Restaurant

**Food ordering site**



Content

[Introduction 3](#_Toc115034044)

[Installation 3](#_Toc115034045)

[Authentication and Roles 3](#_Toc115034046)

[User registration and login 3](#_Toc115034047)

[Adding and Updating Food Cards 5](#_Toc115034048)

[Adding a new food card 5](#_Toc115034049)

[Updating and Deleting a Food Card 6](#_Toc115034050)

[Making orders 8](#_Toc115034051)

[View orders 10](#_Toc115034052)

[Frameworks and Libraries used in the project 11](#_Toc115034053)

## Introduction

This is a food ordering project, which was created for learning purposes.

Project includes the following features:

* User registration and login
* Shopping cart and ordering functionality
* Processing orders and tracking

## Installation

1. Run “Git clone https://github.com/MuradyanA/Restaurant.git”
2. CD .\Restaurant\Restaurant\
3. Run npm install
4. CD ..\Server\
5. Run npm install
6. Run npm run start
7. Visit <http://localhost:8080> address in your browser

## Authentication and Roles

There are two types of available users. First one is a regular user. All the users registered in the system become a regular user.

The administrator is a special type of a user, which is able to

1. View and Process all the orders registered in the site
2. Add, Edit and Disable food cards

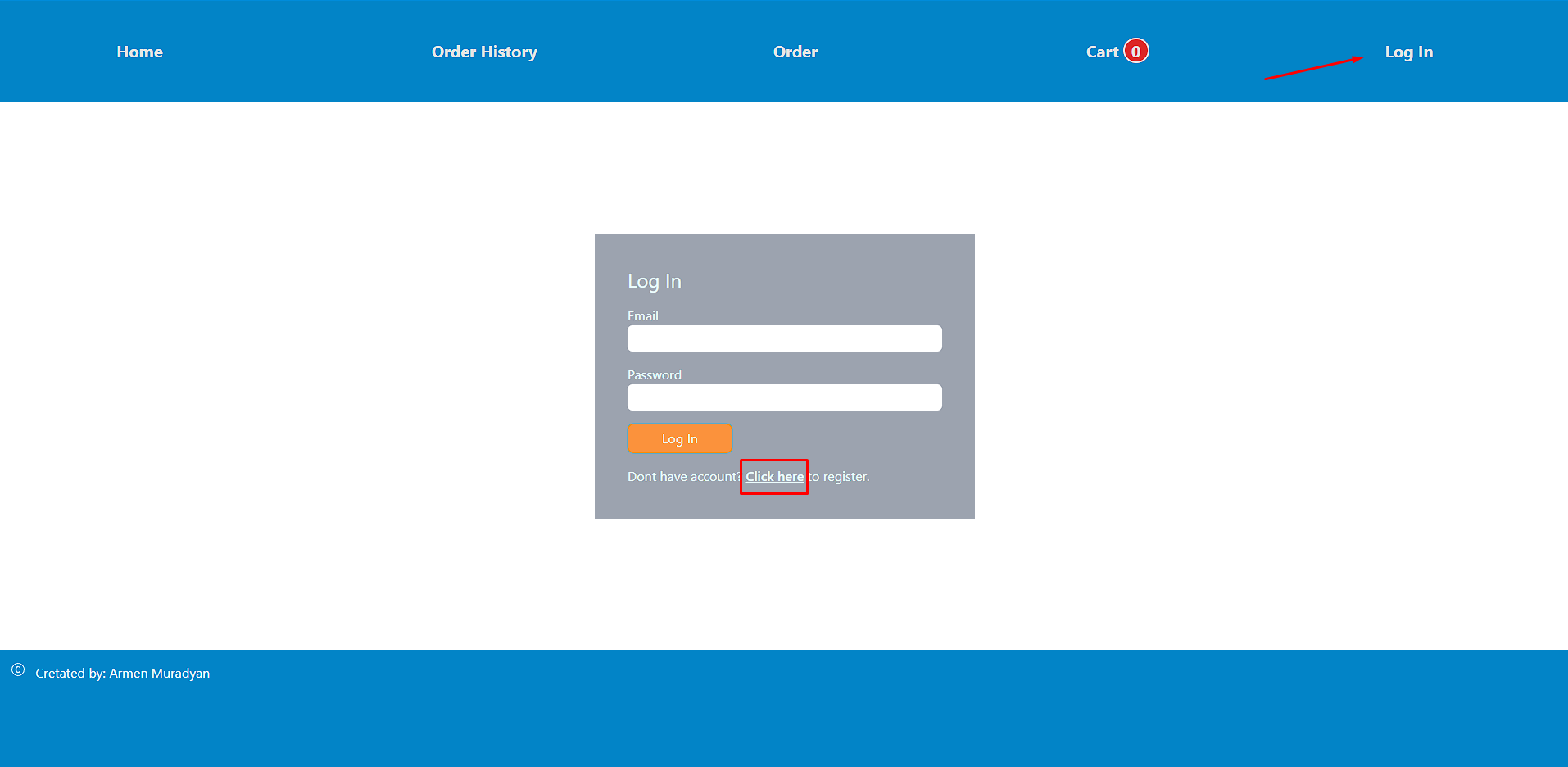
|  |
| --- |
| Database storage - Free computer icons **You can Log In as administrator by using the following credentials.**  **E-mail:** [marmhar@mail.ru](mailto:marmhar@mail.ru)  **Password:** 1234 |

