

DOKUMENTACIJA

<http://php3.bojanmaksimovic.com/>

Web Programiranje PHP 2

Bojan Maksimović 92/19

Smer: Internet tehnologije

Modul: Web programiranje

Predmet: Web Programiranje PHP 2

SADRŽAJ

Contents

1.	Uvod	6
1.1	Korišćeni programski jezici i tehnologije	6
1.2	Opis funkcionalnosti	6
1.3	Template	6
2.	Organizacija	7
2.1	Dizajn baze podataka	7
2.2	Mapa sajta	8
2.3	Slike stranica i opis funkcionalnosti	9
	/ ili /home	9
	/restaurants	10
	/restaurants/{id}	11
	/contact	12
	/author	13
	/login	14
	/register	15
	/admin/ ili /admin/dashboard	16
	/admin/actions	17
	/admin/users	18
	/admin/restaurants	19
	/admin/food	20
	/admin/categories	21
	/admin/orders	22
3.	Kodovi	23
3.1	Routes	23
	web.php	23
3.2	Kernel	25
	Kernel.php	25
3.3	Controllers	27
	AdminController	27
	AuthController	29
	AuthorController	31
	ClientController	32
	ContactController	33
	FoodCategoryController	34
	FoodController	37
	HomeController	40
	MailController	41
	MyController	42
	OrderController	43
	RestaurantController	45

UserController	50
3.4 Middleware	53
IsLoggedIn	53
IsNotLoggedIn	54
IsAdmin	55
3.5 Requests	56
ChangeCartRequest.....	56
ContactRequest	57
LoginUserRequest.....	58
RegisterUserRequest.....	59
StoreCategoryRequest.....	60
StoreFoodRequest	61
StoreOrderRequest	62
StoreRestaurantRequest	63
StoreUserRequest.....	64
UpdateCategoryRequest	65
UpdateFoodRequest.....	66
UpdateRestaurantRequest	67
UpdateUserRequest.....	68
3.6 Models	69
AccessLog.....	69
ActionLog	70
Cart	71
Food	73
FoodCategory.....	74
ImageUpload.....	75
Nav.....	76
Order	77
Restaurant	79
Role.....	82
Social	83
User.....	84
3.7 Migrations.....	85
create_nav_table	85
create_social_table.....	86
create_user_table.....	87
create_role_table	88
create_access_log_table.....	89
create_restaurant_table	90
create_food_category_table	91
create_food_table.....	92
create_action_log_table	93
create_cart_table	94
create_order_table.....	95
create_order_detail_table.....	96

3.8 Seeders	97
FoodCategorySeeder	97
FoodSeeder	99
NavSeeder	116
RestaurantSeeder	118
RoleSeeder	120
SocialSeeder	121
UserSeeder	123
3.9 Components	124
Food	124
Restaurant	125
3.10 Blade	126
components.client.food	126
componentents.client.restaurant	127
emails.contact	128
fixed.admin.footer	129
fixed.admin.head	130
fixed.admin.nav	131
fixed.admin.scripts	132
fixed.admin.sidebar	133
fixed.client.footer	135
fixed.client.head	136
fixed.client.header	137
fixed.client.scripts	139
layouts.admin	140
layouts.client	141
pages.admin.dashboard	142
pages.admin.actions.index	143
pages.admin.food.create	145
pages.admin.food.edit	146
pages.admin.food.index	147
pages.admin.foodCategories.create	149
pages.admin.foodCategories.edit	150
pages.admin.foodCategories.index	151
pages.admin.orders.details	152
pages.admin.orders.index	153
pages.admin.restaurants.create	154
pages.admin.restaurants.edit	155
pages.admin.restaurants.index	156
pages.admin.users.create	158
pages.admin.users.edit	159
pages.users.admin.index	160
pages.client.author	161
pages.client.contact	162
pages.client.home	164

pages.client.login	165
pages.client.register.....	166
pages.client.restaurants.index	167
pages.client.restaurants.show	169
3.11 JavaScript.....	171
client.js.....	171
common.js.....	172
contact.js	173
food.js	174
form-validation.js.....	178
home.js.....	179
login.js	181
register.js.....	182
restaurants.js	183
admin/categories.js.....	186
admin/food.js	187
admin/restaurants.js	188
admin/users.js	189
3.12 SCSS	190
admin.scss	190
app.scss	191

1. Uvod

1.1 Korišćeni programski jezici i tehnologije

Za izradu sajta korišćen je Laravel, PHP, HTML, SASS/CSS, JavaScript/jQuery i XML(sitemap). Radno okruženje Visual Studio Code.

1.2 Opis funkcionalnosti

Sajt predstavlja veb aplikaciju za naručivanje hrane iz različitih restorana u ponudi

- Registracija i logovanje sa validacijom na klijentu i server
- Zaštita stranica uz pomoć middleware-a
- Dinamičko ispisivanje menija, restorana i najzastupljenih kategorija proizvoda u njima i samih proizvoda
- Pretraga restorana na početnoj stranici pomoću ajax-a
- Filtriranje i sortiranje i paginacija restorana pomoću ajax-a
- Filtriranje i sortiranje i paginacija proizvoda svakog restorana pomoću ajax-a
- Dodavanje proizvoda u korpu (porudžbinu) i izmena korpe pomoću ajax-a
- Slanje mejla administratoru preko kontakt forme
- Uvid u bitne statistike sajta u admin panel-u
- Izveštaj o ključnim aktivnostima korisnika u admin panel-u
- Mogućnost upravljanja sadržajem sajta u admin panel-u
- Resize slika pri upload-u uz pomoć Intervention Image-a

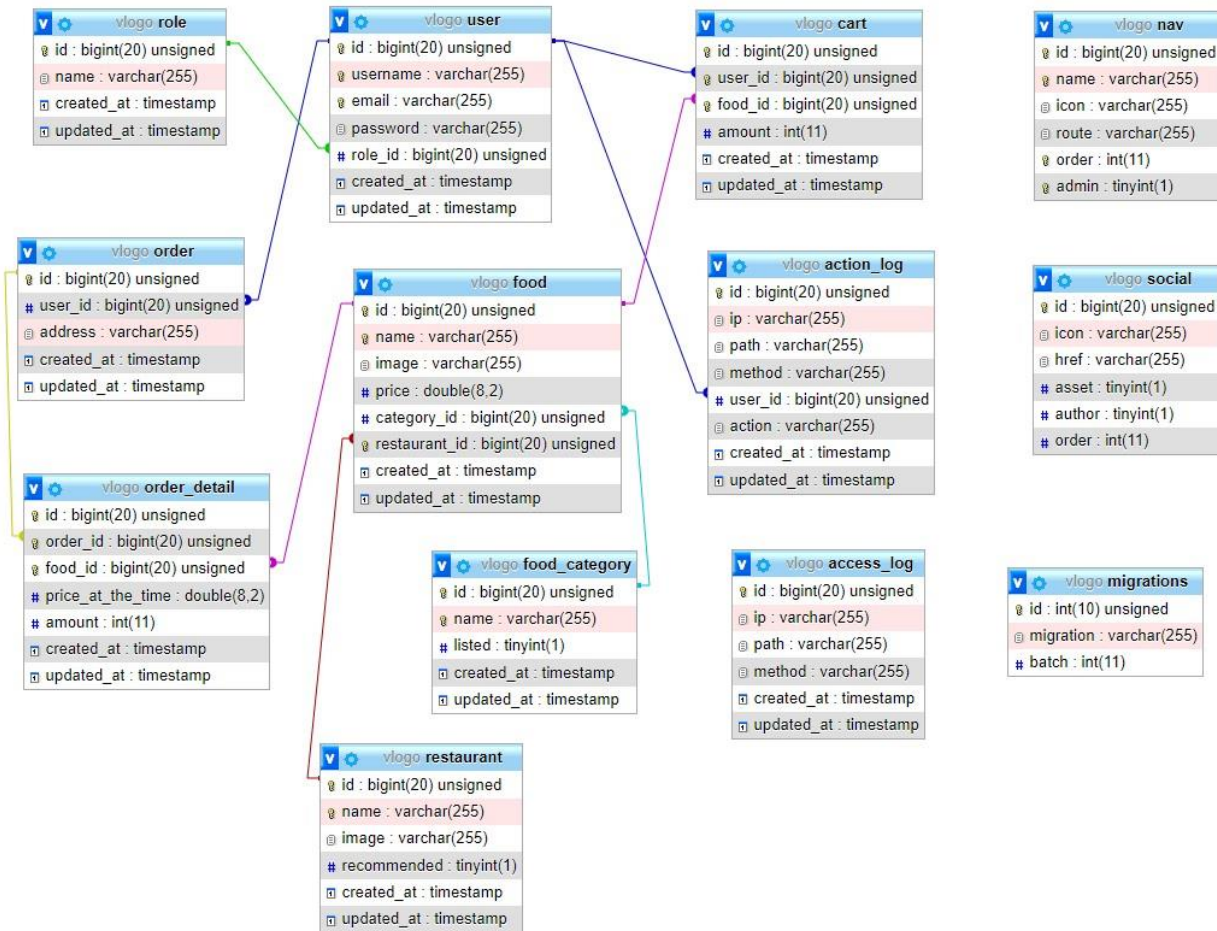
1.3 Template

Korisnički deo: Nije korišćen template

Admin panel: <https://adminlte.io/>

2. Organizacija

2.1 Dizajn baze podataka

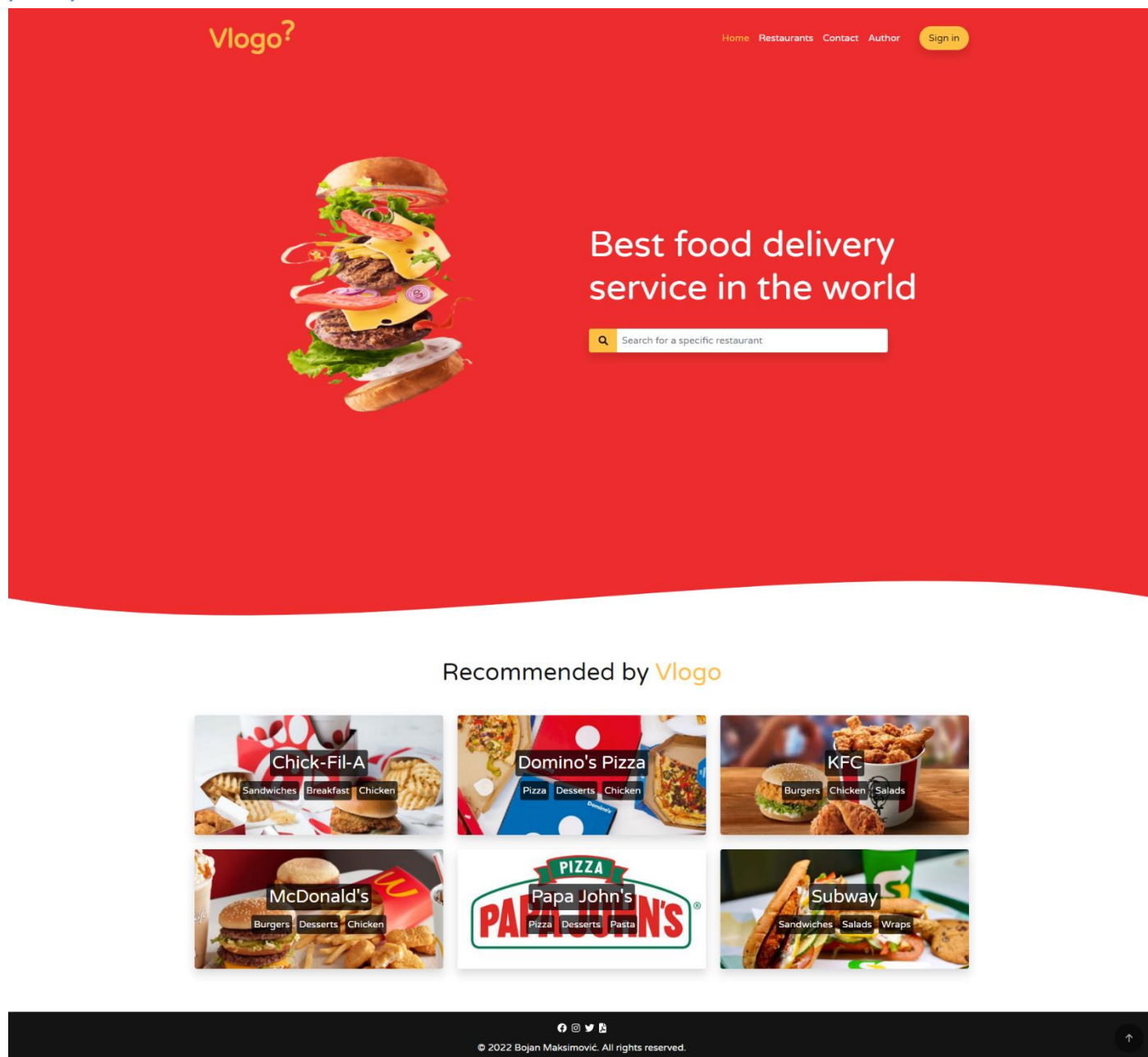


2.2 Mapa sajta

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://php3.bojanmaksimovic.com/</loc>
    <lastmod>2022-03-06</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.8</priority>
  </url>
  <url>
    <loc>http://php3.bojanmaksimovic.com/restaurants</loc>
    <lastmod>2022-03-06</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.8</priority>
  </url>
  <url>
    <loc>http://php3.bojanmaksimovic.com/contact</loc>
    <lastmod>2022-03-06</lastmod>
    <changefreq>yearly</changefreq>
    <priority>0.4</priority>
  </url>
  <url>
    <loc>http://php3.bojanmaksimovic.com/author</loc>
    <lastmod>2022-03-06</lastmod>
    <changefreq>yearly</changefreq>
    <priority>0.4</priority>
  </url>
  <url>
    <loc>http://php3.bojanmaksimovic.com/login</loc>
    <lastmod>2022-03-06</lastmod>
    <changefreq>yearly</changefreq>
    <priority>0.2</priority>
  </url>
  <url>
    <loc>http://php3.bojanmaksimovic.com/register</loc>
    <lastmod>2022-03-06</lastmod>
    <changefreq>yearly</changefreq>
    <priority>0.2</priority>
  </url>
</urlset>
```


2.3 Slike stranica i opis funkcionalnosti

/ ili /home



Na početnoj stranici korisnici imaju mogućnost pretrage restorana u ponudi uz pomoć ajax-a. Na ovoj stranici se takođe ispisuju neki od restorana koje vlasnici sajta preporučuju.

Vlogo?

[Home](#) [Restaurants](#) [Contact](#) [Author](#) [Sign in](#)

All food

Breakfast

Burgers

Chicken

Coffee

Desserts

Mexican

Pasta

Pizza

Salads

Sandwiches

Wraps

Restaurant name

Name ascending

Chick-Fil-A

Sandwiches Breakfast Chicken

Domino's Pizza

Pizza Desserts Chicken

KFC

Burgers Chicken Salads

Chipotle

Mexican Salads

Dunkin Donut's

Coffee Breakfast Desserts

Little Caesars

Pizza Chicken

1 2

© 2022 Bojan Maksimović. All rights reserved.

Na stranici restaurants je odrađeno filtriranje, sortiranje i paginacija restorana uz pomoć ajax-a.

10

Vlogo?

[Home](#) [Restaurants](#) [Contact](#) [Author](#) [korisnik1 ▾](#)


Restaurants / McDonald's

All food


[Beverages](#)
[Burgers](#)
[Chicken](#)
[Combos](#)
[Desserts](#)
[Sides](#)

Search by name


Name ascer ▾




Big Mac
\$2.99
[+ Add to order](#)




Big Mac Combo Meal
\$6.99
[+ Add to order](#)




Cheeseburger
\$1.99
[+ Add to order](#)




Cheeseburger Combo Meal
\$6.99
[+ Add to order](#)




Chicken McNuggets
\$3.99
[+ Add to order](#)



Coca Cola
\$0.99
[+ Add to order](#)



Double Cheeseburger
\$2.99
[+ Add to order](#)



Double Quarter Pounder
\$2.99
[+ Add to order](#)

12

Your order

Delivery address

Delivery address

Big Mac Combo Meal - \$6.99 - 1 + x

Cheeseburger - \$1.99 - 1 + x

Total: \$8.98

Place order

[f](#) [i](#) [t](#) [l](#)

© 2022 Bojan Maksimović. All rights reserved.

↑

Na ovoj stranici ispisuju se sve kategorija proizvoda unutar izabranog restorana i svi proizvodi koji pripadaju njemu. Takođe je omogućeno filtriranje, sortiranje i paginacija proizvoda kao i to da korisnik može da dodaje proizvode u svoju porudžbinu i da ih menja. Sadržaj porudžbine se čuva za svaki restoran zasebno jer korisnik ne može u jednoj porudžbini da naruči proizvode iz više restorana. Ova stranica je zaštićena middleware-om koji proverava da li je korisnik ulogovan.

11

Contact us

If you wish to reach out to us, you can do so via social media, the information listed below, or by filling out the contact form.

☎ +381 60/123-4567

✉ bojan.maxim075@gmail.com

Email address

Subject

Message



© 2022 Bojan Maksimović. All rights reserved.



Na kontakt stranici se nalazi forma za kontaktiranje administratora sa proverom na klijentskoj i serverskoj strani. Ako je provera uspešna administratoru se podaci šalju na mejl.



Bojan Maksimović 92/19
Web Developer

About me

Hello! I am Bojan, an aspiring back-end web developer. I'm a driven individual with the ability to learn and collaborate in rapidly changing environments and compositions. I am eager to tackle web development challenges to achieve lasting impacts on user experience.



[Portfolio Website](#)



© 2022 Bojan Maksimović. All rights reserved.



Stranica sa slikom I nekim podacima o autoru.

Log in

Email address

name@example.com

name@example.com

Password

Password

Log in

[Don't have an account? Register Instead.](#)



© 2022 Bojan Maksimović. All rights reserved.



Stranica sa formom za logovanje kao i linkom koji vodi do stranice za registrovanje. Podaci iz forme se proveravaju na klijentu i serveru. Stranica je zaštićena middleware-om koji proverava da korisnike NIJE ulogovan.

Register

Username

Your username cannot contain special characters other than '.', must contain at least one alphabetical character and be between 8 and 25 characters long

Email address

Password

Confirm password

Register

Already have an account? [Log in instead.](#)

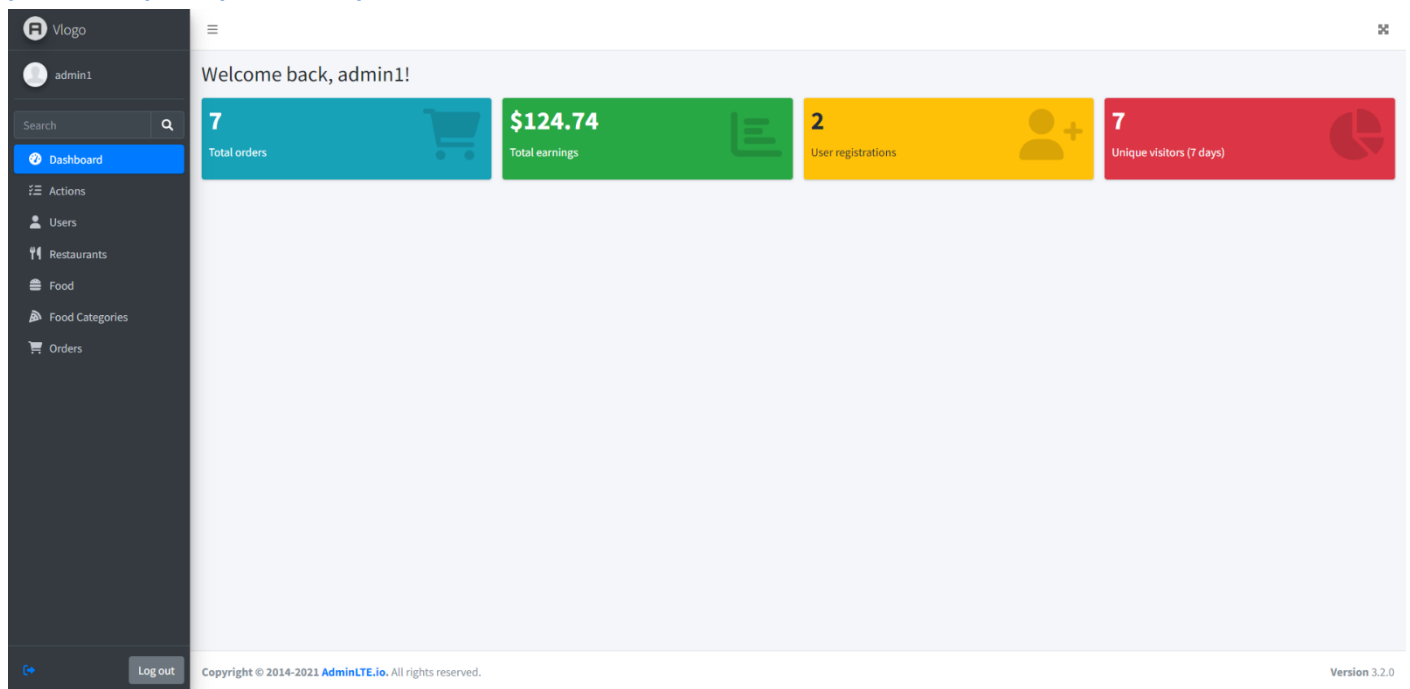


© 2022 Bojan Maksimović. All rights reserved.



Stranica sa formom za registrovanje kao i linkom koji vodi do stranice za logovanje. Podaci iz forme se proveravaju na klijentu i serveru. Stranica je zaštićena middleware-om koji proverava da korisnike NIJE ulogovan.

/admin/ ili /admin/dashboard



Kao i sve stranice u admin panelu, stranica je zaštićena middleware-ovima koji proveravaju da je korisnik ulogovan i da je uloga tog korisnika 'admin'.

Na ovoj stranici korisnik ima uvid u neke ključne statistike sajta kao što su: broj porudžbina, totalna zarada, broj registracija i broj jedinstvenih posetioca u poslednjih 7 dana.

Vlogo

admin1

Search

Dashboard

Actions

Users

Restaurants

Food

Food Categories

Orders

Log out

Food

From mm/dd/yyyy To mm/dd/yyyy Filter

Search:

#	IP	Path	Method	User	Action	Created At
1	127.0.0.1	logout	POST	admin1	User logged out.	2022-03-07 11:29:45
2	127.0.0.1	register	POST	bojanm00	User registered themselves.	2022-03-07 11:29:59
3	127.0.0.1	login	POST	bojanm00	User failed to log in.	2022-03-07 11:30:02
4	127.0.0.1	login	POST	bojanm00	User logged in.	2022-03-07 11:30:08
5	127.0.0.1	cart/add	POST	bojanm00	User added Barbeque Sauce (ID: 74) to their cart.	2022-03-07 11:30:23
6	127.0.0.1	cart/add	POST	bojanm00	User added Barbeque Sauce (ID: 74) to their cart.	2022-03-07 11:30:24
7	127.0.0.1	cart/add	POST	bojanm00	User added Chick-n-Strips (ID: 78) to their cart.	2022-03-07 11:30:25
8	127.0.0.1	cart/add	POST	bojanm00	User added Chick-n-Strips (ID: 78) to their cart.	2022-03-07 11:30:26
9	127.0.0.1	cart/add	POST	bojanm00	User added Chicken Sandwich (ID: 76) to their cart.	2022-03-07 11:30:27
10	127.0.0.1	cart/add	POST	bojanm00	User added Cobb Salad (ID: 79) to their cart.	2022-03-07 11:30:28

Showing 1 to 10 of 126 entries

Previous 1 2 3 4 5 ... 13 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Na ovoj stranici korisnici mogu da vide izveštaj o ključnim aktivnostima svih korisnika kao i da filtriraju ispisane aktivnosti po datumu.

Vlogo

admin1

Search

Dashboard

Actions

Users

Restaurants

Food

Food Categories

Orders

Log out

Users

Add new user

Search:

#	Username	Email	Role	Created At	Updated At	Edit	Delete
1	korisnik1	korisnik1@gmail.com	user	2022-03-07 11:29:31	N/A	Edit	Delete
2	korisnik2	korisnik2@gmail.com	user	2022-03-07 11:29:31	N/A	Edit	Delete
3	admin1	admin1@gmail.com	admin	2022-03-07 11:29:31	N/A	Edit	Delete
4	admin2	admin2@gmail.com	admin	2022-03-07 11:29:31	N/A	Edit	Delete
5	bojanm00	bojanm00@gmail.com	admin	2022-03-07 11:29:59	2022-03-08 00:10:14	Edit	Delete
6	bojanm01	bojanm01@gmail.com	user	2022-03-07 11:35:56	N/A	Edit	Delete
7	bojanm02	bojanm02@gmail.com	user	2022-03-07 20:40:35	2022-03-07 20:41:52	Edit	Delete

Showing 1 to 7 of 7 entries

Previous1Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Na ovoj stranici korisnici mogu da vide sve registroane korisnike na sajtu, da ih menjaju, brišu ili da dodaju nove. Ako korisnik izmeni svoje podatke, sajt će ga izlogovati.

Vlogo

admin1

Search

Dashboard

Actions

Users

Restaurants

Food

Food Categories

























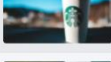





Orders

Log out

Restaurants

Add new restaurant

Search:

#	Name	Image	Active	Created At	Updated At	Edit	Delete
1	Chick-Fil-A		1	2022-03-07 11:29:31	2022-03-07 11:32:28		
2	Chipotle		1	2022-03-07 11:29:31	N/A		
3	Domino's Pizza		1	2022-03-07 11:29:31	N/A		
4	Dunkin Donut's		1	2022-03-07 11:29:31	N/A		
5	KFC		1	2022-03-07 11:29:31	N/A		
6	Little Caesars		1	2022-03-07 11:29:31	N/A		
7	McDonald's		1	2022-03-07 11:29:31	N/A		
8	Papa John's		1	2022-03-07 11:29:31	N/A		
9	Starbucks		1	2022-03-07 11:29:31	N/A		
10	Subway		1	2022-03-07 11:29:31	N/A		

Showing 1 to 10 of 10 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Na ovoj stranici korisnici mogu da vide sve restorane, da ih menjaju, brišu i da kreiraju nove. Pri dodavanju novog restorana ili izmeni postojećeg, uploadovana slika se resize-uje.

Vlogo

admin1

Search

Dashboard

Actions

Users

Restaurants

Food

Food Categories































Orders

Log out

Food

Add new food

Search:

ID	Name	Image	Restaurant	Category	Price	Created At	Updated At	Edit	Delete
1	Big Mac		McDonald's	Burgers	\$2.99	2022-03-07 11:29:31	2022-03-07 11:32:51		
2	Big Mac Combo Meal		McDonald's	Combos	\$6.99	2022-03-07 11:29:31	N/A		
3	Cheeseburger		McDonald's	Burgers	\$1.99	2022-03-07 11:29:31	N/A		
4	Cheeseburger Combo Meal		McDonald's	Combos	\$6.99	2022-03-07 11:29:31	N/A		
5	Chicken McNuggets		McDonald's	Chicken	\$3.99	2022-03-07 11:29:31	N/A		
6	Coca Cola		McDonald's	Beverages	\$0.99	2022-03-07 11:29:31	N/A		
7	Double Cheeseburger		McDonald's	Burgers	\$2.99	2022-03-07 11:29:31	N/A		
8	Double Quarter Pounder		McDonald's	Burgers	\$2.99	2022-03-07 11:29:31	N/A		
9	Fanta		McDonald's	Beverages	\$0.99	2022-03-07 11:29:31	N/A		
10	Hamburger		McDonald's	Burgers	\$1.99	2022-03-07 11:29:31	N/A		

Showing 1 to 10 of 148 entries

Previous12345...15Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0

Na ovoj stranici korisnici mogu da vide sve proizvode iz restorana, da ih menjaju, brišu i da kreiraju nove. Pri dodavanju novog proizvoda ili izmeni postojećeg, upload-ovana slika se resize-uje.

Vlogo

admin1

Search

Dashboard

Actions

Users

Restaurants

Food

Food Categories

Orders

Log out

Food categories

Add new category

Search:

#	Name	Listed	Created At	Updated At	Edit	Delete
1	Beverages	0	2022-03-07 11:29:31	2022-03-07 11:34:00	Edit	Delete
2	Breakfast	1	2022-03-07 11:29:31	N/A	Edit	Delete
3	Burgers	1	2022-03-07 11:29:31	N/A	Edit	Delete
4	Chicken	1	2022-03-07 11:29:31	N/A	Edit	Delete
5	Coffee	1	2022-03-07 11:29:31	N/A	Edit	Delete
6	Combos	0	2022-03-07 11:29:31	N/A	Edit	Delete
7	Desserts	1	2022-03-07 11:29:31	N/A	Edit	Delete
8	Extras	0	2022-03-07 11:29:31	N/A	Edit	Delete
9	Mexican	1	2022-03-07 11:29:31	N/A	Edit	Delete
10	Pasta	1	2022-03-07 11:29:31	N/A	Edit	Delete

Showing 1 to 10 of 15 entries

Previous 1 2 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Na ovoj stranici korisnici mogu da vide sve kategorije proizvoda, da ih menjaju, brišu i da kreiraju nove.

Vlogo

admin1

Search

Dashboard

Actions

Users

Restaurants

Food

Food Categories

Orders

Log out

Orders

Search:

ID	User	Restaurant	Address	Total Price	Created At	Details
1	bojanm00	Chick-Fil-A	Ljubljanska 11/3	\$24.43	2022-03-07 11:30:33	Details
2	bojanm00	Little Caesars	Dimitrija Tucovica 154A	\$26.96	2022-03-07 11:30:47	Details
3	bojanm01	Papa John's	Ljubljanska 11/3	\$19.46	2022-03-07 11:36:17	Details
4	bojanm00	Chick-Fil-A	Ljubljanska 11/3	\$14.95	2022-03-07 20:39:01	Details
5	bojanm02	Dunkin Donut's	Dimitrija Tucovica 154A	\$11.98	2022-03-07 20:41:33	Details
6	bojanm00	Starbucks	Dimitrija Tucovica 154A	\$11.97	2022-03-07 21:05:35	Details
7	admin1	Little Caesars	Bele Bartoka 28	\$14.99	2022-03-07 21:16:35	Details

Showing 1 to 7 of 7 entries

Previous1Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Na ovoj stranici korisnici imaju uvid I sve porudžbine. Mogu da kliknu na ikonicu za detalje da vide šta je sve bilo poručeno.

3. Kodovi

3.1 Routes

web.php

