# Frontend study project

## Intro:

In current project you need to implement frontend part of application for food delivery service.

[**Frontend study project**](#_55jtdbwsh337) **1**

[Intro:](#_a094gjn5nhch) 1

[Subject domain area](#_wegplhgvftr4) 1

[Authorization](#_9ux09njp07ek) 2

[Registration](#_fr6rwaw7hvnm) 2

[Profile](#_urhmpis5gq27) 3

[Main page](#_vptu3kkk8xi2) 4

[Menu item](#_cpuvqf4tkumd) 6

[Cart](#_2k2z4z7ockue) 7

[Orders](#_ks9mlfmppadh) 7

[Order details](#_4ygfxdwd6fik) 8

[Ordering](#_fdzfe52mcp0p) 8

## Subject domain area

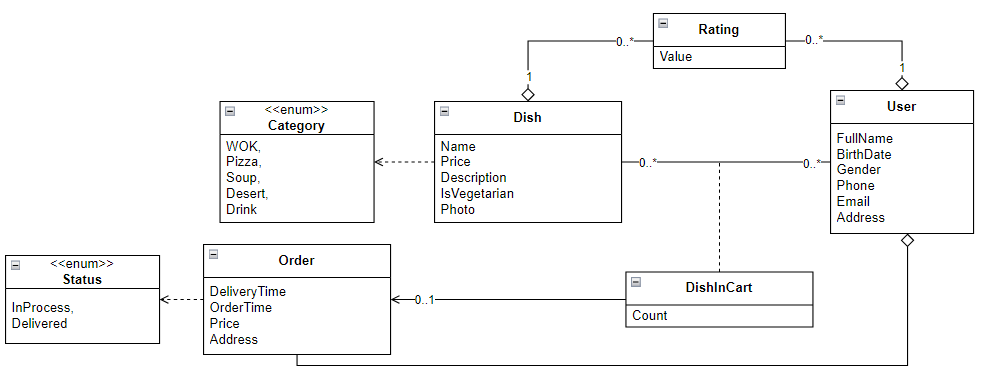
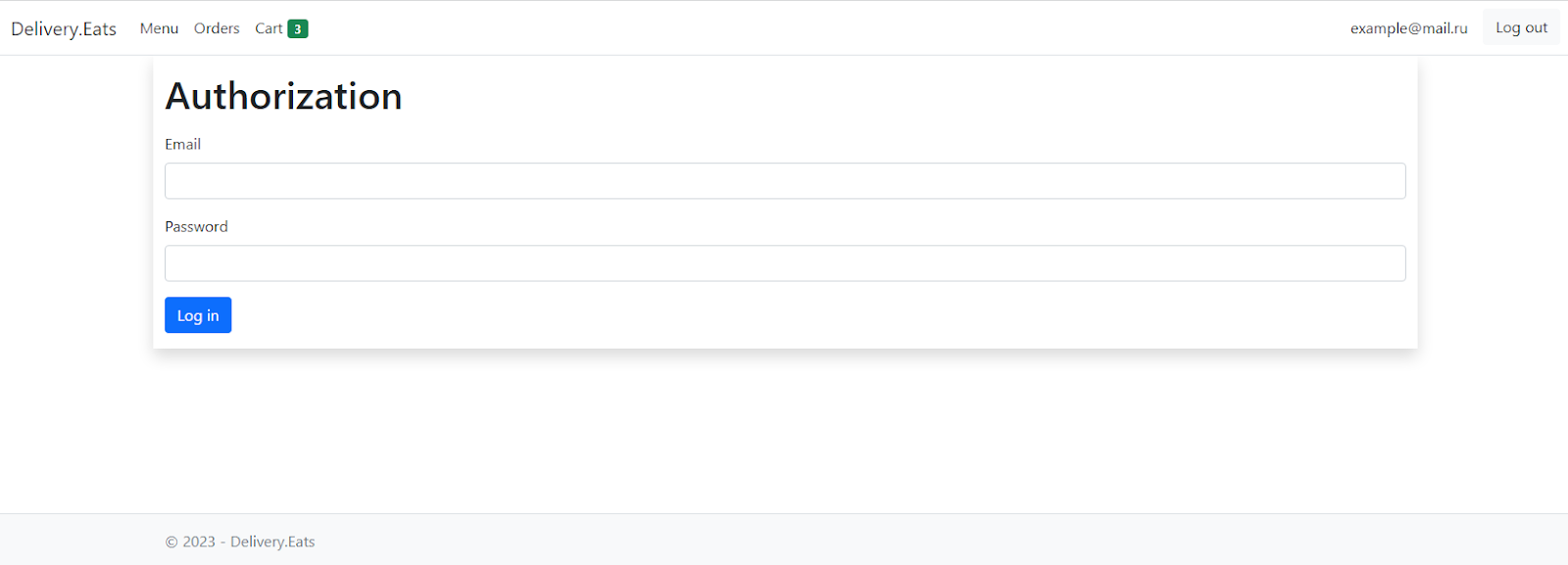


