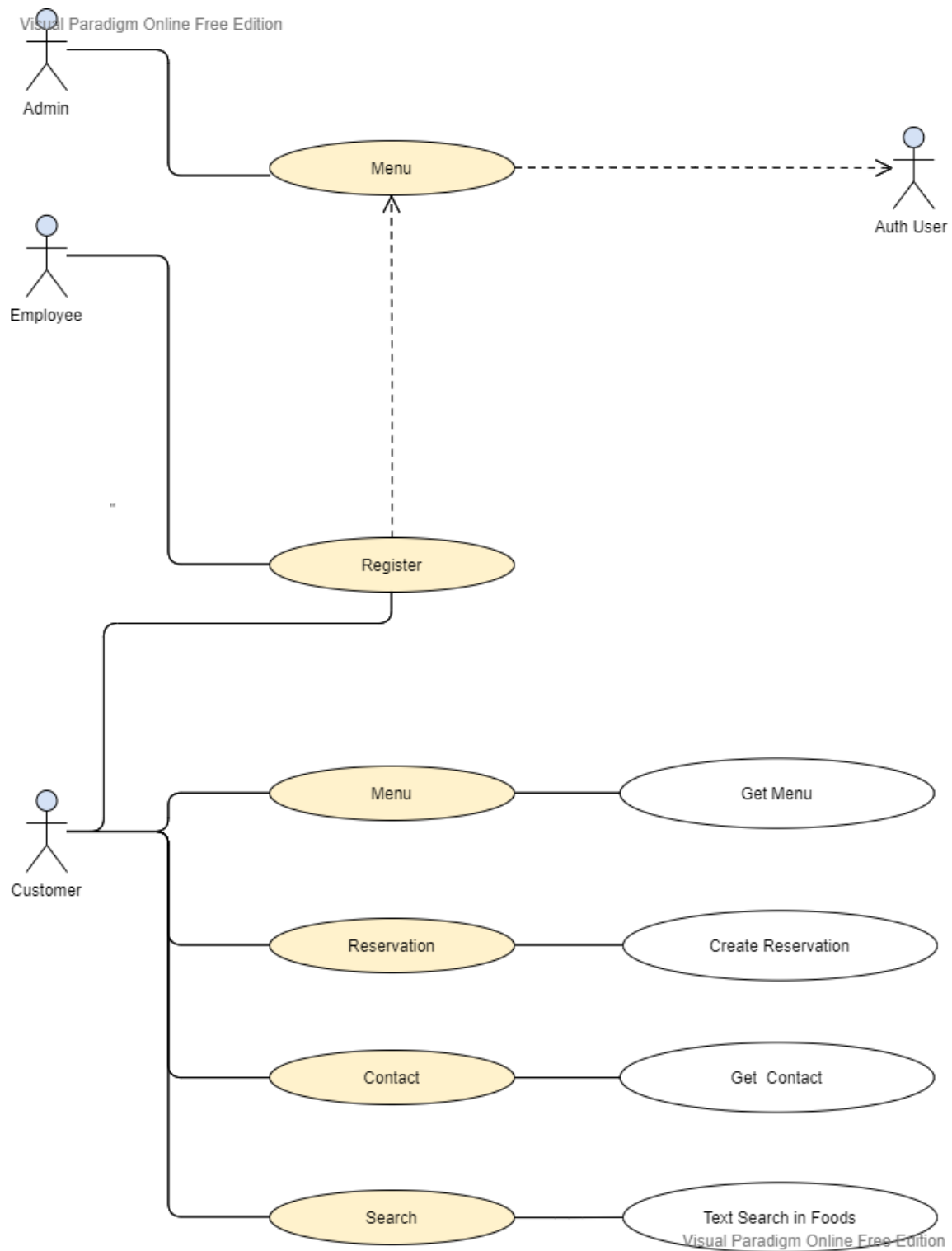
**3.4.1.29. S -29: Get Food Statistics**

Priority:	Essential
Risk:	1-Risk
Functional Area(s):	Employee and Admin
Use Case(s):	S-29
Description:	Employee and Admin can see all foods by category, average rating, average price.

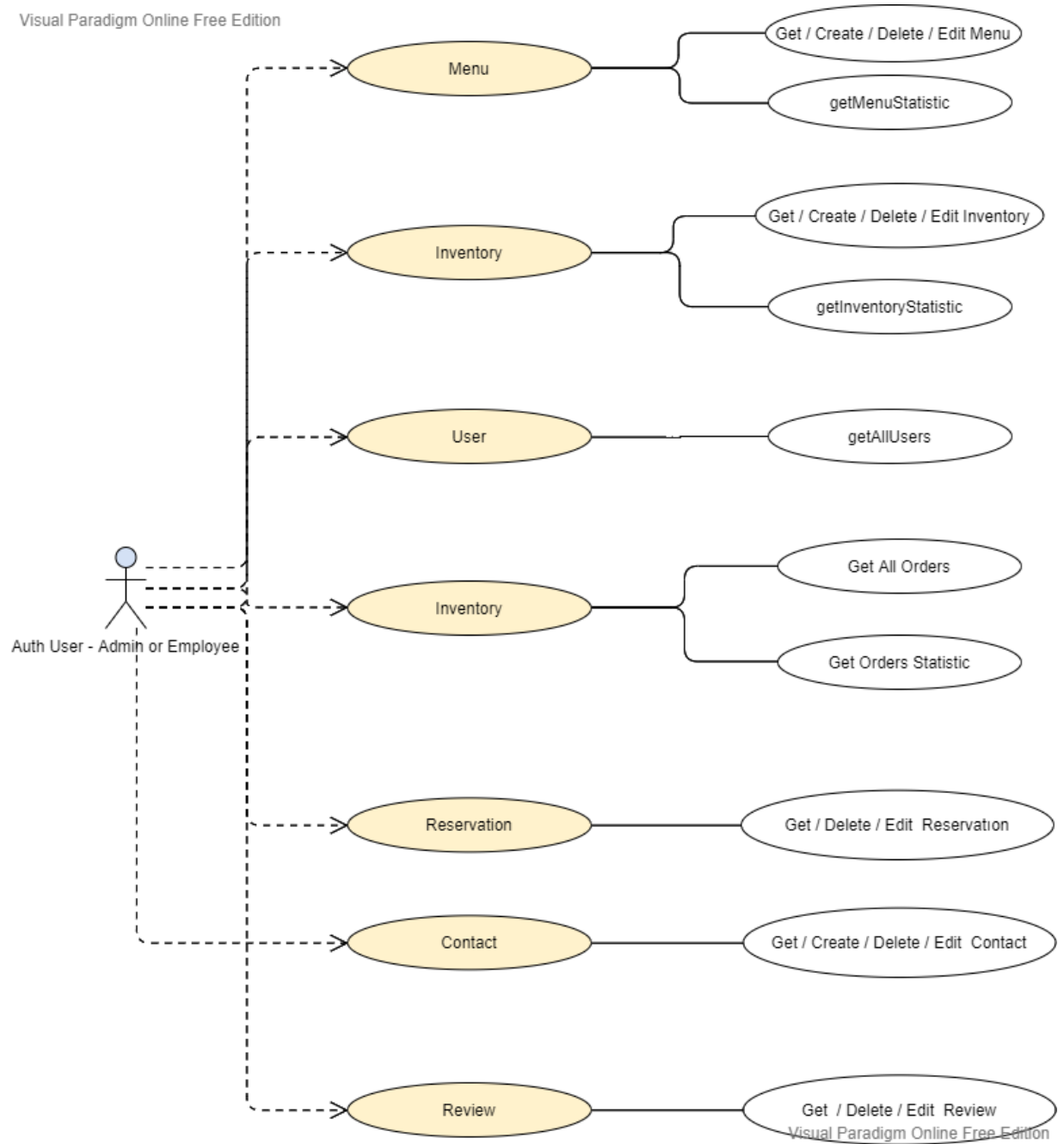
**3.4.1.30. S -30: Search with text in Food**

Priority:	Optional
Risk:	1-Risk
Functional Area(s):	Customer
Use Case(s):	S-30
Description:	Customers can search for items in the food search bar. Food description saved from database in project daily basis.

