**If your program is not compiled, run for a mile and start recompile**

TEAM MEMBERS:

Kattakhodjaev Boiskhon U1910171

Karimov Ravshan, U1910119

Zaynutdinov Zufarbek U1910139

Ismoilov Ibrokhim U1910191

Ulugbek Yarkinov U1910140

Github REPO: <https://github.com/iuthub/group-project-compilemile-1>

Web SITE: compile-mile.uz



Evaluation criteria

You are required to develop a web application which involves both client-side and server-side programming. It should also use database. You should upload your final source code to corresponding GitHub Classroom repository and host your application on a remote server. Web application should be developed using HTML, CSS, PHP, MySQL, JavaScript and other related technologies covered in lectures. You must choose one of the below business scenarios for your web application. If you have a different project idea, please write an email or contact me in my office hours to verify

1. Content (weight 20%)
2. Look and Feel (weight 10%)
3. Forms and Input Validation (weight 10%)

• Username which must contain at least 5 alphanumeric characters

• Password which must contain at least 5 word characters (alphanumeric, %,$, -, \_, etc)

• Confirm password which should be the same as the password

• Email which must match the standard email format

• Phone number which must match the standard phone number format (e.g. +998-99- 1234567)

• Date of birth which must match the standard date format (e.g. 23-03-1993) • Postal code which must contain exactly 7 digits (e.g. 100011)

• City name which must contain words only

• Passport number which match the standard passport number (e.g. AA1234567)

1. Database (weight 10%)
2. Server-side Framework - Laravel (weight 10%)
3. Client-side Scripting (weight 10%)
4. GitHub Activity and Hosting (weight 10%)

About us:

Short story of creation

Compile-mile is a young “commercial business” project that was created by several students at the first year of studying at IUT. Firstly, the brand name was used in C++ project. However, being at the second course it was reused for Java course and furtherly for creating the online shop for a small ‘commercial’ shoes store.

Main activities

According to our registration panel, you may think that our main purpose is to collect your private data such as (passport number, date of birth …), but I swear that these data is collected for your benefits, to personalize your account settings.

For the begging, our aim is to sell all **four** pairs of sneakers!!! Somebody might think that it is not a big deal, but Rome wasn’t built in a day😊. Who knows, may be in future we will make a real competitive market of shoes wear in Uzbekistan, CIS or even the whole world. As the second, but not less important activity is charity, 10% of each pair is sent for charity and subsidizing of non-commercial projects

Workers:

Our ‘~~family’~~ ‘team’ is not so big, but really close-knit

* Kattakhodjaev Boiskhon – a front-end (JavaScript) developer whose purpose is to make our web recourse more dynamical and user friendly.
* Zaynutdinov Zufarbek – a front-end (html/css) developer whose role can’t be overestimated. The design is the highest abstraction that user sees. That’s why the final outlook of the project plays a credential role
* Karimov Ravshan – a front-end (html/css) developer who let our online store to look modern and keep up with all new concepts
* Ismoilov Ibrokhim – a front-end developer who made the first steps in project
* Yarkinov Ulugbek – the only one back-end developer whose aim was to work with database and server/client-side working

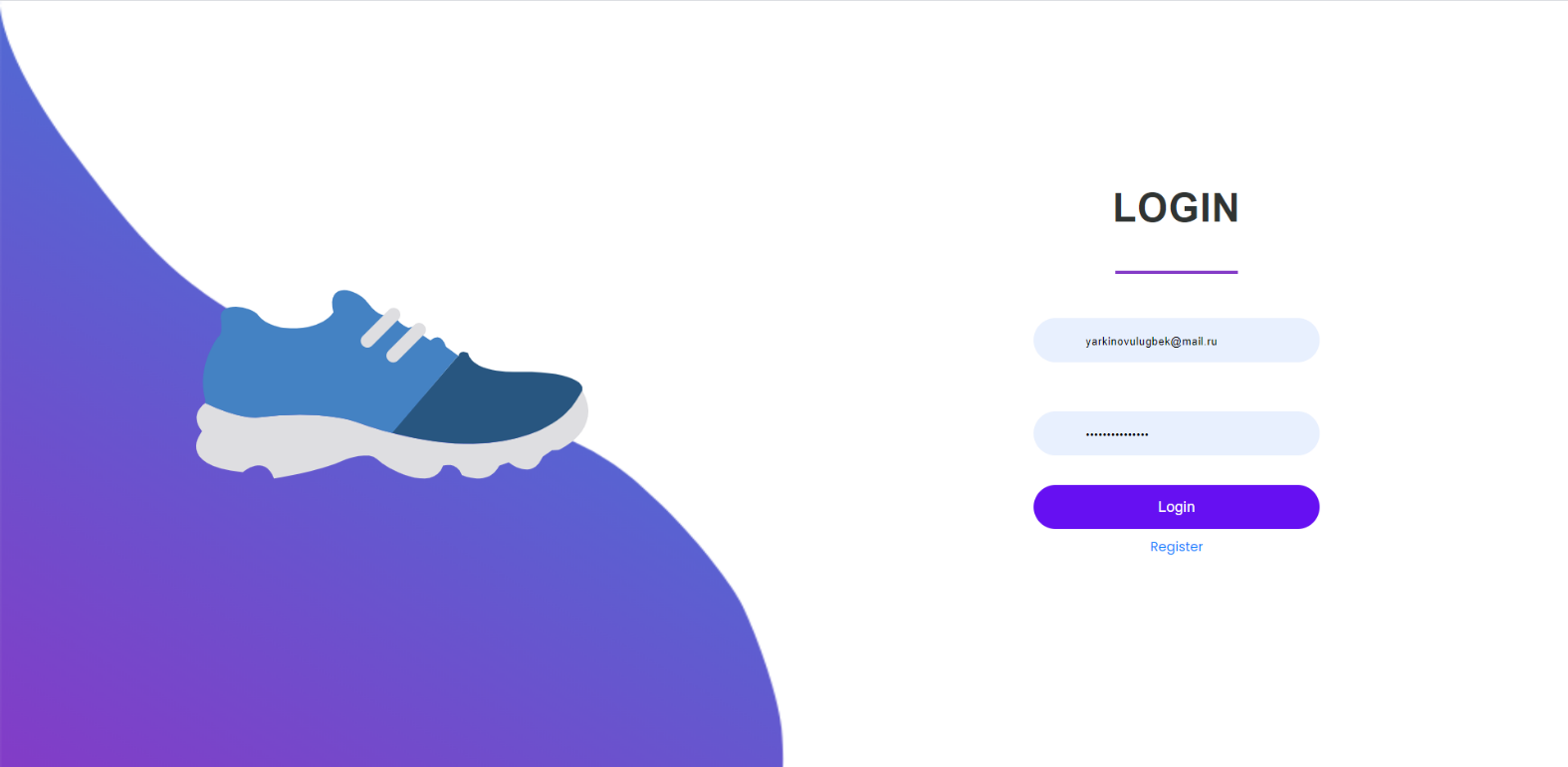
**Remark: All of the workers mentioned above worked together to achieve the goal**