Figure 1 - Subject Domain Area

API Documentation - <https://food-delivery.int.kreosoft.space/swagger/index.html>

Since a lot of the functionality of the site is tied to the user, you need to provide a way to register users and their further authorization.

## Authorization

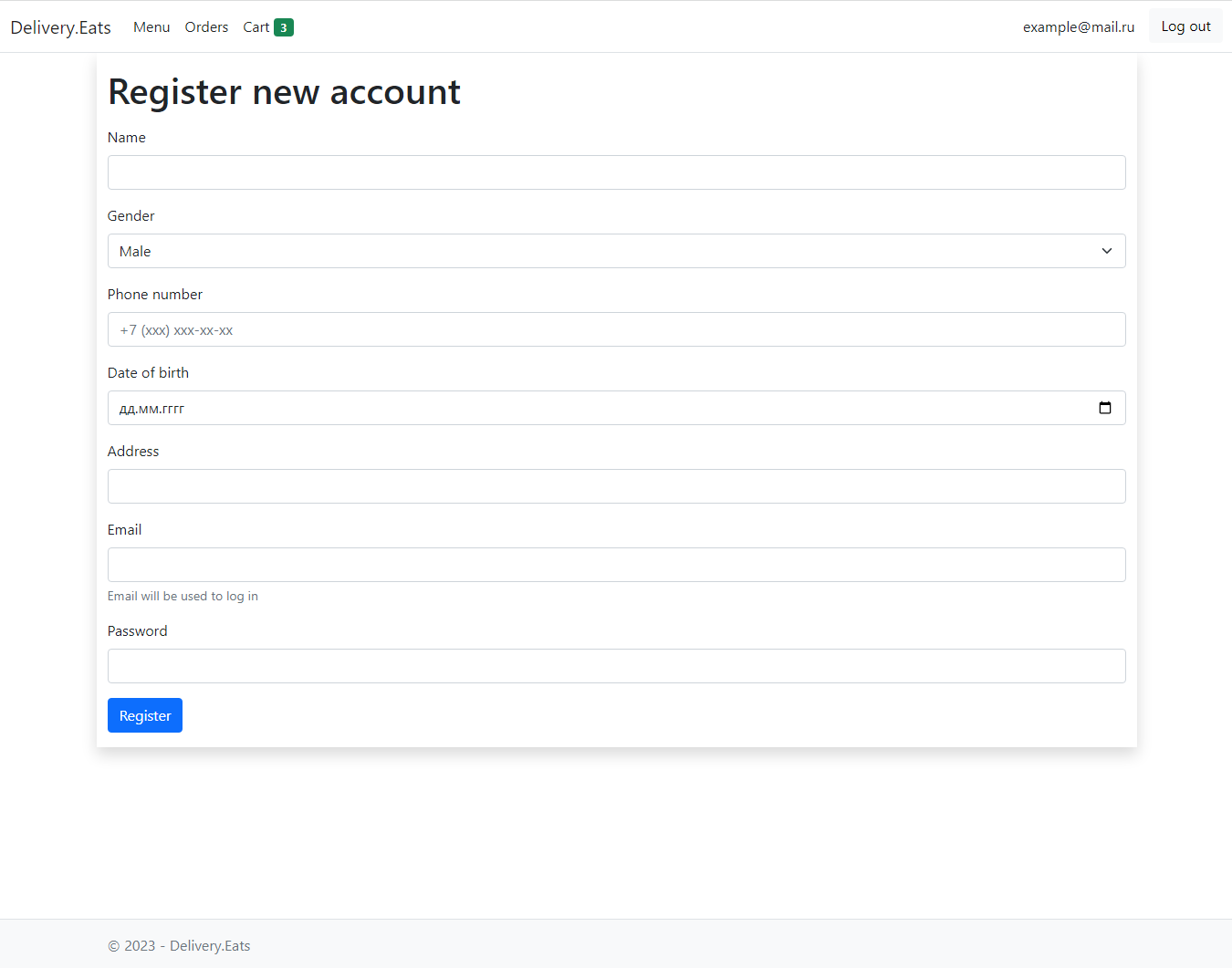


The address you should have in the address bar when you open the authorization page: <http://localhost/login/>

You need to send a POST request with the necessary data to this address. The TOKEN of