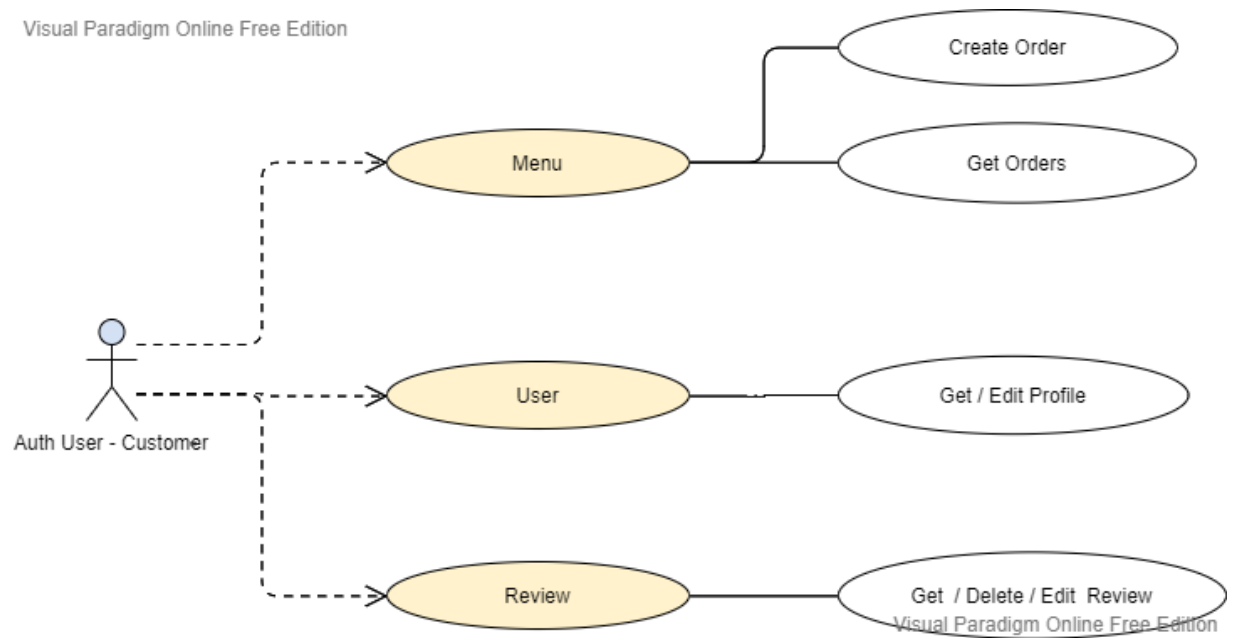
### 3.4.2. Use Case Model



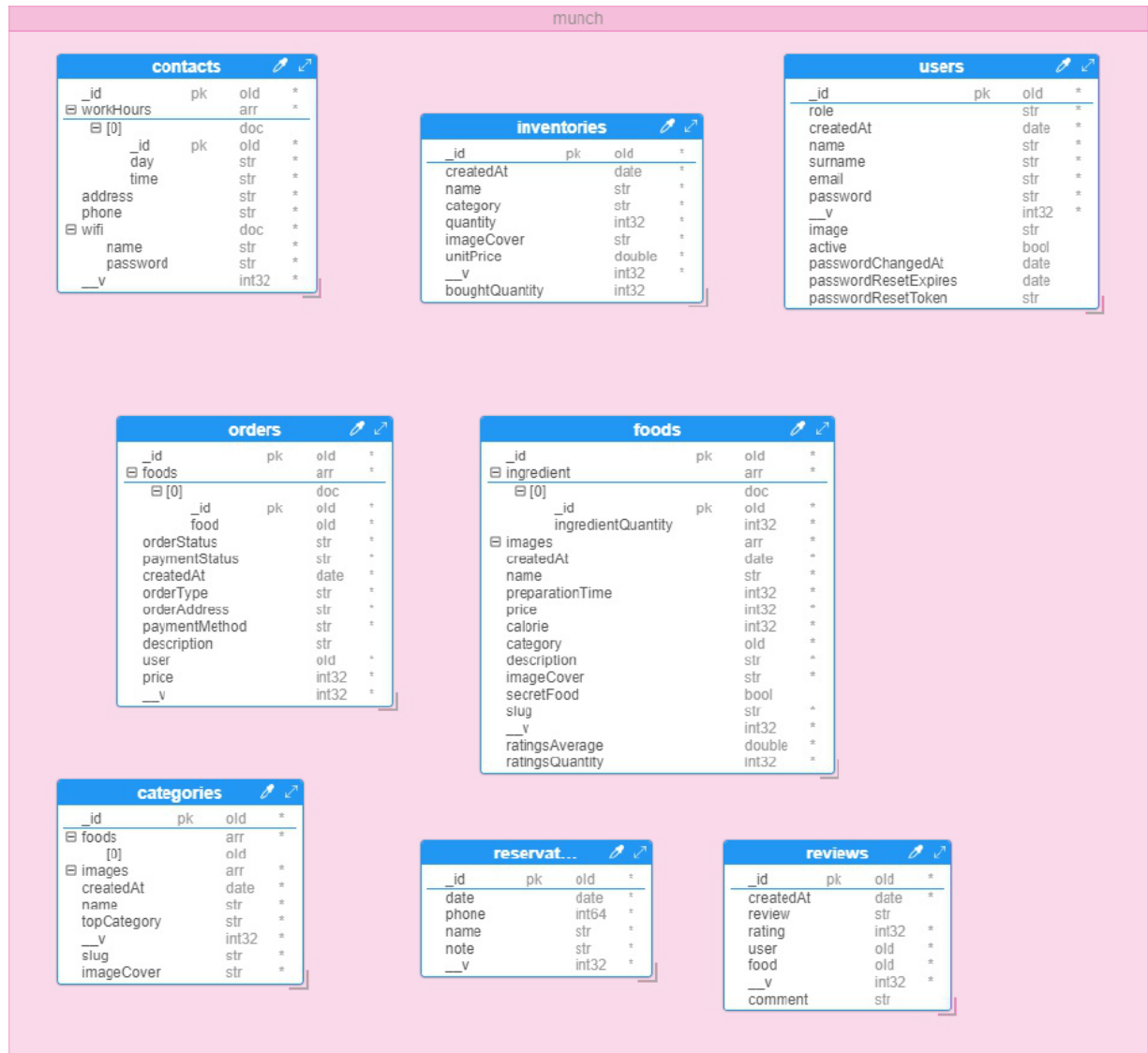
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition