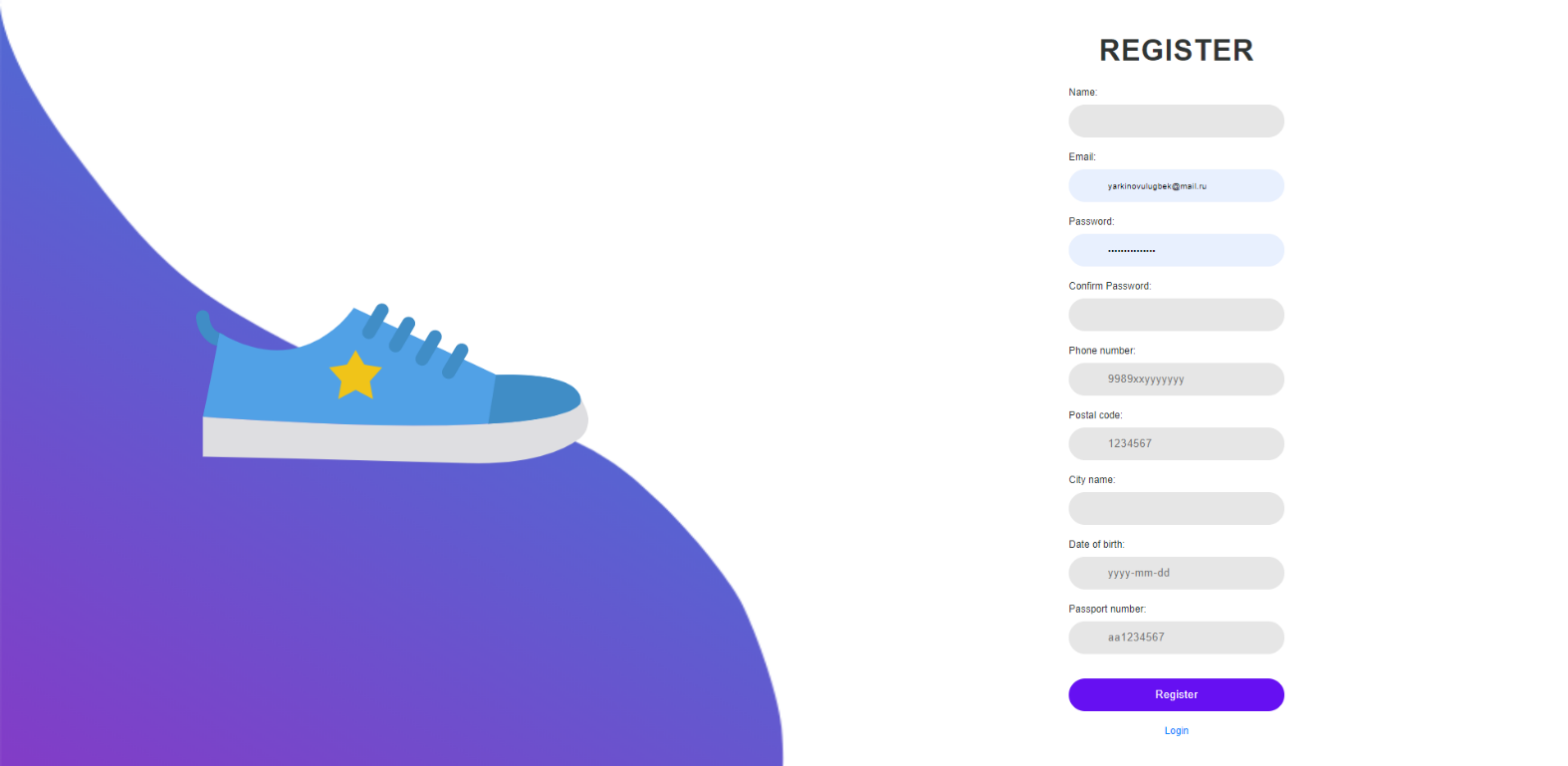
Brief documentation (guide)

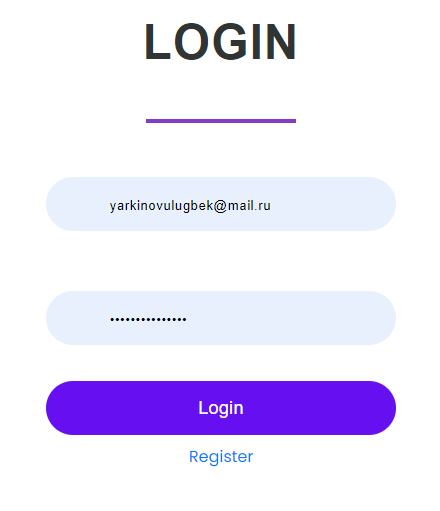
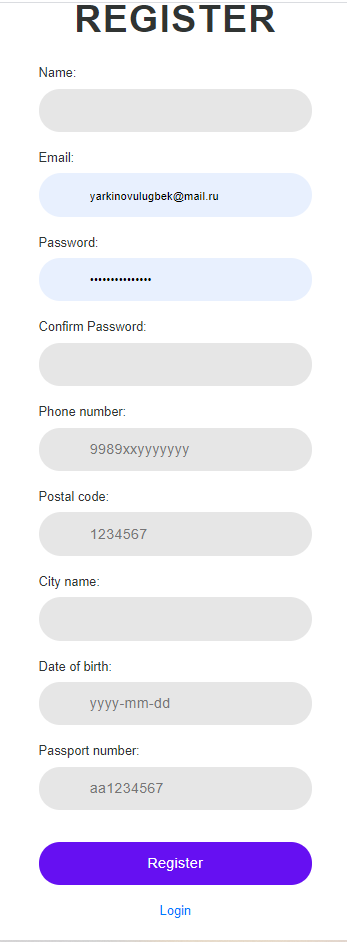
Login/Registration panel

Front-end

Login panel



Registration panel



To Create a Login and Registration form also, we have visited some web apps to see and to test their forms and how they work.

For design as a main login/register photo we have used .svg pictures and just html to create login and registration form. The customization was completely done using css, login.css and register.css respectively (onlineShop/public/css/)

Back-end

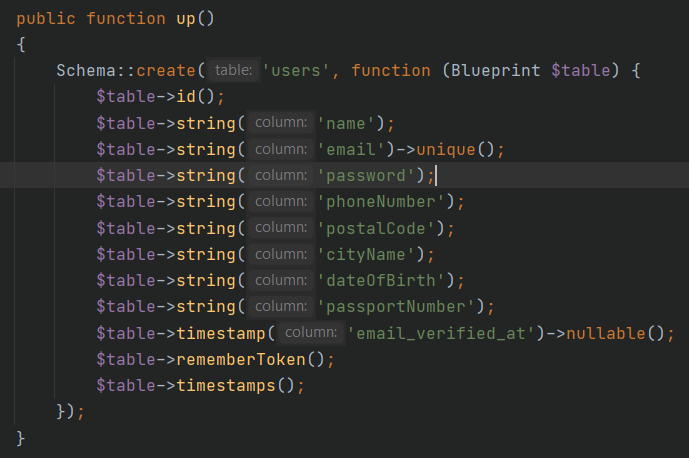
For back-end we have used Fortifier as a base functionality

The variety of prepared files such as (/Actions/Fortify/) folder content allowed to customize validation before seeding the database

onlineShop/database/migrations -> user table.