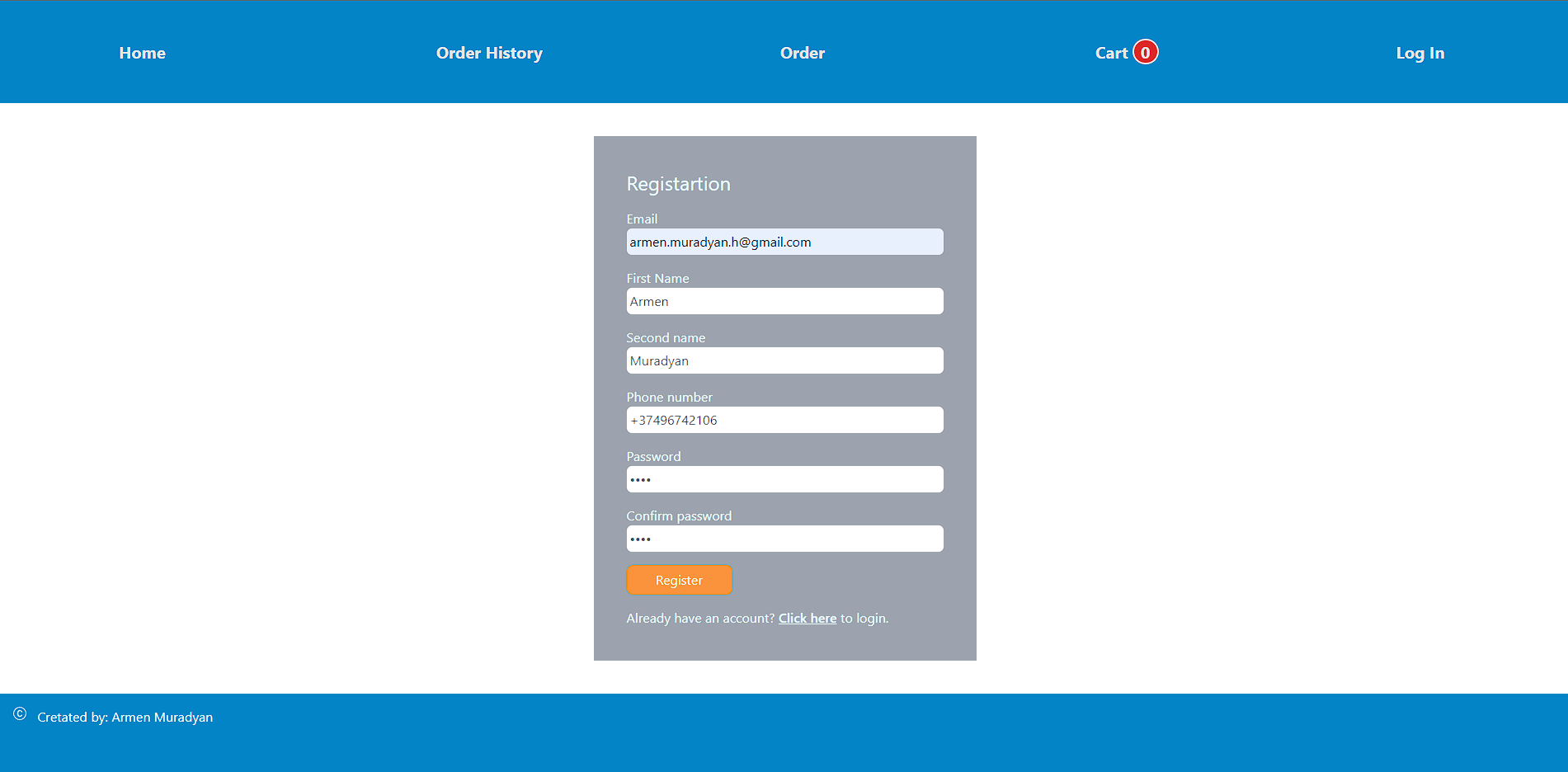
JWT tokens are used for the authentication. During the login a pair of tokens are generated. The first token is the “Access Token” which is set in the request headers, while performing HTTP request.

The second one is a “Refresh Token” which is stored in HttpOnly cookie and used to update “Access Token” when it’s lifetime expires (120 seconds). To prevent CSRF attacks, SameSite attribute was applied for the cookie. The request to update the “Access Token” is performed automatically before processing a regular request.

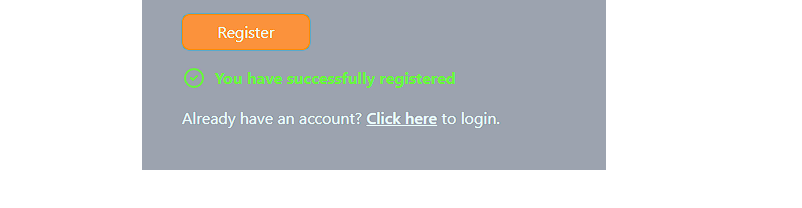
## User registration and login

To register a user visit the following page <http://localhost:8080/#/register> or you can visit Log In page and click the appropriate link.





After filling the required fields, press “Register” button. After the successful registration you will see the appropriate message on your screen.