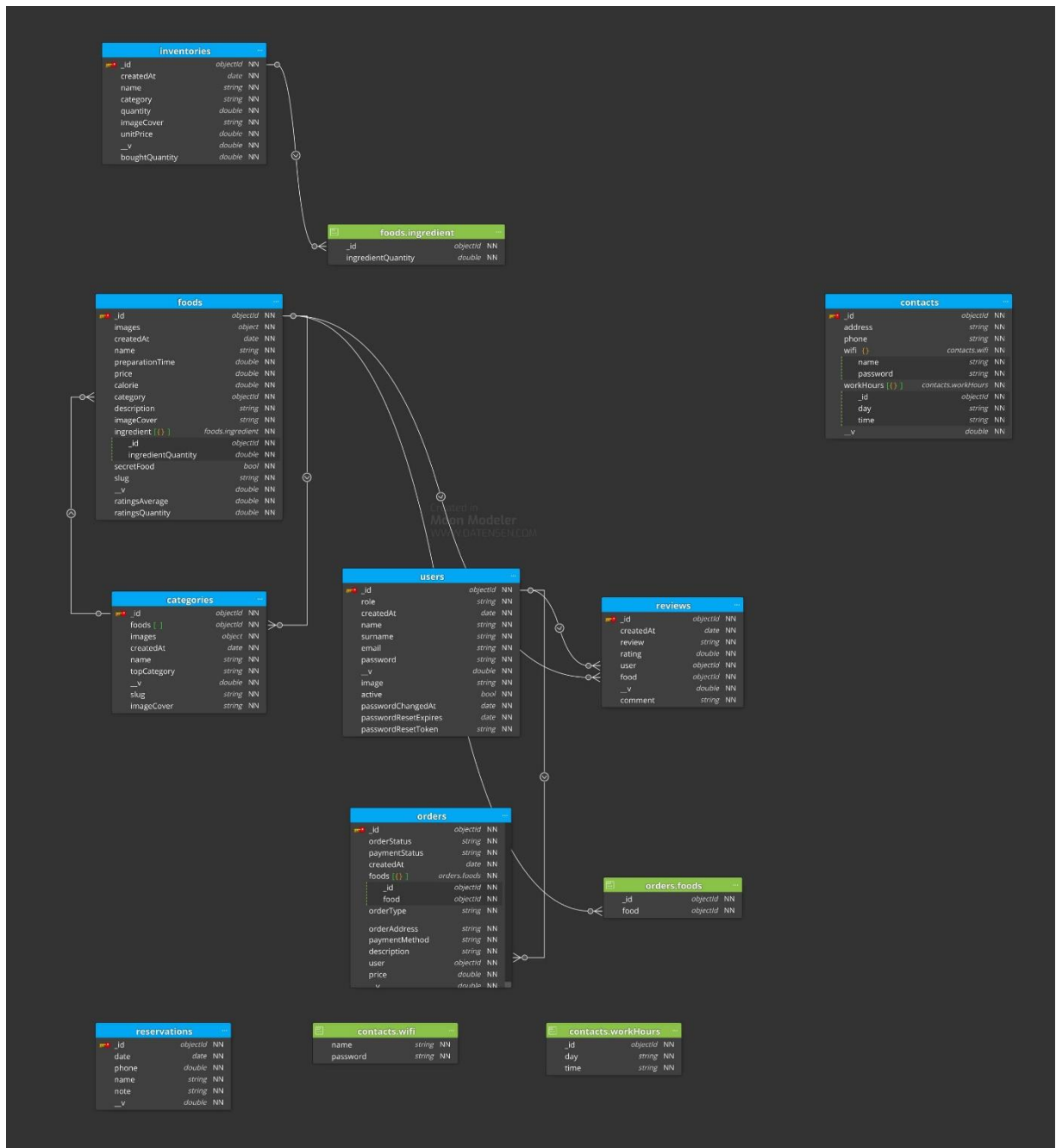
### 3.4.3. Object Model



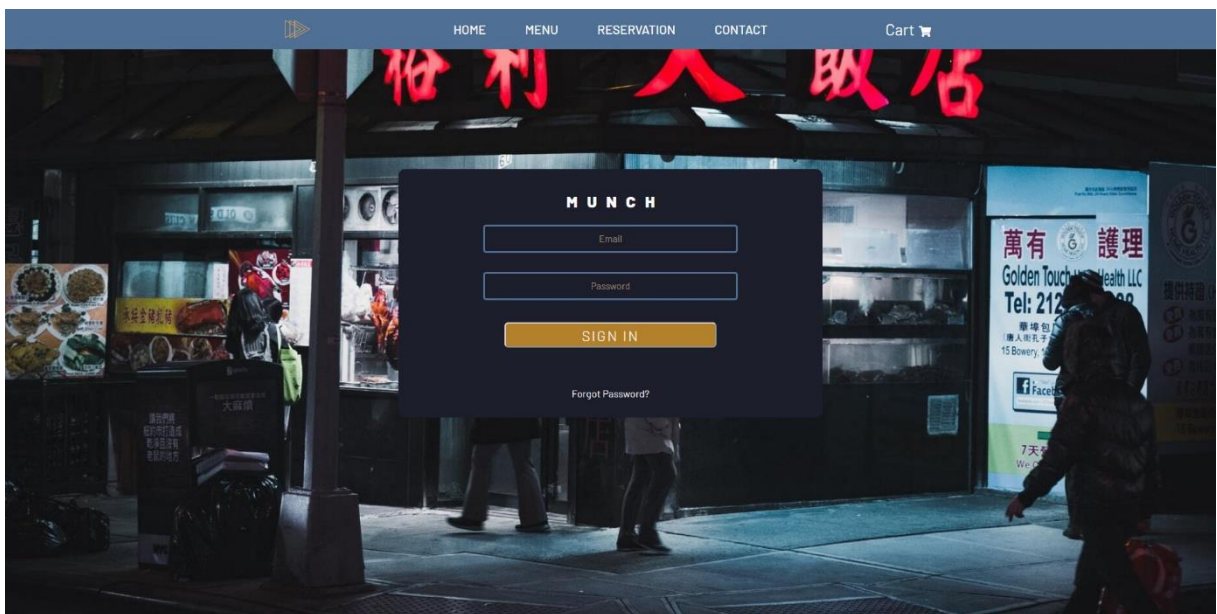
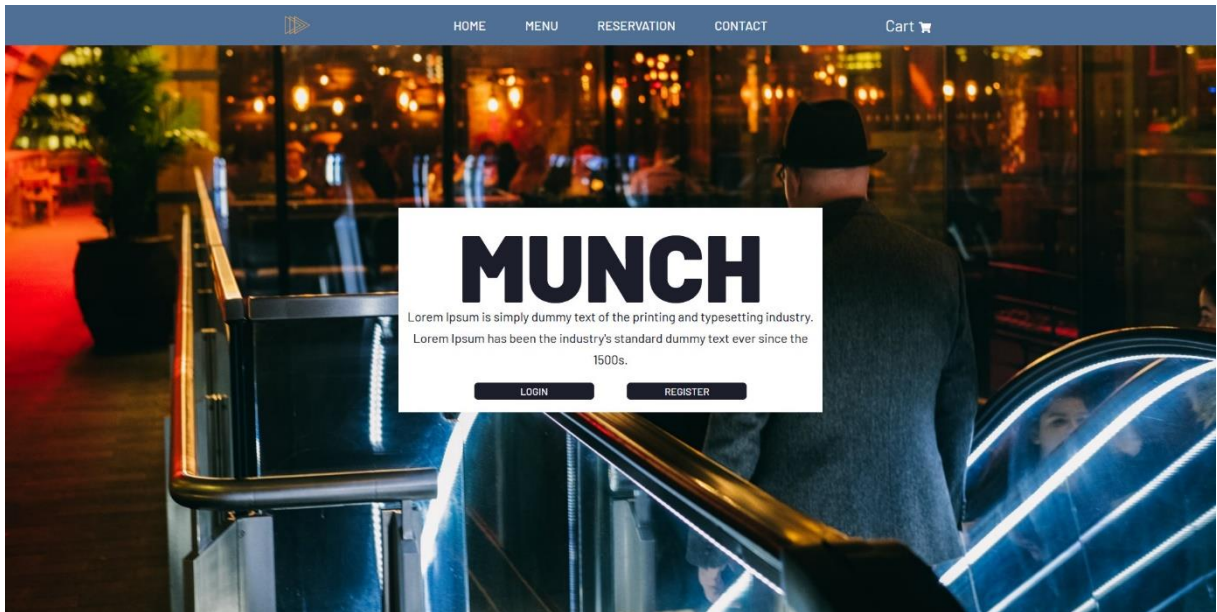




### 3.4.4. Dynamic Model

### 3.4.5. Database model — ER diagram



### 3.4.6. User interface—navigational paths and screen mock-ups



 [HOME](#) [MENU](#) [RESERVATION](#) [CONTACT](#) [Cart](#) 

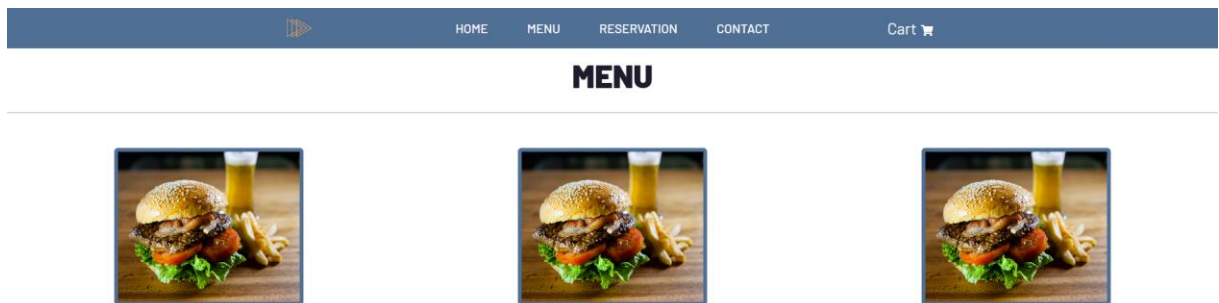
# MUNCH

## CREATE ACCOUNT

### REGISTER

<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

SUBMIT




MENU-ITEM

GO BACK


CheeseBurger   ★ 8,4

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.




CheeseBurger   ★ 8,4

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.




CheeseBurger   ★ 8,4

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.




CheeseBurger   ★ 8,4

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



CheeseBurger   ★ 8,4

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



MENU-ITEM-DETAIL

GO BACK

ADD TO BASKET



Description

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Ingredients

- A

■ A

■ A

■ A
- A

■ A

■ A

■ A
- A

■ A

■ A

■ A

Preparation Time : 15 min   Calorie : 155 kcal

Price : 15.25


## CART


FOOD	PRICE	QUANTITY	AMOUNT	
Hamburger	30.50	<input type="text" value="1"/>	30.50 TL	
Hamburger	30.50 TL	<input type="text" value="1"/>	30.50 TL	
Hamburger	30.50	<input type="text" value="1"/>	30.50 TL	


## ORDER


[GO BACK](#)

### BILLING

 Full Name

 Address







 Delivery Time

☒ Now (20-30 minutes)

☐ Future

### PAYMENT

Accepted Cards



Name on Card


Credit card number

Exp Date

CVV

CONTINUE TO CHECKOUT

### CART

 4

Product 1

\$16

Product 2

\$6

Product 3

\$8

Product 4

\$2

Total

\$30

### ORDER NOTE

#### 4. GLOSSARY

- **API** - Application programming interface. Spell out the abbreviation when it is first used.
- **MongoDB** - Cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.
- **NodeJS** - An open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside a web browser.
- **React** - An open-source JavaScript library for building user interfaces.
- **Express** - Fast, unopinionated, minimalist web framework for Node.js.

## PROJECT BUDGET AND TIME PLAN

### 5. INTRODUCTION

A project plan is one of the most important ingredients for a successful project. It is used to document and communicate expectations, to control schedule and delivery, and to calculate and manage risks. In general, it acts as a roadmap for everyone involved in the project.

#### 5.1. Gantt Chart

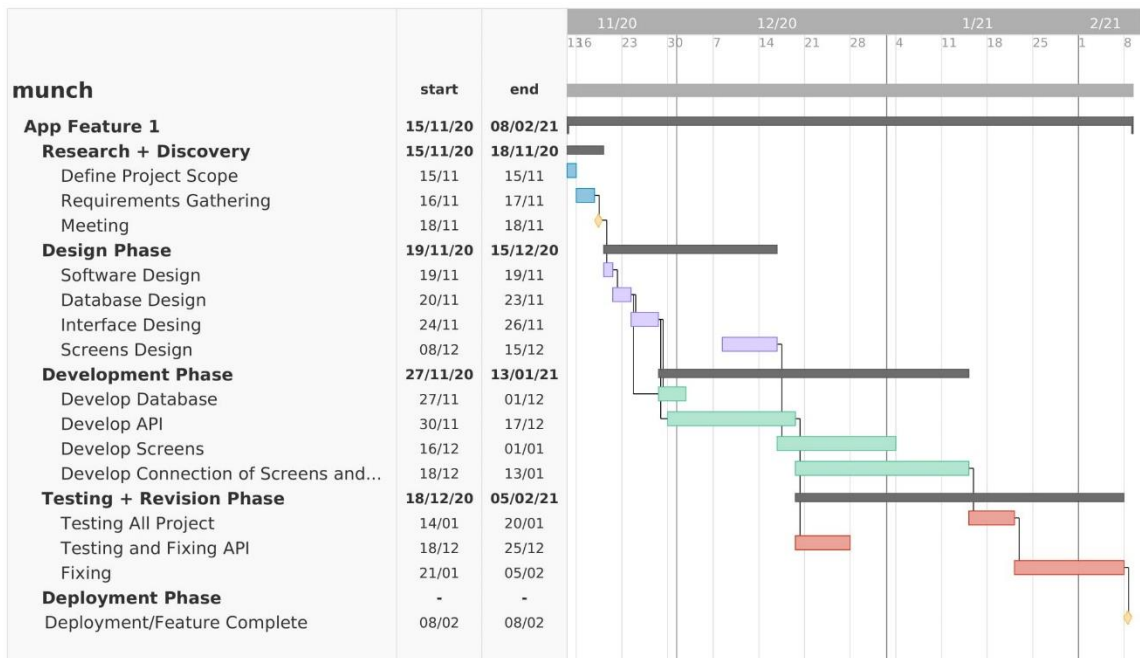


Figure 0.1 Gantt chart sample 1

### 5.1. Project Budget

**Table 0.1 Project budget table**

	Estimated		
Expenses	Period (One time or Recurrent)	Cost per Period	Total Cost
Human resource for support	30 man-day	100 TL	3.000 TL
VS Code	1 Time	free	0
Hosting for 3 months	Monthly	100 TL	300 TL
Domain	1 Time	50 TL	50 TL
TOTAL BUDGET			3.350 TL

According to the plan, estimated project budget is 3.350 TL.