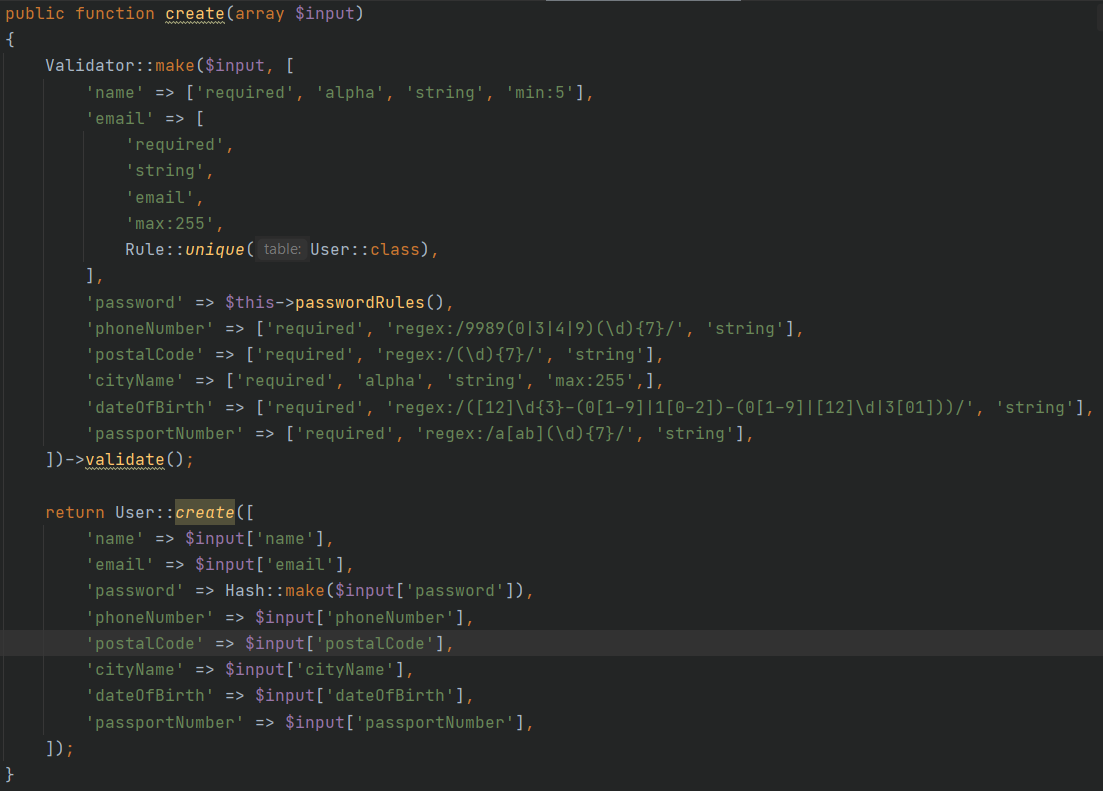
The user table contained several columns that are necessary for registration

Where email is signed as unique because it is used to registration to view the repeated accounts



onlineShop/app/Actions/Fortify/CreateNewUser.php

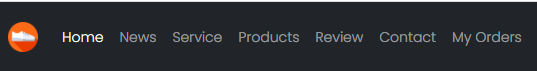
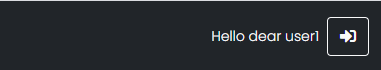
Using the forms to get the values from Request and further validating them.



Main Page

With our team we have decided to make the conceptual design for online store of sneakers. To realize front-end part, we have used some 3rd party frameworks such as Bootstrap to make our web app to be responsive and adaptive. Laravel we used for back-end. Before we created the conceptual design, we have done some mini research, so we have visited some famous websites and web apps to see how their sites structured. After that we came up with the online store of sneakers and the design we have now.

Navigation

For navigation menu we have used bootstrap sticky navbar. Every section of the webpage is divided by ids. Active menu links are highlighted when it is clicked, it is done using JavaScript code.

My Orders section opens the basket.blade.php (onlineShop/resources/views/)

And the right side nav-menu displays the logged in user name and logout icon

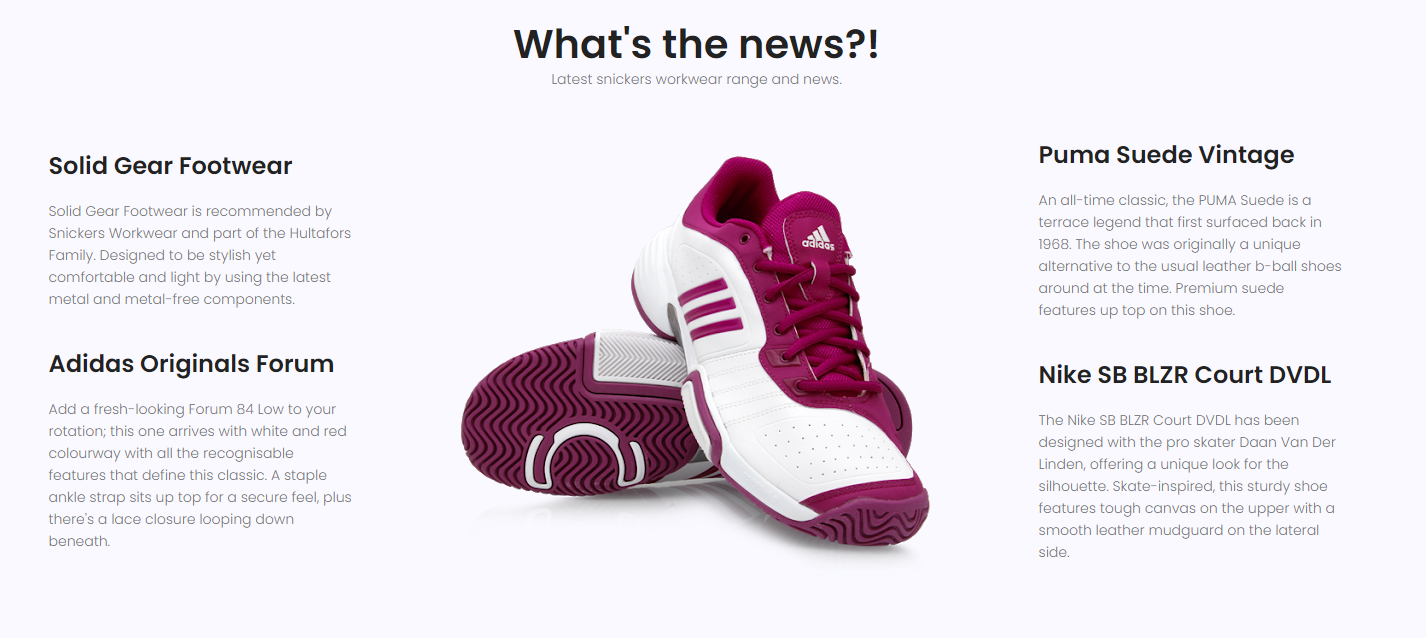
Carousel

Carousel was created in a base of bootstrap library and built with CSS 3D transforms and a bit of JavaScript. It works with a series of images and also includes support for previous/next controls and indicators. .d-block and .img-fluid on carousel images to prevent browser default image alignment. With data-targets we identify which slider-gallery element would come first second or third in particular. The active class needs to be added to one of the slides. Otherwise, the carousel will not be visible.