```
<?php

use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', [\App\Http\Controllers\HomeController::class, "index"]->name("home"));
Route::get('/home', function() {
    return redirect()->route("home");
});
Route::resource('/restaurants', \App\Http\Controllers\RestaurantController::class,
['only' => ['index', 'show']]);
Route::post('/restaurants/search', [\App\Http\Controllers\RestaurantController::class,
'search']->name("restaurants.search"));
Route::post('/restaurants/filter', [\App\Http\Controllers\RestaurantController::class,
'getFiltered']->name("restaurants.filter"));
Route::post('/restaurants/food', [\App\Http\Controllers\RestaurantController::class,
'filterFood']->name("restaurants.food"));
Route::get('/contact', [\App\Http\Controllers>ContactController::class, "index"]->
name("contact"));
Route::post('/contact', [\App\Http\Controllers\MailController::class, "contact"]->
name("mail.contact"));
Route::get('/author', [\App\Http\Controllers\AuthorController::class, "index"]->
name("author"));
Route::middleware('isNotLoggedIn')->group(function() {
    Route::get('/login', [\App\Http\Controllers\AuthController::class, "login"]->
name("form.login"));
    Route::get('/register', [\App\Http\Controllers\AuthController::class, "register"]->
name("form.register"));
    Route::post('/login', [\App\Http\Controllers\AuthController::class, "doLogin"]->
name("login"));
    Route::post('/register', [\App\Http\Controllers\AuthController::class,
"doRegister"]->name("register"));
});
Route::middleware('isLoggedIn')->group(function() {
    Route::post('/logout', [\App\Http\Controllers\AuthController::class, "logout"]->
name("logout"));
    Route::post('/cart/add', [\App\Http\Controllers\OrderController::class,
"addItem"]->name("cart.add"));
    Route::post('/cart/change', [\App\Http\Controllers\OrderController::class,
"change"]->name("cart.change"));
    Route::post('/cart/items', [\App\Http\Controllers\OrderController::class,
"getItems"]->name("cart.items"));
    Route::post('/orders/store', [\App\Http\Controllers\OrderController::class,
"store"]->name('orders.store'));
});
Route::prefix('/admin')->name('admin.')->middleware(['isLoggedIn', 'isAdmin'])->
group(function() {
```

```

Route::get('/', [\App\Http\Controllers\AdminController::class, 'index'])->
>name('dashboard');
Route::get('/dashboard', function() {
    return redirect()->route('admin.dashboard');
});
Route::resource('/users', \App\Http\Controllers\UserController::class, ['except' =>
['show']]);
Route::get('/restaurants', [\App\Http\Controllers\AdminController::class,
'restaurants'])->name('restaurants');
Route::resource('/restaurants', \App\Http\Controllers\RestaurantController::class,
['except' => ['index', 'show']]);
Route::resource('/food', \App\Http\Controllers\FoodController::class, ['except' =>
'show']);
//Route::get('/categories', [\App\Http\Controllers\AdminController::class,
'foodCategories'])->name('categories');
Route::resource('/categories', \App\Http\Controllers\FoodCategoryController::class,
['except' => ['show']]);
Route::get('/orders', [\App\Http\Controllers\AdminController::class, 'orders'])->
>name('orders');
Route::get('/orders/{id}', [\App\Http\Controllers\OrderController::class,
'details'])->name('orders.details');
Route::get('/actions', [\App\Http\Controllers\AdminController::class, 'actions'])->
>name('actions');
});

```


3.2 Kernel

Kernel.php

```
<?php

namespace App\Http;

use App\Http\Middleware\LogAccess;
use App\Http\Middleware\IsAdmin;
use App\Http\Middleware\IsLoggedIn;
use App\Http\Middleware\IsNotLoggedIn;
use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
     * These middleware are run during every request to your application.
     *
     * @var array<int, class-string|string>
     */
    protected $middleware = [
        // \App\Http\Middleware\TrustHosts::class,
        \App\Http\Middleware\TrustProxies::class,
        \Fruitcake\Cors\HandleCors::class,
        \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,
        \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array<string, array<int, class-string|string>>
     */
    protected $middlewareGroups = [
        'web' => [
            \App\Http\Middleware\EncryptCookies::class,
            \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
            \Illuminate\Session\Middleware\StartSession::class,
            // \Illuminate\Session\Middleware\AuthenticateSession::class,
            \Illuminate\View\Middleware\ShareErrorsFromSession::class,
            \App\Http\Middleware\VerifyCsrfToken::class,
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
            LogAccess::class
        ],

        'api' => [
            //
            \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
            'throttle:api',
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],
    ];

    /**
     * The application's route middleware.
     *
     * These middleware may be assigned to groups or used individually.

```

```

*
* @var array<string, class-string|string>
*/
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
    'isLoggedIn' => IsLoggedIn::class,
    'isNotLoggedIn' => IsNotLoggedIn::class,
    'isAdmin' => IsAdmin::class
];
}

```

3.3 Controllers

AdminController

```
<?php

namespace App\Http\Controllers;

use App\Models\AccessLog;
use App\Models\ActionLog;
use App\Models\Food;
use App\Models\FoodCategory;
use App\Models\Nav;
use App\Models\Order;
use App\Models\Restaurant;
use App\Models\Role;
use App\Models\User;
use Illuminate\Http\Request;

class AdminController extends MyController
{
    private $navModel;

    public function __construct()
    {
        parent::__construct();
        $this->navModel = new Nav();
        $this->data['nav'] = $this->navModel->getAdminNav();
    }

    public function index() {
        $orderModel = new Order();
        $accessLogModel = new AccessLog();

        $this->data['infoBoxes'] = [
            [
                "value" => $orderModel->getCount(),
                "text" => "Total orders",
                "colorClass" => "bg-info",
                "icon" => "fas fa-shopping-cart"
            ],
            [
                "value" => "$" . round($orderModel->getTotalEarnings(), 2),
                "text" => "Total earnings",
                "colorClass" => "bg-success",
                "icon" => "fas fa-bar-chart"
            ],
            [
                "value" => $this->actionLogModel->getRegistrationsCount(),
                "text" => "User registrations",
                "colorClass" => "bg-warning",
                "icon" => "fas fa-user-plus"
            ],
            [
                "value" => $accessLogModel->getUniqueVisitors(),
                "text" => "Unique visitors (7 days)",
                "colorClass" => "bg-danger",
                "icon" => "fas fa-chart-pie"
            ]
        ];
        return view('pages.admin.dashboard', $this->data);
    }

    public function actions(Request $request) {
        $actionLogModel = new ActionLog();
    }
}
```

```
$this->data['actions'] = $actionLogModel->get($request);  
  
return view('pages.admin.actions.index', $this->data);  
}  
  
public function restaurants() {  
    $restaurantModel = new Restaurant();  
    $this->data['restaurants'] = $restaurantModel->getAll();  
  
    return view('pages.admin.restaurants.index', $this->data);  
}  
  
public function orders() {  
    $orderModel = new Order();  
    $this->data['orders'] = $orderModel->get();  
  
    return view('pages.admin.orders.index', $this->data);  
}  
}
```

AuthController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\LoginUserRequest;
use App\Http\Requests\RegisterUserRequest;
use App\Models\User;
use Illuminate\Http\Request;

class AuthController extends ClientController
{
    public function login() {
        return view("pages.client.login", $this->data);
    }

    public function register() {
        return view("pages.client.register", $this->data);
    }

    public function doLogin(LoginUserRequest $request) {
        try {
            $email = $request->get("email");
            $password = $request->get("password");

            $userModel = new User();
            $user = $userModel->where("email", "=", $email);

            if(!$user || $user->password !== md5($password)) {
                $userId = $user != null ? $user->id : null;
                $this->logAction("User failed to log in.", $request, $userId);
                return redirect()->back()->withErrors(["Invalid credentials."]);
            }

            $request->session()->put("user", $user);

            $this->logAction("User logged in.", $request);

            return redirect()->route("home");
        } catch (\Exception $ex) {
            $uniqueId = uniqid();
            \Log::error($uniqueId . " " . $ex->getMessage());
            return redirect()->back()->withErrors(["We encountered an error.", 'Error ID: ' . $uniqueId]);
        }
    }

    public function doRegister(RegisterUserRequest $request) {
        try {
            $userModel = new User();
            $userId = $userModel->insert($request->get("username"), $request->get("email"), $request->get("password"), 1);

            $this->logAction("User registered themselves.", $request, $userId);

            return redirect()->route("form.login")->with(["email" => $request->get("email"), "messages" => ["Your account was created successfully. You can log in now."]]);
        } catch (Exception $e) {
            $uniqueId = uniqid();
            \Log::error($uniqueId . " " . $e->getMessage());
            return redirect()->back()->withErrors(["We encountered an error. Please try again later.", 'Error ID: ' . $uniqueId]);
        }
    }
}
```

```
public function logout(Request $request) {  
    try {  
        $userId = $request->session()->get("user")->id;  
        $request->session()->remove("user");  
        $this->logAction("User logged out.", $request, $userId);  
    } catch(\Exception $e) {  
        $uniqueId = uniqid();  
        \Log::error($uniqueId . " " . $e->getMessage());  
    }  
  
    return redirect()->back();  
}
```

AuthorController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AuthorController extends ClientController
{
    public function __construct() {
        parent::__construct();
        $this->data['authorSocials'] = $this->socialsModel->getAuthorSocials();
    }

    public function index() {
        return view("pages.client.author", $this->data);
    }
}
```

ClientController

```
<?php

namespace App\Http\Controllers;

use App\Models\Nav;
use App\Models\Restaurant;
use App\Models\Social;
use Illuminate\Http\Request;

class ClientController extends MyController
{
    protected $socialsModel;
    protected $navModel;

    public function __construct() {
        parent::__construct();

        $this->socialsModel = new Social();
        $this->data["socials"] = $this->socialsModel->getSocials();

        $this->navModel = new Nav();
        $this->data['nav'] = $this->navModel->getClientNav();
    }
}
```


ContactController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ContactController extends ClientController
{
    public function index() {
        return view("pages.client.contact", $this->data);
    }
}
```

FoodCategoryController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\StoreCategoryRequest;
use App\Http\Requests\UpdateCategoryRequest;
use App\Models\FoodCategory;
use Illuminate\Http\Request;

class FoodCategoryController extends AdminController
{
    private $foodCategoryModel;

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View
     */
    public function __construct()
    {
        parent::__construct();
        $this->foodCategoryModel = new FoodCategory();
    }

    public function index() {
        $foodCategoryModel = new FoodCategory();
        $this->data['categories'] = $foodCategoryModel->get();

        return view('pages.admin.foodCategories.index', $this->data);
    }

    public function create()
    {
        $this->data['pages'] = [
            [
                "name" => "Categories",
                "route"=> "admin.categories.index"
            ],
            [
                "name" => "Add new category"
            ]
        ];

        return view('pages.admin.foodCategories.create', $this->data);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\RedirectResponse
     */
    public function store(StoreCategoryRequest $request)
    {
        try {
            $this->foodCategoryModel->insert($request->get('name'), $request->has('listed'));

            $this->logAction("Admin created the " . $request->get('name') . " food category.", $request);

            return redirect()->route('admin.categories.index')->with('messages', ["Category " . $request->get('name') . " was created successfully."]);
        }
    }
}
```

```

    } catch(\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
        return redirect()->back()->withErrors(['We encountered an error.', 'Error
ID: ' . $uniqueId]);
    }
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View
 */
public function edit($id)
{
    $this->data['category'] = $this->foodCategoryModel->find($id);
    $this->data['pages'] = [
        [
            "name" => "Categories",
            "route"=> "admin.categories.index"
        ],
        [
            "name" => "Edit category"
        ]
    ];

    return view('pages.admin.foodCategories.edit', $this->data);
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\RedirectResponse
 */
public function update(UpdateCategoryRequest $request, $id)
{
    try {
        $this->foodCategoryModel->myUpdate($id, $request->get('name'), $request-
>has('listed'));

        $this->logAction("Admin updated the " . $request->get('name') . " food
category.", $request);

        return redirect()->route('admin.categories.index')->with('messages',
['Category was changed successfully.']);
    } catch(\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
        return redirect()->back()->withErrors(["We encountered an error.", "Error
ID: " . $uniqueId]);
    }
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\RedirectResponse
 */
public function destroy($id)

```

```
{
    try {
        $this->foodCategoryModel->myDelete($id);

        $this->logAction("Admin deleted a food category.", request());

        return redirect()->route('admin.categories.index')->with('messages',
["Category was deleted successfully."]);
    } catch(\Exception $ex) {
        return redirect()->back()->withErrors(["We encountered an error."]);
    }
}
```

FoodController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\StoreFoodRequest;
use App\Http\Requests\UpdateFoodRequest;
use App\Models\Food;
use App\Models\FoodCategory;
use App\Models\ImageUpload;
use App\Models\Restaurant;
use Doctrine\DBAL\Driver\Exception;
use Illuminate\Http\Request;

class FoodController extends AdminController
{
    private $foodModel;

    public function __construct() {
        parent::__construct();
        $this->foodModel = new Food();
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View
     */
    public function index()
    {
        $this->data['food'] = $this->foodModel->get();

        return view('pages.admin.food.index', $this->data);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View
     */
    public function create()
    {
        $this->data['pages'] = [
            [
                'name' => "Food",
                'route' => 'admin.food.index'
            ],
            [
                'name' => 'Add new food'
            ]
        ];

        $restaurantsModel = new Restaurant();
        $this->data['restaurants'] = $restaurantsModel->getAll();

        $categoriesModel = new FoodCategory();
        $this->data['categories'] = $categoriesModel->get();

        return view('pages.admin.food.create', $this->data);
    }
}
```

```

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\RedirectResponse
 */
public function store(StoreFoodRequest $request)
{
    try {
        $imageUploadModel = new ImageUpload();
        $imageName = $imageUploadModel->uploadResized($request,
'/assets/images/food/', 150, 150);

        $this->foodModel->insert(
            $request->get('name'),
            $imageName,
            $request->get('restaurant'),
            $request->get('category'),
            round($request->get('price'), 2)
        );

        $this->logAction("Admin created the " . $request->get('name') . " food.",
$request);

        return redirect()->route('admin.food.index')->with('messages', ['Food ' .
$request->get('name') . ' was created successfully.']);
    } catch (\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
        return redirect()->back()->withErrors(['We encountered an error.', 'Error
ID: ' . $uniqueId]);
    }
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Contracts\View\Factory|\Illumi
nate\Contracts\View\View
 */
public function edit($id)
{
    $this->data['pages'] = [
        [
            'name' => "Food",
            'route' => 'admin.food.index'
        ],
        [
            'name' => 'Edit food'
        ]
    ];
    $this->data['food'] = $this->foodModel->find($id);

    $restaurantsModel = new Restaurant();
    $this->data['restaurants'] = $restaurantsModel->getAll();

    $categoriesModel = new FoodCategory();
    $this->data['categories'] = $categoriesModel->get();

    return view('pages.admin.food.edit', $this->data);
}