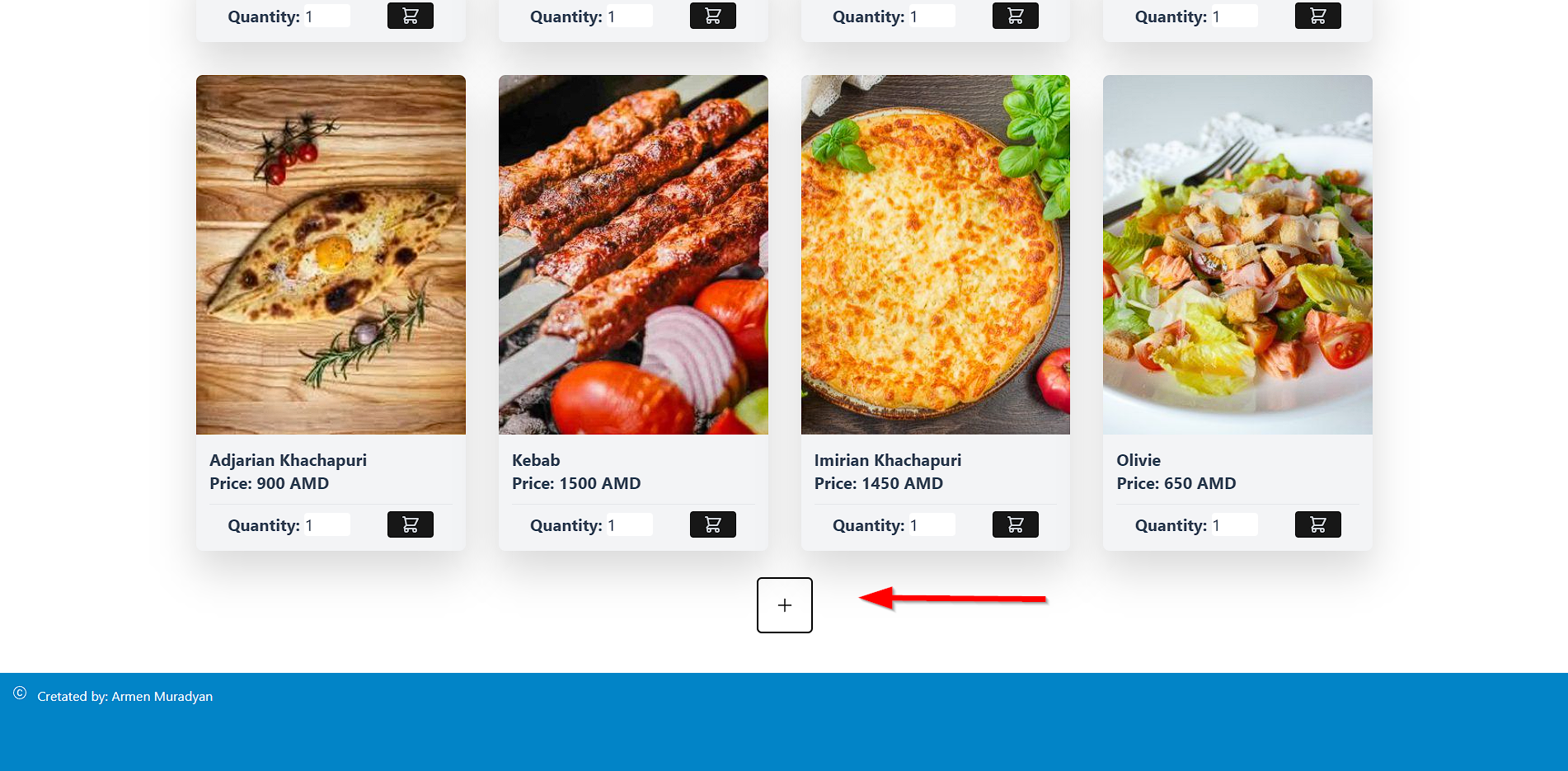


Now on the Log In page you are able to log into the system by using EMAIL and PASSWORD.