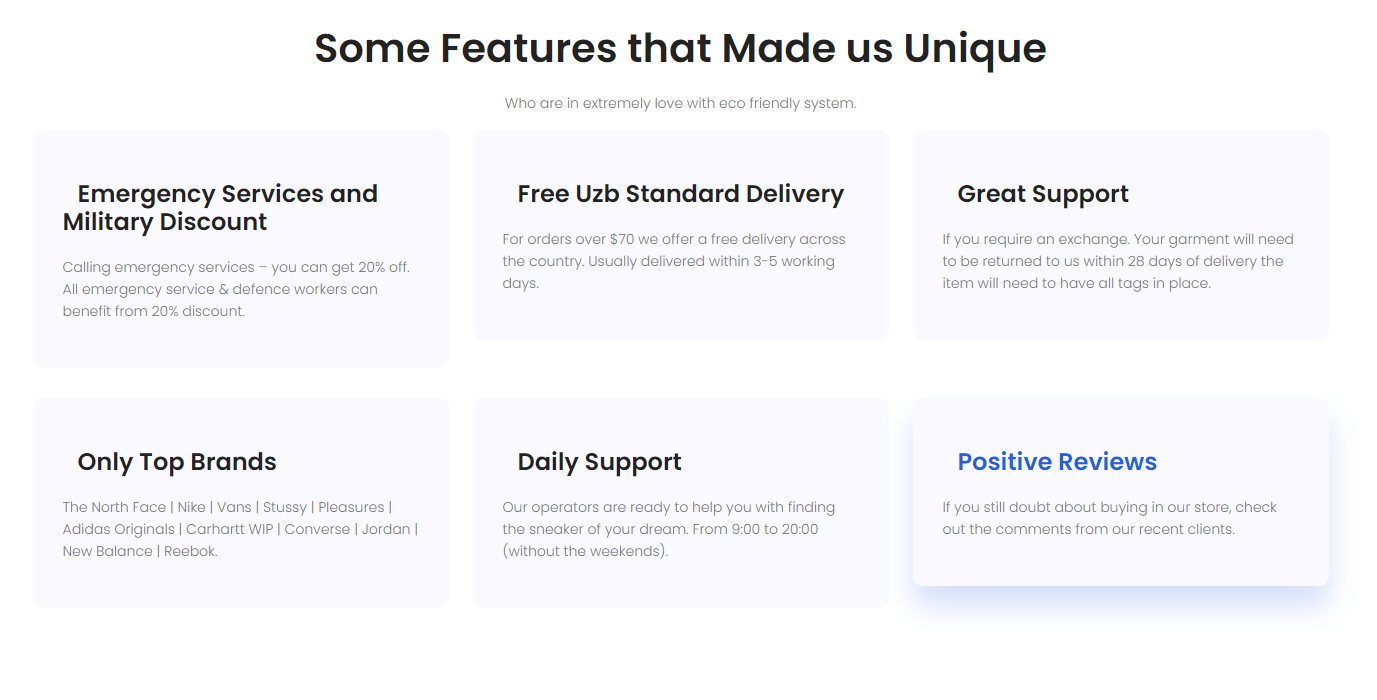
News

News section consists of a container with small cards with the centered image. It is used for showing latest news of the store



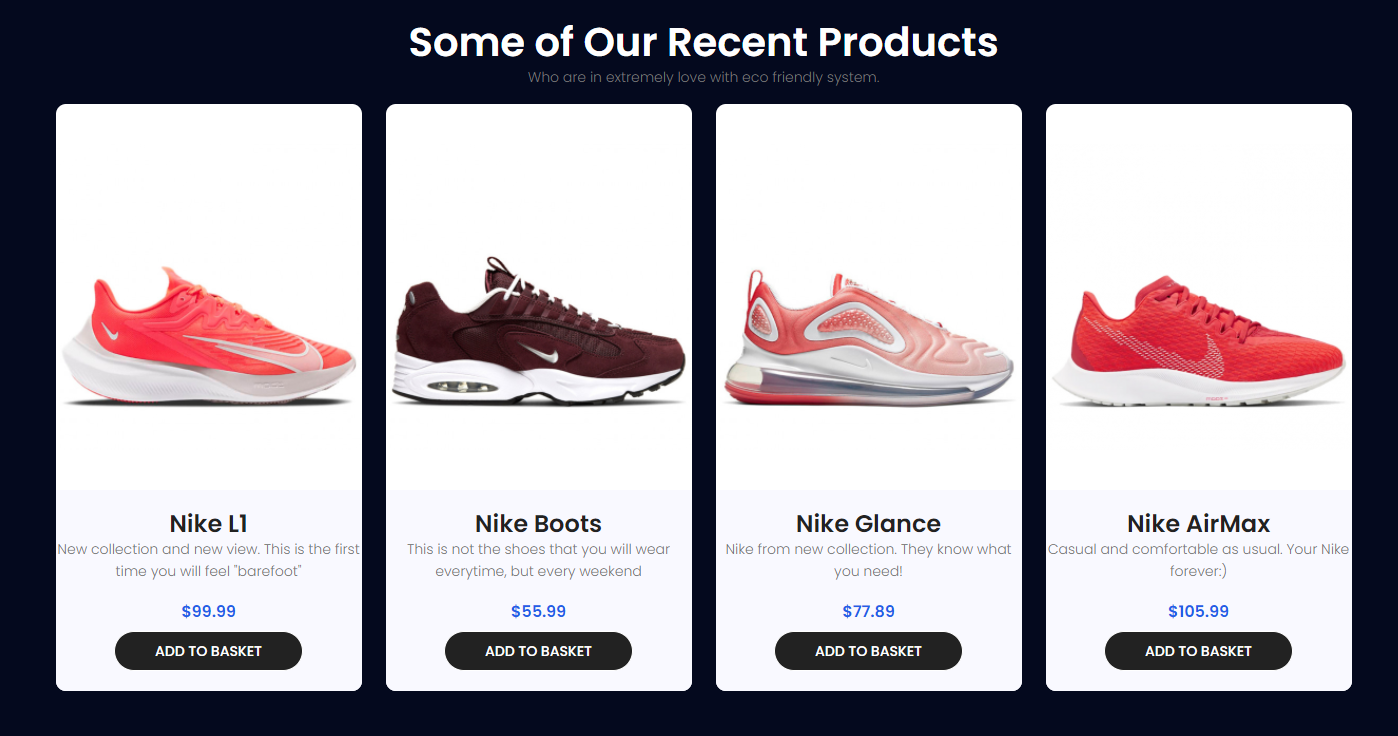
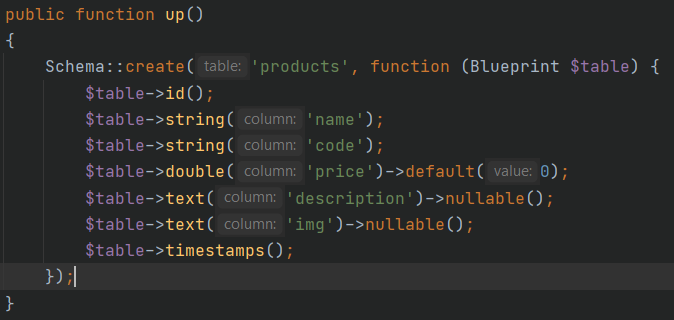
Service

Service container is added to show all the services provided by the store (free delivery, discounts and e t.c.). Cards of the section are responsive (bootstrap flexbox alignment: col-lg-4 col-md-6) and have hover effects (box-shadow: 0px 20px 30px 0px rgba(35, 94, 231, 0.2).