### 5.2. Project Phase Distribution

Implementation took most of our time during project, after that design was second most took time.

**Table 0.2 Project phase distribution**

Planning & Analysis	Design	Implementation	Testing
%10	%35	%40	%15

### 5.3. Other Tables

You can extend this section as 5.3 Log tables, 5.4 Advisor meeting tables, or other tables or diagrams as you wish.



## PROJECT SPECIFIC CONTENT

### 6. INTRODUCTION

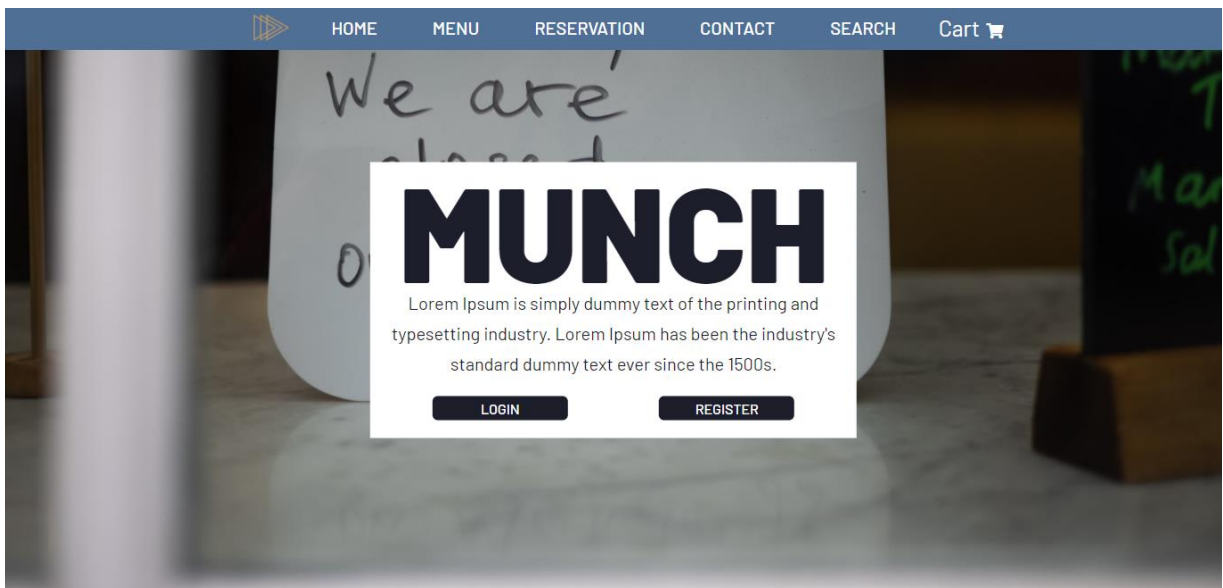
#### 6.1. ACCESSIBILITY

Since it is lightweight application, browser can download web page fast so no problem for user about opening. This app is also built-in responsive web design it means that mobile user can use this app without any problem.

#### 6.2. USER MANUAL

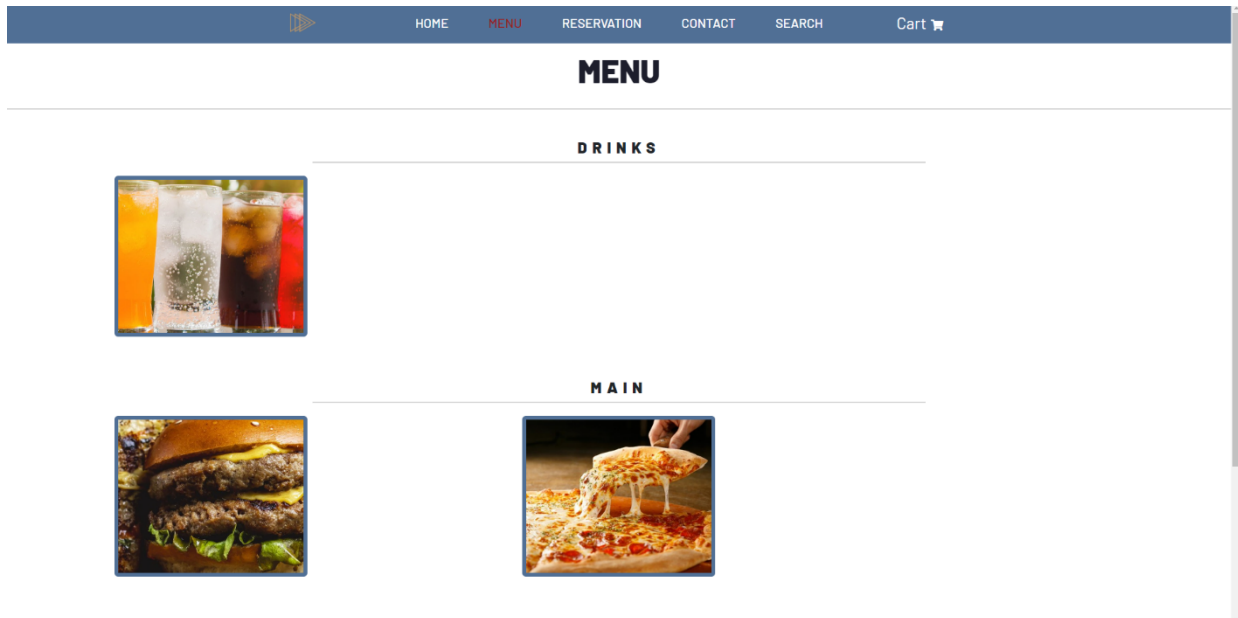
##### 6.2.1. Landing Page

When a user comes from a browser to our website if there is no authentication, the user sees this page. You can select login or register for an account otherwise you can select other options in the navbar to see the menu, create a reservation, see restaurant information, or search food.

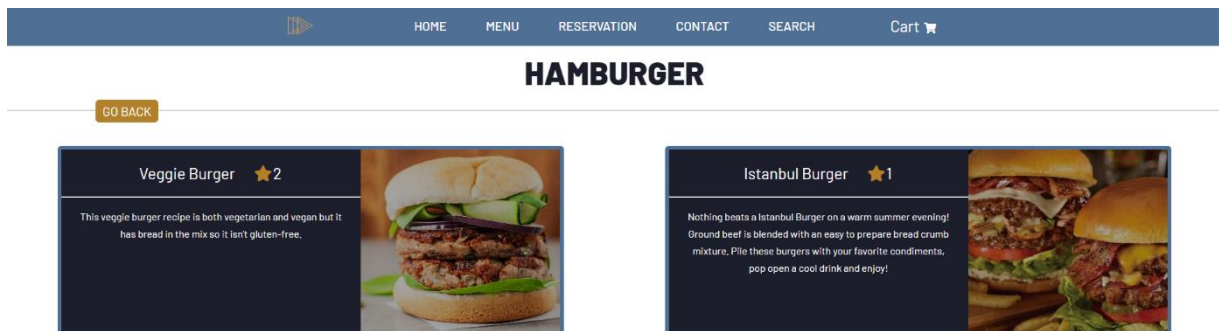


##### 6.2.2. Menu Page


On the menu page, you can select a category that is separated as top categories.



When you select items from the menu page, you can see the foods in that category.



After your selection, you can see the food detail. You can press add cart button to buy this food when you logged in.



**VEGGIE BURGER**

★☆☆☆ 4.5 reviews

Price: \$15

Add To Cart

Description: This veggie burger recipe is both vegetarian and vegan but it has bread in the mix so it isn't gluten-free.

Ingredients:

- \$Meat
- \$Tomato

Go Back

Reviews

**jaga**

★★★★☆

2020-12-11


**ceren**

★★★★★

2021-02-03


### 6.2.3. Reservation Page

You can fill this form to create a reservation. There is the name, date, reservation note, and phone number area. You must fill in all the blanks except the note area.



### 6.2.4. Contact Page

You can see the restaurant address, phone, Wi-Fi, and work hours on this page.


[HOME](#)
[MENU](#)
[RESERVATION](#)
[CONTACT](#)
[SEARCH](#)
[Cart](#)


## CONTACT

---

Address	Phone
854 Lyme Avenue, Smeltertown, Georgia, 846	+1(880)521-3401
Wifi	Days
Wifi Name: munch1	Sunday
Wifi Password: 1234munch	Monday
	Times
	19.00-20.00
	10.00-18.00

### 6.2.5. Search Page

On this page, you can search for anything you want and see matched results.