authorization of the registered user will be sent in reply (if the authorization was successful).

You must use email as your login for authorization.

## Registration

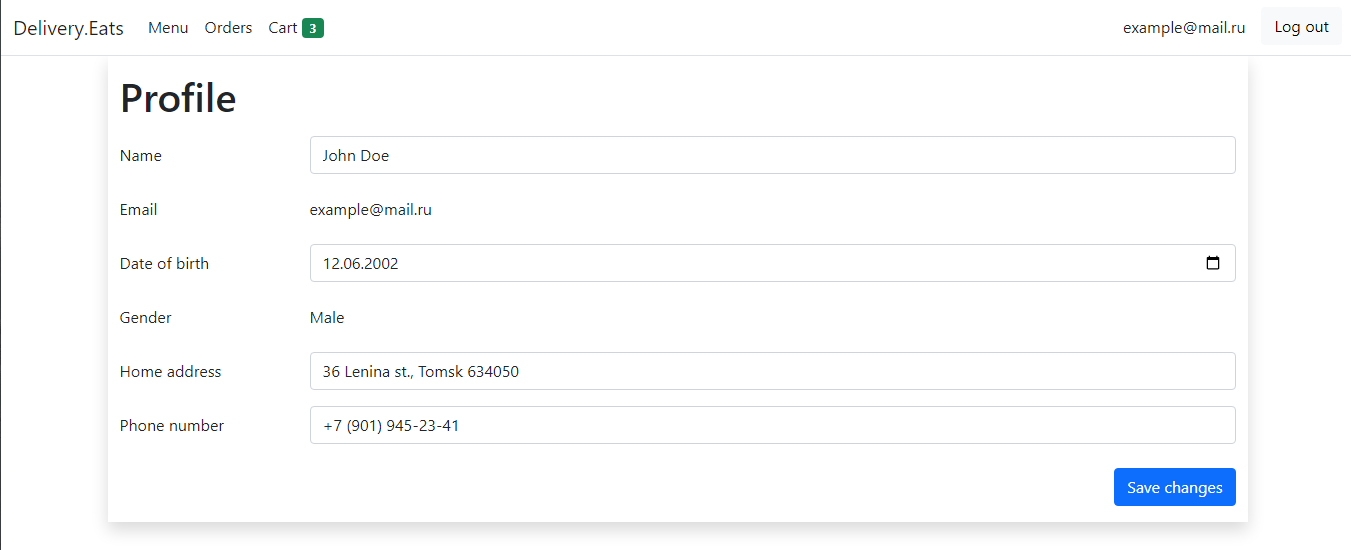


The address you should have in the address bar when you open the registration page: <http://localhost/registration/>

At this address you need to send a POST request with the necessary data. In response the TOKEN of authorization of the registered user will be sent (if the registration was successful).