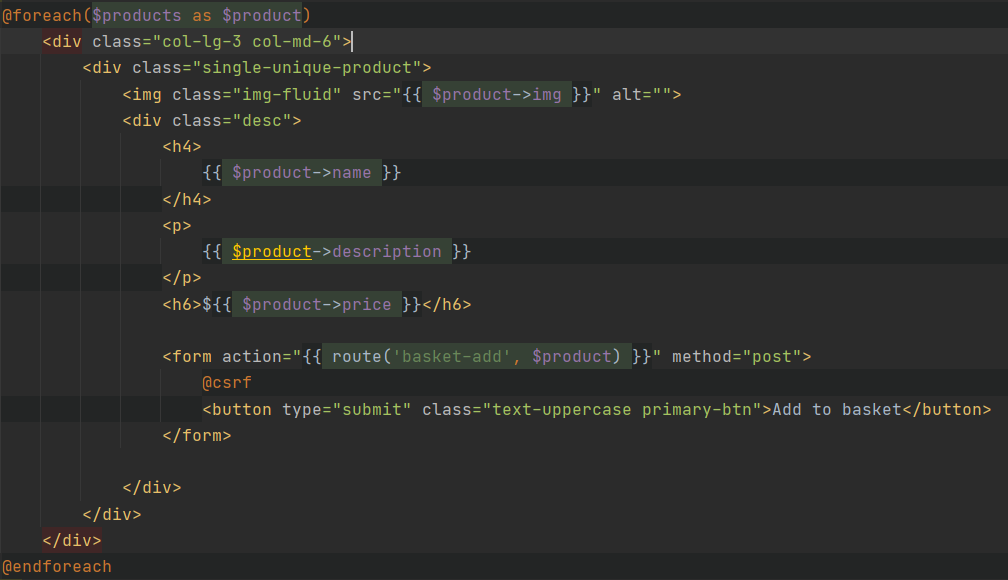
Products

Review section is similar to the service container. It also contains small cards with hover effect. This section contains comments from the clients with the 5 start rank (class: checked of icons)



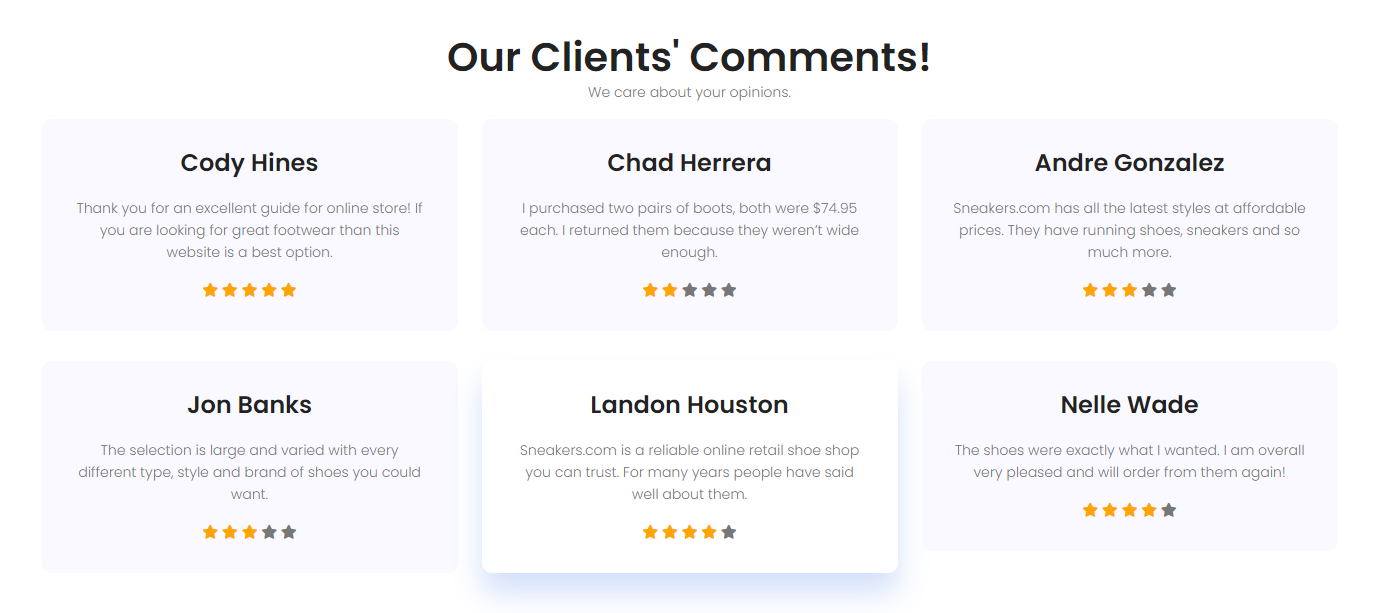
Each product is a card which data is taken from product db table. It contains such columns

Then the products are displayed using foreach() loop



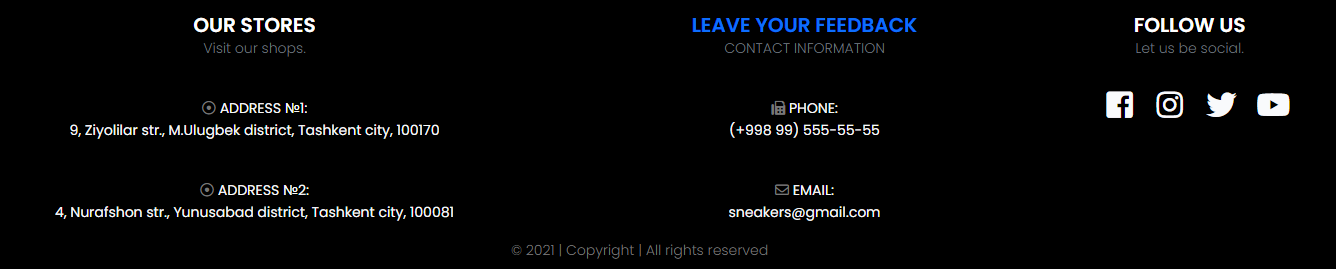
Review

Review section is similar to the service container. It also contains small cards with hover effect. This section contains comments from the clients with the 5 start rank (class: checked of icons)