[HOME](#)
[MENU](#)
[RESERVATION](#)
[CONTACT](#)
[SEARCH](#)
[Cart](#)

## SEARCH

---

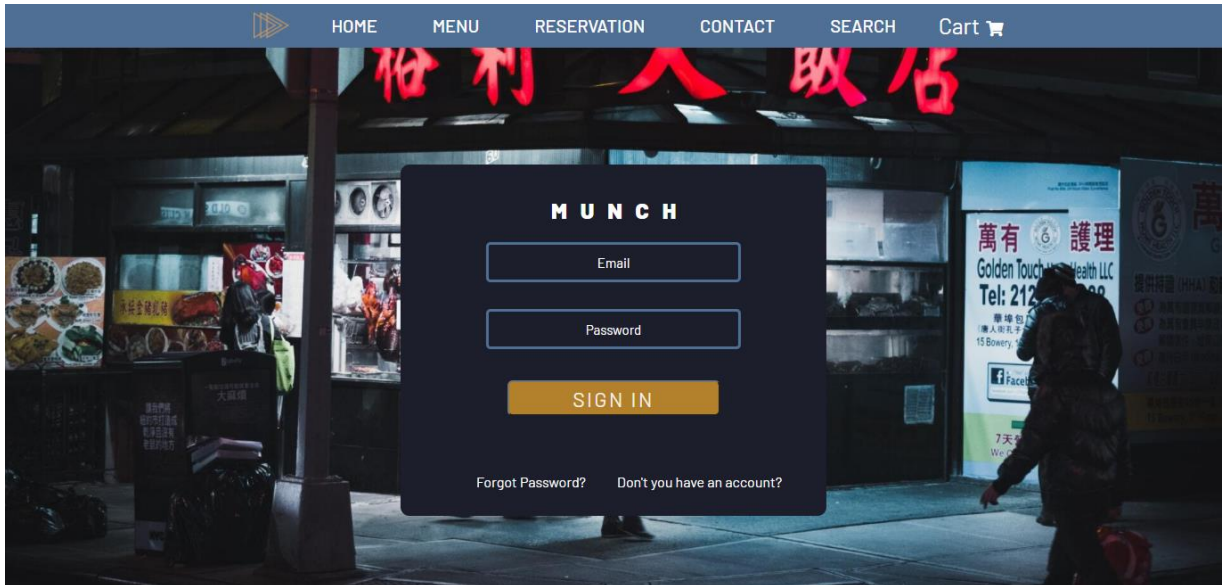
Veggie Burger ★2

This veggie burger recipe is both vegetarian and vegan but it has bread in the mix so it isn't gluten-free.



### 6.2.6. Login Page

On this page, you can be logged in if you have an account.



### 6.2.7. Register Page

On this page, you can create an account if you have not.

CREATE ACCOUNT

REGISTER

Name

Surname

Email

Phone

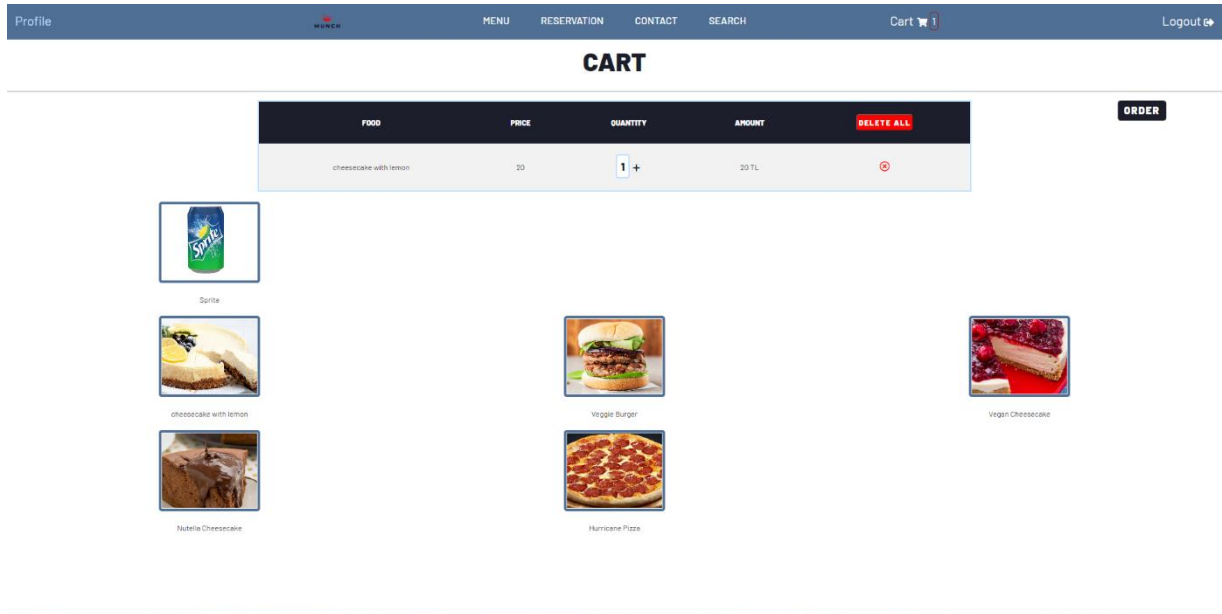
Password

Confirm Password

SUBMIT

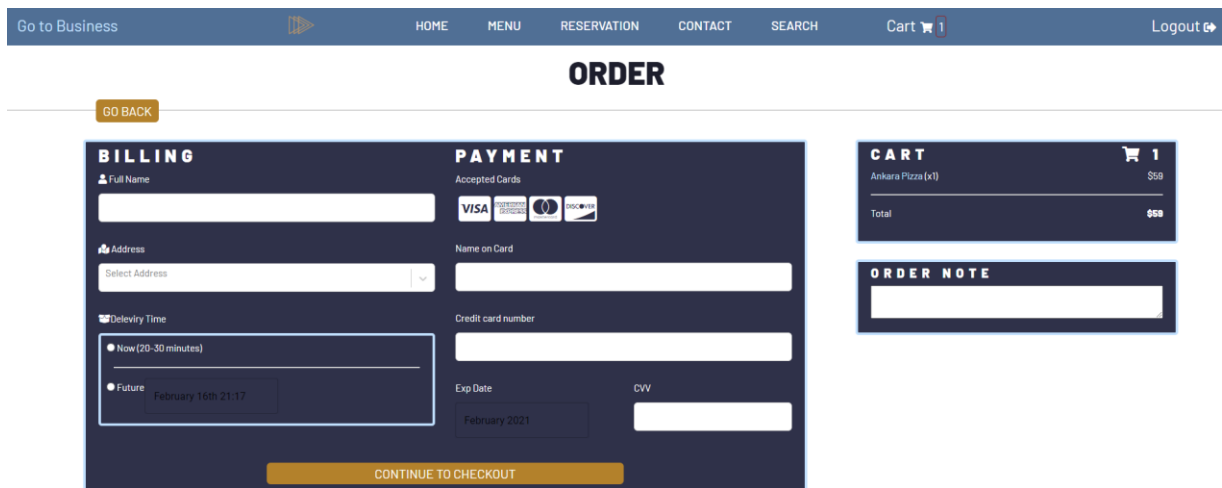
### 6.2.8. Cart Page

After you logged in, you can add an item to your cart, and you can see your cart on this page. You can delete, increase, or decrease items. There is one additional area below the table which is the suggestion part. According to your cart, there is food shown in that area.



### 6.2.9. Order Page

You can check out your cart and create an order in this area.

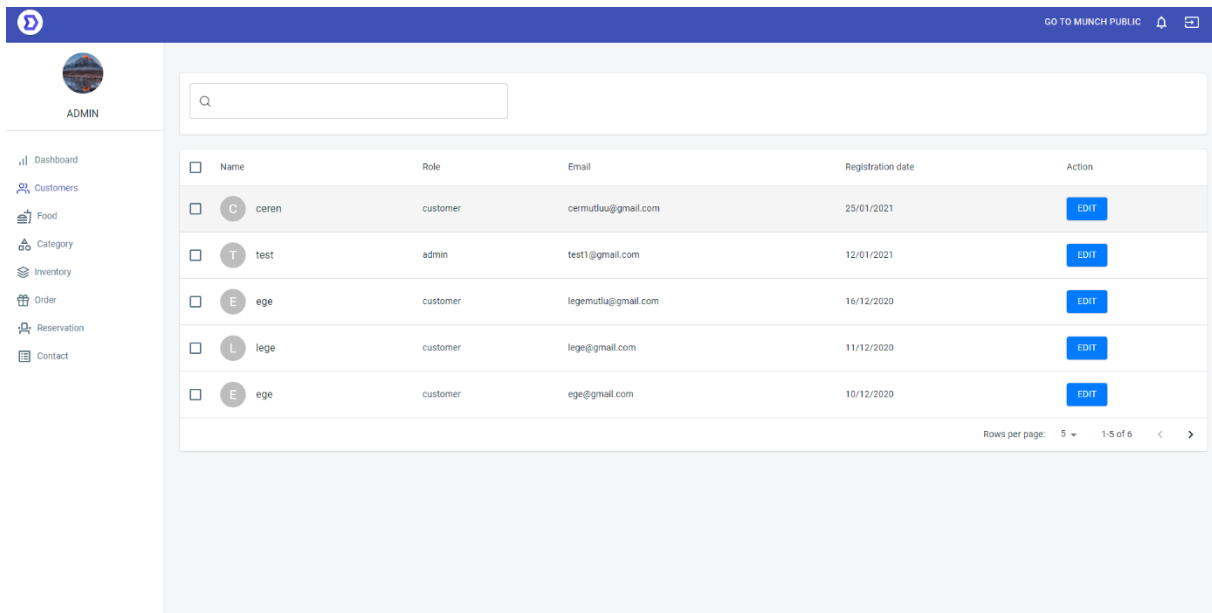







### 6.2.10. Admin Page

If you are logged in as an admin or employee, you can access the admin panel from the navbar.

