



By Andrin, Aryan, Gabriel, Kenneth & Max

Table of Content

1	Introduction	3
2	Members	3
3	Procedure	4
3.1	To-Do List	4
3.2	Timetable	5
3.3	Meetings.....	6
	12.06.2024	6
	13.06.24	7
	14.06.24	8
	19.06.24	8
	20.06.24	8
	21.06.24	8
	25.06.2024	9
	26.06.2024	9
4	Instructions	10
4.1	Setup.....	10
	Cloning the Repository.....	10
	Accessing Docker	10
	Running both programs	10
4.2	How to use our web application.....	11
4.2.1	Instruction	11
4.2.2	Helpful Links	15
4.3	Testing.....	16
5	Graphics	17
5.1	Mock-up	17
5.2	System overview	18
5.3	ERD	18

1 Introduction

Welcome to our documentation about the REST-Foods fullstack project by Andrin, Aryan, Gabriel, Kenneth and Max.

In this project we created a fullstack web application for a restaurant called REST-Foods. The web application offers several menus, the function to create reservations and as admin you have access to the dashboard, which provides everything administrative (editing the menus and reservations).

With our database called "team7" you can work with the data of the restaurant and offer a dynamic web application to the customers.

We hope that you can find use in our project. For more information, see the documentation.

2 Members

Frontend Team:

- Andrin Ramon Rüegg
- Gabriel Arocha Garcia Villalobos

Backend Team:

- Aryan Bisen
- Kenneth Chi Nam Dang
- Maximilian Nöthe

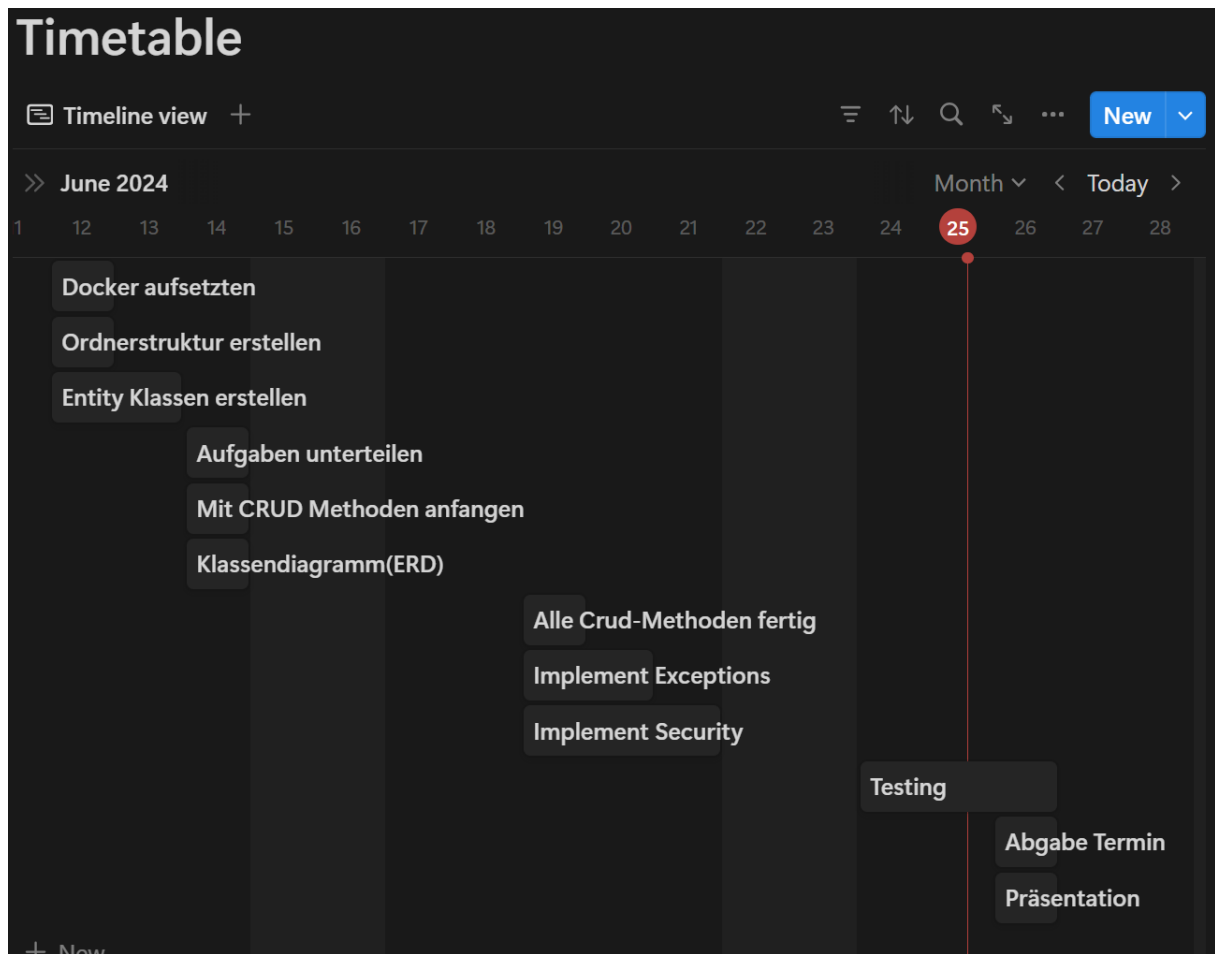
3 Procedure

3.1 To-Do List

Task	Stack	Date	Status
Backend aufsetzen	Backend	12.06.	Done
Structure einzelner Pages erstellen	Frontend	13.06	Done
Columns der Tabellen definieren	Backend	13.06	Done
Security anpassen, damit mit email anstatt u	Backend	13.06	Done
Make API Fetch to Backend	Frontend	13.06	Done
Make Reservation Page (Design)	Frontend	13.06	Done
Make main page	Frontend	13.06	Done
Folderstructure erstellen	Backend	13.06.	Done
Structure der Domains erstellen	Backend	13.06.	Done
Zeitplan erstellen	Admin	13.06.	Done
Aufgaben unterteilen	Backend	13.06.	Done
Struktur der Datenbank erstellen (DDL)	Backend	13.06.	Done
Layers von Dishes erstellen	Backend	13.06.	Done
Layers von Reservation erstellen	Backend	13.06.	Done
Layers von User erstellen	Backend	13.06.	Done
Layers von Role erstellen	Backend	13.06.	Done
Layers von Authority erstellen	Backend	13.06.	Done
Data.sql / Testdaten erstellen (DML)	Backend	13.06.	Done
ERD erstellen	Backend	13.06.	Done
Datentypen bearbeiten der Fields	Backend	14.06	Done
JWT in Backend konfigurieren	Backend	14.06	Done
→ UserRepository & UserService bearbeiten	Backend	14.06	Done
Finish ReservationController	Backend	19.06	Done
Create new table "Table"	Backend	19.06	Done
Implement layers for table domain	Backend	19.06	Done
Create joins with table and reservation	Backend	19.06	Done
Implement all needed exceptions into Dish S	Backend	19.06	Done
Implement all needed exceptions into Resen	Backend	19.06	Done
Implement Javadoc	Backend	25.06	Done
Implement exceptions into Reservationservi	Backend	25.06	Done
Update ReservationController with exceptio	Backend	25.06	Done
Make Dashboard Page	Frontend	tbd	Done
Make validation(s)	Both	tbd	Done
Make general desgin better	Frontend	tbd	In progress
Improve general code	Both	tbd	In progress
Anpassungen im Controller für Übereinstim	Backend		Done
Make valid image links on database for dishe	Backend		Done
Push your work	Backend		In progress
Communicate when work is ready to pull	Backend		In progress
Work	Both		In progress
make crud	Frontend		Done
make code better	Both		In progress
make design better	Frontend		In progress
Finish To-Do List	Admin		Done
Finish timetable	Admin		Done
Create instruction on how to use our web ap	Admin		Done
Testing	Both		Done
Implement mockup into documentation	Frontend		Done
Create a system overview	Backend		Done
Swagger	Backend		Done

3.2 Timetable

v



3.3 Meetings

12.06.2024

Current Activity of All Team Members

Frontend:

- **Folder Structure:**
 - Created and organized folder structure.
- **Mock-up:**
 - Completed the design and layout of the following components:
 - Header for reservation in the navbar
 - Landing page
 - Homepage
 - Display of all dishes
 - Detail page to show individual dishes with details
 - Reservation functionality (Delete function for users is ongoing)
 - Dashboard

Backend:

- **Setup and Configuration:**
 - Set up the backend environment.
 - Created data.sql to configure the database.
- **Domain Creation:**
 - **Dish Domain:**
 - Implemented 3 layers (Controller, Service, Repository)
 - **Role Domain:**
 - Implemented 2 layers (Service, Repository)
 - **Authority Domain:**
 - Implemented 2 layers (Service, Repository)
 - **User Domain:**
 - Implemented 2 layers (Service, Repository)
 - **Reservation Domain:**
 - Implemented 3 layers (Controller, Service, Repository)

Problems:

- **Backend Setup Issue:**
 - Database name conflict with Docker container name.