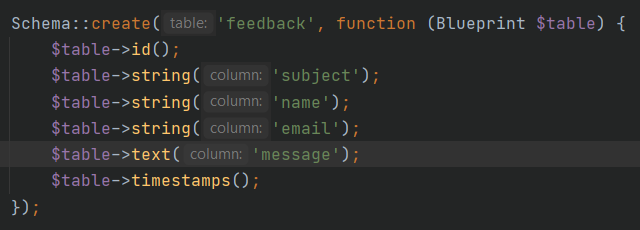
Contact

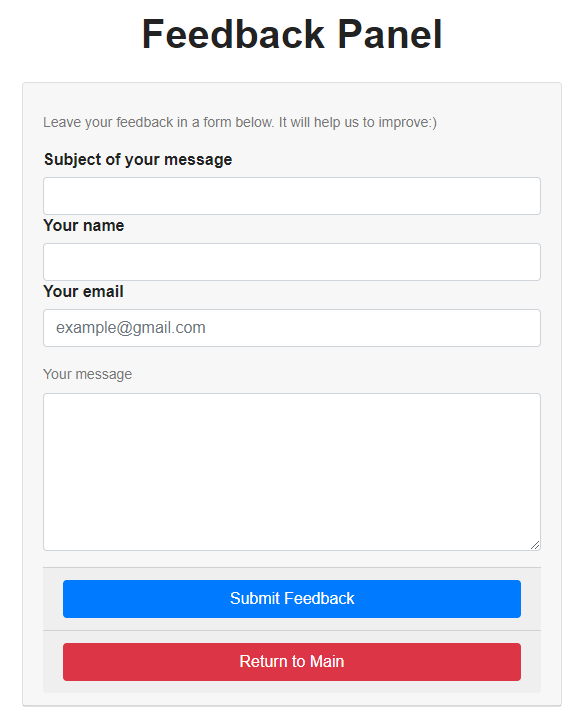
Footer is divided to 3 parts; it is done using bootstrap flexbox alignment utilities to vertically (class="col-lg-5 col-md-6 col-sm-6"). By using this utilities, we got a responsive footer section.



Feedback Panel

The blue link on the center LEAVE YOUR FEEDBACK redirects user to a panel where he or she can leave their comment and suggestions about our online shop. Our feedback panel has its own table where all suggestions of our clients are stored, model and controller to represent the work between view and model keeping the concept of MVC architecture



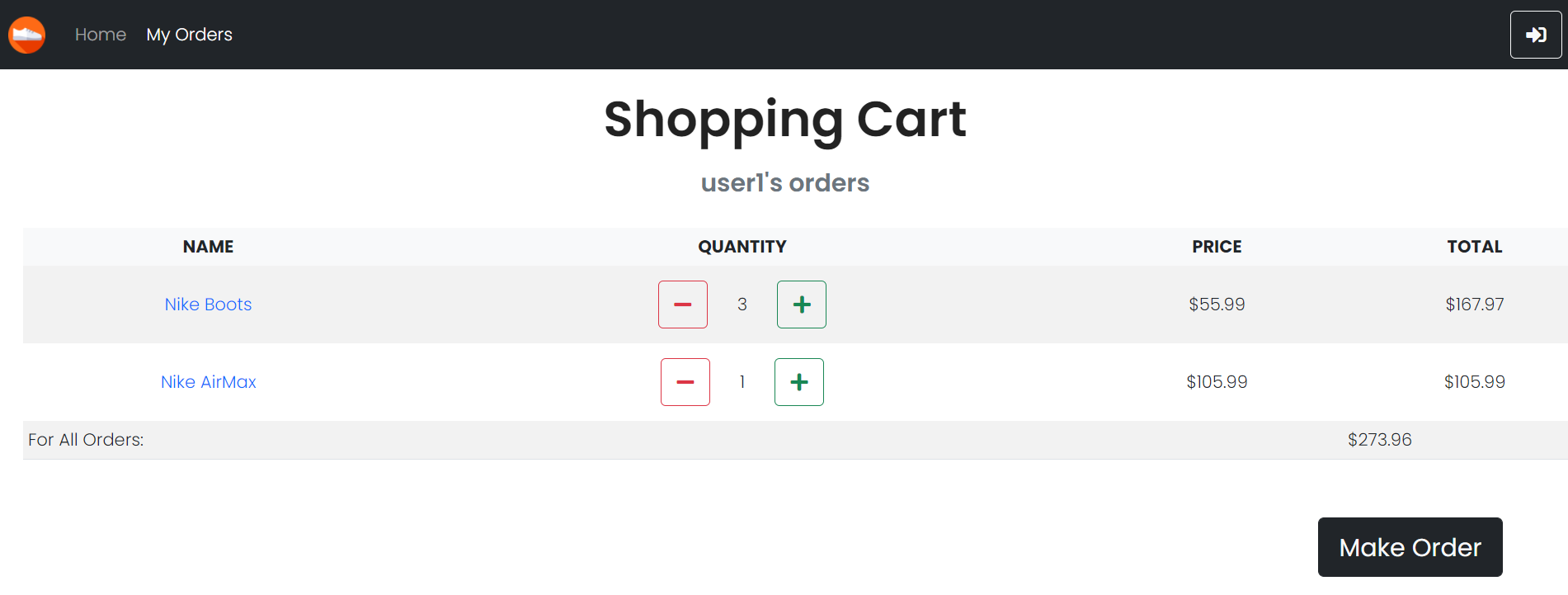


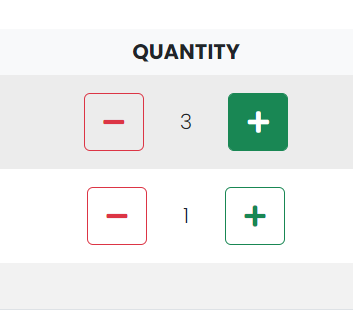
When Submit Feedback button is pressed the feedbackConfirm() method is called. The concept of its work is approximately the same as in basket case. The validation is applied to email only and data take from Request are saved in feedback table. When Return to Main is pressed it redirects to idex

My Orders (detailed review)

Front-end

Basket webpage is done using table html tag with bootstrap classes. It has a dynamic total price calculation. Also users can specify the quantity of the product. Returning to the main page is also done without the data lost. When user click «Make Order» button it will redirect to Payment webpage.

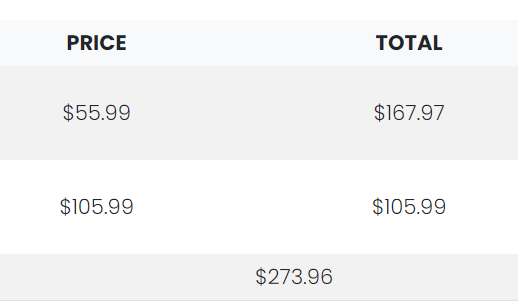




These two buttons call methods basketAdd and basketRemove respectfully. The number between them shows the quantity and is stored in order db.

It is incremented and decremented in the same methods mentioned above. When a product is added or removed successfully the message about it is displayed in a navigation menu above using sessions and flash

Below is the code for them



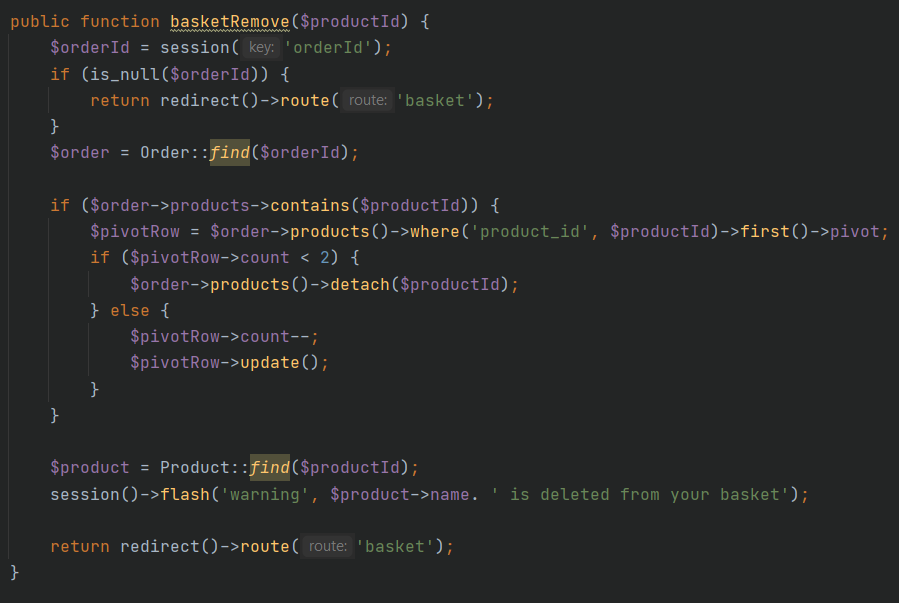
Here using methods getPriceForCount() in Product model we could calculate the price for N products