Please note, the pages where the user enters the phone number must have the following mask validation +7 (xxx) xxx-xx-xx-xx.

## Profile

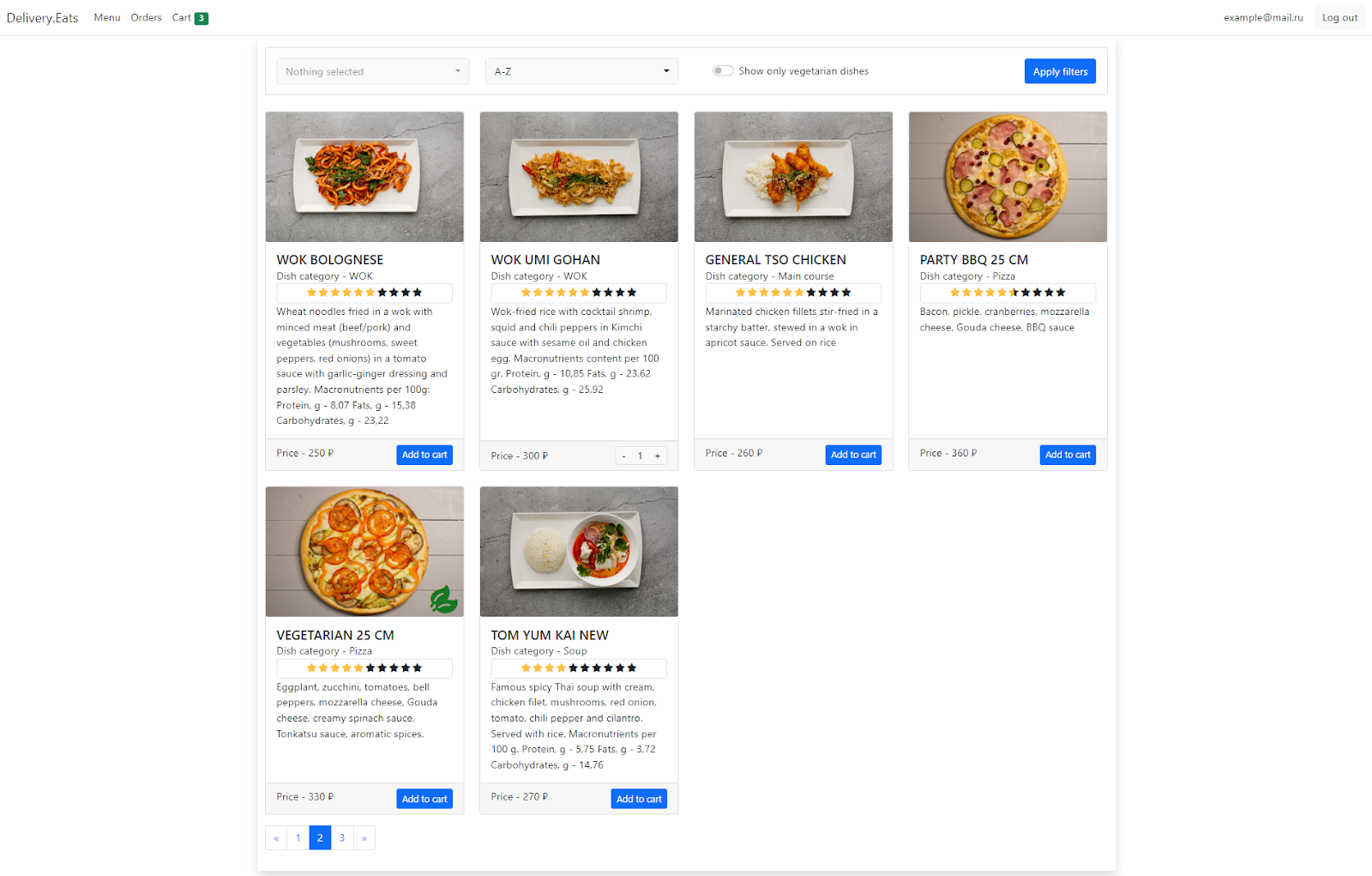


The page should be accessible at <http://localhost/profile>

As you can see from screenshot of layout, editable fields in profile is:

* Name
* BirthDate
* Address
* Phone

## Main page



Since there is pagination on the site, you need to provide a mechanism that does not just send a request to the server with the page number and updates the list of dishes, you need to make sure that when the page changes, the number is displayed in the URL and when you reload the page, the correct list of dishes for the last selected page is displayed.

What the URL should look like when paginating through the menu:

* localhost/ - first page (default)
* localhost/?page=1 - also the first page
* localhost/?page=N - page N

In order to implement this mechanism the pagination elements should not be just buttons with event handlers, but the full-fledged navigation elements.

The site has a mechanism for filtering and sorting dishes by several criteria that can be combined together:

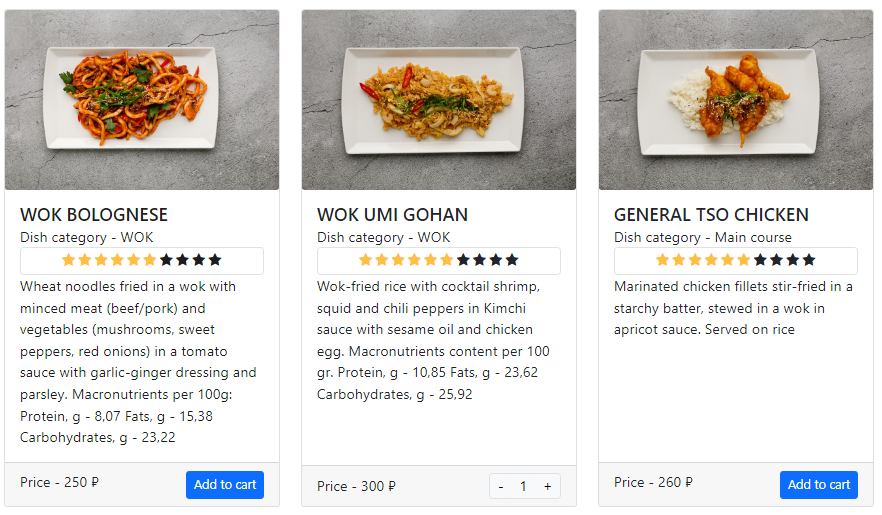
* Dish category (multiple categories can be specified when filtering)
* Display of vegetarian dishes only
* Dishes sorting (by name from A-Y, by name from Y-A, price ascending, price descending, rating ascending, rating descending)

URL examples for pagination and filtering:

* localhost/?page=1 - also the first page
* localhost/?page=N - page N
* localhost/?categories=Wok&categories=Pizza&vegetarian=false&sorting=NameDesc&page=1
* localhost/?categories=Wok&categories=Pizza&vegetarian=false&sorting=NameDesc
* localhost/?categories=&vegetarian=true&sorting=PriceDesc

Please note!