## Adding and Updating Food Cards

### Adding a new food card

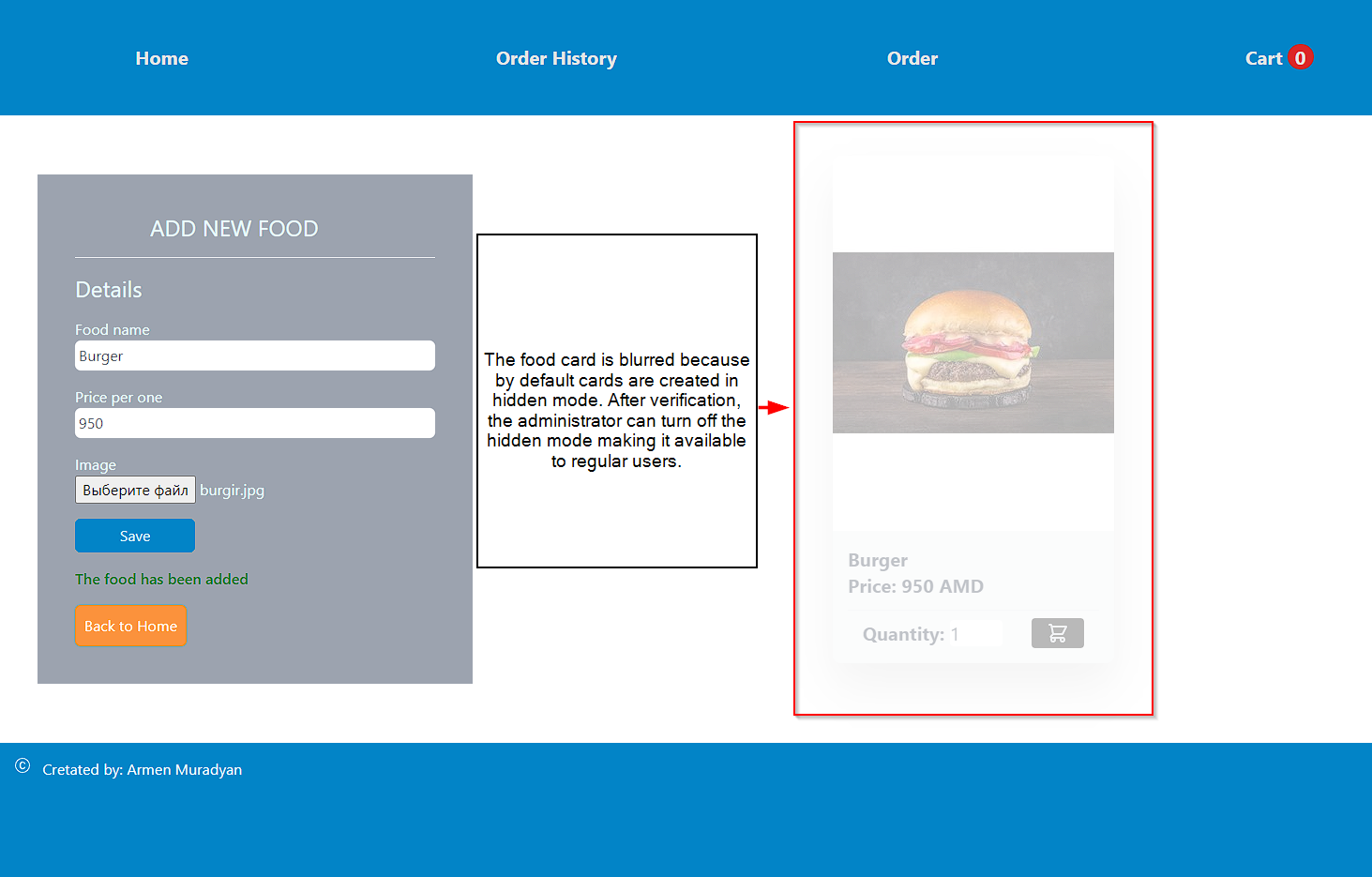
To add and update food cards user must have administrator privileges. If you are logged in as an Administrator the following button is available in the home page.



By pressing the button, you will be redirected to the page where you can add a new food card.

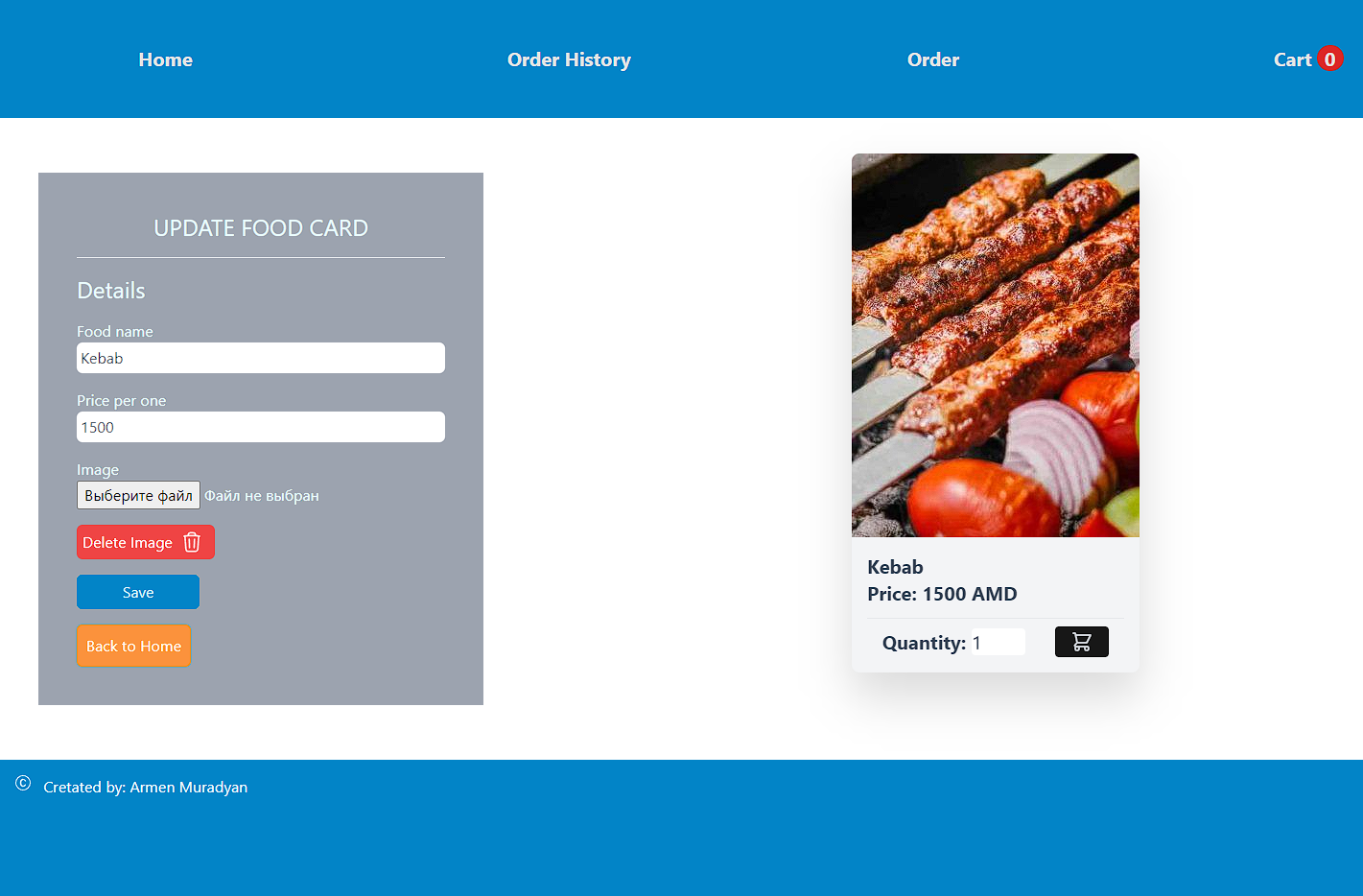
After filling the form fields and choosing the appropriate image the new food card will be added by pressing the “Save“ button (If the image isn’t provided the default image will be shown).