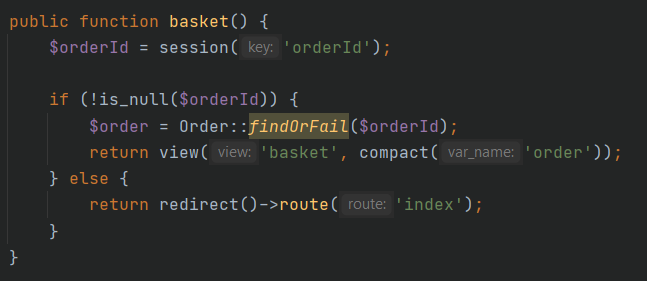
getFullPrice() in Order Product shows the price for all products added to basket

While your basket is empty it is not accessible and redirects you to the index page, when at least one product is added to basket, basket.blade fills up with the order

It happens because we decided to use sessions to make the basket add feature and when the user has just logged is the session of orders is undefined and when the product is added the session product is created.

Two methods are responsible for adding and removing item to/from a basket



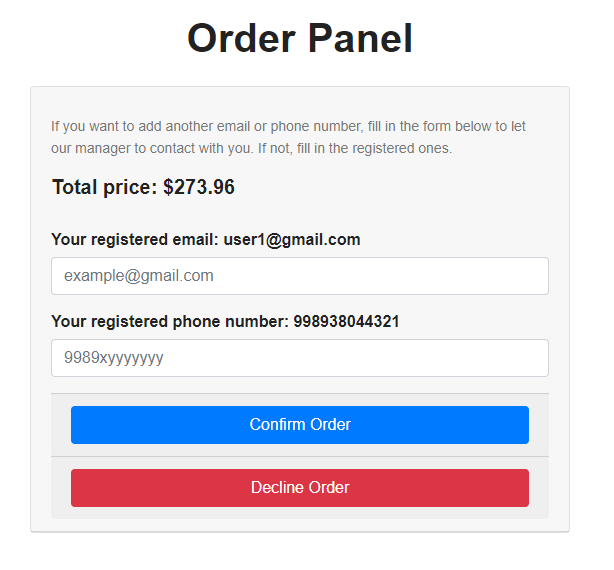


Pressing button **Make Order** calls method basket-place that displace the final order stage:

href="{{ route('basket-place', $order->getFullPrice()) }}"

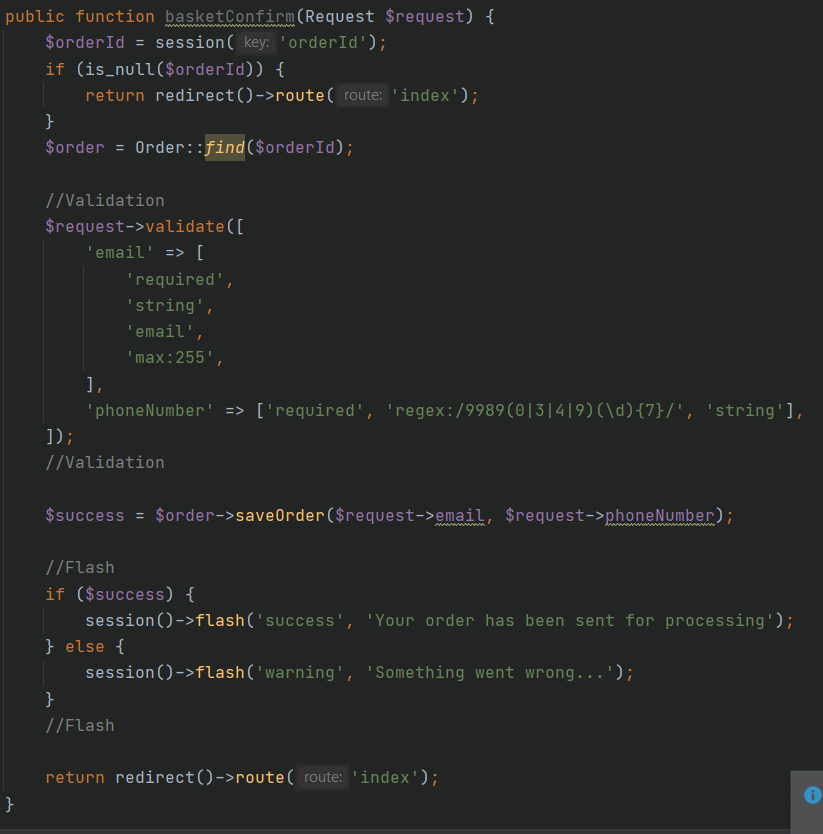
Here href is gets the parameter getFullPrice for basket-place method to not allow user to make an order if his/her full price is 0

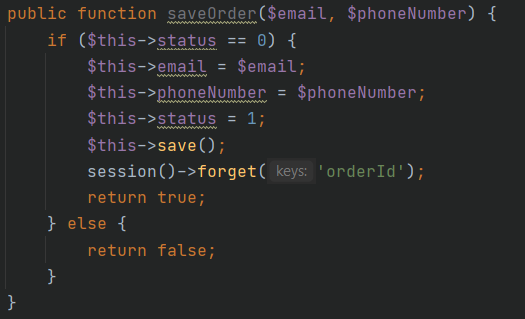
Make order Panel



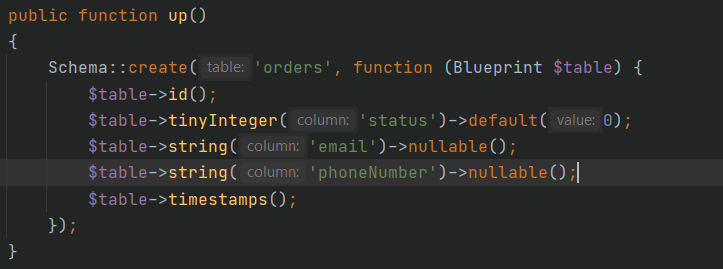
In this panel the variable with getFullPrice() method’s value is taken by method compact() – collection.