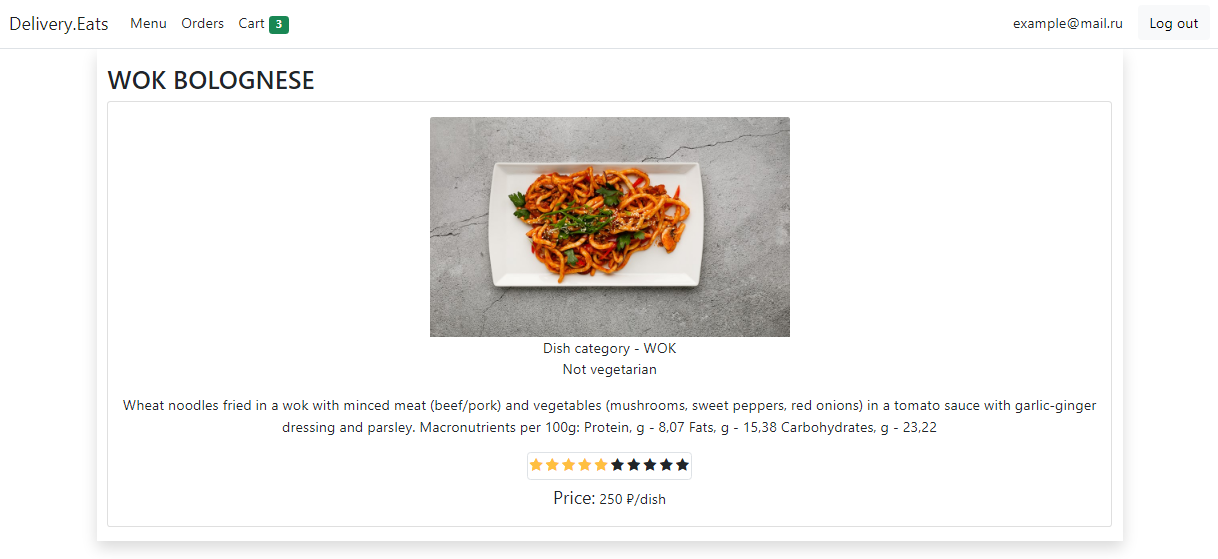
* Vegetarian dishes are marked with a special symbol
* If a dish is already in the cart, it must be possible to change its quantity on the menu page
* Rating of a dish is not necessarily an integer number



If you wish, you can change the display of the filtering and sorting block by using libraries. An example of using the selectpicker library^

When you click on a menu item (a specific dish), the menu item page opens.

## Menu item

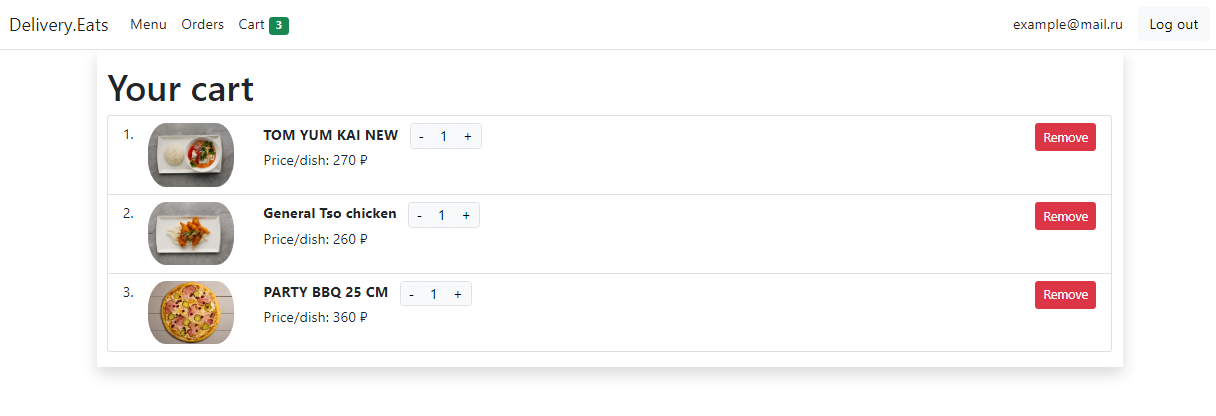


The address you should have in the address bar when you open the page:

[http://localhost/item/{id](http://localhost/item/%7Bid)} , where {id} is the identifier of the menu position on the server.

For the authorized user there is an opportunity to put his rating for the dish if he has ordered this dish at least once before, while the unauthorized user can only view the current rating.