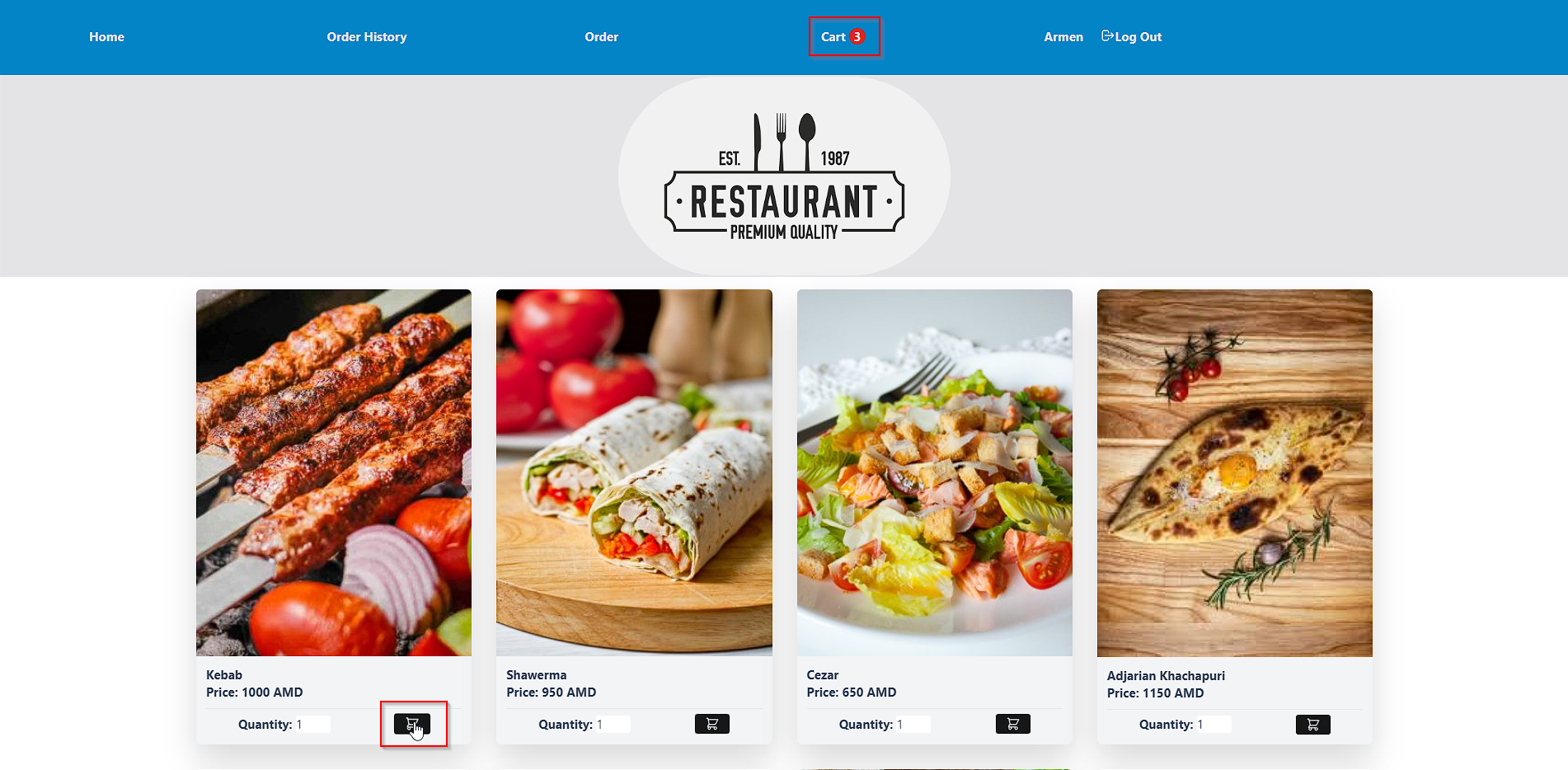
### Updating and Deleting a Food Card

|  |  |
| --- | --- |
| To delete or edit the food card use the toolbar which is shown while pointing on food card using the mouse.  The toolbar is only available for administrators.   * By clicking the eye icon, you can toggle the hidden mode. * The pen icon used for editing the food card. After clicking the mentioned icon, the administrator will be redirected to the page where the editing form is available. The editing form is similar to the form used for adding a new food card. An additional button for deleting the current image is available in the form. The name of the food and the price are mandatory. If the user doesn’t provide a new image, the current image will be preserved. * To delete the food card, click the Trash icon. The application will ask for confirmation before deleting the food card. |  |