Goals for the next few days:

- **Dashboard:**
 - Ensure that the dashboard is only accessible and usable by admins (GET request).
- **Reservation:**
 - Allow everyone to view the reservation page.
 - Implement a pop-up for user login when creating a reservation.
 - Make sure that the work at the frontend is not too heavy, if this happens take measures.
 - Users with the admin role should see the dashboard button; others should not.

13.06.24

Current activity of all team members

Frontend:

- Functionality of the individual pages
- Configuration of the individual pages

Backend:

- Create ERD
- Create fields for the individual entities

Goals for the next few days

- Registration via Postman
- Implement registration functionality in the frontend
- Create ERD and structure of the database

14.06.24

Frontend:

- Make dynamic fetching

Backend:

- Implement HTTP-Methods for dish
- Begin with Exceptions

19.06.24

The meeting was not held due to the information afternoon.

Frontend:

- Make pages responsive

Backend:

- Create Restaurant Table entity
- Implement HTTP-methods for reservation
- Add exceptions to dish service
- Implement JOINS between reservation and restaurant table

20.06.24

- Meeting was not held because all orders were clear, and the problems did not affect the whole team

21.06.24

Frontend:

- Login to be implemented using JWT
- Aryan support frontend with implementation of security

Backend:

- Max update database for image links

25.06.2024

Frontend:

- Fix CORS Policy problem
- Finish Dashboard
- Improve CRUD and Design

Backend:

- Add Validation (reservation and dish fields)
- Add Swagger documentation
- Start implementing reservation logic
- Add Javadoc

26.06.2024

Frontend:

- Begin with Testing
- Finish logic for Reservation

Backend:

- Add more dishes to data.sql
- Start Testing
- Adjust reservation logic

4 Instructions

4.1 Setup

Cloning the Repository

- To begin, clone the repository to your local machine.

Accessing Docker

- Create a docker container by entering the following command in CMD.

```
docker run -d -p 5432:5432 -e POSTGRES_PASSWORD=password --name team7 postgres
```

- Make sure you run your docker container.