```

```

    * Update the specified resource in storage.
    *
    * @param \Illuminate\Http\Request $request
    * @param int $id
    * @return \Illuminate\Http\RedirectResponse
    */
    public function update(UpdateFoodRequest $request, $id)
    {
        try {
            $imageName = null;

            if($request->has('image') && $request->file('image')) {
                $imageUploadModel = new ImageUpload();
                $imageName = $imageUploadModel->uploadResized($request,
'/assets/images/food/', 150, 150);
            }

            $this->foodModel->myUpdate(
                $id,
                $request->get('name'),
                $request->get('restaurant'),
                $request->get('category'),
                round($request->get('price'), 2),
                $imageName
            );

            $this->logAction("Admin updated the " . $request->get('name') . " food.",
$request);

            return redirect()->route('admin.food.index')->with('messages', ['Food was
updated successfully.']);
        } catch (\Exception $ex) {
            $uniqueId = uniqid();
            \Log::error($uniqueId . " " . $ex->getMessage());
            return redirect()->back()->withErrors(['We encountered an error.', 'Error
ID: ' . $uniqueId]);
        }
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\RedirectResponse
     */
    public function destroy($id)
    {
        try {
            $this->foodModel->myDelete($id);

            $this->logAction("Admin deleted a food.", request());

            return redirect()->route('admin.food.index')->with('messages', ['Food was
deleted successfully.']);
        } catch (\Exception $ex) {
            $uniqueId = uniqid();
            \Log::error($uniqueId . " " . $ex->getMessage());
            return redirect()->back()->withErrors(['We encountered an error.', 'Error
ID: ' . $uniqueId]);
        }
    }
}

```

HomeController

```
<?php

namespace App\Http\Controllers;

use App\Models\Restaurant;
use Illuminate\Http\Request;

class HomeController extends ClientController
{
    private $restaurantModel;

    public function __construct()
    {
        parent::__construct();
        $this->restaurantModel = new Restaurant();
        $this->data["recommendedRestaurants"] = $this->restaurantModel->get(true);
    }

    public function index() {
        return view("pages.client.home", $this->data);
    }
}
```


MailController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\ContactRequest;
use App\Mail\ContactMail;
use Illuminate\Http\Request;

class MailController extends MyController
{
    public function contact(ContactRequest $request) {
        try {
            \Mail::to("bojanm245@gmail.com")->send(new ContactMail($request-
>get("email"), $request->get("subject"), $request->get("message")));

            $this->logAction("User sent a contact message.", $request);

            return redirect()->back()->with("messages", ["Your email was sent
successfully."]);
        } catch(Exception $ex) {
            $uniqueId = uniqid();
            \Log::error($uniqueId . " " . $ex->getMessage());
            return redirect()->back()->withErrors(["We encountered an error.", "Error
ID: " . $uniqueId]);
        }
    }
}
```

MyController

```
<?php

namespace App\Http\Controllers;

use App\Models\ActionLog;
use Illuminate\Http\Request;

class MyController extends Controller
{
    protected $data;
    protected $actionLogModel;

    public function __construct() {
        $this->actionLogModel = new ActionLog();
    }

    protected function logAction($action, Request $request, $user_id = null) {
        $ip = $request->ip();
        $path = $request->path();
        $method = $request->method();
        $user_id = $request->session()->has("user") ? $request->session()->get("user")->id : $user_id;

        try {
            $this->actionLogModel->insert($ip, $path, $method, $action, $user_id);
        } catch (\Exception $ex) {
            \Log::error($ex->getMessage());
        }
    }
}
```

OrderController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\ChangeCartRequest;
use App\Http\Requests\StoreOrderRequest;
use App\Models\Cart;
use App\Models\Food;
use App\Models\Order;
use Illuminate\Http\Request;

class OrderController extends ClientController
{
    private $orderModel;
    private $cartModel;

    public function __construct()
    {
        parent::__construct();
        $this->orderModel = new Order();
        $this->cartModel = new Cart();
    }

    public function addItem(Request $request) {
        try {
            $foodId = $request->get('id');
            $userId = $request->session()->get('user')->id;

            $this->cartModel->addItem($userId, $foodId);

            $foodModel = new Food();
            $this->logAction('User added ' . $foodModel->find($foodId)->name . " (ID: $foodId)" . ' to their cart.', $request, $userId);

            return response()->json('Item was added to your order successfully.');
```

```
        } catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return response('We encountered an error.', 500);
        }
    }

    public function change(ChangeCartRequest $request) {
        try {
            $cartId = $request->get('id');
            $action = $request->get('action');
            $userId = $request->session()->get('user')->id;

            $result = $this->cartModel->change($userId, $cartId, $action);

            if($result) {
                $this->logAction('User altered their cart.', $request, $userId);

                return response()->json('Action completed successfully.');
```

```
            }

            return response('We encountered an error.', 500);
        } catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return response('We encountered an error.', 500);
        }
    }

    public function getItem(Request $request) {
```

```

        try {
            $userId = $request->session()->get('user')->id;
            $restaurantId = $request->get('restaurantId');

            $items = $this->cartModel->getItems($userId, $restaurantId);

            return response()->json($items);
        } catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return response('We encountered an error.', 500);
        }
    }

    public function store(StoreOrderRequest $request) {
        try {
            $userId = $request->session()->get('user')->id;
            $restaurantId = $request->get('restaurantId');
            $address = $request->get('address');
            $items = $this->cartModel->getItems($userId, $restaurantId);

            if(!count($items))
                return redirect()->back()->withErrors(['Your order is empty.']);

            \DB::beginTransaction();
            $orderId = $this->orderModel->store($userId, $address, $items);
            $this->cartModel->empty($items);
            \DB::commit();

            $this->logAction('User created an order. Order id: ' . $orderId, $request,
$userId);

            return redirect()->back()->with('messages', ['Your order was created
successfully.']);
        } catch (\Exception $ex) {
            \DB::rollBack();
            \Log::error($ex->getMessage());
            return redirect()->back()->withErrors(['We encountered an error.']);
        }
    }

    public function details($id) {
        $this->data['details'] = $this->orderModel->details($id);
        $this->data['nav'] = $this->navModel->getAdminNav();
        $this->data['pages'] = [
            [
                "name" => "Orders",
                "route"=> "admin.orders"
            ],
            [
                "name" => "Order details"
            ]
        ];

        return view('pages.admin.orders.details', $this->data);
    }
}

```

RestaurantController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\StoreRestaurantRequest;
use App\Http\Requests\UpdateRestaurantRequest;
use App\Models\FoodCategory;
use App\Models\ImageUpload;
use App\Models\Restaurant;
use Illuminate\Http\Request;
use Intervention\Image\Facades\Image;

class RestaurantController extends ClientController
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    private $restaurantModel;
    private $foodCategoryModel;

    public function __construct()
    {
        parent::__construct();

        $this->middleware('isLoggedIn', ['except' => ['index', 'getFiltered',
'search']]);
        $this->middleware(['isLoggedIn', 'isAdmin'], ['except' => ['index', 'show',
'filterFood', 'getFiltered', 'search']]);

        $this->restaurantModel = new Restaurant();
        $this->foodCategoryModel = new FoodCategory();

        $this->data['foodCategories'] = $this->foodCategoryModel->get(true);
    }

    public function index()
    {
        $this->data["restaurants"] = $this->restaurantModel->get();

        return view("pages.client.restaurants.index", $this->data);
    }

    public function search(Request $request) {
        try {
            $keyword = $request->get("keyword") == null ? "" : $request-
>input("keyword");

            $restaurants = $this->restaurantModel->search($keyword);

            return response()->json($restaurants);
        } catch (\Exception $ex) {
            $uniqueId = uniqid();
            \Log::error($uniqueId . " " . $ex->getMessage());

            return response("We encountered an error.", 500);
        }
    }

    public function getFiltered(Request $request) {
        try {
            $keyword = $request->get("keyword") == null ? "" : $request-
```

```

>input("keyword");
    $sortOrder = $request->get("sortOrder");
    $categories = $request->get("categories") == null ? [] : $request-
>input("categories");

    $restaurants = $this->restaurantModel->getFiltered($keyword, $sortOrder,
$categories);

    $response = [
        "restaurants" => $restaurants,
        "views" => []
    ];

    foreach($restaurants->items() as $r) {
        $response['views'][] = \View::make("components.client.restaurant", [
            "id" => $r->id,
            "name" => $r->name,
            "image" => $r->image,
            "categories" => $r->topCategories,
            "lg" => 6,
            "md" => 6
        ])->render();
    }

    return response()->json($response);
} catch(\Exception $ex) {
    $uniqueId = uniqid();
    \Log::error($uniqueId . " " . $ex->getMessage());

    return response("We encountered an error.", 500);
}

public function filterFood(Request $request) {
    try {
        $id = $request->get("id");
        $keyword = $request->get("keyword") == null ? "" : $request-
>input("keyword");
        $sortOrder = $request->get("sortOrder");
        $categories = $request->get("categories") == null ? [] : $request-
>input("categories");

        $food = $this->restaurantModel->filterFood($id, $keyword, $sortOrder,
$categories);

        $response = [
            "food" => $food,
            "views" => []
        ];

        foreach($food->items() as $f) {
            $response['views'][] = \View::make("components.client.food", [
                "id" => $f->id,
                "name" => $f->name,
                "image" => $f->image,
                "price" => $f->price,
                "md" => 6,
                "lg" => 6
            ])->render();
        }

        return response()->json($response);
    } catch(\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
    }
}

```

```

        return response("We encountered an error.", 500);
    }
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View
 */
public function create()
{
    $this->data['nav'] = $this->navModel->getAdminNav();
    $this->data['restaurants'] = $this->restaurantModel->getAll();
    $this->data['pages'] = [
        [
            'name' => 'Restaurants',
            'route' => 'admin.restaurants'
        ],
        [
            'name' => 'Add new restaurant'
        ]
    ];

    return view('pages.admin.restaurants.create', $this->data);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\RedirectResponse
 */
public function store(StoreRestaurantRequest $request)
{
    try {
        $imageUploadModel = new ImageUpload();
        $imageName = $imageUploadModel->uploadResized($request,
'/assets/images/restaurants/', 405);

        $this->restaurantModel->insert(
            $request->get('name'),
            $imageName,
            $request->has('recommended'),
            $request->has('active')
        );

        $this->logAction("Admin created the " . $request->get('name') . "
restaurant.", $request);

        return redirect()->route('admin.restaurants')->with('messages',
['Restaurant ' . $request->get('name') . ' was created successfully.']);
    } catch (\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
        return redirect()->back()->withErrors(['We encountered an error.', 'Error
ID: ' . $uniqueId]);
    }
}

/**
 * Display the specified resource.
 *

```

```

        * @param int $id
        * @return
\Illuminate\Contracts\Foundation\Application|Illuminate\Contracts\View\Factory|Illuminate\Contracts\View\View
    */
    public function show($id)
    {
        $restaurant = $this->restaurantModel->find($id);
        $this->data['restaurant'] = $restaurant;

        return view("pages.client.restaurants.show", $this->data);
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return
\Illuminate\Contracts\Foundation\Application|Illuminate\Contracts\View\Factory|Illuminate\Contracts\View\View
    */
    public function edit($id)
    {
        $this->data['nav'] = $this->navModel->getAdminNav();
        $this->data['pages'] = [
            [
                'name' => 'Restaurants',
                'route' => 'admin.restaurants'
            ],
            [
                'name' => 'Edit restaurant'
            ]
        ];
        $this->data['restaurant'] = $this->restaurantModel->find($id);

        return view('pages.admin.restaurants.edit', $this->data);
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\RedirectResponse
     */
    public function update(UpdateRestaurantRequest $request, $id)
    {
        try {
            $imageName = null;

            if($request->has('image')) {
                $imageUploadModel = new ImageUpload();
                $imageName = $imageUploadModel->uploadResized($request,
'/assets/images/restaurants/', 405);
            }

            $this->restaurantModel->myUpdate(
                $id,
                $request->get('name'),
                $request->has('recommended'),
                $request->has('active'),
                $imageName
            );

            $this->logAction("Admin updated the " . $request->get('name') . "

```



```

restaurant.", $request);

        return redirect()->route('admin.restaurants')->with('messages',
['Restaurant was updated successfully.']);
    } catch(\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
        return redirect()->back()->withErrors(['We encountered an error.', 'Error
ID: ' . $uniqueId]);
    }
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\RedirectResponse
 */
public function destroy($id)
{
    try {
        $this->restaurantModel->myDelete($id);

        $this->logAction("Admin deleted a restaurant.", request());

        return redirect()->route('admin.restaurants')->with('messages',
['Restaurant was deleted successfully.']);
    } catch(\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
        return redirect()->back()->withErrors(['We encountered an error.', 'Error
ID: ' . $uniqueId]);
    }
}
}

```

UserController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\StoreUserRequest;
use App\Http\Requests\UpdateUserRequest;
use App\Models\Role;
use App\Models\User;
use Illuminate\Http\Request;

class UserController extends AdminController
{
    private $userModel;
    private $roleModel;
    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View
     */
    public function __construct() {
        parent::__construct();
        $this->userModel = new User();
        $this->roleModel = new Role();
    }

    public function index() {
        $this->data['users'] = $this->userModel->get();

        return view('pages.admin.users.index', $this->data);
    }

    public function create()
    {
        $this->data['roles'] = $this->roleModel->get();
        $this->data['pages'] = [
            [
                "name" => "Users",
                "route"=> "admin.users.index"
            ],
            [
                "name" => "Add new user"
            ]
        ];

        return view('pages.admin.users.create', $this->data);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\RedirectResponse
     */
    public function store(StoreUserRequest $request)
    {
        try {
            $this->userModel->insert(
                $request->get('username'),
                $request->get('email'),
                $request->get('password'),
                $request->get('role')
            );
        } catch (\Exception $e) {
            return response()->json([
                'message' => $e->getMessage()
            ], 400);
        }

        return response()->json([
            'message' => 'User created successfully'
        ], 201);
    }
}
```

```

    );

    $this->logAction("Admin created the " . $request->get('username') . "
user.", $request);

    return redirect()->route('admin.users.index')->with('messages', ['User ' .
$request->get('username') . ' was created successfully.']);
    } catch(\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
        return redirect()->back()->withErrors(["We encountered an error.", "Error
ID: " . $uniqueId]);
    }
}
/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Contracts\View\Factory|\Illumi
nate\Contracts\View\View
 */
public function edit($id)
{
    $this->data['user'] = $this->userModel->find($id);
    $this->data['roles'] = $this->roleModel->get();
    $this->data['pages'] = [
        [
            "name" => "Users",
            "route"=> "admin.users.index"
        ],
        [
            "name" => "Edit user"
        ]
    ];

    return view('pages.admin.users.edit', $this->data);
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\RedirectResponse
 */
public function update(UpdateUserRequest $request, $id)
{
    try {
        $this->userModel->myUpdate(
            $id,
            $request->get('username'),
            $request->get('email'),
            $request->get('password'),
            $request->get('role')
        );

        $this->logAction("Admin updated the " . $request->get('username') . "
user.", $request);

        if(session()->get('user')->id == $id)
            session()->remove('user');

        return redirect()->route('admin.users.index')->with('messages', ['User was
updated successfully.']);
    }
}

```

```

        } catch(\Exception $ex) {
            $uniqueId = uniqid();
            \Log::error($uniqueId . " " . $ex->getMessage());
            return redirect()->back()->withErrors(["We encountered an error.", "Error
ID: " . $uniqueId]);
        }
    }

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\RedirectResponse
 */
public function destroy($id)
{
    try {
        $this->userModel->myDelete($id);

        $this->logAction("Admin deleted a user.", request());

        if(session()->get('user')->id == $id)
            session()->remove('user');

        return redirect()->route('admin.users')->with('messages', ['User was
deleted successfully.']);
    } catch(\Exception $ex) {
        $uniqueId = uniqid();
        \Log::error($uniqueId . " " . $ex->getMessage());
        return redirect()->back()->withErrors(["We encountered an error.", "Error
ID: " . $uniqueId]);
    }
}
}

```

3.4 Middleware

IsLoggedIn

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class IsLoggedIn
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if ($request->session()->has("user"))
            return $next($request);
        else
            return redirect()->route("form.login")->withErrors(["You need to be logged in in order to access that page."]);
    }
}
```

IsNotLoggedIn

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class IsNotLoggedIn
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request):
     * (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if($request->session()->has("user"))
            return redirect()->route("home");
        else
            return $next($request);
    }
}
```

IsAdmin

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class IsAdmin
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if($request->session()->get("user")->role_name == "admin")
            return $next($request);
        else
            return redirect()->route("home");
    }
}
```

3.5 Requests

ChangeCartRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Validation\Rule;

class ChangeCartRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'id' => 'required|exists:cart',
            'action' => ['required', Rule::in(['increase', 'decrease', 'remove'])]
        ];
    }
}
```


ContactRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class ContactRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $emailPattern = "/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/";

    public function rules()
    {
        return [
            "email" => ["required", "regex:" . $this->emailPattern],
            "subject" => "required",
            "message" => "required"
        ];
    }
}
```

LoginUserRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class LoginUserRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $emailPattern = "/^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/";
    private $passwordPattern = "/^(?=.*\\d)(?=.*[a-z]).{8,}$/";

    public function rules()
    {
        return [
            "email" => ["required", "regex:" . $this->emailPattern],
            "password" => ["required", "regex:" . $this->passwordPattern]
        ];
    }
}
```

RegisterUserRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegisterUserRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $usernamePattern = "/^(?=.*[a-z])[a-zA-Z0-9\\_]{8,25}$/";
    private $emailPattern = "/^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/";
    private $passwordPattern = "/^(?=.*\\d)(?=.*[a-z]).{8,}$/";

    public function rules()
    {
        return [
            "username" => ["required", "min:8", "max:25", "unique:user", "regex:" .
$this->usernamePattern],
            "email" => ["required", "regex:" . $this->emailPattern, "unique:user"],
            "password" => ["required", "regex:" . $this->passwordPattern],
            "confirmPassword" => "required|in:" . request()->input("password")
        ];
    }

    public function messages() {
        return [
            "confirmPassword.in" => "Your passwords don't match."
        ];
    }
}
```

StoreCategoryRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class StoreCategoryRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $namePattern = "/^[A-Z][a-z]{2,}(\s[a-zA-Z][a-z]{2,})*$/";

    public function rules()
    {
        return [
            "name" => 'required|unique:food_category|regex:' . $this->namePattern
        ];
    }
}
```

StoreFoodRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Validation\Rule;

class StoreFoodRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $pricePattern = "/^(?=.+) (?:[1-9]\d*)?(?: (\.\d+) | (0\.\d*[1-9]+\d*))?$/";

    public function rules()
    {
        return [
            'name' => ['required', Rule::unique('food')->where(function($query) {
                return $query->where('restaurant_id', request()->get('restaurant'));
            })],
            'image' => 'required|image|mimes:jpg,jpeg,png,svg,gif|max:2048',
            'restaurant' => 'required|exists:restaurant,id',
            'category' => 'required|exists:food_category,id',
            'price' => ['required', 'regex:' . $this->pricePattern]
        ];
    }
}
```

StoreOrderRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class StoreOrderRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $addressPattern = "/^[A-ZŠĐČĆŽ][a-zšđčćž]{2,}(\s[A-ZŠĐČĆŽa-zšđčćž][a-zšđčćž]{2,})*\s\d+[A-Z]?(\s\d+)*$/";

    public function rules()
    {
        return [
            'restaurantId' => 'required|exists:restaurant,id',
            'address' => ['required', 'regex:' . $this->addressPattern]
        ];
    }
}
```

StoreRestaurantRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class StoreRestaurantRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'name' => 'required|unique:restaurant',
            'image' => 'required|image|mimes:jpg,jpeg,png,svg,gif|max:2048'
        ];
    }
}
```

StoreUserRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class StoreUserRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $usernamePattern = "/^(?=.*[a-z])[a-zA-Z0-9\\_]{8,25}$/";
    private $emailPattern = "/^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}$/";
    private $passwordPattern = "/^((?=.*\\d)(?=.*[a-z]).{8,})$/";

    public function rules()
    {
        return [
            "username" => ["required", "regex:" . $this->usernamePattern,
"unique:user"],
            "email" => ["required", "regex:" . $this->emailPattern, "unique:user"],
            "password" => ["required", "regex:" . $this->passwordPattern],
            "role" => ["required", "exists:role,id"]
        ];
    }
}
```


UpdateCategoryRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Validation\Rule;

class UpdateCategoryRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $namePattern = "/^[A-Z][a-z]{2,}(\s[a-zA-Z][a-z]{2,})*/";

    public function rules()
    {
        return [
            "name" => [
                "required",
                "regex:" . $this->namePattern,
                Rule::unique('food_category')->ignore($this->category)
            ],
        ];
    }
}
```

UpdateFoodRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Validation\Rule;

class UpdateFoodRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $pricePattern = "/^(?=.+) (?:[1-9]\d*)?(?: (\.\d+) | (0\.\d*[1-9]+\d*))?$/";

    public function rules()
    {
        return [
            'name' => ['required', Rule::unique('food')->where(function($query) {
                return $query->where('restaurant_id', request()->get('restaurant'));
            })->ignore($this->food)],
            'image' => 'nullable|image|mimes:jpg,jpeg,png,svg,gif|max:2048',
            'restaurant' => 'required|exists:restaurant,id',
            'category' => 'required|exists:food_category,id',
            'price' => ['required', 'regex:' . $this->pricePattern]
        ];
    }
}
```

UpdateRestaurantRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Validation\Rule;

class UpdateRestaurantRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'name' => ['required', Rule::unique('restaurant')->ignore($this->restaurant)],
            'image' => 'nullable|image|mimes:jpg,jpeg,png,svg,gif|max:2048'
        ];
    }
}
```

UpdateUserRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Validation\Rule;

class UpdateUserRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    private $usernamePattern = "/^(?=.*[a-z])[a-zA-Z0-9\\_]{8,25}$/";
    private $emailPattern = "/^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/";
    private $passwordPattern = "/^$|^((?=.*\\d)(?=.*[a-z]).{8,})$/";

    public function rules()
    {
        return [
            "username" => ["required", "regex:" . $this->usernamePattern,
                Rule::unique('user')->ignore($this->user)],
            "email" => ["required", "regex:" . $this->emailPattern,
                Rule::unique('user')->ignore($this->user)],
            "password" => ["nullable", "regex:" . $this->passwordPattern],
            "role" => ["required", "exists:role,id"]
        ];
    }
}
```

3.6 Models

AccessLog

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class AccessLog extends Model
{
    public function create($ip, $path, $method) {
        \DB::table("access_log")->insert([
            "ip" => $ip,
            "path" => $path,
            "method" => $method,
            "created_at" => now()
        ]);
    }

    public function getUniqueVisitors() {
        return \DB::table('access_log')
            ->distinct('ip')
            ->count();
    }
}
```

ActionLog

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Http\Request;

class ActionLog extends Model
{
    public function get(Request $request) {
        $query = \DB::table("action_log as al")
            ->leftJoin('user as u', 'al.user_id', '=', 'u.id')
            ->select(['al.*', 'u.username']);

        if($request->has('from') && $request->get('from'))
            $query = $query->where('al.created_at', '>', $request->get('from'));

        if($request->has('to') && $request->get('to'))
            $query = $query->where('al.created_at', '<', $request->get('to'));

        return $query->get();
    }

    public function insert($ip, $path, $method, $action, $user_id = null) {
        \DB::table("action_log")->insert([
            "ip" => $ip,
            "path" => $path,
            "method" => $method,
            "action" => $action,
            "user_id" => $user_id,
            "created_at" => now()
        ]);
    }

    public function getRegistrationsCount() {
        return \DB::table('action_log')
            ->where('path', 'register')
            ->where('method', 'POST')
            ->where('action', 'User registered themselves.')
            ->count();
    }
}
```

Cart

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cart extends Model
{
    private $cartLimit = 10;

    public function addItem($userId, $foodId) {
        $alreadyInCart = \DB::table('cart')
            ->where('user_id', $userId)
            ->where('food_id', $foodId)
            ->first();

        if($alreadyInCart) {
            $newAmount = $alreadyInCart->amount + 1;

            if($alreadyInCart->amount >= $this->cartLimit)
                $newAmount = $this->cartLimit;

            \DB::table('cart')
                ->where('id', $alreadyInCart->id)
                ->update([
                    'amount' => $newAmount,
                    'updated_at' => now()
                ]);
        } else {
            \DB::table('cart')
                ->insert([
                    'user_id' => $userId,
                    'food_id' => $foodId,
                    'amount' => 1,
                    'created_at' => now()
                ]);
        }
    }

    public function change($userId, $cartId, $action) {
        $alreadyInCart = \DB::table('cart')
            ->where('id', $cartId)
            ->where('user_id', $userId)
            ->first();

        if(!$alreadyInCart)
            return false;

        $oldAmount = $alreadyInCart->amount;

        if($action === 'increase') {
            if($oldAmount >= $this->cartLimit)
                $newAmount = $this->cartLimit;
            else
                $newAmount = $oldAmount + 1;

            \DB::table('cart')
                ->where('id', $cartId)
                ->update([
                    'amount' => $newAmount,
                    'updated_at' => now()
                ]);
        }
    }
}
```

```

    } else if($action === 'decrease') {
        if($oldAmount === 1)
            $newAmount = 1;
        else
            $newAmount = $oldAmount - 1;

        \DB::table('cart')
            ->where('id', $cartId)
            ->update([
                'amount' => $newAmount,
                'updated_at' => now()
            ]);
    } else if($action === 'remove') {
        \DB::table('cart')->delete($cartId);
    } else {
        return false;
    }

    return true;
}

public function getItems($userId, $restaurantId) {
    return \DB::table('cart as c')
        ->join('food as f', 'c.food_id', '=', 'f.id')
        ->where('f.restaurant_id', $restaurantId)
        ->where('c.user_id', $userId)
        ->select(["f.*", 'c.amount', 'c.id as cart_id'])
        ->get();
}

public function empty($items) {
    $idsToRemove = [];

    foreach($items as $i)
        $idsToRemove[] = $i->cart_id;

    \DB::table('cart')
        ->whereIn('id', $idsToRemove)
        ->delete();
}
}

```


Food

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Food extends Model
{
    public function get() {
        return \DB::table('food as f')
            ->join('food_category as fc', 'f.category_id', '=', 'fc.id')
            ->join('restaurant as r', 'f.restaurant_id', '=', 'r.id')
            ->orderBy('f.name')
            ->select(['f.*', 'fc.name as category_name', 'r.name as restaurant_name'])
            ->get();
    }

    public function find($id) {
        return \DB::table('food')->find($id);
    }

    public function insert($name, $image, $restaurant, $category, $price) {
        \DB::table('food')->insert([
            'name' => $name,
            'image' => $image,
            'restaurant_id' => $restaurant,
            'category_id' => $category,
            'price' => $price,
            'created_at' => now()
        ]);
    }

    public function myUpdate($id, $name, $restaurant, $category, $price, $image = null)
    {
        $valuesArray = [
            'name' => $name,
            'restaurant_id' => $restaurant,
            'category_id' => $category,
            'price' => $price,
            'updated_at' => now()
        ];

        if($image != null)
            $valuesArray['image'] = $image;

        \DB::table('food')
            ->where('id', $id)
            ->update($valuesArray);
    }

    public function myDelete($id) {
        \DB::table('food')->delete($id);
    }
}
```

FoodCategory

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class FoodCategory extends Model
{
    public function get($onlyListed = false) {
        $query = \DB::table("food_category")->orderBy("name");

        if($onlyListed)
            $query->where("listed", true);

        return $query->get();
    }

    public function find($id) {
        return \DB::table('food_category')->find($id);
    }

    public function insert($name, $listed) {
        \DB::table('food_category')->insert([
            "name" => $name,
            "listed" => $listed,
            "created_at" => now()
        ]);
    }

    public function myUpdate($id, $name, $listed) {
        \DB::table('food_category')
            ->where('id', $id)
            ->update([
                "name" => $name,
                "listed" => $listed,
                "updated_at" => now()
            ]);
    }

    public function myDelete($id) {
        \DB::table('food_category')->delete($id);
    }
}
```

ImageUpload

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Http\Request;
use Intervention\Image\Facades\Image;

class ImageUpload extends Model
{
    public function uploadResized(Request $request, $path, $width, $height = null) {
        $image = $request->file('image');
        $imageName = rand(1, 99999) . time() . '.' . $image->extension();

        $filePath = public_path($path);
        $img = Image::make($image->path());
        $img->resize($width, $height, function ($const) {
            $const->aspectRatio();
        })->save($filePath . $imageName);

        if($width != null && $height != null) {
            $canvas = Image::canvas($width, $height);
            $canvas->insert($img, 'center');
            $canvas->save($filePath . $imageName);
        }

        return $imageName;
    }
}
```

Nav

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Nav extends Model
{
    public static function getClientNav() {
        return \DB::table("nav")->where("admin", false)->orderBy("order")->get();
    }

    public static function getAdminNav() {
        return \DB::table("nav")->where("admin", true)->orderBy("order")->get();
    }
}
```

Order

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    public function get() {
        return \DB::table('order as o')
            ->join('user as u', 'o.user_id', '=', 'u.id')
            ->join('order_detail as od', 'o.id', '=', 'od.order_id')
            ->join('food as f', 'od.food_id', '=', 'f.id')
            ->join('restaurant as r', 'f.restaurant_id', '=', 'r.id')
            ->groupBy(['o.id', 'o.address', 'u.username', 'r.name', 'o.created_at',
'o.updated_at'])
            ->select(['o.id', 'o.address', 'u.username', 'r.name as restaurant_name',
'o.created_at', 'o.updated_at', \DB::raw('SUM(od.price_at_the_time * od.amount) as
total_price')])
            ->get();
    }

    public function store($userId, $address, $items) {
        $orderId = \DB::table('order')->insertGetId([
            'user_id' => $userId,
            'address' => $address,
            'created_at' => now()
        ]);

        $itemsToInsert = [];

        foreach($items as $i) {
            $itemsToInsert[] = [
                'order_id' => $orderId,
                'food_id' => $i->id,
                'price_at_the_time' => $i->price,
                'amount' => $i->amount,
                'created_at' => now()
            ];
        }

        \DB::table('order_detail')->insert($itemsToInsert);

        return $orderId;
    }

    public function getCount() {
        return \DB::table('order')
            ->count();
    }

    public function getTotalEarnings() {
        return \DB::table('order as o')
            ->join('order_detail as od', 'o.id', '=', 'od.order_id')
            ->sum(\DB::raw('od.price_at_the_time * od.amount'));
    }

    public function details($id) {
        return \DB::table('order_detail as od')
            ->join('food as f', 'od.food_id', '=', 'f.id')
            ->where('od.order_id', $id)
            ->get();
    }
}
```

```
}  
}
```

Restaurant

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use phpDocumentor\Reflection\Types\Self_;

class Restaurant extends Model
{
    protected $perPage = 6;
    private $perPageFood = 8;
    private $perPageSearch = 5;

    public function getAll() {
        return \DB::table('restaurant')
            ->orderBy('name')
            ->get();
    }

    public function get($recommended = false) {
        $query = \DB::table("restaurant")
            ->orderBy("name")
            ->where('active', true);

        if($recommended)
            $query = $query->where("recommended", true);

        $restaurants = $query->get();

        foreach($restaurants as $r) {
            $this->getCategoriesForRestaurant($r);
        }

        return $restaurants;
    }

    public function find($id) {
        $query = \DB::table("restaurant as r")
            ->where("r.id", $id);

        $restaurant = $query->first();

        $this->getCategoriesForRestaurant($restaurant);

        return $restaurant;
    }

    private function getCategoriesForRestaurant($restaurant) {
        $restaurant->categories = \DB::table("food as f")
            ->join("food_category as fc", "f.category_id", "=", "fc.id")
            ->select(["fc.id", "fc.name"])
            ->where("f.restaurant_id", $restaurant->id)
            ->groupBy(["fc.id", "fc.name"])
            ->orderBy("fc.name", "asc")
            ->get();

        $restaurant->topCategories = \DB::table("food as f")
            ->join("food_category as fc", "f.category_id", "=", "fc.id")
            ->select(["fc.id", "fc.name"])
            ->where("f.restaurant_id", $restaurant->id)
            ->groupBy(["fc.id", "fc.name"])
            ->orderBy(\DB::raw("count(f.id)"), "desc")
    }
}
```

```

        ->orderBy("fc.name", "asc")
        ->where("fc.listed", true)
        ->limit(3)
        ->get();
    }

    public function search($keyword) {
        return \DB::table("restaurant")
            ->where('name', 'LIKE', '%' . $keyword . '%')
            ->where('active', true)
            ->orderBy("name")
            ->paginate($this->perPageSearch);
    }

    public function getFiltered($keyword, $sortOrder, $categories) {
        $query = \DB::table("restaurant as r")
            ->leftJoin("food as f", "r.id", "=", "f.restaurant_id")
            ->where("r.name", "LIKE", "%" . $keyword . "%")
            ->where('r.active', true);

        if(count($categories))
            $query->whereIn("f.category_id", $categories);

        if($sortOrder == 'nameDesc')
            $query->orderBy("r.name", "desc");
        else
            $query->orderBy("r.name", "asc");

        $columns = ["r.id", "r.name", "r.image"];

        $restaurants = $query->groupBy($columns)->select($columns)->paginate($this->perPage);

        foreach($restaurants as $r) {
            $this->getCategoriesForRestaurant($r);
        }

        return $restaurants;
    }

    public function filterFood($id, $keyword, $sortOrder, $categories) {
        $query = \DB::table("food as f")
            ->where("f.restaurant_id", $id)
            ->where("f.name", "LIKE", "%" . $keyword . "%");

        if(count($categories))
            $query->whereIn("f.category_id", $categories);

        if($sortOrder == 'nameDesc')
            $query->orderBy("f.name", "desc");
        else if($sortOrder == 'priceAsc')
            $query->orderBy("f.price", "asc");
        else if($sortOrder == 'priceDesc')
            $query->orderBy("f.price", "desc");
        else
            $query->orderBy("f.name", "asc");

        return $query->paginate($this->perPageFood);
    }

    public function insert($name, $imageName, $recommended, $active) {
        \DB::table('restaurant')->insert([
            'name' => $name,
            'image' => $imageName,
            'recommended' => $recommended,

```



```

        'active' => $active,
        'created_at' => now()
    ]);
}

public function myUpdate($id, $name, $recommended, $active, $imageName = null) {
    $valuesArray = [
        'name' => $name,
        'recommended' => $recommended,
        'active' => $active,
        'updated_at' => now()
    ];

    if($imageName != null)
        $valuesArray['image'] = $imageName;

    \DB::table('restaurant')
        ->where('id', $id)
        ->update($valuesArray);
}

public function myDelete($id) {
    \DB::table('restaurant')->delete($id);
}
}

```

Role

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Role extends Model
{
    public function get() {
        return \DB::table("role")->get();
    }
}
```

Social

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Social extends Model
{
    public function getSocials() {
        return \DB::table("social")->where("author", false)->orderBy("order")->get();
    }

    public function getAuthorSocials() {
        return \DB::table("social")->where("author", true)->orderBy("order")->get();
    }
}
```

User

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    public function get() {
        return \DB::table("user as u")
            ->join("role as r", "u.role_id", "=", "r.id")
            ->select("u.*", "r.name as role_name")
            ->get();
    }

    public function find($id) {
        return \DB::table('user')->find($id);
    }

    public function where($column, $operator, $value, $multiple = false) {
        $query = \DB::table("user as u")->join("role as r", "u.role_id", "=", "r.id")->
        >where("u." . $column, $operator, $value)->select("u.*", "r.name as role_name");

        if($multiple)
            return $query->get();

        return $query->first();
    }

    public function insert($username, $email, $password, $role) {
        return \DB::table("user")->insertGetId([
            "username" => $username,
            "email" => $email,
            "password" => md5($password),
            "role_id" => $role,
            "created_at" => now()
        ]);
    }

    public function myUpdate($id, $username, $email, $password, $role) {
        $valuesArray = [
            'username' => $username,
            'email' => $email,
            'role_id' => $role,
            'updated_at' => now()
        ];

        if($password != '')
            $valuesArray['password'] = md5($password);

        \DB::table('user')
            ->where('id', $id)
            ->update($valuesArray);
    }

    public function myDelete($id) {
        \DB::table('user')->delete($id);
    }
}
```

3.7 Migrations

create_nav_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('nav', function (Blueprint $table) {
            $table->id();
            $table->string("name");
            $table->string('icon')->nullable()->default(null);
            $table->string("route");
            $table->integer("order");
            $table->boolean("admin");

            $table->unique(['order', 'admin']);
            $table->unique(['name', 'admin']);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('nav_links');
    }
};
```

create_social_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('social', function (Blueprint $table) {
            $table->id();
            $table->string("icon");
            $table->string("href");
            $table->boolean("asset");
            $table->boolean("author");
            $table->integer("order");

            $table->unique(['order', 'author']);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('Social');
    }
};
```

create_user_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('user', function (Blueprint $table) {
            $table->id();
            $table->string("username")->unique();
            $table->string("email")->unique();
            $table->string("password");
            $table->foreignId("role_id")->index();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
};
```

create_role_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('role', function (Blueprint $table) {
            $table->id();
            $table->string("name");
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('role');
    }
};
```


create_access_log_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('access_log', function (Blueprint $table) {
            $table->id();
            $table->string("ip");
            $table->string("path");
            $table->string("method");
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('access_log');
    }
};
```

create_restaurant_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('restaurant', function (Blueprint $table) {
            $table->id();
            $table->string("name")->unique();
            $table->string("image");
            $table->boolean("recommended")->default(false);
            $table->boolean('active')->default(true);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('restaurant');
    }
};
```

create_food_category_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('food_category', function (Blueprint $table) {
            $table->id();
            $table->string("name")->unique();
            $table->boolean("listed")->default(true);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('food_category');
    }
};
```

create_food_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('food', function (Blueprint $table) {
            $table->id();
            $table->string("name");
            $table->string("image");
            $table->float("price");
            $table->foreignId("category_id")->index();
            $table->foreignId("restaurant_id")->index();
            $table->timestamps();

            $table->unique(['name', 'restaurant_id']);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('food');
    }
};
```

create_action_log_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('action_log', function (Blueprint $table) {
            $table->id();
            $table->string("ip");
            $table->string("path");
            $table->string("method");
            $table->foreignId("user_id")->nullable()->default(null)->index();
            $table->string("action");
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('action_log');
    }
};
```

create_cart_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('cart', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id');
            $table->foreignId('food_id');
            $table->integer('amount');
            $table->timestamps();

            $table->unique(['user_id', 'food_id']);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('cart');
    }
};
```

create_order_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('order', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')->index();
            $table->string('address');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('order');
    }
};
```

create_order_detail_table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('order_detail', function (Blueprint $table) {
            $table->id();
            $table->foreignId('order_id');
            $table->foreignId('food_id');
            $table->float('price_at_the_time');
            $table->integer('amount');
            $table->timestamps();

            $table->unique(['order_id', 'food_id']);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('order_detail');
    }
};
```


3.8 Seeders

FoodCategorySeeder

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class FoodCategorySeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    private $foodCategories = [
        [
            "name" => "Pizza",
            "listed" => true
        ],
        [
            "name" => "Burgers",
            "listed" => true
        ],
        [
            "name" => "Sides",
            "listed" => false
        ],
        [
            "name" => "Desserts",
            "listed" => true
        ],
        [
            "name" => "Coffee",
            "listed" => true
        ],
        [
            "name" => "Chicken",
            "listed" => true
        ],
        [
            "name" => "Mexican",
            "listed" => true
        ],
        [
            "name" => "Combos",
            "listed" => false
        ],
        [
            "name" => "Breakfast",
            "listed" => true
        ],
        [
            "name" => "Beverages",
            "listed" => false
        ],
        [
            "name" => "Sandwiches",
            "listed" => true
        ],
    ]
}
```

```

        "name" => "Salads",
        "listed" => true
    ],
    [
        "name" => "Wraps",
        "listed" => true
    ],
    [
        "name" => "Extras",
        "listed" => false
    ],
    [
        "name" => "Pasta",
        "listed" => true
    ]
];

public function run()
{
    foreach($this->foodCategories as $fc) {
        $fc["created_at"] = now();
        \DB::table("food_category")->insert($fc);
    }
}
}

```

FoodSeeder

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class FoodSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    private $food = [
        [ // McDonalds
            "name" => "Big Mac",
            "image" => "mcdonalds_big_mac.jpg",
            "price" => 2.99,
            "category_id" => 2,
            "restaurant_id" => 1
        ],
        [
            "name" => "Big Mac Combo Meal",
            "image" => "mcdonalds_big_mac_combo_meal.jpg",
            "price" => 6.99,
            "category_id" => 8,
            "restaurant_id" => 1
        ],
        [
            "name" => "Cheeseburger",
            "image" => "mcdonalds_cheeseburger.jpg",
            "price" => 1.99,
            "category_id" => 2,
            "restaurant_id" => 1
        ],
        [
            "name" => "Cheeseburger Combo Meal",
            "image" => "mcdonalds_cheeseburger_combo_meal.jpg",
            "price" => 6.99,
            "category_id" => 8,
            "restaurant_id" => 1
        ],
        [
            "name" => "Chicken McNuggets",
            "image" => "mcdonalds_chicken_mcnuggets.jpg",
            "price" => 3.99,
            "category_id" => 6,
            "restaurant_id" => 1
        ],
        [
            "name" => "Coca Cola",
            "image" => "mcdonalds_coca_cola.jpg",
            "price" => 0.99,
            "category_id" => 10,
            "restaurant_id" => 1
        ],
        [
            "name" => "Double Cheeseburger",
            "image" => "mcdonalds_double_cheeseburger.jpg",
            "price" => 2.99,
            "category_id" => 2,
            "restaurant_id" => 1
        ]
    ];
}
```

```

],
[
  "name" => "Double Quarter Pounder",
  "image" => "mcdonalds_double_quarter_pounder.jpg",
  "price" => 2.99,
  "category_id" => 2,
  "restaurant_id" => 1
],
[
  "name" => "Fanta",
  "image" => "mcdonalds_fanta.jpg",
  "price" => 0.99,
  "category_id" => 10,
  "restaurant_id" => 1
],
[
  "name" => "Hamburger",
  "image" => "mcdonalds_hamburger.jpg",
  "price" => 1.99,
  "category_id" => 2,
  "restaurant_id" => 1
],
[
  "name" => "Hot Caramel Sundae",
  "image" => "mcdonalds_hot_caramel_sundae.jpg",
  "price" => 1.99,
  "category_id" => 4,
  "restaurant_id" => 1
],
[
  "name" => "Hot Fudge Sundae",
  "image" => "mcdonalds_hot_fudge_sundae.jpg",
  "price" => 1.99,
  "category_id" => 4,
  "restaurant_id" => 1
],
[
  "name" => "McChicken",
  "image" => "mcdonalds_mcchicken.jpg",
  "price" => 1.99,
  "category_id" => 2,
  "restaurant_id" => 1
],
[
  "name" => "Quarter Pounder",
  "image" => "mcdonalds_quarter_pounder.jpg",
  "price" => 1.99,
  "category_id" => 2,
  "restaurant_id" => 1
],
[
  "name" => "Sprite",
  "image" => "mcdonalds_sprite.jpg",
  "price" => 0.99,
  "category_id" => 10,
  "restaurant_id" => 1
],
[
  "name" => "World Famous Fries",
  "image" => "mcdonalds_world_famous_fries.jpg",
  "price" => 1.99,
  "category_id" => 3,
  "restaurant_id" => 1
],
[ // KFC