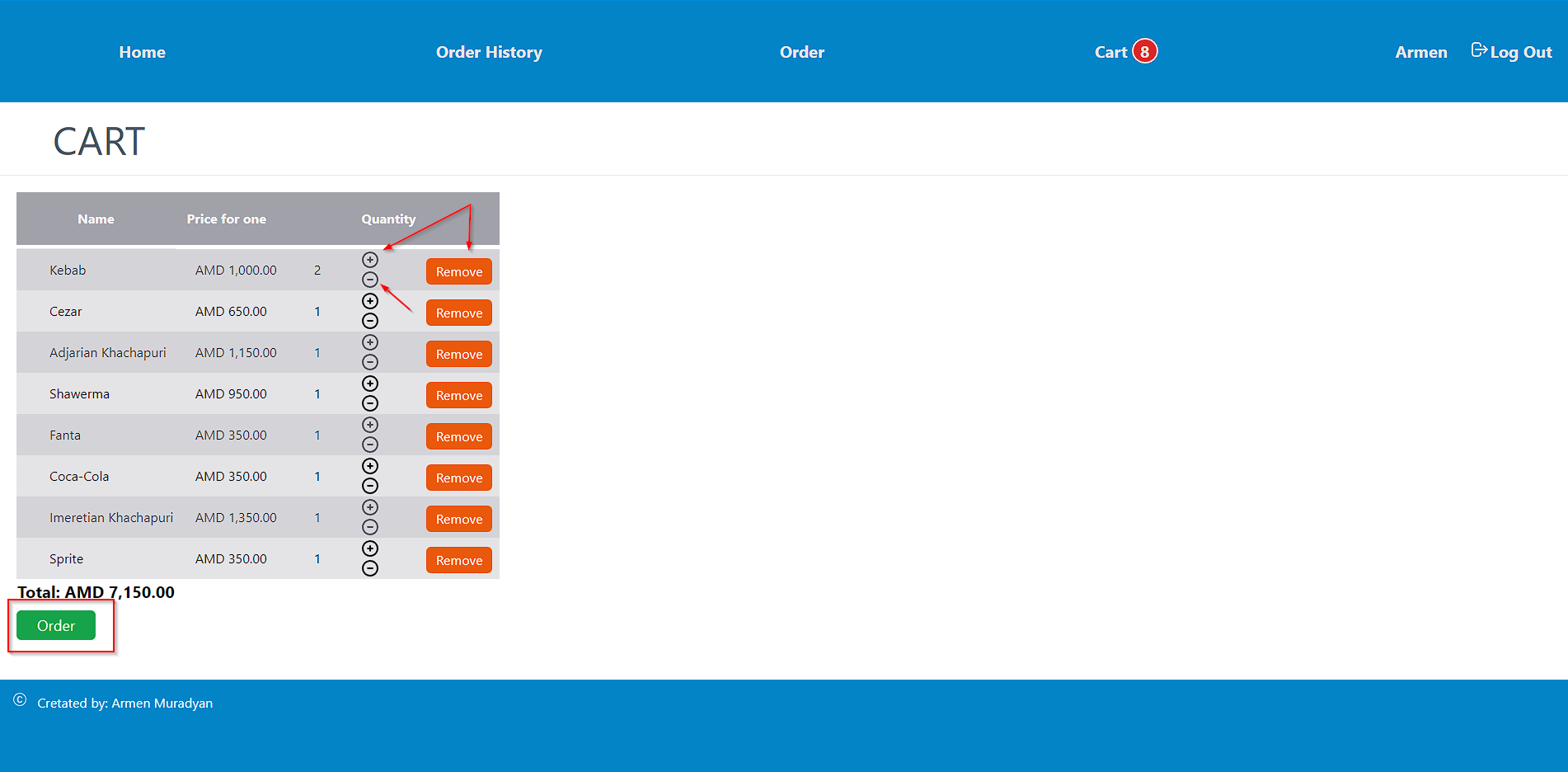
## Making orders

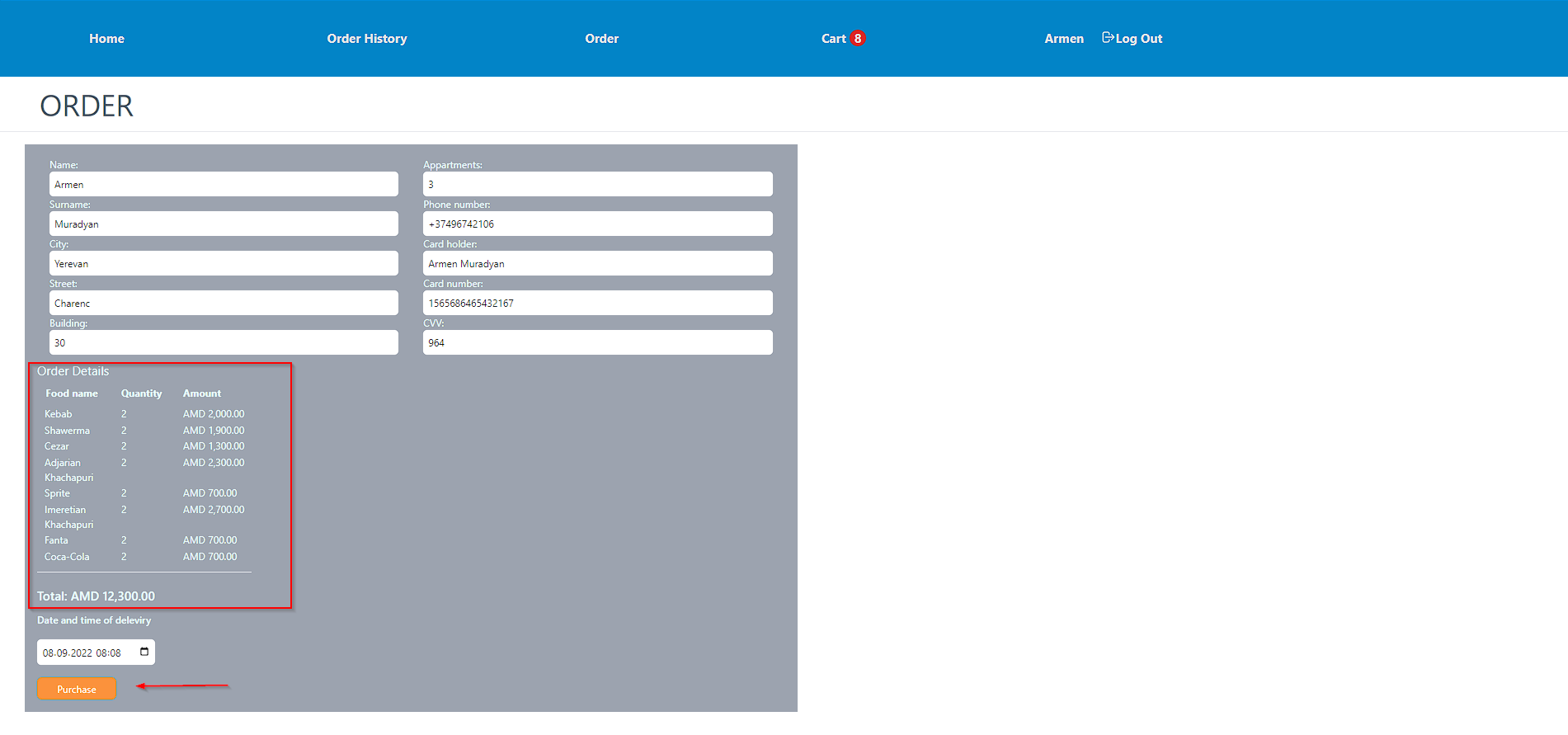
To see the food list, visit the home page <http://localhost:8080/#/>. To make an order first you need to add desired foods to your cart by clicking the cart icon. The quantity of the selected foods will be shown next to “Cart” menu item.