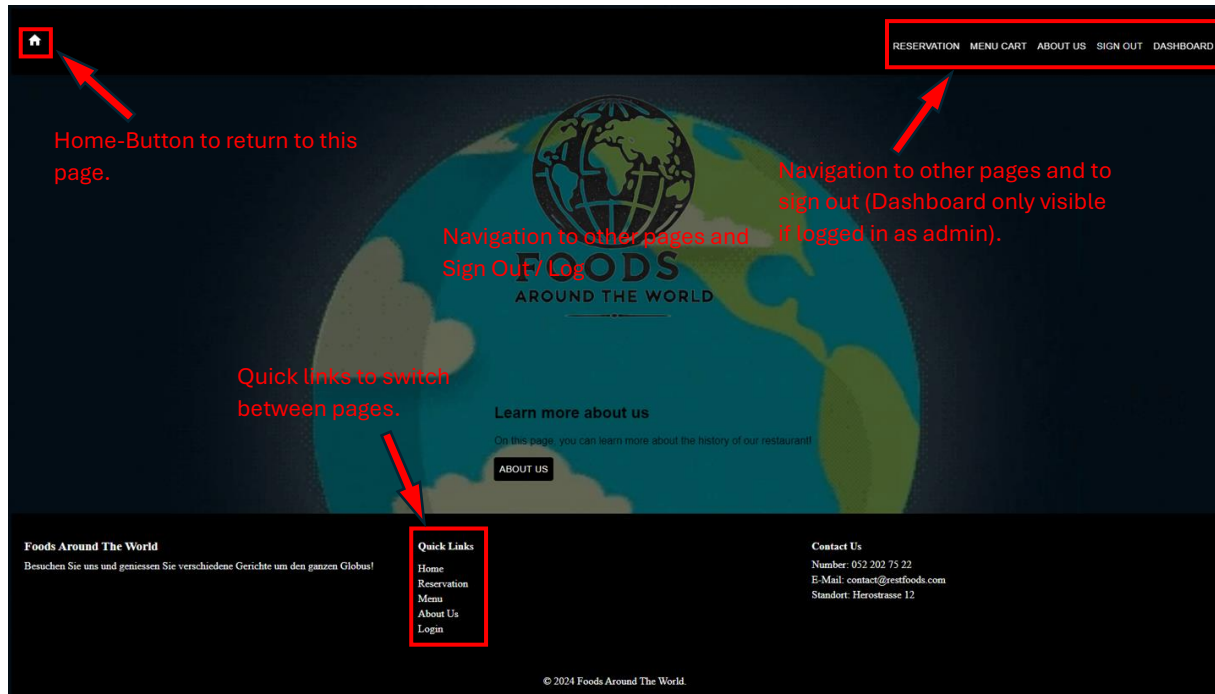
Running both programs

- Backend: Open IntelliJ for the backend and run the program.
- Frontend: Go to my-app directory and type npm run dev in your Terminal.