```

```

    "name" => "330ml 7up",
    "image" => "kfc_330ml_7up.jpg",
    "price" => 0.99,
    "category_id" => 10,
    "restaurant_id" => 2
  ],
  [
    "name" => "330ml Pepsi",
    "image" => "kfc_330ml_pepsi.jpg",
    "price" => 0.99,
    "category_id" => 10,
    "restaurant_id" => 2
  ],
  [
    "name" => "BBQ Sauce",
    "image" => "kfc_bbq_sauce.jpg",
    "price" => 0.99,
    "category_id" => 14,
    "restaurant_id" => 2
  ],
  [
    "name" => "Colonel Salad",
    "image" => "kfc_colonel_salad.jpg",
    "price" => 3.99,
    "category_id" => 12,
    "restaurant_id" => 2
  ],
  [
    "name" => "Fries",
    "image" => "kfc_fries.jpg",
    "price" => 2.99,
    "category_id" => 3,
    "restaurant_id" => 2
  ],
  [
    "name" => "Hot Wings Bucket",
    "image" => "kfc_hot_wings_bucket.jpg",
    "price" => 8.99,
    "category_id" => 6,
    "restaurant_id" => 2
  ],
  [
    "name" => "Smartster",
    "image" => "kfc_smartster.jpg",
    "price" => 2.99,
    "category_id" => 3,
    "restaurant_id" => 2
  ],
  [
    "name" => "Sweet & Sour Sauce",
    "image" => "kfc_sweet_&_sour_sauce.jpg",
    "price" => 0.99,
    "category_id" => 14,
    "restaurant_id" => 2
  ],
  [
    "name" => "Twister",
    "image" => "kfc_twister.jpg",
    "price" => 3.99,
    "category_id" => 13,
    "restaurant_id" => 2
  ],
  [
    "name" => "Twister Meal",
    "image" => "kfc_twister_meal.jpg",

```

```

    "price" => 7.99,
    "category_id" => 8,
    "restaurant_id" => 2
  ],
  [
    "name" => "Variety Meal",
    "image" => "kfc_variety_meal.jpg",
    "price" => 8.99,
    "category_id" => 8,
    "restaurant_id" => 2
  ],
  [
    "name" => "Zinger Box Meal",
    "image" => "kfc_zinger_box_meal.jpg",
    "price" => 8.99,
    "category_id" => 8,
    "restaurant_id" => 2
  ],
  [
    "name" => "Zinger Burger Meal",
    "image" => "kfc_zinger_burger_meal.jpg",
    "price" => 8.99,
    "category_id" => 8,
    "restaurant_id" => 2
  ],
  [
    "name" => "Zinger Burger Single",
    "image" => "kfc_zinger_burger_single.jpg",
    "price" => 2.99,
    "category_id" => 2,
    "restaurant_id" => 2
  ],
  [
    "name" => "Zinger Tower Burger",
    "image" => "kfc_zinger_tower_burger.jpg",
    "price" => 3.99,
    "category_id" => 2,
    "restaurant_id" => 2
  ],
  [ // Subway
    "name" => "Baja Chicken & Bacon Wrap",
    "image" => "subway_baja_chicken_&_bacon_wrap.jpg",
    "price" => 5.99,
    "category_id" => 13,
    "restaurant_id" => 4
  ],
  [
    "name" => "Baja Turkey Avocado Salad",
    "image" => "subway_baja_turkey_avocado_salad.jpg",
    "price" => 3.99,
    "category_id" => 12,
    "restaurant_id" => 4
  ],
  [
    "name" => "Black Forest Ham Salad",
    "image" => "subway_black_forest_ham_salad.jpg",
    "price" => 3.99,
    "category_id" => 12,
    "restaurant_id" => 4
  ],
  [
    "name" => "Buffalo Chicken Sandwich",
    "image" => "subway_buffalo_chicken_sandwich.jpg",
    "price" => 4.49,
    "category_id" => 11,

```

```

    "restaurant_id" => 4
  ],
  [
    "name" => "Cold Cut Combo Sandwich",
    "image" => "subway_cold_cut_combo_sandwich.jpg",
    "price" => 3.99,
    "category_id" => 11,
    "restaurant_id" => 4
  ],
  [
    "name" => "Meatball Marinara Sandwich",
    "image" => "subway_meatball_marinara_sandwich.jpg",
    "price" => 5.99,
    "category_id" => 11,
    "restaurant_id" => 4
  ],
  [
    "name" => "Mozza Meat Sandwich",
    "image" => "subway_mozza_meat_sandwich.jpg",
    "price" => 4.99,
    "category_id" => 11,
    "restaurant_id" => 4
  ],
  [
    "name" => "Mozza Meat Wrap",
    "image" => "subway_mozza_meat_wrap.jpg",
    "price" => 4.99,
    "category_id" => 13,
    "restaurant_id" => 4
  ],
  [
    "name" => "Spicy Italian Salad",
    "image" => "subway_spicy_italian_salad.jpg",
    "price" => 3.99,
    "category_id" => 12,
    "restaurant_id" => 4
  ],
  [
    "name" => "Subway Club Sandwich",
    "image" => "subway_subway_club_sandwich.jpg",
    "price" => 4.99,
    "category_id" => 11,
    "restaurant_id" => 4
  ],
  [
    "name" => "Subway Club Wrap",
    "image" => "subway_subway_club_wrap.jpg",
    "price" => 4.99,
    "category_id" => 13,
    "restaurant_id" => 4
  ],
  [
    "name" => "Supreme Meats Sandwich",
    "image" => "subway_supreme_meats_sandwich.jpg",
    "price" => 5.99,
    "category_id" => 11,
    "restaurant_id" => 4
  ],
  [ // Domino's Pizza
    "name" => "3 Meats",
    "image" => "dominos_3_meats.jpg",
    "price" => 8.99,
    "category_id" => 1,
    "restaurant_id" => 3
  ],

```

```

[
  "name" => "7up",
  "image" => "dominos_7up.jpg",
  "price" => 1.99,
  "category_id" => 10,
  "restaurant_id" => 3
],
[
  "name" => "Boneless Chicken Tenders",
  "image" => "dominos_boneless_chicken_tenders.jpg",
  "price" => 3.99,
  "category_id" => 6,
  "restaurant_id" => 3
],
[
  "name" => "Buttermilk Chicken Pieces",
  "image" => "dominos_buttermilk_chicken_pieces.jpg",
  "price" => 4.99,
  "category_id" => 6,
  "restaurant_id" => 3
],
[
  "name" => "Chicken Supreme",
  "image" => "dominos_chicken_supreme.jpg",
  "price" => 8.99,
  "category_id" => 1,
  "restaurant_id" => 3
],
[
  "name" => "Choc Lava Cake",
  "image" => "dominos_choc_lava_cake.jpg",
  "price" => 5.99,
  "category_id" => 4,
  "restaurant_id" => 3
],
[
  "name" => "Choc Lava Puff Roll",
  "image" => "dominos_choc_lava_puff_roll.jpg",
  "price" => 4.99,
  "category_id" => 4,
  "restaurant_id" => 3
],
[
  "name" => "Double Bacon Cheeseburger",
  "image" => "dominos_double_bacon_cheeseburger.jpg",
  "price" => 8.99,
  "category_id" => 1,
  "restaurant_id" => 3
],
[
  "name" => "Fire Breather",
  "image" => "dominos_fire_breather.jpg",
  "price" => 7.99,
  "category_id" => 1,
  "restaurant_id" => 3
],
[
  "name" => "Garlic Bread",
  "image" => "dominos_garlic_bread.jpg",
  "price" => 2.99,
  "category_id" => 3,
  "restaurant_id" => 3
],
[
  "name" => "Godfather",

```



```

    "image" => "dominos_godfather.jpg",
    "price" => 7.99,
    "category_id" => 1,
    "restaurant_id" => 3
  ],
  [
    "name" => "Oven Baked Chips",
    "image" => "dominos_oven_baked_chips.jpg",
    "price" => 7.99,
    "category_id" => 3,
    "restaurant_id" => 3
  ],
  [
    "name" => "Pepperoni",
    "image" => "dominos_pepperoni.jpg",
    "price" => 7.99,
    "category_id" => 1,
    "restaurant_id" => 3
  ],
  [
    "name" => "Pepsi",
    "image" => "dominos_pepsi.jpg",
    "price" => 1.99,
    "category_id" => 10,
    "restaurant_id" => 3
  ],
  [
    "name" => "Vanilla Custard Puff Roll",
    "image" => "dominos_vanilla_custard_puff_roll.jpg",
    "price" => 4.99,
    "category_id" => 4,
    "restaurant_id" => 3
  ],
  [ // Papa John's
    "name" => "All The Meats",
    "image" => "papa_johns_all_the_meats.jpg",
    "price" => 8.99,
    "category_id" => 1,
    "restaurant_id" => 6
  ],
  [
    "name" => "Breadsticks",
    "image" => "papa_johns_breadsticks.jpg",
    "price" => 2.99,
    "category_id" => 3,
    "restaurant_id" => 6
  ],
  [
    "name" => "Cheese",
    "image" => "papa_johns_cheese.jpg",
    "price" => 6.99,
    "category_id" => 1,
    "restaurant_id" => 6
  ],
  [
    "name" => "Cheesesticks",
    "image" => "papa_johns_cheesesticks.jpg",
    "price" => 2.99,
    "category_id" => 3,
    "restaurant_id" => 6
  ],
  [
    "name" => "Chicken Florentine Pasta",
    "image" => "papa_johns_chicken_florentine_pasta.jpg",
    "price" => 7.99,

```

```

    "category_id" => 15,
    "restaurant_id" => 6
  ],
  [
    "name" => "Chocolate Chip Cookie",
    "image" => "papa_johns_chocolate_chip_cookie.jpg",
    "price" => 2.99,
    "category_id" => 4,
    "restaurant_id" => 6
  ],
  [
    "name" => "Cinnamon Pullaparts",
    "image" => "papa_johns_cinnamon_pullaparts.jpg",
    "price" => 3.99,
    "category_id" => 4,
    "restaurant_id" => 6
  ],
  [
    "name" => "Creamy Red Sauce Pasta With Italian Sausage",
    "image" => "papa_johns_creamy_red_sauce_pasta_with_italian_sausage.jpg",
    "price" => 9.99,
    "category_id" => 15,
    "restaurant_id" => 6
  ],
  [
    "name" => "Garden Fresh",
    "image" => "papa_johns_garden_fresh.jpg",
    "price" => 6.99,
    "category_id" => 1,
    "restaurant_id" => 6
  ],
  [
    "name" => "Hawaiian",
    "image" => "papa_johns_hawaiian.jpg",
    "price" => 6.99,
    "category_id" => 1,
    "restaurant_id" => 6
  ],
  [
    "name" => "Pepperoni",
    "image" => "papa_johns_pepperoni.jpg",
    "price" => 7.99,
    "category_id" => 1,
    "restaurant_id" => 6
  ],
  [
    "name" => "Pepsi - 2l Bottle",
    "image" => "papa_johns_pepsi_2l.jpg",
    "price" => 3.49,
    "category_id" => 10,
    "restaurant_id" => 6
  ],
  [
    "name" => "Pepsi - 500ml Bottle",
    "image" => "papa_johns_pepsi_500ml.jpg",
    "price" => 1.99,
    "category_id" => 10,
    "restaurant_id" => 6
  ],
  [
    "name" => "Sausage and Bacon Supreme",
    "image" => "papa_johns_sausage_and_bacon_supreme.jpg",
    "price" => 8.99,
    "category_id" => 1,
    "restaurant_id" => 6
  ]
]

```

```

],
[
  "name" => "The Works",
  "image" => "papa_johns_the_works.jpg",
  "price" => 8.99,
  "category_id" => 1,
  "restaurant_id" => 6
],
[ // Chick-fil-a
  "name" => "Barbeque Sauce",
  "image" => "chickfila_barbeque_sauce.jpg",
  "price" => 1.99,
  "category_id" => 14,
  "restaurant_id" => 5
],
[
  "name" => "Chicken Biscuit",
  "image" => "chickfila_chicken_biscuit.jpg",
  "price" => 2.99,
  "category_id" => 9,
  "restaurant_id" => 5
],
[
  "name" => "Chicken Sandwich",
  "image" => "chickfila_chicken_sandwich.jpg",
  "price" => 3.99,
  "category_id" => 11,
  "restaurant_id" => 5
],
[
  "name" => "Chick-Fil-A Sauce",
  "image" => "chickfila_chick-fil-a_sauce.jpg",
  "price" => 1.99,
  "category_id" => 14,
  "restaurant_id" => 5
],
[
  "name" => "Chick-n-Strips",
  "image" => "chickfila_chick-n-strips.jpg",
  "price" => 3.99,
  "category_id" => 6,
  "restaurant_id" => 5
],
[
  "name" => "Cobb Salad",
  "image" => "chickfila_cobb_salad.jpg",
  "price" => 3.99,
  "category_id" => 12,
  "restaurant_id" => 5
],
[
  "name" => "Coca-Cola",
  "image" => "chickfila_coca-cola.jpg",
  "price" => 2.99,
  "category_id" => 10,
  "restaurant_id" => 5
],
[
  "name" => "Cool Wrap",
  "image" => "chickfila_cool_wrap.jpg",
  "price" => 3.99,
  "category_id" => 13,
  "restaurant_id" => 5
],
[

```

```

    "name" => "Deluxe Sandwich",
    "image" => "chickfila_deluxe_sandwich.jpg",
    "price" => 4.49,
    "category_id" => 11,
    "restaurant_id" => 5
  ],
  [
    "name" => "Dr Pepper",
    "image" => "chickfila_dr_pepper.jpg",
    "price" => 2.99,
    "category_id" => 10,
    "restaurant_id" => 5
  ],
  [
    "name" => "Egg White Grill",
    "image" => "chickfila_egg_white_grill.jpg",
    "price" => 2.49,
    "category_id" => 9,
    "restaurant_id" => 5
  ],
  [
    "name" => "Fruit Cup",
    "image" => "chickfila_fruit_cup.jpg",
    "price" => 2.49,
    "category_id" => 3,
    "restaurant_id" => 5
  ],
  [
    "name" => "Grilled Chicken Sandwich",
    "image" => "chickfila_grilled_chicken_sandwich.jpg",
    "price" => 4.49,
    "category_id" => 11,
    "restaurant_id" => 5
  ],
  [
    "name" => "Hash Brown Scramble Bowl",
    "image" => "chickfila_hash_brown_scramble_bowl.jpg",
    "price" => 2.99,
    "category_id" => 9,
    "restaurant_id" => 5
  ],
  [
    "name" => "Nuggets",
    "image" => "chickfila_nuggets.jpg",
    "price" => 3.99,
    "category_id" => 6,
    "restaurant_id" => 5
  ],
  [
    "name" => "Spicy Chicken Sandwich",
    "image" => "chickfila_spicy_chicken_sandwich.jpg",
    "price" => 3.99,
    "category_id" => 11,
    "restaurant_id" => 5
  ],
  [
    "name" => "Spicy Deluxe Sandwich",
    "image" => "chickfila_spicy_deluxe_sandwich.jpg",
    "price" => 4.49,
    "category_id" => 11,
    "restaurant_id" => 5
  ],
  [
    "name" => "Spicy Southwest Salad",
    "image" => "chickfila_spicy_southwest_salad.jpg",

```

```

    "price" => 3.99,
    "category_id" => 12,
    "restaurant_id" => 5
  ],
  [
    "name" => "Waffle Potato Fries",
    "image" => "chickfila_waffle_potato_fries.jpg",
    "price" => 2.99,
    "category_id" => 3,
    "restaurant_id" => 5
  ],
  [ // Starbucks
    "name" => "Bacon, Gouda & Egg Sandwich",
    "image" => "starbucks_bacon_gouda_egg_sandwich.jpg",
    "price" => 2.99,
    "category_id" => 9,
    "restaurant_id" => 8
  ],
  [
    "name" => "Caffe Americano",
    "image" => "starbucks_caffe_americano.jpg",
    "price" => 3.99,
    "category_id" => 5,
    "restaurant_id" => 8
  ],
  [
    "name" => "Caffe Latte",
    "image" => "starbucks_caffe_latte.jpg",
    "price" => 4.99,
    "category_id" => 5,
    "restaurant_id" => 8
  ],
  [
    "name" => "Caffe Misto",
    "image" => "starbucks_caffe_misto.jpg",
    "price" => 4.99,
    "category_id" => 5,
    "restaurant_id" => 8
  ],
  [
    "name" => "Caffe Mocha",
    "image" => "starbucks_caffe_mocha.jpg",
    "price" => 4.99,
    "category_id" => 5,
    "restaurant_id" => 8
  ],
  [
    "name" => "Cappuccino",
    "image" => "starbucks_cappuccino.jpg",
    "price" => 4.49,
    "category_id" => 5,
    "restaurant_id" => 8
  ],
  [
    "name" => "Caramel Macchiato",
    "image" => "starbucks_caramel_macchiato.jpg",
    "price" => 5.99,
    "category_id" => 5,
    "restaurant_id" => 8
  ],
  [
    "name" => "Cinnamon Dolce Latte",
    "image" => "starbucks_cinnamon_dolce_latte.jpg",
    "price" => 4.99,
    "category_id" => 5,

```

```

    "restaurant_id" => 8
  ],
  [
    "name" => "Crispy Grilled Cheese on Sourdough",
    "image" => "starbucks_crispy_grilled_cheese_on_sourdough.jpg",
    "price" => 4.99,
    "category_id" => 11,
    "restaurant_id" => 8
  ],
  [
    "name" => "Double-Smoked Bacon, Cheddar & Egg Sandwich",
    "image" => "starbucks_double-smoked_bacon_cheddar_&_egg_sandwich.jpg",
    "price" => 3.99,
    "category_id" => 9,
    "restaurant_id" => 8
  ],
  [
    "name" => "Ham & Swiss on Baguette",
    "image" => "starbucks_ham_&_swiss_on_baguette.jpg",
    "price" => 4.99,
    "category_id" => 11,
    "restaurant_id" => 8
  ],
  [
    "name" => "Impossible Breakfast Sandwich",
    "image" => "starbucks_impossible_breakfast_sandwich.jpg",
    "price" => 3.99,
    "category_id" => 9,
    "restaurant_id" => 8
  ],
  [
    "name" => "Spinach, Feta & Egg White Wrap",
    "image" => "starbucks_spinach_feta_&_egg_white_wrap.jpg",
    "price" => 3.99,
    "category_id" => 9,
    "restaurant_id" => 8
  ],
  [
    "name" => "White Chocolate Mocha",
    "image" => "starbucks_white_chocolate_mocha.jpg",
    "price" => 5.99,
    "category_id" => 5,
    "restaurant_id" => 8
  ],
  [ // Chipotle
    "name" => "Blackberry Izze",
    "image" => "chipotle_blackberry_izze.jpg",
    "price" => 1.99,
    "category_id" => 10,
    "restaurant_id" => 9
  ],
  [
    "name" => "Burrito",
    "image" => "chipotle_burrito.jpg",
    "price" => 10.99,
    "category_id" => 7,
    "restaurant_id" => 9
  ],
  [
    "name" => "Burrito Bowl",
    "image" => "chipotle_burrito_bowl.jpg",
    "price" => 12.99,
    "category_id" => 7,
    "restaurant_id" => 9
  ],
],

```

```
[
  "name" => "Chips & Guacamole",
  "image" => "chipotle_chips_&_guacamole.jpg",
  "price" => 4.99,
  "category_id" => 3,
  "restaurant_id" => 9
],
[
  "name" => "Chips & Gueso Blanco",
  "image" => "chipotle_chips_&_queso_blanco.jpg",
  "price" => 4.99,
  "category_id" => 3,
  "restaurant_id" => 9
],
[
  "name" => "Guacamole",
  "image" => "chipotle_guacamole.jpg",
  "price" => 2.99,
  "category_id" => 14,
  "restaurant_id" => 9
],
[
  "name" => "High Protein Bowl",
  "image" => "chipotle_high_protein_bowl.jpg",
  "price" => 11.99,
  "category_id" => 12,
  "restaurant_id" => 9
],
[
  "name" => "Keto Salad Bowl",
  "image" => "chipotle_keto_salad_bowl.jpg",
  "price" => 11.99,
  "category_id" => 12,
  "restaurant_id" => 9
],
[
  "name" => "Kid's Quesadilla",
  "image" => "chipotle_kids_quesadilla.jpg",
  "price" => 8.99,
  "category_id" => 7,
  "restaurant_id" => 9
],
[
  "name" => "Mexical Coca-Cola",
  "image" => "chipotle_mexican_coca-cola.jpg",
  "price" => 1.99,
  "category_id" => 10,
  "restaurant_id" => 9
],
[
  "name" => "Mexican Sprite",
  "image" => "chipotle_mexican_sprite.jpg",
  "price" => 1.99,
  "category_id" => 10,
  "restaurant_id" => 9
],
[
  "name" => "Quesadilla",
  "image" => "chipotle_quesadilla.jpg",
  "price" => 10.99,
  "category_id" => 7,
  "restaurant_id" => 9
],
[
  "name" => "Queso Blanco",
```

```

    "image" => "chipotle_queso_blanco.jpg",
    "price" => 2.99,
    "category_id" => 14,
    "restaurant_id" => 9
  ],
  [
    "name" => "Tacos",
    "image" => "chipotle_tacos.jpg",
    "price" => 6.99,
    "category_id" => 7,
    "restaurant_id" => 9
  ],
  [
    "name" => "Whole30 Salad Bowl",
    "image" => "chipotle_whole30_salad_bowl.jpg",
    "price" => 11.99,
    "category_id" => 12,
    "restaurant_id" => 9
  ],
  [ // Dunkin' Donuts
    "name" => "Americano",
    "image" => "dunkindonuts_americano.jpg",
    "price" => 6.99,
    "category_id" => 5,
    "restaurant_id" => 7
  ],
  [
    "name" => "Bacon, Egg & Cheese",
    "image" => "dunkindonuts_bacon_egg_&_cheese.jpg",
    "price" => 6.99,
    "category_id" => 9,
    "restaurant_id" => 7
  ],
  [
    "name" => "Cappuccino",
    "image" => "dunkindonuts_cappuccino.jpg",
    "price" => 5.99,
    "category_id" => 5,
    "restaurant_id" => 7
  ],
  [
    "name" => "Coffe",
    "image" => "dunkindonuts_coffe.jpg",
    "price" => 5.99,
    "category_id" => 5,
    "restaurant_id" => 7
  ],
  [
    "name" => "Donut",
    "image" => "dunkindonuts_donut.jpg",
    "price" => 4.99,
    "category_id" => 4,
    "restaurant_id" => 7
  ],
  [
    "name" => "Egg & Cheese",
    "image" => "dunkindonuts_egg_&_cheese.jpg",
    "price" => 5.99,
    "category_id" => 9,
    "restaurant_id" => 7
  ],
  [
    "name" => "Iced Americano",
    "image" => "dunkindonuts_iced_americano.jpg",
    "price" => 6.99,