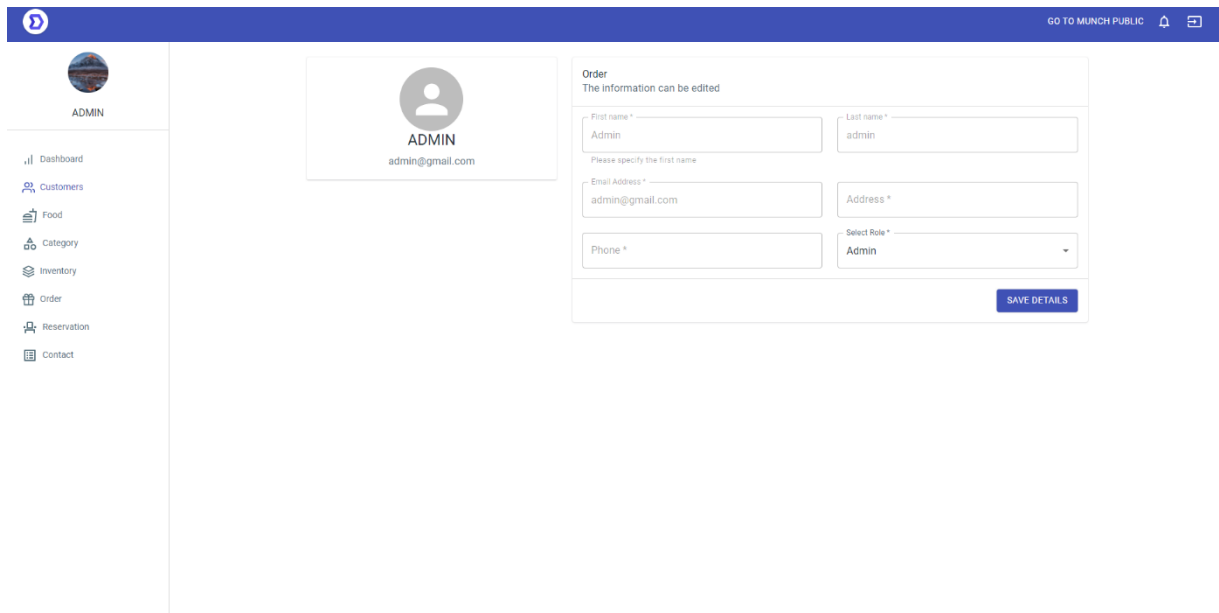
### 6.2.11. Admin Customer Page

If you are an admin you can access the customer page from the sidebar and edit the customer role. Admin can change role of user.



<input type="checkbox"/>	Name	Role	Email	Registration date	Action
<input type="checkbox"/>	 ceren	customer	cermutlu@gmail.com	25/01/2021	<a href="#">EDIT</a>
<input type="checkbox"/>	 test	admin	test1@gmail.com	12/01/2021	<a href="#">EDIT</a>
<input type="checkbox"/>	 ege	customer	legemutlu@gmail.com	16/12/2020	<a href="#">EDIT</a>
<input type="checkbox"/>	 lege	customer	lege@gmail.com	11/12/2020	<a href="#">EDIT</a>
<input type="checkbox"/>	 ege	customer	ege@gmail.com	10/12/2020	<a href="#">EDIT</a>

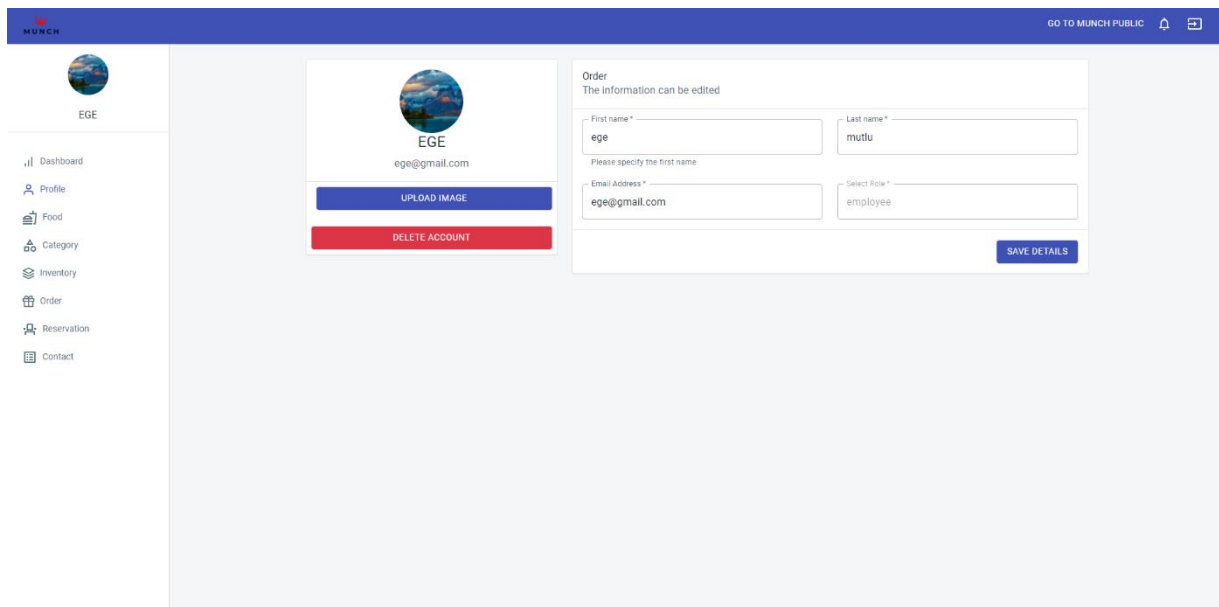
Rows per page: 5 1-5 of 6



The screenshot shows the Admin Profile Page. The top navigation bar is blue with the MUNCH logo on the left and a 'GO TO MUNCH PUBLIC' link on the right. The left sidebar is white and contains a list of menu items: Dashboard, Customers, Food, Category, Inventory, Order, Reservation, and Contact. The main content area is white and features a profile card on the left with a placeholder image, the name 'ADMIN', and the email 'admin@gmail.com'. To the right of the profile card is a form titled 'Order' with the subtitle 'The information can be edited'. The form contains several input fields: 'First name \*' (Admin), 'Last name \*' (admin), 'Email Address \*' (admin@gmail.com), 'Address \*', 'Phone \*', and a 'Select Role \*' dropdown menu set to 'Admin'. A 'SAVE DETAILS' button is located at the bottom right of the form.

### 6.2.12. Admin Profile Page

You can access your profile page if you are logged in as an employee. You can edit your profile except for your role on this page.

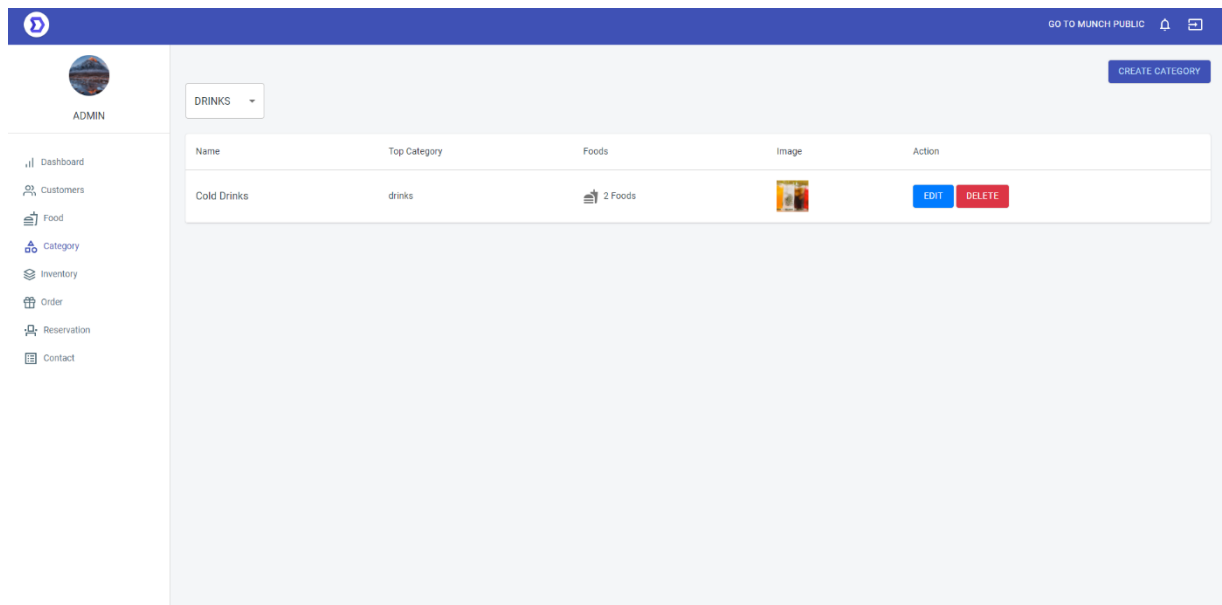


The screenshot shows the Employee Profile Page. The top navigation bar is blue with the MUNCH logo on the left and a 'GO TO MUNCH PUBLIC' link on the right. The left sidebar is white and contains a list of menu items: Dashboard, Profile, Food, Category, Inventory, Order, Reservation, and Contact. The main content area is white and features a profile card on the left with a placeholder image, the name 'EGE', and the email 'ege@gmail.com'. Below the profile card are two buttons: 'UPLOAD IMAGE' (blue) and 'DELETE ACCOUNT' (red). To the right of the profile card is a form titled 'Order' with the subtitle 'The information can be edited'. The form contains several input fields: 'First name \*' (ege), 'Last name \*' (mutlu), 'Email Address \*' (ege@gmail.com), and a 'Select Role \*' dropdown menu set to 'employee'. A 'SAVE DETAILS' button is located at the bottom right of the form.