Here again the form input email and phone number is validated using regex



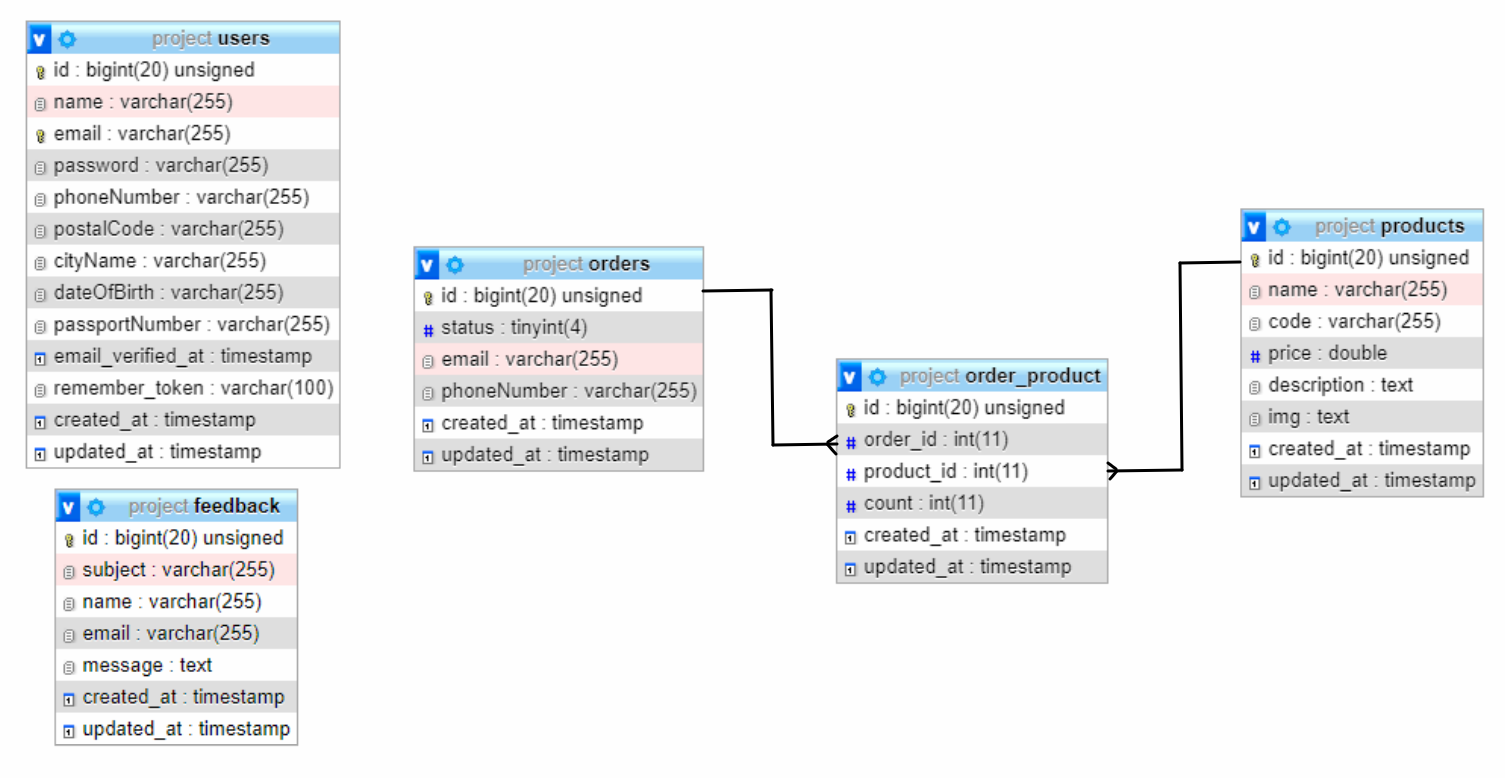
And stores the data in orders table calling the method in Order model. As arguments it takes the data taken form Request form. Pressing Confirm Order button calls basketConfirm() method and either stores them id database or prints the validation error. Decline Order just redirects you to index page in case if you changed your mind.

Orders table. Email and phone number were taken as nullable to check the new feature



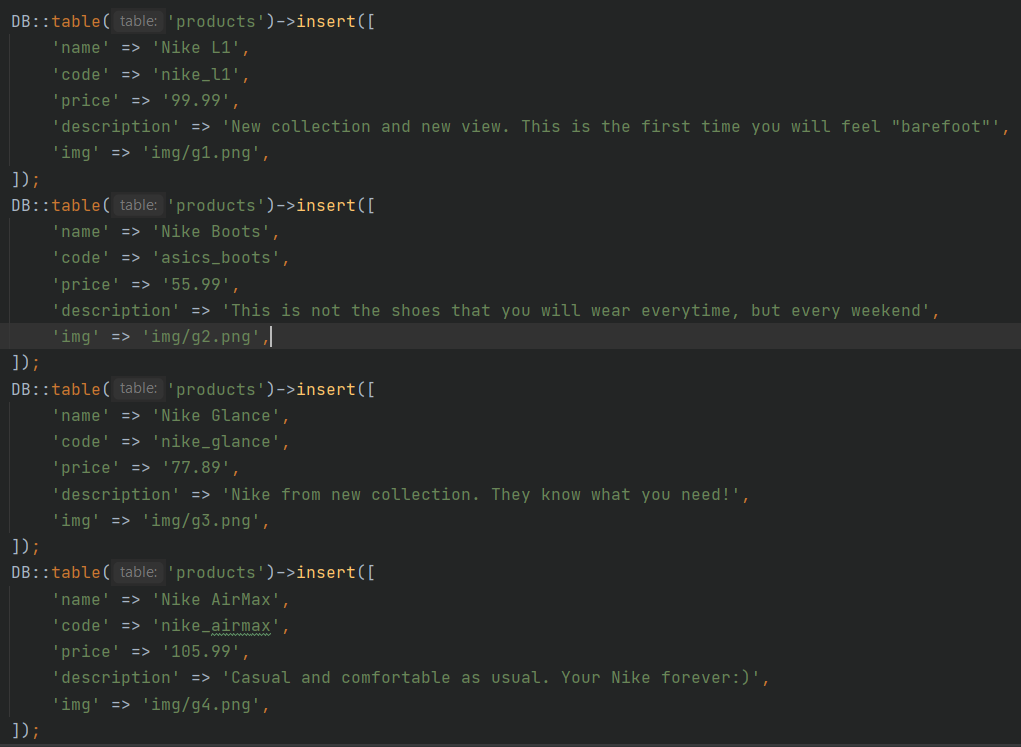
When an order is successfully confirmed the message on index page where you were redirected prints the required message in navigation bar using sessions and flash

Database



Seeders

Product Seeder



User seeder

name: user1

email: [user1@gmail.com](mailto:user1@gmail.com)

password: 12345678

phone number: 998938044321

postal code: 1234567

city name: Tashkent

date of birth: 2000-04-26

passport number: AB1233211

Contributors

**Zufarbek Zaynutdinov**- navigation menu, content filling, footer section, basket page design, carousel design, code review, login panel fixing, report contribution

**Ismoilov Ibrokhim** - main.css and index.html (main page) creation and development, content filling, code review, report contribution

**Kattakhodjaev Boiskhon** - JavaScript code review, carousel implementation and styling, content filling, code review, report contribution

**Karimov Ravshan** - login and register page design, form processing, developing user friendly interface, content filling, code review, report contribution

**Ulugbek Yarkinov** – login/registration and basket back-end logic, developing order and feedback front and back, database architecture and filing layouts from database, content filling, code review, report contribution