```



```

        "category_id" => 5,
        "restaurant_id" => 7
    ],
    [
        "name" => "Iced Coffe",
        "image" => "dunkindonuts_iced_coffe.jpg",
        "price" => 5.99,
        "category_id" => 5,
        "restaurant_id" => 7
    ],
    [
        "name" => "Latte",
        "image" => "dunkindonuts_latte.jpg",
        "price" => 6.99,
        "category_id" => 5,
        "restaurant_id" => 7
    ],
    [
        "name" => "Macchiato",
        "image" => "dunkindonuts_macchiato.jpg",
        "price" => 6.99,
        "category_id" => 5,
        "restaurant_id" => 7
    ],
    [
        "name" => "Munchkins Donut Hole Treats",
        "image" => "dunkindonuts_munchkins_donut_hole_treats.jpg",
        "price" => 5.99,
        "category_id" => 4,
        "restaurant_id" => 7
    ],
    [
        "name" => "Sausage, Egg & Cheese",
        "image" => "dunkindonuts_sausage_egg_&_cheese.jpg",
        "price" => 6.99,
        "category_id" => 9,
        "restaurant_id" => 7
    ],
    [
        "name" => "Wake-Up Wrap",
        "image" => "dunkindonuts_wake-up_wrap.jpg",
        "price" => 4.99,
        "category_id" => 9,
        "restaurant_id" => 7
    ],
    [ // Little Caesars
        "name" => "3 Meat Treat",
        "image" => "littlecaesars_3_meat_treat.jpg",
        "price" => 10.99,
        "category_id" => 1,
        "restaurant_id" => 10
    ],
    [
        "name" => "4-N-1 Pizza",
        "image" => "littlecaesars_4-n-1_pizza.jpg",
        "price" => 11.99,
        "category_id" => 1,
        "restaurant_id" => 10
    ],
    [
        "name" => "Caesar Wings",
        "image" => "littlecaesars_caesar_wings.jpg",
        "price" => 5.99,
        "category_id" => 6,
        "restaurant_id" => 10
    ]

```

```

],
[
  "name" => "Cheese",
  "image" => "littlecaesars_cheese.jpg",
  "price" => 9.99,
  "category_id" => 1,
  "restaurant_id" => 10
],
[
  "name" => "Coca-Cola - 2l Bottle",
  "image" => "littlecaesars_coca-cola_2l_bottle.jpg",
  "price" => 2.99,
  "category_id" => 10,
  "restaurant_id" => 10
],
[
  "name" => "Complete Meal",
  "image" => "littlecaesars_complete_meal.jpg",
  "price" => 14.99,
  "category_id" => 8,
  "restaurant_id" => 10
],
[
  "name" => "Crazy Combo",
  "image" => "littlecaesars_crazy_combo.jpg",
  "price" => 2.99,
  "category_id" => 8,
  "restaurant_id" => 10
],
[
  "name" => "Fun Meal",
  "image" => "littlecaesars_fun_meal.jpg",
  "price" => 17.99,
  "category_id" => 8,
  "restaurant_id" => 10
],
[
  "name" => "Hula Hawaiian",
  "image" => "littlecaesars_hula_hawaiian.jpg",
  "price" => 10.99,
  "category_id" => 1,
  "restaurant_id" => 10
],
[
  "name" => "Party Meal",
  "image" => "littlecaesars_party_meal.jpg",
  "price" => 20.99,
  "category_id" => 8,
  "restaurant_id" => 10
],
[
  "name" => "Pepperoni",
  "image" => "littlecaesars_pepperoni.jpg",
  "price" => 9.99,
  "category_id" => 1,
  "restaurant_id" => 10
],
[
  "name" => "The Batman Pizza",
  "image" => "littlecaesars_the_batman_pizza.jpg",
  "price" => 11.99,
  "category_id" => 1,
  "restaurant_id" => 10
],
[

```

```

        "name" => "Ultimate Supreme",
        "image" => "littlecaesars_ultimate_supreme.jpg",
        "price" => 10.99,
        "category_id" => 1,
        "restaurant_id" => 10
    ],
    [
        "name" => "Veggie",
        "image" => "littlecaesars_veggie.jpg",
        "price" => 10.99,
        "category_id" => 1,
        "restaurant_id" => 10
    ]
];

public function run()
{
    foreach($this->food as $f) {
        $f['created_at'] = now();
        \DB::table("food")->insert($f);
    }
}
}

```

NavSeeder

```
<?php

namespace Database\Seeders;

use App\Models\Nav;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class NavSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    private $navLinks = [
        [
            "name" => "Home",
            "route" => "home",
            "order" => 1,
            "admin" => 0
        ],
        [
            "name" => "Restaurants",
            "route" => "restaurants.index",
            "order" => 2,
            "admin" => 0
        ],
        [
            "name" => "Contact",
            "route" => "contact",
            "order" => 3,
            "admin" => 0
        ],
        [
            "name" => "Author",
            "route" => "author",
            "order" => 4,
            "admin" => 0
        ],
        [
            "name" => "Dashboard",
            "icon" => 'fas fa-tachometer-alt',
            "route" => "admin.dashboard",
            "order" => 1,
            "admin" => 1
        ],
        [
            "name" => "Actions",
            "icon" => 'fas fa-tasks',
            "route" => "admin.actions",
            "order" => 2,
            "admin" => 1
        ],
        [
            "name" => "Users",
            "icon" => 'fa fa-user',
            "route" => "admin.users.index",
            "order" => 3,
            "admin" => 1
        ],
        [
            "name" => "Restaurants",
```

```

        "icon" => 'fa fa-cutlery',
        "route" => "admin.restaurants",
        "order" => 4,
        "admin" => 1
    ],
    [
        "name" => "Food",
        "icon" => 'fa-solid fa-burger',
        "route" => "admin.food.index",
        "order" => 5,
        "admin" => 1
    ],
    [
        "name" => "Food Categories",
        "icon" => 'fas fa-pizza-slice',
        "route" => "admin.categories.index",
        "order" => 6,
        "admin" => 1
    ],
    [
        "name" => "Orders",
        "icon" => 'fas fa-shopping-cart',
        "route" => "admin.orders",
        "order" => 7,
        "admin" => 1
    ]
];

public function run()
{
    foreach($this->navLinks as $nl)
        \DB::table("nav")->insert($nl);
}
}

```

RestaurantSeeder

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class RestaurantSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    private $restaurants = [
        [
            "name" => "McDonald's",
            "image" => "mcdonalds.jpg",
            "recommended" => true
        ],
        [
            "name" => "KFC",
            "image" => "kfc.jpg",
            "recommended" => true
        ],
        [
            "name" => "Domino's Pizza",
            "image" => "dominos.jpg",
            "recommended" => true
        ],
        [
            "name" => "Subway",
            "image" => "subway.jpg",
            "recommended" => true
        ],
        [
            "name" => "Chick-Fil-A",
            "image" => "chick_fil_a.jpg",
            "recommended" => true
        ],
        [
            "name" => "Papa John's",
            "image" => "papa_johns.jpg",
            "recommended" => true
        ],
        [
            "name" => "Dunkin Donut's",
            "image" => "dunkin_donuts.jpg",
            "recommended" => false
        ],
        [
            "name" => "Starbucks",
            "image" => "starbucks.jpg",
            "recommended" => false
        ],
        [
            "name" => "Chipotle",
            "image" => "chipotle.jpg",
            "recommended" => false
        ],
        [
            "name" => "Little Caesars",
            "image" => "little_caesars.jpg",
```

```
        "recommended" => false
    ]
};

public function run()
{
    foreach($this->restaurants as $r) {
        $r["created_at"] = now();
        \DB::table("restaurant")->insert($r);
    }
}
```

RoleSeeder

```
<?php

namespace Database\Seeders;

use App\Models\Role;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class RoleSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    private $roles = [
        [
            "name" => "user"
        ],
        [
            "name" => "admin"
        ]
    ];

    public function run()
    {
        \DB::table("role")->insert($this->roles);
    }
}
```



```
<?php

namespace Database\Seeders;

use App\Models\Social;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class SocialSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    private $socials = [
        [
            "icon" => "fab fa-facebook",
            "asset" => false,
            "href" => "https://www.facebook.com/bojan.maksimovic.908",
            "author" => false,
            "order" => 1
        ],
        [
            "icon" => "fab fa-instagram",
            "asset" => false,
            "href" => "https://www.instagram.com/bojanm____/",
            "author" => false,
            "order" => 2
        ],
        [
            "icon" => "fab fa-twitter",
            "asset" => false,
            "href" => "https://twitter.com/bojanm_",
            "author" => false,
            "order" => 3
        ],
        [
            "icon" => "fas fa-file-pdf",
            "asset" => true,
            "href" => "documentation.pdf",
            "author" => false,
            "order" => 4
        ],
        [
            "icon" => "fab fa-facebook-f",
            "asset" => false,
            "href" => "https://www.facebook.com/bojan.maksimovic.908",
            "author" => true,
            "order" => 1
        ],
        [
            "icon" => "fab fa-twitter",
            "asset" => false,
            "href" => "https://twitter.com/bojanm_",
            "author" => true,
            "order" => 2
        ],
        [
            "icon" => "fab fa-github",
            "asset" => false,
            "href" => "https://github.com/bojanmks",
            "author" => true,
```

```
        "order" => 3
    ],
    [
        "icon" => "fas fa-file-pdf",
        "asset" => true,
        "href" => "documentation.pdf",
        "author" => true,
        "order" => 4
    ]
];

public function run()
{
    \DB::table("social")->insert($this->socials);
}
}
```

UserSeeder

```
<?php

namespace Database\Seeders;

use App\Models\User;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $users = [
            [
                "username" => "korisnik1",
                "email" => "korisnik1@gmail.com",
                "password" => md5(env('USER_PASSWORD')),
                "role_id" => 1
            ],
            [
                "username" => "korisnik2",
                "email" => "korisnik2@gmail.com",
                "password" => md5(env('USER_PASSWORD')),
                "role_id" => 1
            ],
            [
                "username" => "admin1",
                "email" => "admin1@gmail.com",
                "password" => md5(env('ADMIN_PASSWORD')),
                "role_id" => 2
            ],
            [
                "username" => "admin2",
                "email" => "admin2@gmail.com",
                "password" => md5(env('ADMIN_PASSWORD')),
                "role_id" => 2
            ]
        ];

        foreach($users as $u) {
            $u["created_at"] = now();
            \DB::table("user")->insert($u);
        }
    }
}
```

3.9 Components

Food

```
<?php

namespace App\View\Components;

use Illuminate\View\Component;

class Food extends Component
{
    /**
     * Create a new component instance.
     *
     * @return void
     */
    public $id;
    public $name;
    public $image;
    public $price;
    public $md;
    public $lg;

    public function __construct($id, $name, $image, $price, $md = 6, $lg = 4)
    {
        $this->id = $id;
        $this->name = $name;
        $this->image = $image;
        $this->price = $price;
        $this->md = $md;
        $this->lg = $lg;
    }

    /**
     * Get the view / contents that represent the component.
     *
     * @return \Illuminate\Contracts\View\View|\Closure|string
     */
    public function render()
    {
        return view('components.client.food');
    }
}
```

Restaurant

```
<?php

namespace App\View\Components;

use Illuminate\View\Component;

class Restaurant extends Component
{
    /**
     * Create a new component instance.
     *
     * @return void
     */
    public $id;
    public $name;
    public $image;
    public $categories;
    public $md;
    public $lg;

    public function __construct($id, $name, $image, $categories, $md = "6", $lg = "4")
    {
        $this->id = $id;
        $this->name = $name;
        $this->image = $image;
        $this->categories = $categories;
        $this->md = $md;
        $this->lg = $lg;
    }

    /**
     * Get the view / contents that represent the component.
     *
     * @return \Illuminate\Contracts\View\View|\Closure|string
     */
    public function render()
    {
        return view('components.client.restaurant');
    }
}
```

3.10 Blade

components.client.food

```
<div class="foodCol col-12 col-md-{{ $md }} col-lg-{{ $lg }}" mb-4">
  <div class="foodDiv d-flex align-items-center justify-content-between align-items-center">
    <div class="food-image d-flex align-items-center">
      
    </div>
    <div class="food-text">
      <h3 class="font-medium">{{ $name }}</h3>
      <span class="font-small">${ $price }</span>
      <br/>
      <a data-id="{{ $id }}" class="add-food color-primary d-flex align-items-center" href="#"><i class="fas fa-plus me-1"></i>Add to order</a>
    </div>
  </div>
</div>
```

components.client.restaurant

```
<div class="restaurantCol col-12 col-md-{{ $md }} col-lg-{{ $lg }} mb-4">
  <a href="{{ route("restaurants.show", ["restaurant" => $id]) }}">
    <div class="restaurantDiv rounded position-relative shadow" style="background-
image: url('{{ asset("assets/images/restaurants/" . $image) }}');">
      <div class="restaurantText position-absolute top-0 bottom-0 start-0 end-0
d-flex flex-column align-items-center justify-content-center">
        <h3 class="rounded font-large p-1 text-center fw-normal">{{ $name
}}</h3>
        <ul class="d-flex">
          @foreach($categories as $c)
            <li class="font-small text-center p-1 rounded mx-1">{{ $c->name
}}</li>
          @endforeach
        </ul>
      </div>
    </div>
  </a>
</div>
```

emails.contact

```
@component('mail:message')
# {{ $subject }}

{{ $message }}
<br/><br/>
<span>Sent from: {{ $email }}</span>
@endcomponent
```


fixed.admin.footer

```
<footer class="main-footer">
  <div class="float-right d-none d-sm-block">
    <b>Version</b> 3.2.0
  </div>
  <strong>Copyright &copy; 2014-2021 <a
href="https://adminlte.io">AdminLTE.io</a>.</strong> All rights reserved.
</footer>
```

fixed.admin.head

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Vlogo | Admin</title>
<link rel="icon" href="{{ asset("favicon.ico") }}" />
<!-- Google Font: Source Sans Pro -->
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=
fallback">
<!-- Font Awesome -->
{{--<link rel="stylesheet" href="{{ asset('assets/vendor/AdminLTE/plugins/fontawesome-
free/css/all.css') }}">--}}
<link rel="stylesheet" href="{{ asset('assets/vendor/fontawesome/css/all.min.css') }}">
<!-- overlayScrollbars -->
<link rel="stylesheet" href="{{
asset('assets/vendor/AdminLTE/plugins/overlayScrollbars/css/OverlayScrollbars.min.css')
 }}">
<!-- Theme style -->
<link rel="stylesheet" href="{{
asset('assets/vendor/AdminLTE/dist/css/adminlte.min.css') }}">
@yield('additionalCss')
<link rel="stylesheet" href="{{ asset('assets/css/admin.min.css') }}">
```

fixed.admin.nav

```
<!-- Navbar -->
<nav class="main-header navbar navbar-expand navbar-white navbar-light">
  <!-- Left navbar links -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i
class="fas fa-bars"></i></a>
    </li>
  </ul>

  <!-- Right navbar links -->
  <ul class="navbar-nav ml-auto">
    <li class="nav-item">
      <a class="nav-link" data-widget="fullscreen" href="#" role="button">
        <i class="fas fa-expand-arrows-alt"></i>
      </a>
    </li>
  </ul>
</nav>
<!-- /.navbar -->
```

fixed.admin.scripts

```
<!-- jQuery -->
<script src="{{ asset('assets/vendor/AdminLTE/plugins/jquery/jquery.min.js') }}"></script>
<!-- Bootstrap 4 -->
<script src="{{ asset('assets/vendor/AdminLTE/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<!-- overlayScrollbars -->
<script src="{{ asset('assets/vendor/AdminLTE/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js') }}"></script>
<!-- FontAwesome -->
<script src="{{ asset('assets/vendor/fontawesome/js/all.min.js') }}"></script>
<!-- AdminLTE App -->
<script src="{{ asset('assets/vendor/AdminLTE/dist/js/adminlte.min.js') }}"></script>
<!-- AdminLTE for demo purposes -->
<script src="{{ asset('assets/js/common.min.js') }}"></script>
@yield('additionalScripts')
{{--<script src="{{ asset('assets/vendor/AdminLTE/dist/js/demo.js') }}"></script>--}}
```

fixed.admin.sidebar

```
<!-- Main Sidebar Container -->
<aside class="main-sidebar main-sidebar-custom sidebar-dark-primary elevation-4">
  <!-- Brand Logo -->
  <a href="{{ route('admin.dashboard') }}" class="brand-link">
    
    <span class="brand-text font-weight-light">Vlogo</span>
  </a>

  <!-- Sidebar -->
  <div class="sidebar">
    <!-- Sidebar user (optional) -->
    <div class="user-panel mt-3 pb-3 mb-3 d-flex">
      <div class="image">
        
      </div>
      <div class="info">
        <a href="#" class="d-block">{{ session()->get('user')->username }}</a>
      </div>
    </div>

    <!-- SidebarSearch Form -->
    <div class="form-inline">
      <div class="input-group" data-widget="sidebar-search">
        <input class="form-control form-control-sidebar" type="search"
placeholder="Search" aria-label="Search">
        <div class="input-group-append">
          <button class="btn btn-sidebar">
            <i class="fas fa-search fa-fw"></i>
          </button>
        </div>
      </div>
    </div>

    <!-- Sidebar Menu -->
    <nav class="mt-2">
      <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview"
role="menu" data-accordion="false">
        <!-- Add icons to the links using the .nav-icon class
with font-awesome or any other icon font library -->
        @foreach($nav as $n)
          <li class="nav-item">
            <a href="{{ route($n->route) }}" class="nav-link @if(request()-
>routeIs($n->route)) active @endif">
              @if($n->icon) <i class="{{ $n->icon }}" nav-icon"></i>
            @endif
              <p>{{ $n->name }}</p>
            </a>
          </li>
        @endforeach
      </ul>
    </nav>
    <!-- /.sidebar-menu -->
  </div>
  <!-- /.sidebar -->

  <div class="sidebar-custom">
    <form action="{{ route('logout') }}" method="POST">
      @csrf
      <a href="{{ route('home') }}" class="btn btn-link"><i class="fas fa-sign-
out-alt"></i></a>
      <input type="submit" class="btn btn-secondary hide-on-collapse pos-right">
```

```
value="Log out"/>
    </form>
    {{--      <a href="{{ route('logout') }}" class="btn btn-secondary hide-on-
collapse">Log out</a>--}}
    {{--      <a href="#" class="btn btn-secondary hide-on-collapse pos-right">Help</a>--
    }}
  </div>
  <!-- /.sidebar-custom -->
</aside>
```

fixed.client.footer

```
<div id="footer" class="py-3">
  <div class="container d-flex flex-column align-items-center justify-content-
center">
    <ul id="socials" class="d-flex">
      @foreach($socials as $s)
        <li>
          <a href="{{ $s->asset ? asset($s->href) : $s->href }}"
target="_blank" class="p-1">
            <i class="{{ $s->icon }}"></i>
          </a>
        </li>
      @endforeach
    </ul>
    <span class="font-small mt-2">&copy; 2022 Bojan Maksimović. All rights
reserved.</span>
  </div>
</div>
```

fixed.client.head

```
<meta charset="UTF-8"/>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<title>
  @section("title")
    Vlogo
  @show
</title>
<meta name="description" content="Vlogo is the food delivery site that will get you
anything you want to your doorstep. You order online, you'll have it!"/>
<meta name="keywords" content="Vlogo, food delivery, Restaurants, Recommended, Home"/>
<meta name="csrf-token" content="{{ csrf_token() }}" />
<link rel="icon" href="{{ asset("favicon.ico") }}" />
<link rel="stylesheet" href="{{ asset("assets/vendor/fontawesome/css/all.min.css") }}"
crossorigin="anonymous"/>
<link rel="stylesheet" href="{{ asset("assets/vendor/bootstrap/css/bootstrap.min.css")
}}"/>
<link rel="stylesheet" href="{{ asset("assets/css/app.min.css") }}" />
```


fixed.client.header

```
<div id="header" class="py-2 shadow">
  <div class="container d-flex align-items-center justify-content-between">
    <div id="logo" class="mt-2">
      <a href="{{ route('home') }}">
        
      </a>
    </div>
    <nav id="nav" class="d-flex align-items-center">
      <ul class="d-flex me-4">
        @foreach($nav as $n)
          <li>
            <a href="{{ route($n->route) }}" class="p-2 @if(request()->routeIs($n->route)) active @endif">{{ $n->name }}</a>
          </li>
        @endforeach
      </ul>
      @if(!session("user"))
        <a href="{{ route('form.login') }}" id="btnSignIn" class="px-3 py-2
rounded-pill shadow font-small">Sign in</a>
      @else
        <span id="userDivFront" class="font-small position-relative user-div p-
1">{{ session("user")->username }}&nbsp;<i class="fas fa-caret-down"></i>
        <div id="userHoverMenu" class="position-absolute shadow rounded py-1">
          <ul>
            @if(session("user")->role_name == "admin")
              <li>
                <a href="{{ route('admin.dashboard') }}" class="px-1 py-
2">Admin</a>
              </li>
            @endif
            <li>
              <form action="{{ route('logout') }}" method="post">
                @csrf
                <input type="submit" value="Log out" class="px-1 py-
2"/>
              </form>
            </li>
          </ul>
        </div>
      </span>
    @endif
    <span id="hamburgerMenu" class="ms-4 me-2">
      <a href="#" class="font-large">
        <i class="fas fa-bars"></i>
      </a>
    </span>
  </nav>
</div>
<div id="sideNav">
  <ul class="d-flex flex-column">
    @foreach($nav as $n)
      <li>
        <a href="{{ route($n->route) }}" class="d-block py-2 ps-3 ps-sm-5
@if(request()->routeIs($n->route)) active @endif">
          {{ $n->name }}
        </a>
      </li>
    @endforeach
  </ul>
</div>
</div>
```

```
<div id="sidenavCover" class="cover">  
</div>
```