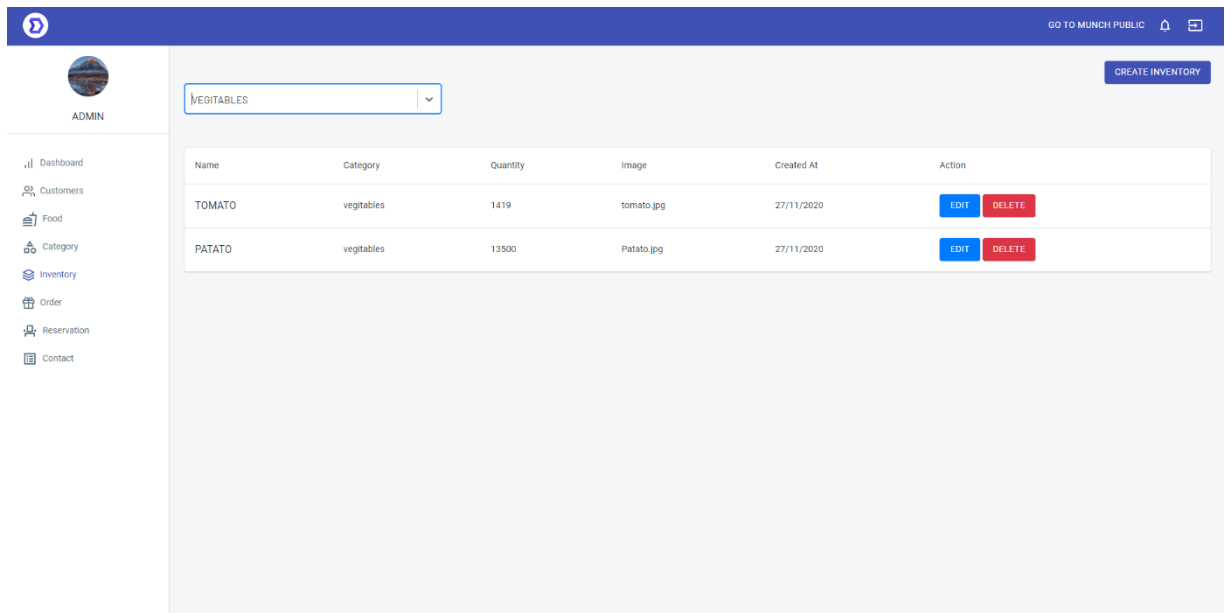
### 6.2.13. Admin Category Page

You can access the category page in the admin panel. You can see a list of categories, create a category, edit the category, or delete a category on this page. Also, you must select a top category from the dropdown menu which is top of the list.



### 6.2.14. Admin Inventory Page

You can access the inventory page in the sidebar. You must select an inventory category and a list of inventory items will be able to be shown. You can edit or delete created inventory, or you can create a new inventory.

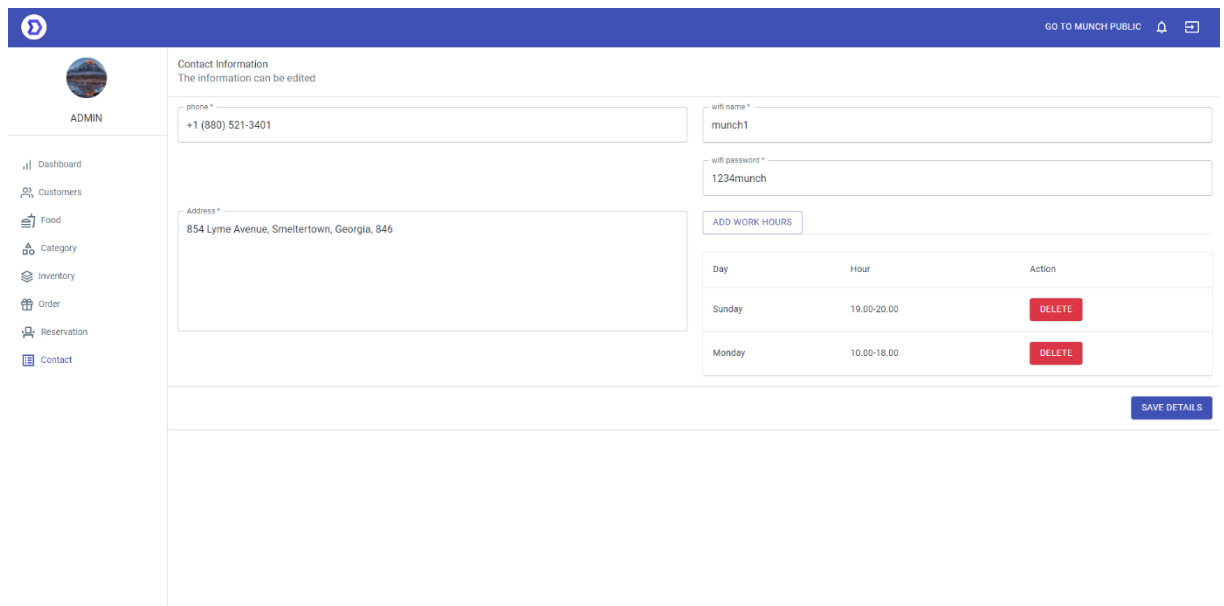


The screenshot shows the Admin Inventory Page. At the top, there's a blue header with a logo and a "GO TO MUNCH PUBLIC" link. Below the header, a sidebar on the left lists navigation options: Dashboard, Customers, Food, Category, Inventory, Order, Reservation, and Contact. The main content area has a "VEGETABLES" dropdown menu and a "CREATE INVENTORY" button. Below this is a table with columns: Name, Category, Quantity, Image, Created At, and Action.

Name	Category	Quantity	Image	Created At	Action
TOMATO	vegetables	1419	tomato.jpg	27/11/2020	<a href="#">EDIT</a> <a href="#">DELETE</a>
PATATO	vegetables	13500	Patato.jpg	27/11/2020	<a href="#">EDIT</a> <a href="#">DELETE</a>

### 6.2.15. Admin Contact Page

You can access the contact page from the sidebar. You can edit contact information on this page.



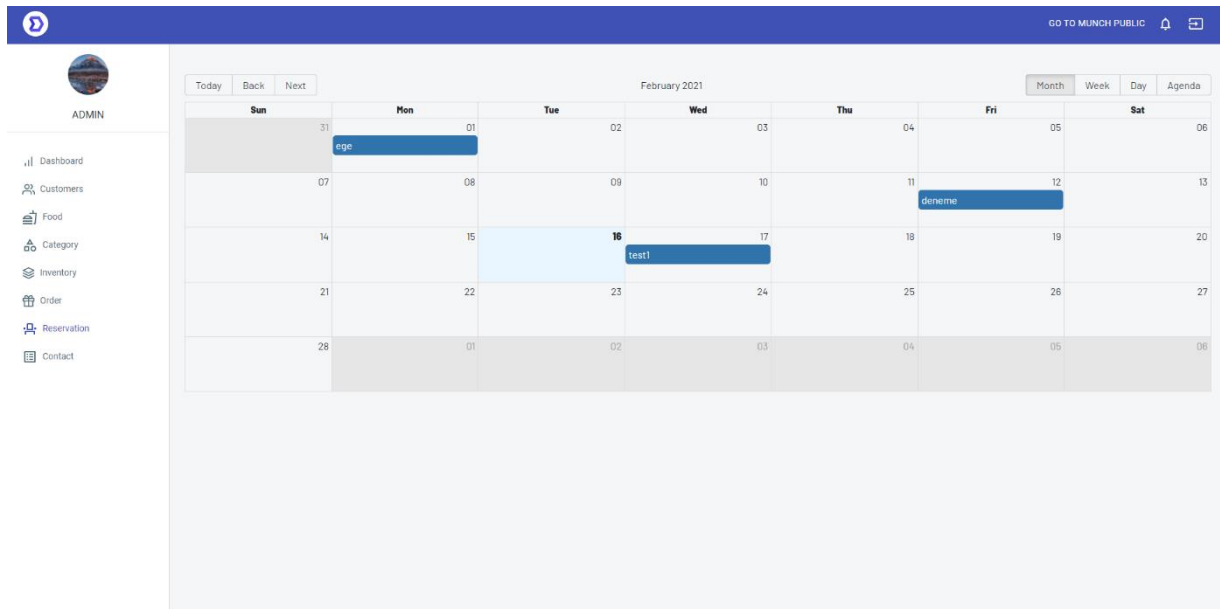
The screenshot shows the Admin Contact Page. The sidebar is the same as in the previous screenshot. The main content area has a "Contact Information" section with the subtext "The information can be edited". It contains form fields for "phone \*" (+1 (880) 521-3401), "wifi name \*" (munch1), and "wifi password \*" (1234munch). There is also an "Address \*" field (854 Lyme Avenue, Smeltertown, Georgia, 846) and an "ADD WORK HOURS" button. Below these is a table for work hours with columns: Day, Hour, and Action.

Day	Hour	Action
Sunday	19.00-20.00	<a href="#">DELETE</a>
Monday	10.00-18.00	<a href="#">DELETE</a>

At the bottom right of the main content area is a "SAVE DETAILS" button.