You can update the quantity of foods or delete them from the cart, by visiting the “Cart” page <http://localhost:8080/#/cart>.

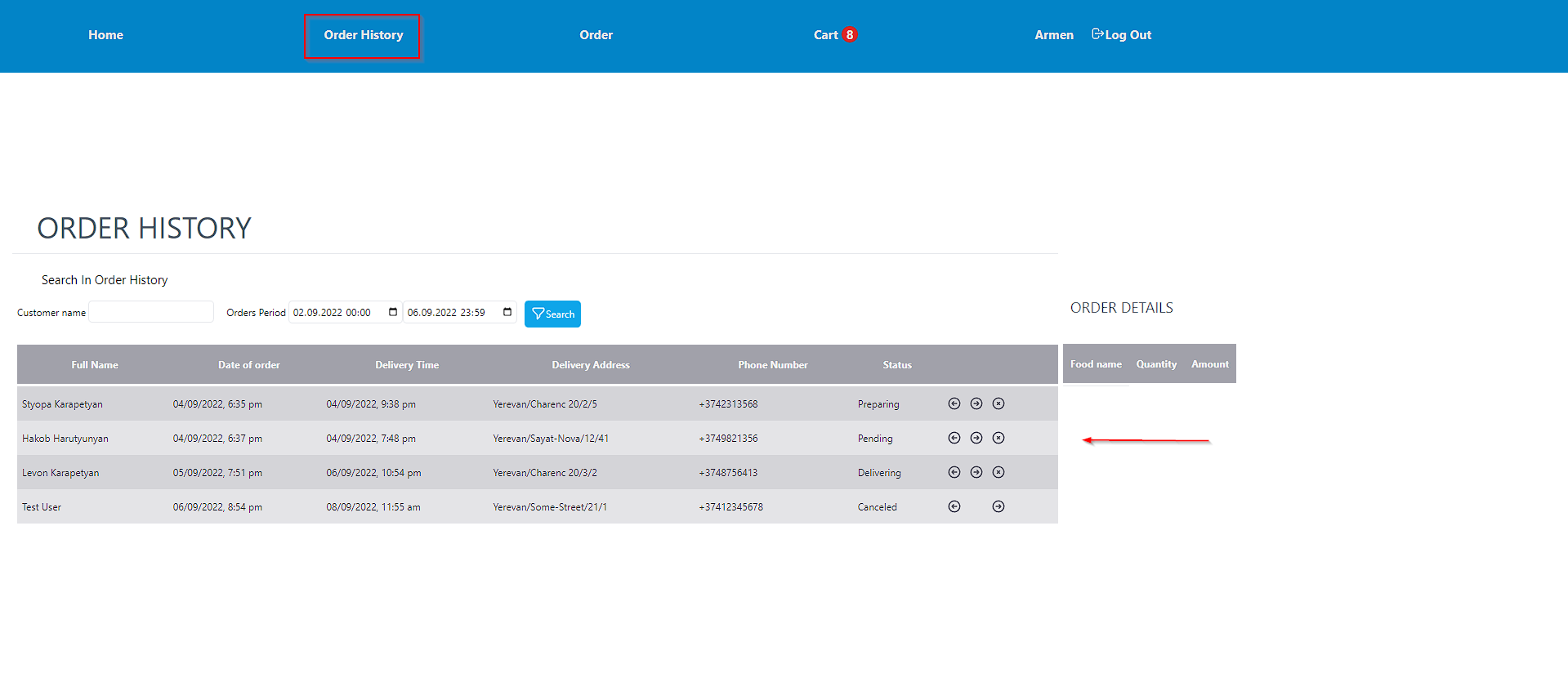
To order selected foods, click on the “Order” button or click on the “Order” menu item.