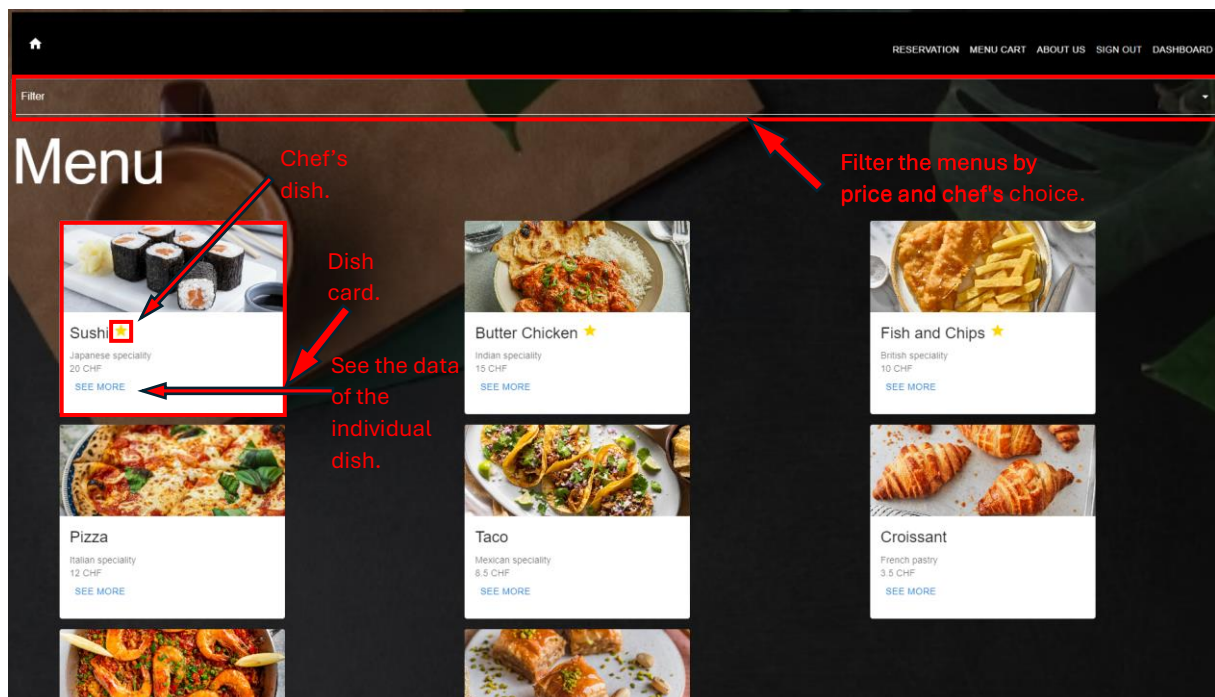
4.2 How to use our web application

4.2.1 Instruction

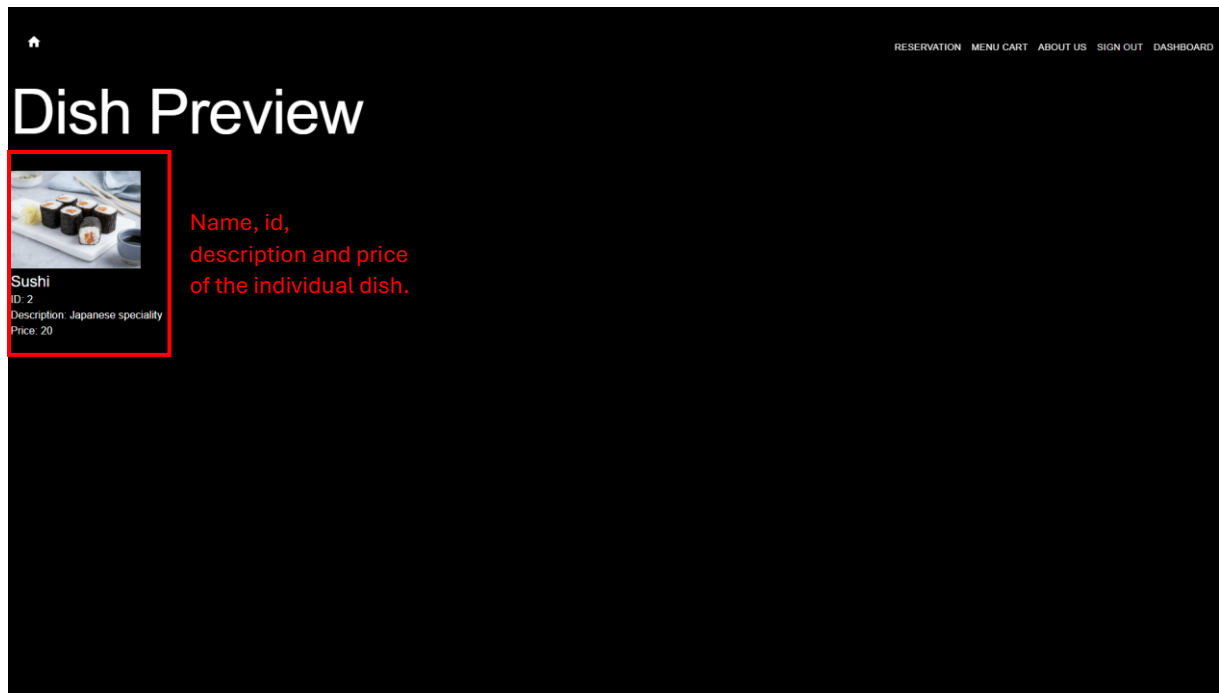
4.2.1.1 Homepage



4.2.1.2 Menu Card - Page



4.2.1.3 Individual dish – Page



4.2.1.4 Login – Page

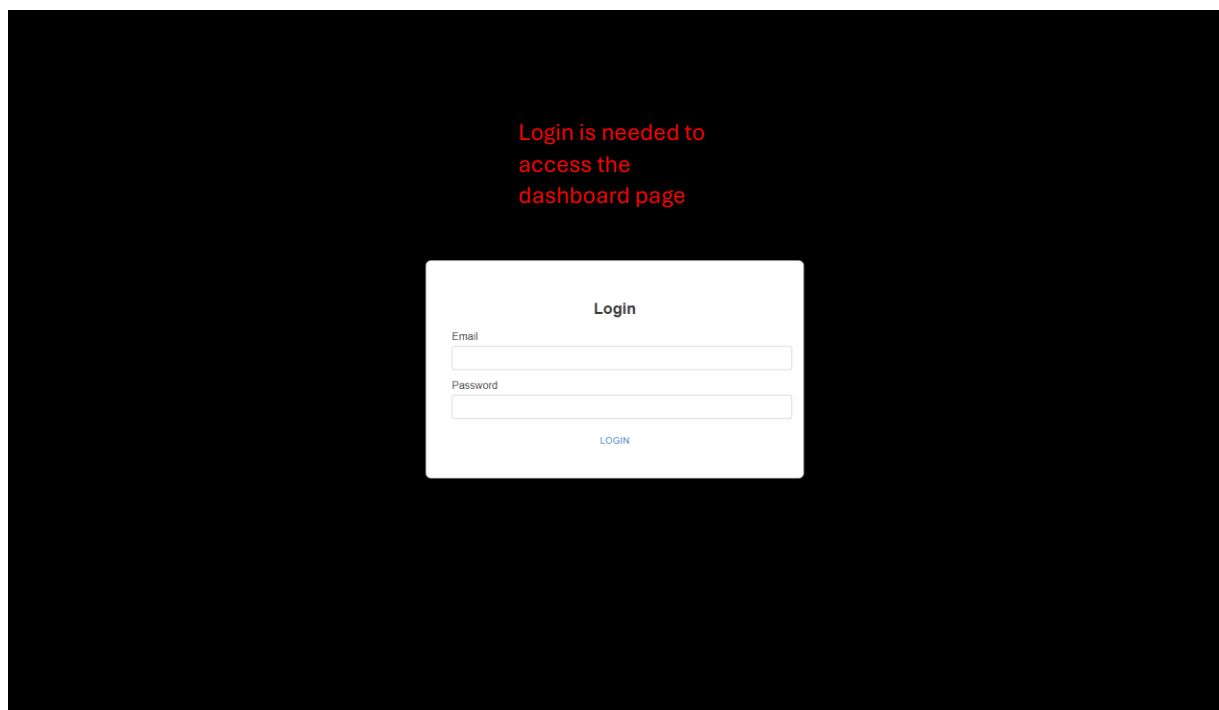
Use the following credentials:

User Credentials:

- Username: user@user.com
- Password: 1234

Admin Credentials:

- Username: admin@admin.com
- Password: 1234



Login is needed to
access the
dashboard page

Login

Email

Password

LOGIN

4.2.1.5 Dashboard – Page

Dashboard

Dishes

Menu ID Actions ADD

Pizza	1	DELETE PREVIEW UPDATE
Sushi	2	DELETE PREVIEW UPDATE
Taco	3	DELETE PREVIEW UPDATE
Croissant	4	DELETE PREVIEW UPDATE
Paella	5	DELETE PREVIEW UPDATE
Butter Chicken	6	DELETE PREVIEW UPDATE
Fish and Chips	7	DELETE PREVIEW UPDATE
Baklava	8	DELETE PREVIEW UPDATE

Menu names Menu id's CRUD operations for dishes

Reservation

ID: Description: Price:

Reservations	ID	Actions	ADD
2024-06-26T13:00:00	1	DELETE PREVIEW UPDATE	
2024-06-26T15:00:00	2	DELETE PREVIEW UPDATE	
2024-06-26T18:00:00	3	DELETE PREVIEW UPDATE	
2024-06-26T15:00:00	4	DELETE PREVIEW UPDATE	
2024-06-26T19:00:00	5	DELETE PREVIEW UPDATE	

Reservation times Reservation id's CRUD operations for reservation

4.2.1.6 About us – Page

About Us

Our Foundation

Foods Around The World was founded in 2024 by a group of friends who wanted to bring the taste of the world to the people of the world. We have been serving our customers with the best foods from around the world ever since. Our mission is to bring the world closer together through the power of food.

Our Philosophy

At Foods Around The World, we believe that food is more than just sustenance. Food is a way to bring people together, to share culture and tradition, and to create memories that last a lifetime. That's why we are committed to serving the best foods from around the world, made with love and care.

Join Us

We invite you to join us on our journey to bring the world closer together through the power of food. Whether you're a food lover, a world traveler, or just someone who enjoys a good meal, there's a place for you at Foods Around The World. Come visit us today and experience the taste of the world!

Information about "Foods Around The World"

Foods Around The World
Besuchen Sie uns und genießen Sie verschiedene Gerichte um den ganzen Globus!

Quick Links
Home
Reservation
Menu
About Us
Login

Contact Us
Number: 052 202 75 22
E-Mail: contact@grestfoods.com
Standort: Herostrasse 12

© 2024 Foods Around The World.

4.2.2 Helpful Links

- **Swagger Documentation:** [Swagger](#).

Feel free to explore and have fun with the program!