fixed.client.scripts

```
<script src="{{ asset("assets/vendor/bootstrap/js/bootstrap.min.js") }}"></script>
<script src="{{ asset("assets/vendor/jquery/jquery.min.js") }}"></script>
<script src="{{ asset("assets/vendor/jquery/jquery.easing.min.js") }}"></script>
<script src="{{ asset('assets/vendor/fontawesome/js/all.min.js') }}"></script>
<script src="{{ asset("assets/js/common.min.js") }}"></script>
<script src="{{ asset("assets/js/client.min.js") }}"></script>
```

layouts.admin

```
<!DOCTYPE html>
<html>
  <head>
    @include('fixed.admin.head')
  </head>
  <body class="hold-transition sidebar-mini layout-fixed">
    <div class="wrapper">
      @include('fixed.admin.nav')
      @include('fixed.admin.sidebar')
      <div class="content-wrapper">
        <section class="content-header">
          <div class="container-fluid">
            @if(isset($pages))
              <nav aria-label="breadcrumb" class="mb-4">
                <ol class="breadcrumb">
                  @for($i = 0; $i < count($pages) - 1; $i++)
                    <li class="breadcrumb-item"><a href="{ route($pages[$i]['route']) }}">{{ $pages[$i]['name'] }}</a></li>
                  @endfor
                  <li class="breadcrumb-item active" aria-current="page">{{ $pages[count($pages) - 1]['name'] }}</li>
                </ol>
              </nav>
            @endif
            @yield('contentHeader')
          </div>
        </section>
        <section class="content">
          <div class="container-fluid">
            @yield('content')
          </div>
        </section>
      </div>
      @include('fixed.admin.footer')
    </div>
    @include('fixed.admin.scripts')
  </body>
</html>
```

layouts.client

```
<!DOCTYPE html>
<html>
  <head>
    @include("fixed.client.head")
  </head>
  <body>
    @include("fixed.client.header")

    <div id="content" class="min-vh-100">
      @yield("content")
    </div>

    @include("fixed.client.footer")

    @section("scripts")
      @include("fixed.client.scripts")
    @show
  </body>
</html>
```

pages.admin.dashboard

```
@extends('layouts.admin')

@section('contentHeader')
    <h1>Welcome back, {{ session()->get('user')->username }}!</h1>
@endsection

@section('content')
    <div class="row">
        @foreach($infoBoxes as $i)
            <div class="col-lg-3 col-6">
                <!-- small card -->
                <div class="small-box {{ $i['colorClass'] }}">
                    <div class="inner">
                        <h3>{{ $i['value'] }}</h3>

                        <p>{{ $i['text'] }}</p>
                    </div>
                    <div class="icon">
                        <i class="{{ $i['icon'] }}"></i>
                    </div>
                </div>
            </div>
        @endforeach
    </div>
@endsection
```

```

@extends('layouts.admin')

@section('additionalCss')
    @include('partials.datatables-css')
@endsection

@section('contentHeader')
    <h1>Food</h1>
@endsection

@section('content')
    @include('partials.messages')
    <form id="actionsForm" class="mt-4 mb-4" action="{{ route('admin.actions') }}"
method="GET">
        @csrf
        <div class="row">
            <div class="col-12 col-md-5 mb-2">
                <div class="input-group">
                    <div class="input-group-prepend">
                        <span class="input-group-text">From</span>
                    </div>
                    <input class="form-control" type="date" name="from" @if(request()-
>has('from')) value="{{ request()->get('from') }}" @endif/>
                </div>
            </div>
            <div class="col-12 col-md-5 mb-2">
                <div class="input-group">
                    <div class="input-group-prepend">
                        <span class="input-group-text">To</span>
                    </div>
                    <input class="form-control" type="date" name="to" @if(request()-
>has('to')) value="{{ request()->get('to') }}" @endif/>
                </div>
            </div>
            <div class="col-12 col-md-2 mb-2">
                <input type="submit" value="Filter" class="btn btn-secondary"/>
            </div>
        </div>
    </form>
    <hr/>
    <div class="table-responsive">
        <table id="actionsTable" class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>#</th>
                    <th>IP</th>
                    <th>Path</th>
                    <th>Method</th>
                    <th>User</th>
                    <th class="large-column">Action</th>
                    <th class="large-column">Created At</th>
                </tr>
            </thead>
            <tbody>
                @foreach($actions as $key=>$a)
                    <tr>
                        <td>{{ $key + 1 }}</td>
                        <td>{{ $a->ip }}</td>
                        <td>{{ $a->path }}</td>
                        <td>{{ strtoupper($a->method) }}</td>
                        <td>{{ $a->username ? : 'N/A' }}</td>
                        <td>{{ $a->action }}</td>
                        <td>{{ $a->created_at }}</td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>

```

```
        </tr>
      @endforeach
    </tbody>
  </table>
</div>
@endsection

@section('additionalScripts')
  <script>
    const table = "actionsTable";
  </script>
  @include('partials.datatables-scripts')
@endsection
```



```

@extends('layouts.admin')

@section('contentHeader')
    <h1>Add new food</h1>
@endsection

@section('content')
    <form id="foodForm" action="{{ route('admin.food.store') }}" method="POST"
class="standard-form" enctype="multipart/form-data">
        @csrf
        <div class="mb-3">
            <label for="name">Name</label>
            <input class="form-control" type="text" name="name" id="name"
placeholder="Name"/>
            <span class="font-small error-message">This field is required</span>
        </div>
        <div class="mb-3">
            <label for="image">Image (1:1 for best results)</label>
            <input class="form-control-file" type="file" name="image" id="image"/>
            <span class="font-small error-message">Image is required</span>
        </div>
        <div class="mb-3">
            <label for="restaurant">Restaurant</label>
            <select class="form-control" name="restaurant" id="restaurant">
                @foreach($restaurants as $r)
                    <option value="{{ $r->id }}">{{ $r->name }}</option>
                @endforeach
            </select>
        </div>
        <div class="mb-3">
            <label for="category">Category</label>
            <select class="form-control" name="category" id="category">
                @foreach($categories as $c)
                    <option value="{{ $c->id }}">{{ $c->name }}</option>
                @endforeach
            </select>
        </div>
        <div class="mb-3">
            <label for="price">Price ($)</label>
            <input class="form-control" type="number" name="price" id="price"
placeholder="Price"/>
            <span class="font-small error-message">This field is required</span>
        </div>
        <div class="text-center">
            <input id="btnSubmit" class="btn btn-primary" type="submit" value="Submit"
/>
        </div>
        @include('partials.messages')
    </form>
@endsection

@section('additionalScripts')
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script>
        const edit = false;
    </script>
    <script src="{{ asset('assets/js/admin/food.min.js') }}"></script>
@endsection

```

```

@extends('layouts.admin')

@section('contentHeader')
    <h1>Edit food</h1>
@endsection

@section('content')
    <form id="foodForm" action="{{ route('admin.food.update', ['food' => $food->id]) }}" method="POST" class="standard-form" enctype="multipart/form-data">
        @csrf
        @method('PUT')
        <div class="mb-3">
            <label for="name">Name</label>
            <input class="form-control" type="text" name="name" id="name"
placeholder="Name" value="{{ $food->name }}" />
            <span class="font-small error-message">This field is required</span>
        </div>
        <div class="mb-3">
            <label for="image">Image (1:1 for best results, Leave empty if
unchanged)</label>
            <input class="form-control-file" type="file" name="image" id="image" />
            <span class="font-small error-message">Image is required</span>
        </div>
        <div class="mb-3">
            <label for="restaurant">Restaurant</label>
            <select class="form-control" name="restaurant" id="restaurant">
                @foreach($restaurants as $r)
                    <option @if($r->id == $food->restaurant_id) selected @endif
value="{{ $r->id }}">{{ $r->name }}</option>
                @endforeach
            </select>
        </div>
        <div class="mb-3">
            <label for="category">Category</label>
            <select class="form-control" name="category" id="category">
                @foreach($categories as $c)
                    <option @if($c->id == $food->category_id) selected @endif value="{{
$c->id }}">{{ $c->name }}</option>
                @endforeach
            </select>
        </div>
        <div class="mb-3">
            <label for="price">Price ($)</label>
            <input class="form-control" type="number" name="price" id="price"
placeholder="Price" step=".01" value="{{ $food->price }}" />
            <span class="font-small error-message">This field is required</span>
        </div>
        <div class="text-center">
            <input id="btnSubmit" class="btn btn-primary" type="submit" value="Submit"
/>
        </div>
        @include('partials.messages')
    </form>
@endsection

@section('additionalScripts')
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script>
        const edit = true;
    </script>
    <script src="{{ asset('assets/js/admin/food.min.js') }}"></script>
@endsection

```

```

@extends('layouts.admin')

@section('additionalCss')
    @include('partials.datatables-css')
@endsection

@section('contentHeader')
    <h1>Food</h1>
@endsection

@section('content')
    <a href="{{ route('admin.food.create') }}" class="btn btn-primary">Add new food</a>
    @include('partials.messages')
    <div class="table-responsive">
        <table id="foodTable" class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Image</th>
                    <th>Restaurant</th>
                    <th>Category</th>
                    <th>Price</th>
                    <th class="large-column">Created At</th>
                    <th class="large-column">Updated At</th>
                    <th>Edit</th>
                    <th>Delete</th>
                </tr>
            </thead>
            <tbody>
                @foreach($food as $key=>$f)
                    <tr>
                        <td>{{ $f->id }}</td>
                        <td>{{ $f->name }}</td>
                        <td>
                            <div class="food-image">
                                name }}" />
                            </div>
                        </td>
                        <td>{{ $f->restaurant_name }}</td>
                        <td>{{ $f->category_name }}</td>
                        <td>{{ $f->price }}</td>
                        <td>{{ $f->created_at }}</td>
                        <td>{{ $f->updated_at ? : 'N/A' }}</td>
                        <td><a href="{{ route('admin.food.edit', ['food' => $f->id]) }}"><i
class="fas fa-edit"></i></a></td>
                        <td>
                            <form action="{{ route('admin.food.destroy', ['food' => $f-
>id]) }}" method="POST">
                                @csrf
                                @method('DELETE')
                                <button class="btn btn-link"><i class="fa fa-trash" aria-
hidden="true"></i></button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
@endsection

```

```
@section('additionalScripts')
  <script>
    const table = "foodTable";
  </script>
  @include('partials.datatables-scripts')
@endsection
```

pages.admin.foodCategories.create

```
@extends('layouts.admin')

@section('contentHeader')
    <h1>Add new food category</h1>
@endsection

@section('content')
    <form id="categoryForm" action="{{ route('admin.categories.store') }}"
method="POST" class="standard-form">
        @csrf
        <div class="mb-3">
            <label for="name">Name</label>
            <input class="form-control" type="text" name="name" id="name"
placeholder="Name"/>
            <span class="error-message">Needs to start with a capital letter and each
word has to be at least 3 characters long</span>
        </div>
        <div class="form-check mb-3">
            <input class="form-check-input" type="checkbox" name="listed" id="listed"/>
            <label class="form-check-label" for="listed">Listed</label>
        </div>
        <div class="text-center">
            <input id="btnSubmit" class="btn btn-primary" type="submit" value="Submit"
/>
        </div>
        @include('partials.messages')
    </form>
@endsection

@section('additionalScripts')
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script src="{{ asset('assets/js/admin/categories.min.js') }}"></script>
@endsection
```

```

@extends('layouts.admin')

@section('contentHeader')
    <h1>Edit category</h1>
@endsection

@section('content')
    <form id="categoryForm" action="{{ route('admin.categories.update', ["category" =>
$category->id]) }}" method="POST" class="standard-form">
        @csrf
        @method('PUT')
        <div class="mb-3">
            <label for="name">Name</label>
            <input class="form-control" type="text" name="name" id="name"
placeholder="Name" value="{{ $category->name }}" />
            <span class="error-message">Needs to start with a capital letter and each
word has to be at least 3 characters long</span>
        </div>
        <div class="form-check mb-3">
            <input class="form-check-input" type="checkbox" name="listed" id="listed"
@if($category->listed) checked @endif/>
            <label class="form-check-label" for="listed">Listed</label>
        </div>
        <div class="text-center">
            <input id="btnSubmit" class="btn btn-primary" type="submit" value="Submit"
/>
        </div>
        @include('partials.messages')
    </form>
@endsection

@section('additionalScripts')
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script src="{{ asset('assets/js/admin/categories.min.js') }}"></script>
@endsection

```

```

@extends('layouts.admin')

@section('additionalCss')
    @include('partials.datatables-css')
@endsection

@section('contentHeader')
    <h1>Food categories</h1>
@endsection

@section('content')
    <a href="{{ route('admin.categories.create') }}" class="btn btn-primary">Add new
category</a>
    @include('partials.messages')
    <div class="table-responsive">
        <table id="categoriesTable" class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Name</th>
                    <th>Listed</th>
                    <th class="large-column">Created At</th>
                    <th class="large-column">Updated At</th>
                    <th>Edit</th>
                    <th>Delete</th>
                </tr>
            </thead>
            <tbody>
                @foreach($categories as $key=>$c)
                    <tr>
                        <td>{{ $key + 1 }}</td>
                        <td>{{ $c->name }}</td>
                        <td>{{ $c->listed }}</td>
                        <td>{{ $c->created_at }}</td>
                        <td>{{ $c->updated_at ? : 'N/A' }}</td>
                        <td><a href="{{ route('admin.categories.edit', ["category" => $c-
>id]) }}"><i class="fas fa-edit"></i></a></td>
                        <td>
                            <form action="{{ route('admin.categories.destroy', ["category"
=> $c->id]) }}" method="POST">
                                @csrf
                                @method('DELETE')
                                <button class="btn btn-link"><i class="fa fa-trash" aria-
hidden="true"></i></button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
@endsection

@section('additionalScripts')
    <script>
        const table = "categoriesTable";
    </script>
    @include('partials.datatables-scripts')
@endsection

```

```

@extends('layouts.admin')

@section('additionalCss')
    @include('partials.datatables-css')
@endsection

@section('contentHeader')
    <h1>Order details</h1>
@endsection

@section('content')
    @include('partials.messages')
    <div class="table-responsive">
        <table id="detailsTable" class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Food</th>
                    <th>Image</th>
                    <th class="large-column">Price at the time</th>
                    <th>Amount</th>
                    <th class="large-column">Created At</th>
                </tr>
            </thead>
            <tbody>
                @foreach($details as $key=>$d)
                    <tr>
                        <td>{{ $key + 1 }}</td>
                        <td>{{ $d->name }}</td>
                        <td>
                            <div class="food-image">
                                name }}" />
                            </div>
                        </td>
                        <td>${{ $d->price_at_the_time }}</td>
                        <td>x{{ $d->amount }}</td>
                        <td>{{ $d->created_at }}</td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
@endsection

@section('additionalScripts')
    <script>
        const table = "detailsTable";
    </script>
    @include('partials.datatables-scripts')
@endsection

```



```

@extends('layouts.admin')

@section('additionalCss')
    @include('partials.datatables-css')
@endsection

@section('contentHeader')
    <h1>Orders</h1>
@endsection

@section('content')
    @include('partials.messages')
    <div class="table-responsive">
        <table id="ordersTable" class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>User</th>
                    <th>Restaurant</th>
                    <th>Address</th>
                    <th>Total Price</th>
                    <th class="large-column">Created At</th>
                    <th>Details</th>
                </tr>
            </thead>
            <tbody>
                @foreach($orders as $key=>$o)
                    <tr>
                        <td>{{ $o->id }}</td>
                        <td>{{ $o->username }}</td>
                        <td>{{ $o->restaurant_name }}</td>
                        <td>{{ $o->address }}</td>
                        <td>{{ $o->total_price }}</td>
                        <td>{{ $o->created_at }}</td>
                        <td>
                            <a href="{{ route('admin.orders.details', ['id' => $o->id]) }}">
                                <i class="fas fa-receipt"></i>
                            </a>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
@endsection

@section('additionalScripts')
    <script>
        const table = "ordersTable";
    </script>
    @include('partials.datatables-scripts')
@endsection

```

```

@extends('layouts.admin')

@section('contentHeader')
    <h1>Add new restaurant</h1>
@endsection

@section('content')
    <form id="restaurantForm" action="{{ route('admin.restaurants.store') }}"
method="POST" class="standard-form" enctype="multipart/form-data">
        @csrf
        <div class="mb-3">
            <label for="name">Name</label>
            <input class="form-control" type="text" name="name" id="name"
placeholder="Name"/>
            <span class="font-small error-message">This field is required</span>
        </div>
        <div class="mb-3">
            <label for="image">Image</label>
            <input class="form-control-file" type="file" name="image" id="image"/>
            <span class="font-small error-message">Image is required</span>
        </div>
        <div class="form-check mb-3">
            <input class="form-check-input" type="checkbox" name="recommended"
id="recommended"/>
            <label class="form-check-label" for="recommended">Recommended</label>
        </div>
        <div class="form-check mb-3">
            <input class="form-check-input" type="checkbox" name="active" id="active"/>
            <label class="form-check-label" for="active">Active</label>
        </div>
        <div class="text-center">
            <input id="btnSubmit" class="btn btn-primary" type="submit" value="Submit"
/>
        </div>
        @include('partials.messages')
    </form>
@endsection

@section('additionalScripts')
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script>
        const edit = false;
    </script>
    <script src="{{ asset('assets/js/admin/restaurants.min.js') }}"></script>
@endsection

```

```

@extends('layouts.admin')

@section('contentHeader')
    <h1>Edit restaurant</h1>
@endsection

@section('content')
    <form id="restaurantForm" action="{{ route('admin.restaurants.update',
["restaurant" => $restaurant->id]) }}" method="POST" class="standard-form"
enctype="multipart/form-data">
        @csrf
        @method('PUT')
        <div class="mb-3">
            <label for="name">Name</label>
            <input class="form-control" type="text" name="name" id="name"
placeholder="Name" value="{{ $restaurant->name }}" />
            <span class="font-small error-message">This field is required</span>
        </div>
        <div class="mb-3">
            <label for="image">Image (Leave empty if unchanged)</label>
            <input class="form-control-file" type="file" name="image" id="image" />
            <span class="font-small error-message">Image is required</span>
        </div>
        <div class="form-check mb-3">
            <input class="form-check-input" type="checkbox" name="recommended"
id="recommended" @if($restaurant->recommended) checked @endif/>
            <label class="form-check-label" for="recommended">Recommended</label>
        </div>
        <div class="form-check mb-3">
            <input class="form-check-input" type="checkbox" name="active" id="active"
@if($restaurant->active) checked @endif/>
            <label class="form-check-label" for="active">Active</label>
        </div>
        <div class="text-center">
            <input id="btnSubmit" class="btn btn-primary" type="submit" value="Submit"
/>
        </div>
        @include('partials.messages')
    </form>
@endsection

@section('additionalScripts')
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script>
        const edit = true;
    </script>
    <script src="{{ asset('assets/js/admin/restaurants.min.js') }}"></script>
@endsection

```

```

@extends('layouts.admin')

@section('additionalCss')
    @include('partials.datatables-css')
@endsection

@section('contentHeader')
    <h1>Restaurants</h1>
@endsection

@section('content')
    <a href="{{ route('admin.restaurants.create') }}" class="btn btn-primary">Add new
restaurant</a>
    @include('partials.messages')
    <div class="table-responsive">
        <table id="restaurantsTable" class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Name</th>
                    <th>Image</th>
                    <th>Active</th>
                    <th class="large-column">Created At</th>
                    <th class="large-column">Updated At</th>
                    <th>Edit</th>
                    <th>Delete</th>
                </tr>
            </thead>
            <tbody>
                @foreach($restaurants as $key=>$r)
                    <tr>
                        <td>{{ $key + 1 }}</td>
                        <td>{{ $r->name }}</td>
                        <td>
                            <div class="restaurant-image">
                                name }}" />
                            </div>
                        </td>
                        <td>{{ $r->active }}</td>
                        <td>{{ $r->created_at }}</td>
                        <td>{{ $r->updated_at ? : 'N/A' }}</td>
                        <td><a href="{{ route('admin.restaurants.edit', ["restaurant" =>
$r->id]) }}"><i class="fas fa-edit"></i></a></td>
                        <td>
                            <form action="{{ route('admin.restaurants.destroy',
["restaurant" => $r->id]) }}" method="POST">
                                @csrf
                                @method('DELETE')
                                <button class="btn btn-link"><i class="fa fa-trash" aria-
hidden="true"></i></button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
@endsection

@section('additionalScripts')
    <script>
        const table = "restaurantsTable";

```

```
</script>  
@include('partials.datatables-scripts')  
@endsection
```

```

@extends('layouts.admin')

@section('contentHeader')
    <h1>Add new user</h1>
@endsection

@section('content')
    <form id="userForm" action="{{ route('admin.users.store') }}" method="POST"
class="standard-form">
        @csrf
        <div class="mb-3">
            <label for="username">Username</label>
            <input class="form-control" type="text" name="username" id="username"
placeholder="Username"/>
            <span class="font-small error-message">User's username cannot contain
special characters other than '_', must contain at least one alphabetical character and
be between 8 and 25 characters long</span>
        </div>
        <div class="mb-3">
            <label for="email">Email</label>
            <input class="form-control" type="email" name="email" id="email"
placeholder="Email"/>
            <span class="font-small error-message">name@example.com</span>
        </div>
        <div class="mb-3">
            <label for="password">Password</label>
            <input class="form-control" type="password" name="password" id="password"
placeholder="Password"/>
            <span class="font-small error-message">User's password needs to be at least
8 characters long and contain at least one letter and one number</span>
        </div>
        <div class="mb-3">
            <label for="role">Role</label>
            <select class="form-control" name="role" id="role">
                @foreach($roles as $r)
                    <option @if($r->name === 'user') selected @endif value="{{ $r->id
}}">{{ $r->name }}</option>
                @endforeach
            </select>
        </div>
        <div class="text-center">
            <input id="btnSubmit" class="btn btn-primary" type="submit" value="Submit"
/>
        </div>
        @include('partials.messages')
    </form>
@endsection

@section('additionalScripts')
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script>
        const edit = false;
    </script>
    <script src="{{ asset('assets/js/admin/users.min.js') }}"></script>
@endsection