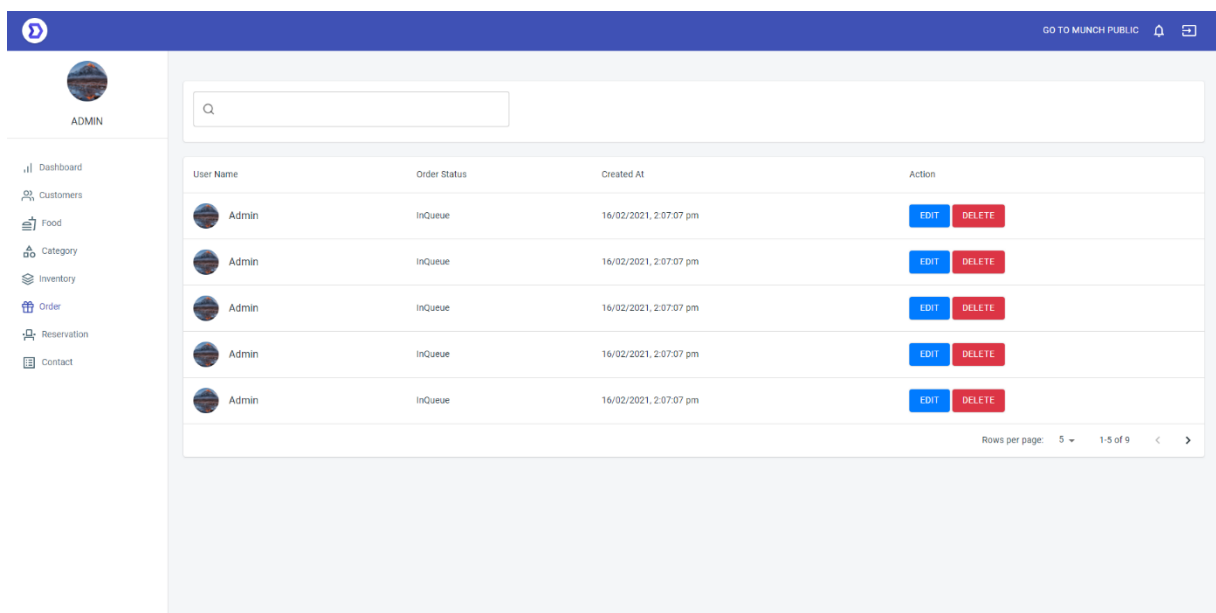
### 6.2.16. Admin Reservation Page

You can access this page from the sidebar. You can see all created reservations on this page as a calendar. You cannot edit this page.

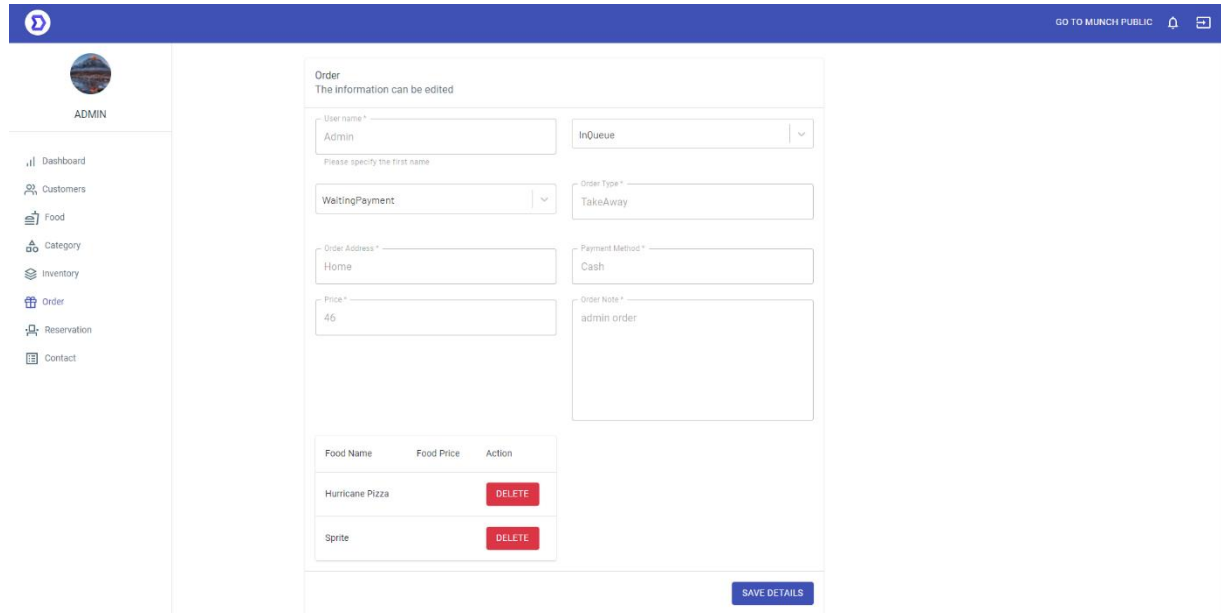


### 6.2.17. Admin Order Page

You can access this page from the sidebar. You can edit or delete orders.



In order edit page you can edit order status, payment status or you can delete food in order.



**Order**  
The information can be edited

User name \*  
Admin

Please specify the first name

Order Type \*  
InQueue

Order Address \*  
Home

Payment Method \*  
Cash

Price \*  
46

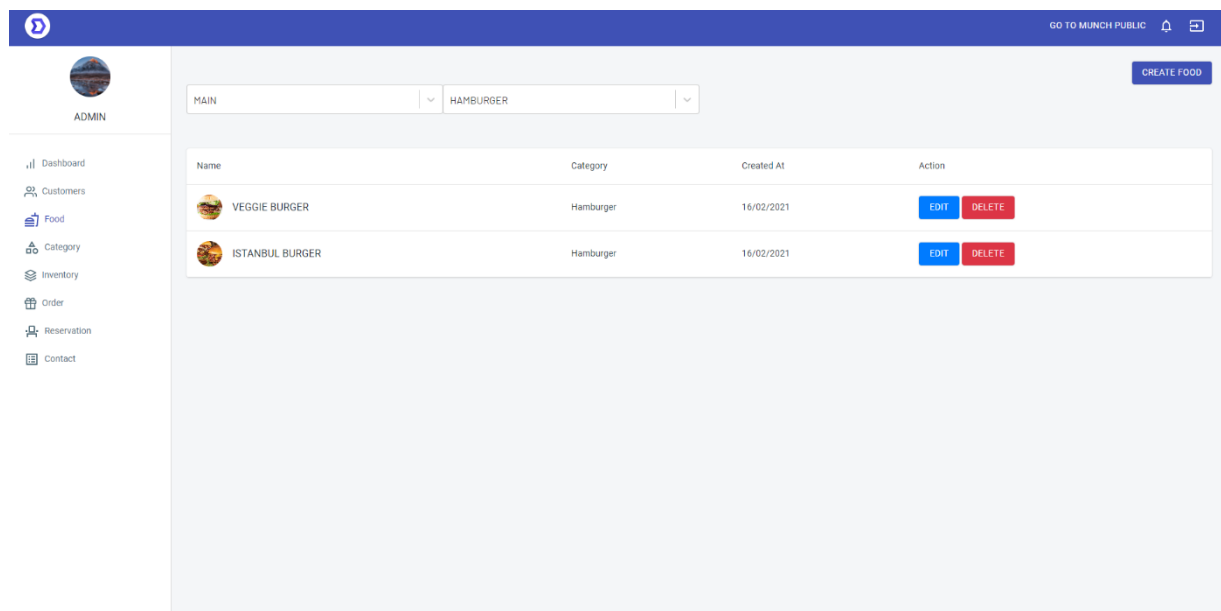
Order Note \*  
admin order

Food Name	Food Price	Action
Hurricane Pizza		DELETE
Sprite		DELETE

SAVE DETAILS

### 6.2.18. Admin Foods Page

You can access this page from the sidebar. You must select a top category and after that selection, you must select a category according to the top category. Then food list will be shown on the page. You can edit or delete listed foods or create new food.



CREATE FOOD

MAIN HAMBURGER

Name	Category	Created At	Action
VEGGIE BURGER	Hamburger	16/02/2021	EDIT DELETE
ISTANBUL BURGER	Hamburger	16/02/2021	EDIT DELETE

You can edit images, ingredients, description, name, calorie, price on this page.

**ADMIN**

- Dashboard
- Customers
- Food
- Category
- Inventory
- Order
- Reservation
- Contact

**Category**  
The information can be edited

**Name \***  
Veggie Burger

**Rating Average \***  
2

**Price \***  
15

**Preparation Time \***  
15

**Calorie \***  
391

**Secret Category \***  
NO

**Description \***  
This veggie burger recipe is both vegetarian and vegan but it has bread in the mix so it isn't gluten-free.

**Secret Category \***  
Hamburger

**ADD INGREDIENT**

Ingredient	Ingredient Quantity	Action
Meat	220	DELETE
Tomato	20	DELETE

**SAVE DETAILS**

### 6.3. TEST PLAN

There will be a two tests phase. First one is from 18/12 to 25/12. I will try to test and fix bugs in API. After that, second test phase will begin in 14/01 and it will continue until 20/01. All application will be tested in this phase.

### 6.4. MAINTENANCE PLAN

I am going to set daily backups on server. If something goes wrong, I will have chance to recovery it.

## CONCLUSION AND DISCUSSION

As an overall conclusion, I would say that I am quite happy to make real this project as I thought. The latest version of the project will be able to quite useful with some little change and addition. I learned technologies about web programming such as NodeJS, ReactJS and I used a NoSQL database in the backend which is MongoDB. I tried to implement the MVC design pattern to my project, backend created with model and controller. I used JavaScript back to front. It was a different experience for me because the first time I create a full-stack project by myself. I believe this project can be useful in real life with some additions. I also implement an algorithm that is a text similarity checker with the dot product. That was a new thing for me, and I also enjoy it when I implement it. I also implement a suggestion algorithm. That project taught me to work in the tidiest way and if there is a work partner in this project it would be better than that moment for me. This project took me through the various phases of project development and gave me a real insight into the world of software engineering. The joy of working and the thrill involved while tackling the various problems and challenges gave me a feel of the developer's industry.

## REFERENCES

- 1 <https://expressjs.com/>
- 2 <https://www.npmjs.com/>
- 3 <https://reactjs.org/>
- 4 <https://react-redux.js.org/>
- 5 <https://docs.mongodb.com/manual/>

**EKLER**

EK I – <https://github.com/legemutlu/munch>



**BIOGRAPHY**

LEVENT EGE MUTLU