In “Order” page you can see the food you added in you cart. To create an order, fill the required fields and choose the date and time of delivery (Time and date of the order must not exceed 30 days from the ordering time) and click “Purchase” button.



## View orders

You can view your orders by clicking on the “Order History” menu item.

By default, you can see orders which were made during the last 3 days. You are able to filter orders by the date and time period they were created or by customer name.

The regular use will see only their own orders. User can cancel orders while they don’t have “delivering” status. 

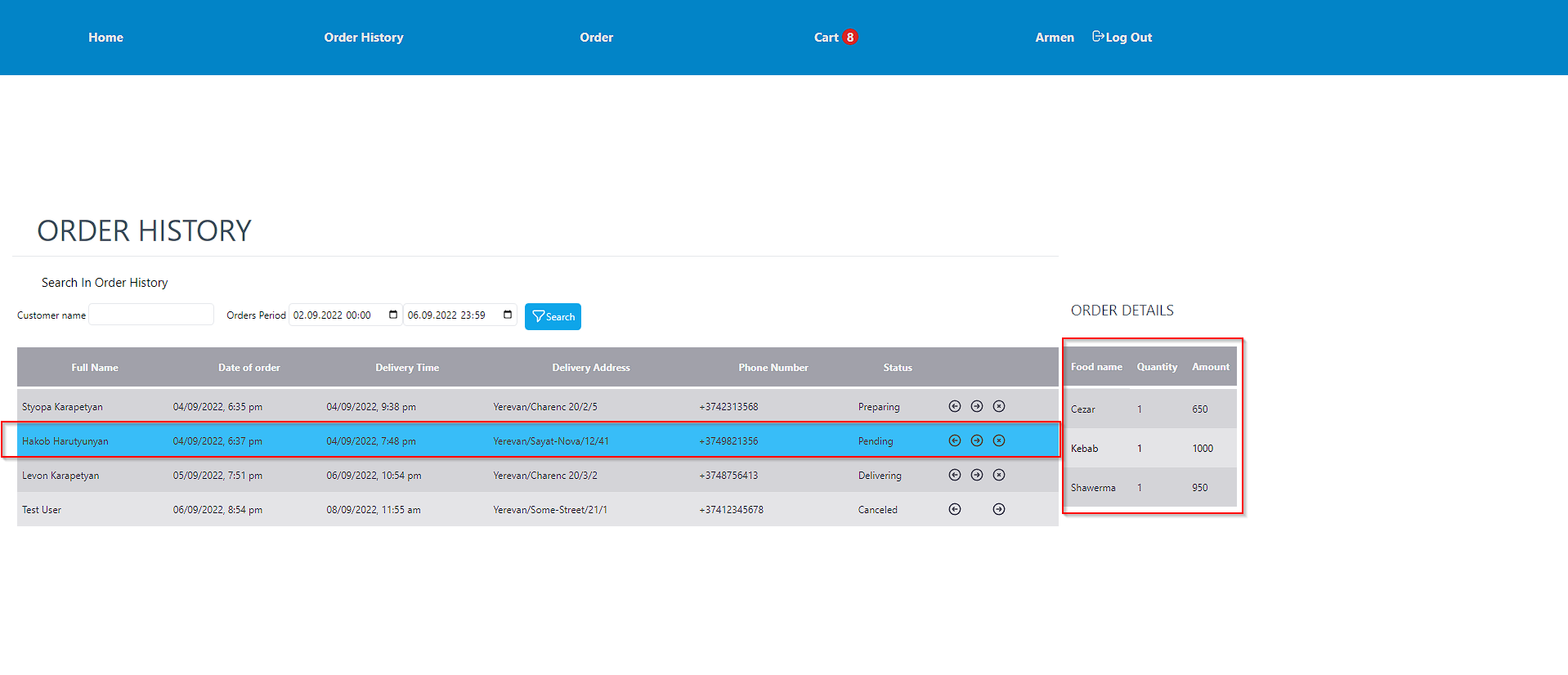
Administrators are able to view and process all the orders registered on the site. They can change status by clicking the arrow-left and arrow-right buttons  also they can cancel the order.

By pressing the arrow buttons the state will change in both directions in the following order

**Note**: Status won’t be changed by pressing left-arrow button if the current status is “Pending”. The same is true in case of clicking right-arrow when the status is “Finished”

1. Pending (The first status that order automatically get)
2. Preparing
3. Delivering
4. Finished

By clicking on the order, on the right side you can see the list of the food bought by the current order in the “Order Details” table. “Order Details” has the sticky position and it will be available on the screen while scrolling.



|  |
| --- |
| Database storage - Free computer icons **There are some orders available in the database for the period 20.08.2022 – 23.08.2022.** |

## Frameworks and Libraries used in the project

* VueJS
* ExpressJS
* Tailwind CSS
* Sequelize
* JWT (JSON Web Token)
* Bycript
* Sharp