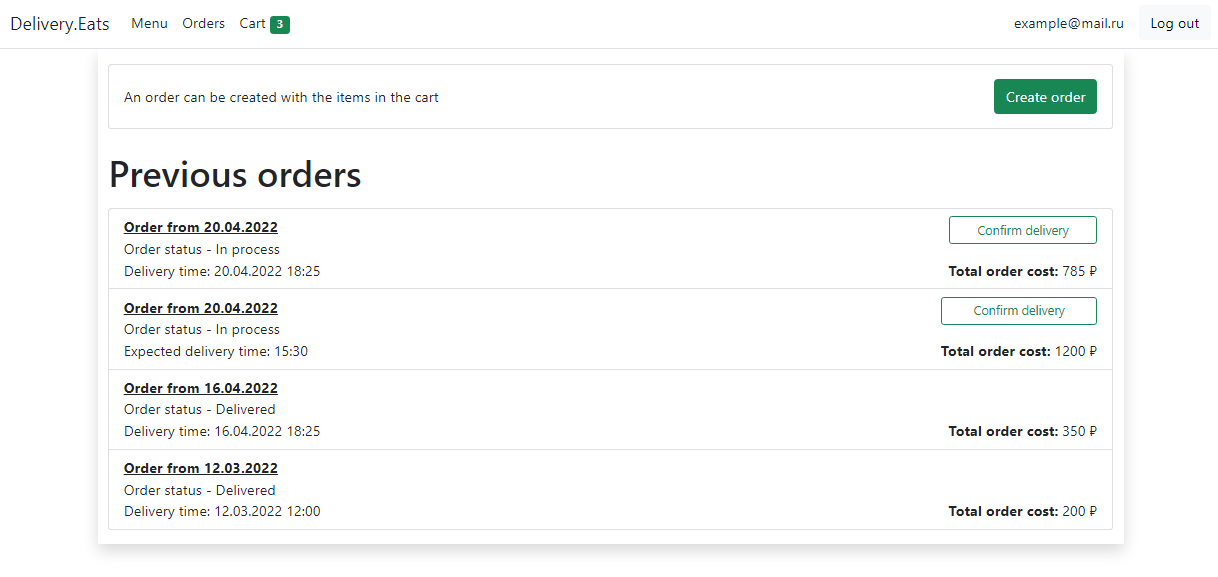
## Cart



The address you should have in the address bar when you open the page:

<http://localhost/cart/>

## Orders

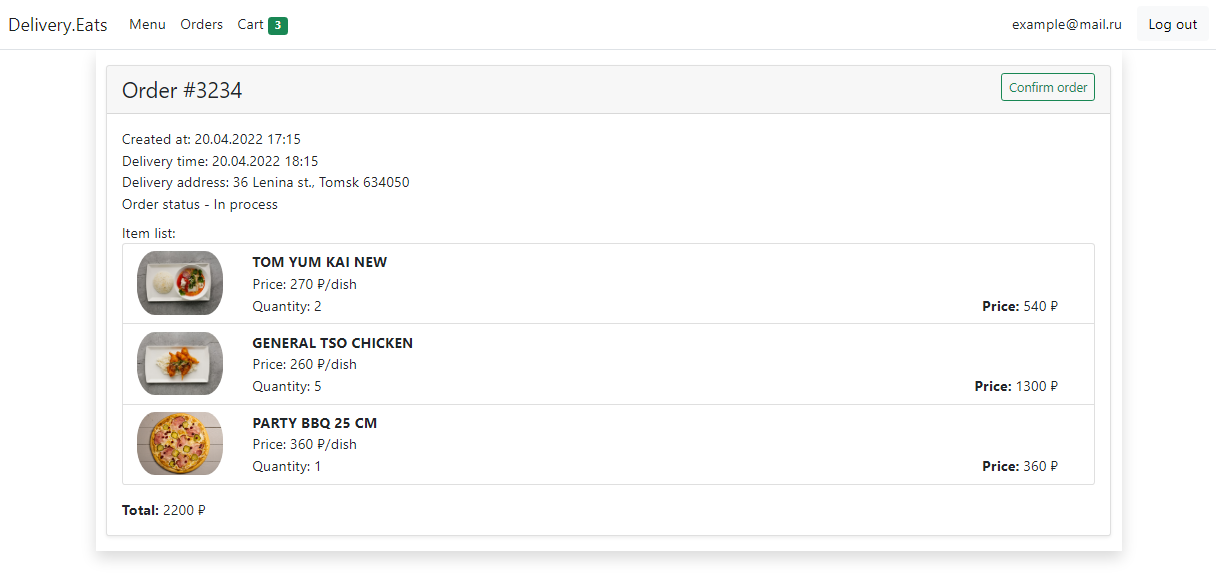


The address you should have in the address bar when you open the page:

<http://localhost/orders>

Please note that you can confirm delivery only for those orders whose status is "In Process".

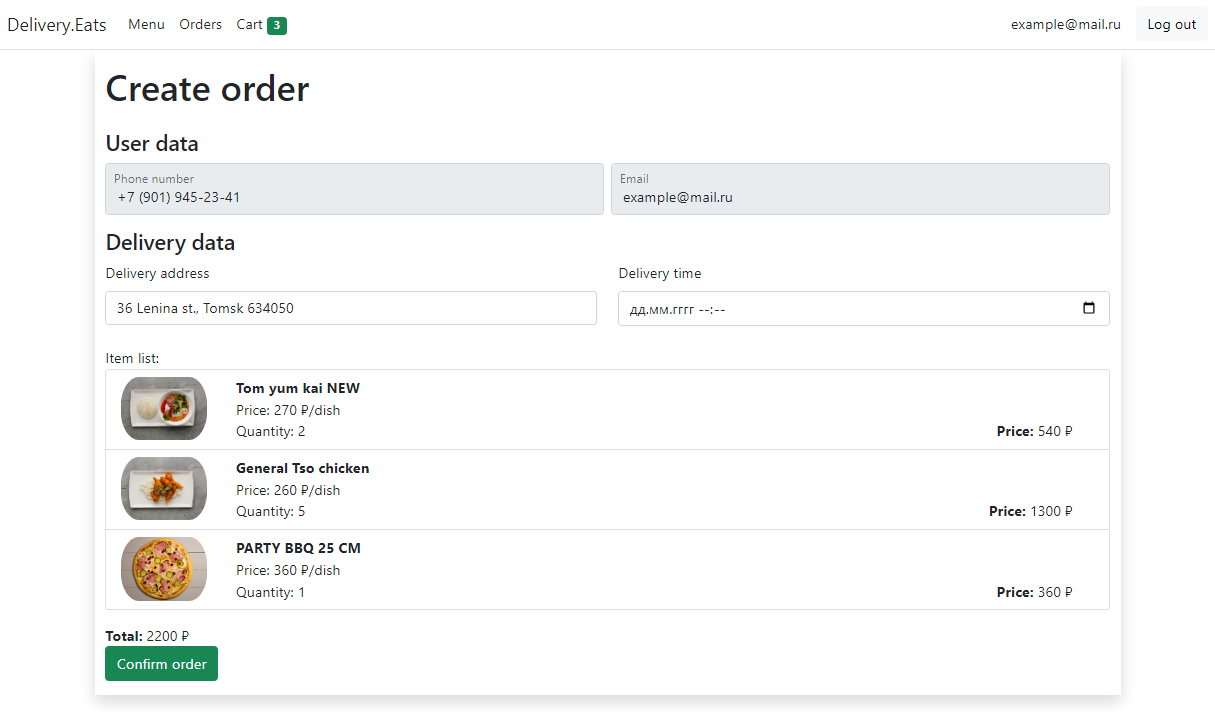
## Order details



The address you should have in the address bar when you open the page:

[http://localhost/order/{id](http://localhost/order/%7Bid)} , where {id} is the order ID on the server

## Ordering



The address you should have in the address bar when you open the page:

<http://localhost/purchase>

When placing an order, all the dishes currently in the user's cart must be taken. After successful checkout the user's cart should be cleared.

Pay attention to the validation of delivery time and date. The order delivery time must be greater than the current time by the variable defined on the server.