4.3 Testing

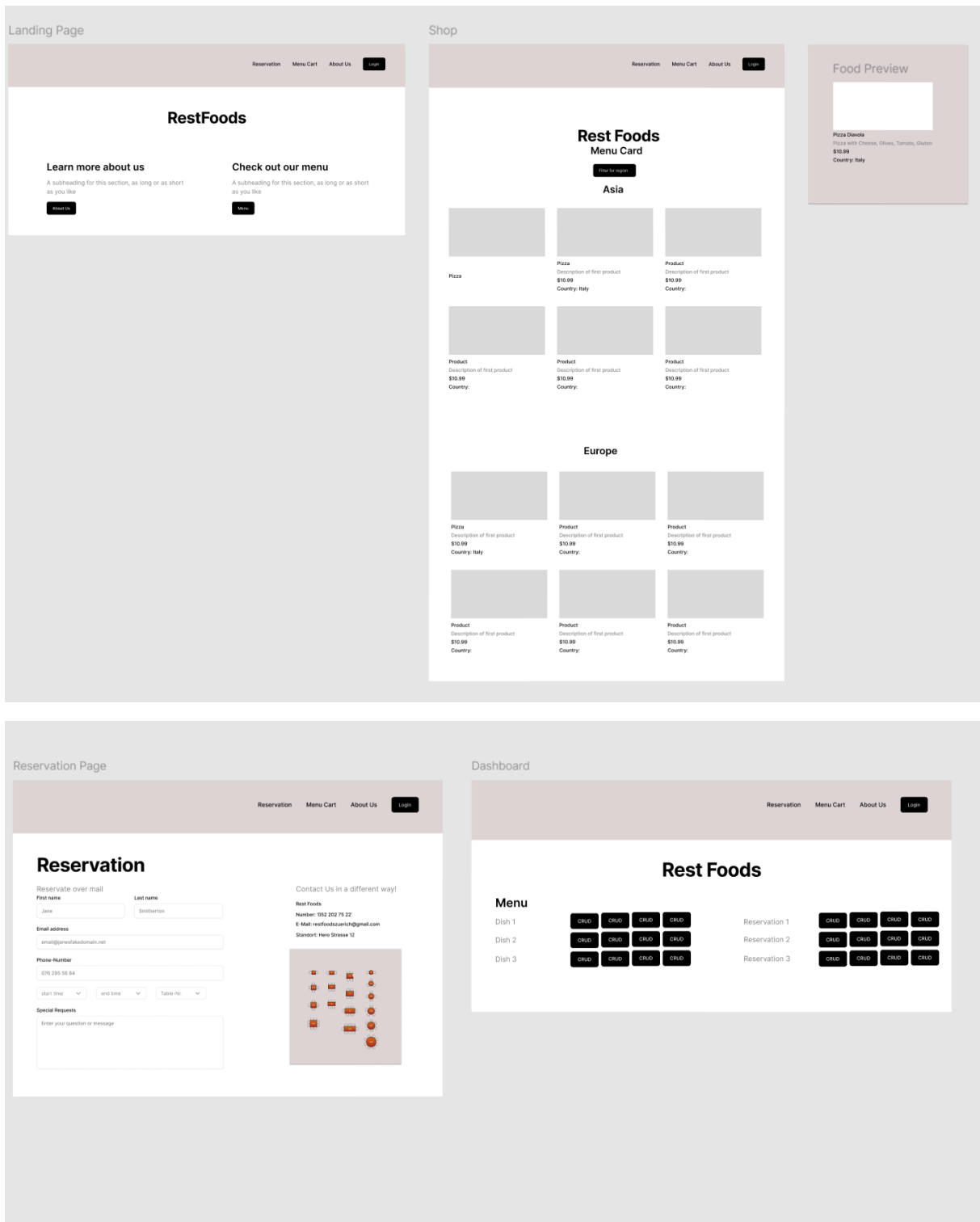
Function	True/Error
Print all Menus	True
Print by ID	True
Delete a Menu	True
Add a Menu	True
Header Links	True
Responsive	Both
Reservation input fields	True
Reservation submission	Error

Reservation Submit: We couldn't make a complete working submission. A user can enter his data, but nothing will happen on submitting.

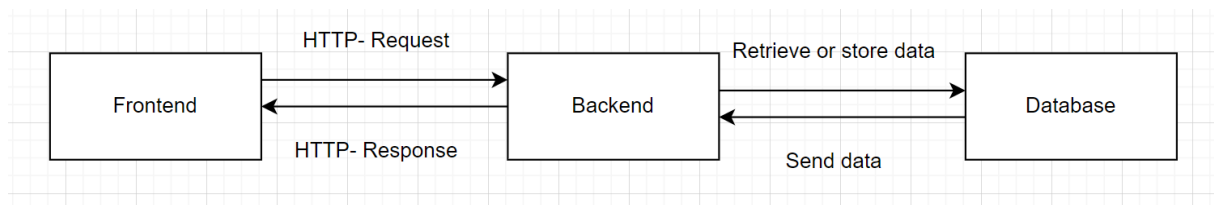
Responsiveness: The page is responsive to 90% but some objects look weird when changing sizes.

5 Graphics

5.1 Mock-up



5.2 System overview



5.3 ERD