```

```

@extends('layouts.admin')

@section('contentHeader')
    <h1>Edit user</h1>
@endsection

@section('content')
    <form id="userForm" action="{{ route('admin.users.update', ['user' => $user->id]) }}" method="POST" class="standard-form">
        @csrf
        @method('PUT')
        <div class="mb-3">
            <label for="username">Username</label>
            <input class="form-control" type="text" name="username" id="username"
placeholder="Username" value="{{ $user->username }}" />
            <span class="font-small error-message">User's username cannot contain
special characters other than '_', must contain at least one alphabetical character and
be between 8 and 25 characters long</span>
        </div>
        <div class="mb-3">
            <label for="email">Email</label>
            <input class="form-control" type="email" name="email" id="email"
placeholder="Email" value="{{ $user->email }}" />
            <span class="font-small error-message">name@example.com</span>
        </div>
        <div class="mb-3">
            <label for="password">Password (Leave empty if unchanged)</label>
            <input class="form-control" type="password" name="password" id="password"
placeholder="Password" />
            <span class="font-small error-message">User's password needs to be at least
8 characters long and contain at least one letter and one number</span>
        </div>
        <div class="mb-3">
            <label for="role">Role</label>
            <select class="form-control" name="role" id="role">
                @foreach($roles as $r)
                    <option @if($user->role_id === $r->id) selected @endif value="{{
$r->id }}">{{ $r->name }}</option>
                @endforeach
            </select>
        </div>
        <div class="text-center">
            <input id="btnSubmit" class="btn btn-primary" type="submit" value="Submit"
/>
        </div>
        @include('partials.messages')
    </form>
@endsection

@section('additionalScripts')
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script>
        const edit = true;
    </script>
    <script src="{{ asset('assets/js/admin/users.min.js') }}"></script>
@endsection

```

```

@extends('layouts.admin')

@section('additionalCss')
    @include('partials.datatables-css')
@endsection

@section('contentHeader')
    <h1>Users</h1>
@endsection

@section('content')
    <a href="{{ route('admin.users.create') }}" class="btn btn-primary">Add new
user</a>
    @include('partials.messages')
    <div class="table-responsive">
        <table id="usersTable" class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Username</th>
                    <th>Email</th>
                    <th>Role</th>
                    <th class="large-column">Created At</th>
                    <th class="large-column">Updated At</th>
                    <th>Edit</th>
                    <th>Delete</th>
                </tr>
            </thead>
            <tbody>
                @foreach($users as $key=>$u)
                    <tr>
                        <td>{{ $key + 1 }}</td>
                        <td>{{ $u->username }}</td>
                        <td>{{ $u->email }}</td>
                        <td>{{ $u->role_name }}</td>
                        <td>{{ $u->created_at }}</td>
                        <td>{{ $u->updated_at ? : 'N/A' }}</td>
                        <td><a href="{{ route('admin.users.edit', ['user' => $u->id])
}}"><i class="fas fa-edit"></i></a></td>
                        <td>
                            <form action="{{ route('admin.users.destroy', ['user' => $u-
>id]) }}" method="POST">
                                @csrf
                                @method('DELETE')
                                <button class="btn btn-link"><i class="fa fa-trash" aria-
hidden="true"></i></button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
@endsection

@section('additionalScripts')
    <script>
        const table = "usersTable";
    </script>
    @include('partials.datatables-scripts')
@endsection

```



```

@extends("layouts.client")

@section("title")
    Vlogo | Author
@endsection

@section("content")
    <div class="section-padding min-vh-100 d-flex align-items-center">
        <div class="container">
            <div class="row">
                <div class="col-12 col-lg-5 d-flex flex-column align-items-center justify-content-center">
                    <div id="authorImage" class="text-center">
                        
                    </div>
                    <span class="mt-3 font-medium">Bojan Maksimović 92/19</span>
                    <span class="font-small opacity-75">Web Developer</span>
                </div>
                <div class="col-12 col-lg-7 d-flex align-items-center align-items-lg-start justify-content-center flex-column mt-3 text-center text-lg-start">
                    <br class="d-block d-lg-none"/>
                    <h2 class="font-large">About me</h2>
                    <hr class="w-100 mt-0"/>
                    <br/>
                    <p class="font-small">Hello! I am Bojan, an aspiring back-end web developer. I'm a driven individual with the ability to learn and collaborate in rapidly changing environments and compositions. I am eager to tackle web development challenges to achieve lasting impacts on user experience.</p>
                    <div id="authorSocials" class="mx-auto mx-lg-0">
                        <ul class="d-flex">
                            @foreach($authorSocials as $sas)
                                <li>
                                    <a href="{{ $sas->asset ? asset($sas->href) : $sas->href }}" target="_blank" class="rounded me-1 font-small d-flex align-items-center justify-content-center"><i class="{{ $sas->icon }}"></i></a>
                                </li>
                            @endforeach
                        </ul>
                    </div>
                    <a id="portfolioLink" href="http://wd2.bojanmaksimovic.com/" target="_blank" class="py-2 mt-2">Portfolio Website</a>
                </div>
            </div>
        </div>
    </div>
@endsection

```

```

@extends("layouts.client")

@section("title")
    Vlogo | Contact
@endsection

@section("content")
    <div class="section-padding">
        <div class="container">
            <h2 class="text-center font-xl">Contact us</h2>
            <br/>
            <div class="row">
                <div class="col-12 col-lg-5">
                    <p class="font-small">
                        If you wish to reach out to us, you can do so via social media,
the information listed below, or by filling out the contact form.
                    </p>
                    <br/>
                    <ul class="font-small">
                        <li>
                            <i class="fas fa-phone me-1"></i> +381 60/123-4567
                        </li>
                        <li>
                            <i class="fas fa-envelope me-1"></i>
bojan.maxim075@gmail.com
                        </li>
                    </ul>
                    <hr class="d-block d-lg-none w-100 my-5"/>
                </div>
                <div class="col-12 col-lg-7">
                    <form id="contactForm" class="font-small mx-auto" action="{{
route("mail.contact") }}" method="POST">
                        @csrf
                        <div class="mb-3">
                            <label for="email" class="form-label">Email address</label>
                            <input type="email" class="form-control font-small"
id="email" name="email" placeholder="name@example.com" value="{{ old("email") }}" />
                            <span class="error-message font-
small">name@example.com</span>
                        </div>
                        <div class="mb-3">
                            <label for="subject" class="form-label">Subject</label>
                            <input type="text" class="form-control font-small"
id="subject" name="subject" placeholder="Subject" value="{{ old("subject") }}" />
                            <span class="error-message font-small">This field is
required</span>
                        </div>
                        <div class="mb-3">
                            <label for="message" class="form-label">Message</label>
                            <textarea class="form-control font-small" id="message"
name="message" rows="10" placeholder="Your message" value="{{ old("message")
}} "></textarea>
                            <span class="error-message font-small">This field is
required</span>
                        </div>
                        <button id="btnSendMessage" class="btn btn-primary text-center
d-block mx-auto px-4">
                            <i class="fas fa-paper-plane"></i>
                        </button>
                        @include("partials.messages")
                    </form>
                </div>
            </div>
        </div>
    </div>

```

```
        </div>
    </div>
@endsection

@section("scripts")
    @parent
    <script src="{{ asset("assets/js/form-validation.min.js") }}"></script>
    <script src="{{ asset("assets/js/contact.min.js") }}"></script>
@endsection
```

```

@extends("layouts.client")

@section("title")
    Vlogo | Home
@endsection

@section("content")
    <div id="mainSlider" class="vh-100 d-flex align-items-center">
        <div class="container">
            <div class="row">
                <div class="col-12 col-md-6 d-none d-lg-flex justify-content-center">
                    <div id="homepageImage">
                        
                    </div>
                </div>
                <div class="col-12 col-lg-6 text-center text-lg-start d-flex flex-
column justify-content-center align-items-center align-items-lg-start">
                    <h1 class="font-xxl">Best food delivery service in the world</h1>
                    <form id="homepageForm" class="mt-4" onsubmit="return false;">
                        <div class="input-group shadow position-relative">
                            <span class="input-group-text font-small py-2 px-3 rounded-
start"><i class="fa-solid fa-magnifying-glass"></i></span>
                            <input autocomplete="off" id="tbSearch" type="text"
class="form-control font-small p-2 rounded-end" placeholder="Search for a specific
restaurant"/>
                            <div id="searchResults" class="position-absolute top-100
left-0 w-100 rounded">
                                <ul>
                                </ul>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
        <div id="mainSliderWave" style="height: 150px; overflow: hidden;" ><svg viewBox="0
0 500 150" preserveAspectRatio="none" style="height: 100%; width: 100%;"><path
d="M0.00,49.98 C149.99,150.00 350.88,-49.98 500.00,49.98 L500.00,0.00 L0.00,0.00 Z"
style="stroke: none;"></path></svg></div>
        <div id="recommendedRestaurants" class="pb-5">
            <h2 class="font-xl text-center mb-5">Recommended by <span class="color-
secondary">Vlogo</span></h2>
            <div class="container">
                <div class="row">
                    @foreach($recommendedRestaurants as $r)
                        <x-restaurant :id="$r->id" :name="$r->name" :image="$r->image"
:categories="$r->topCategories" />
                    @endforeach
                </div>
            </div>
        </div>
    </div>
    @endsection

@section("scripts")
    @parent
    <script src="{{ asset("assets/js/home.min.js") }}"></script>
    @endsection

```

```

@extends("layouts.client")

@section("title")
    Vlogo | Login
@endsection

@section("content")
    <div class="section-padding vh-100 d-flex align-items-center justify-content-center">
        <div class="container">
            <h2 class="text-center font-xl">Log in</h2>
            <form id="loginForm" class="font-small mx-auto" action="{{ route('login') }}" method="POST">
                @csrf
                <div class="mb-3">
                    <label for="email" class="form-label">Email address</label>
                    <input type="email" class="form-control font-small" id="email"
name="email" placeholder="name@example.com" @if(session("email")) value="{{ session("email") }}" @endif/>
                    <span class="font-small error-message">name@example.com</span>
                </div>
                <div class="mb-3">
                    <label for="password" class="form-label">Password</label>
                    <input type="password" class="form-control font-small"
id="password" name="password" placeholder="Password"/>
                    <span class="font-small error-message">Your password needs to be at
least 8 characters long and contain at least one letter and one number</span>
                </div>
                <button id="btnLogin" class="btn btn-primary text-center d-block mx-
auto px-4">
                    Log in
                </button>
                <br/>
                <a class="signInFormLink" href="{{ route('form.register') }}"
class="font-small">Don't have an account? Register instead.</a>
                @include("partials.messages")
            </form>
        </div>
    </div>
@endsection

@section("scripts")
    @parent
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script src="{{ asset('assets/js/login.min.js') }}"></script>
@endsection

```

```

@extends("layouts.client")

@section("title")
    Vlogo | Register
@endsection

@section("content")
    <div class="section-padding vh-100 d-flex align-items-center justify-content-center">
        <div class="container">
            <h2 class="text-center font-xl">Register</h2>
            <form id="registerForm" class="font-small mx-auto" action="{{ route("register") }}" method="POST">
                @csrf
                <div class="mb-3">
                    <label for="username" class="form-label">Username</label>
                    <input type="text" class="form-control font-small" id="username"
name="username" placeholder="Username" maxlength="25" value="{{ old("username") }}" />
                    <span class="font-small error-message">Your username cannot contain
special characters other than '_', must contain at least one alphabetical character and
be between 8 and 25 characters long</span>
                </div>
                <div class="mb-3">
                    <label for="email" class="form-label">Email address</label>
                    <input type="email" class="form-control font-small" id="email"
name="email" placeholder="name@example.com" value="{{ old("email") }}" />
                    <span class="font-small error-message">name@example.com</span>
                </div>
                <div class="mb-3">
                    <label for="password" class="form-label">Password</label>
                    <input type="password" class="form-control font-small"
id="password" name="password" placeholder="Password" />
                    <span class="font-small error-message">Your password needs to be at
least 8 characters long and contain at least one letter and one number</span>
                </div>
                <div class="mb-3">
                    <label for="confirmPassword" class="form-label">Confirm
password</label>
                    <input type="password" class="form-control font-small"
id="confirmPassword" name="confirmPassword" placeholder="Confirm password" />
                    <span class="font-small error-message">Your passwords don't
match</span>
                </div>
                <button id="btnRegister" class="btn btn-primary text-center d-block mx-
auto px-4">
                    Register
                </button>
                <br/>
                <a class="signInFormLink" href="{{ route("form.login") }}" class="font-
small">Already have an account? Log in instead.</a>
                @include("partials.messages")
            </form>
        </div>
    </div>
@endsection

@section("scripts")
    @parent
    <script src="{{ asset("assets/js/form-validation.min.js") }}"></script>
    <script src="{{ asset("assets/js/register.min.js") }}"></script>
@endsection

```

```

@extends("layouts.client")

@section("title")
    Vlogo | Restaurants
@endsection

@section("content")
    <div class="section-padding container">
        <div class="row">
            <div class="col-12 col-lg-3">
                <ul class="food-categories font-medium">
                    <li>
                        <a href="#" id="allCategories" class="active">All food</a>
                    </li>
                    <li>
                        <hr class="my-1"/>
                    </li>
                    @foreach($foodCategories as $fc)
                        <li>
                            <a href="#" class="food-category" data-id="{{ $fc->id }}"
class="py-2">{{ $fc->name }}</a>
                        </li>
                    @endforeach
                </ul>
            </div>
            <div class="col-12 col-lg-9">
                <form>
                    <div class="row">
                        <div class="col-12 col-lg-9 mt-3 mt-lg-0">
                            <input type="text" id="tbName" class="form-control font-
small" autocomplete="off" placeholder="Restaurant name" @if(request()->has('keyword'))
value="{{ request()->get('keyword') }}" @endif/>
                        </div>
                        <div class="col-12 col-lg-3 mt-3 mt-lg-0">
                            <select id="ddlSort" class="form-select font-small">
                                <option value="nameAsc">Name ascending</option>
                                <option value="nameDesc">Name descending</option>
                            </select>
                        </div>
                    </div>
                </form>
                <div class="mt-4 position-relative">
                    <div id="restaurantsWrapper" class="row pt-4">
                    </div>
                    <div id="restaurantsLoader" class="loader">
                        <div class="loader-image">
                            
                        </div>
                    </div>
                </div>
                <div id="restaurantsPagination" class="mt-4 d-flex justify-content-
center">
                </div>
            </div>
        </div>
    </div>
@endsection

@section("scripts")
    @parent
    <script src="{{ asset('assets/js/restaurants.min.js') }}"></script>
@endsection

```



```

@extends("layouts.client")

@section("title")
    Vlogo | {{ $restaurant->name }}
@endsection

@section("content")
    <div class="section-padding container">
        <nav aria-label="breadcrumb">
            <ol class="breadcrumb font-small">
                <li class="breadcrumb-item"><a class="color-primary" href="{{
route("restaurants.index") }}">Restaurants</a></li>
                <li class="breadcrumb-item active" aria-current="page">{{ $restaurant-
>name }}</li>
            </ol>
        </nav>
        <div class="row">
            <div class="col-12 col-xl-3 order-2 order-xl-1">
                <ul class="food-categories font-medium">
                    <li>
                        <a href="#" id="allCategories" class="active">All food</a>
                    </li>
                    <li>
                        <hr class="my-1"/>
                    </li>
                    @foreach($restaurant->categories as $fc)
                        <li>
                            <a href="#" class="food-category" data-id="{{ $fc->id }}"
class="py-2">{{ $fc->name }}</a>
                        </li>
                    @endforeach
                </ul>
            </div>
            <div class="col-12 col-xl-6 mt-4 mt-xl-0 order-3 order-xl-2">
                <form>
                    <input type="hidden" id="idRestaurant" value="{{ $restaurant->id
}}"/>
                    <div class="row">
                        <div class="col-12 col-lg-9 mt-3 mt-lg-0">
                            <input type="text" id="tbName" class="form-control font-
small" autocomplete="off" placeholder="Search by name"/>
                        </div>
                        <div class="col-12 col-lg-3 mt-3 mt-lg-0">
                            <select id="ddlSort" class="form-select font-small">
                                <option value="nameAsc">Name ascending</option>
                                <option value="nameDesc">Name descending</option>
                                <option value="priceAsc">Price low > high</option>
                                <option value="priceDesc">Price high > low</option>
                            </select>
                        </div>
                    </div>
                </form>
                <div class="mt-4 position-relative">
                    <div id="foodWrapper" class="row pt-4">
                    </div>
                    <div id="foodLoader" class="loader">
                        <div class="loader-image">
                            
                        </div>
                    </div>
                </div>
                <div id="foodPagination" class="mt-4 d-flex justify-content-center">

```

```

        </div>
    </div>
    <div class="col-12 col-xl-3 mt-4 mt-xl-0 order-1 order-xl-3 mb-4 mb-xl-0
border-start-0">
        <form action="{{ route('orders.store') }}" method="POST" id="orderForm"
class="p-3 border rounded">
            @csrf
            <input type="hidden" name="restaurantId" value="{{ $restaurant->id
}}"/>
            <h3 class="font-medium text-center">Your order</h3>
            <div class="mb-3">
                <label for="address" class="form-label">Delivery
address</label>
                <input type="text" class="form-control font-small" id="address"
name="address" placeholder="Delivery address"/>
                <span class="error-message font-small">(Street Name) (Home
Number) / [Apartment Number]</span>
            </div>
            <div class="mb-3">
                <div id="orderItems">
                    <ul>
                    </ul>
                </div>
            </div>
            <div class="text-center">
                <input type="submit" value="Place order" id="btnOrder"
class="btn btn-primary"/>
            </div>
            @include('partials.messages')
        </form>
    </div>
</div>
</div>
@endsection

@section("scripts")
    @parent
    <script src="{{ asset('assets/js/form-validation.min.js') }}"></script>
    <script src="{{ asset('assets/js/food.min.js') }}"></script>
@endsection

```

3.11 JavaScript

client.js

```
$(document).ready(() => {  
  // sidenav  
  $("#sideNav").hide();  
  $("#sidenavCover").hide();  
  
  $("#hamburgerMenu a, #sidenavCover").click(function(e) {  
    e.preventDefault();  
  
    $("#sideNav").slideToggle("fast");  
    $("#sidenavCover").fadeToggle("fast");  
  });  
  
  // user div  
  $('#userHoverMenu').hide();  
  
  $(window).click(() => {  
    $('#userHoverMenu').slideUp("fast");  
  });  
  
  $('.user-div').click((e) => {  
    e.stopPropagation();  
    $('#userHoverMenu').slideToggle("fast");  
  });  
});
```

common.js

```
$(document).ready(() => {
  // scrollTop
  $("body").append(`<div id="scrollTop">
    <a href="#" class="d-flex align-items-center justify-content-center rounded-circle shadow">
      <i class="fas fa-arrow-up font-small"></i>
    </a>
  </div>`);

  $(window).scroll(displayScrollTopBtn);

  $("#scrollTop").hide();
  displayScrollTopBtn();

  function displayScrollTopBtn() {
    if($(window).scrollTop() > $("#header .container").outerHeight()) {
      $("#scrollTop").fadeIn("fast");
    } else {
      $("#scrollTop").fadeOut("fast");
    }
  }

  $("#scrollTop a").click(function(e) {
    e.preventDefault();

    $("html, body").animate({
      scrollTop: $("html").offset().top
    }, 500, "easeInOutExpo");
  });

  // csrf
  const token = $('meta[name="csrf-token"]').attr('content');

  $.ajaxSetup({
    headers: {
      'X-CSRF-TOKEN': token
    }
  });
});

const base_url = window.location.origin;

async function ajaxCallback(fileName, method = 'GET', data = {}) {
  let returnValue = null;

  await $.ajax({
    url: fileName,
    method: method,
    dataType: 'json',
    data: data,
    success: data => {
      returnValue = data;
    },
    error: err => {
      console.error(err.responseText);
    }
  });

  return returnValue;
}
```

contact.js

```
$(document).ready(() => {  
  const emailPattern = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/;  
  
  function checkEmail() {  
    validateField($('#email'), emailPattern);  
  }  
  
  function checkSubject() {  
    if($('#subject').val().trim() !== '')  
      fieldCorrect($('#subject'));  
    else  
      fieldIncorrect($('#subject'));  
  }  
  
  function checkMessage() {  
    if($('#message').val().trim() !== '')  
      fieldCorrect($('#message'));  
    else  
      fieldIncorrect($('#message'));  
  }  
  
  $('#email').blur(checkEmail);  
  $('#subject').blur(checkSubject);  
  $('#message').blur(checkMessage);  
  
  $('#btnSendMessage').click(function() {  
    const result = submitForm([checkEmail, checkSubject, checkMessage],  
    'contactForm');  
  
    if(result)  
      $(this).addClass('disabled');  
  
    return result;  
  });  
});
```

```

$(document).ready(() => {
  getFood();
  getCart();

  $('#allCategories').click(function(e) {
    e.preventDefault();

    $('.food-category').removeClass("active");
    $(this).addClass("active");

    getFood();
  });

  $('.food-category').click(function(e) {
    e.preventDefault();

    $('#allCategories').removeClass("active");
    if($(this).hasClass('active')) {
      $(this).removeClass('active')

      if(!$('.food-category.active').length) {
        $('#allCategories').addClass("active");
      }
    } else {
      $(this).addClass("active");
    }

    getFood();
  });

  $('#tbName').on('input', () => {
    getFood();
  });

  $('#ddlSort').on('change', () => {
    getFood();
  });

  async function getFood(page = 1) {
    const restaurantId = $('#idRestaurant').val();
    const keyword = $('#tbName').val().trim();
    const sortOrder = $('#ddlSort').val();
    let categories = [];

    for(let el of $('.food-category.active'))
      categories.push($(el).data('id'));

    $('.loader').fadeIn('fast');
    const response = await ajaxCallback(base_url + "/restaurants/food", "POST", {
      id: restaurantId,
      keyword: keyword,
      sortOrder: sortOrder,
      categories: categories,
      page: page
    });
    $('.loader').fadeOut('fast');

    printFood(response);
  }

  function printFood(data) {
    if(data.views.length) {
      let html = "";

```

```

        for(let r of data.views)
            html += r;

        $('#foodWrapper').html(html);
    } else {
        $('#foodWrapper').html("<span class='no-results col-12 font-medium'>No food
matches your search criteria.</span>");
    }
    updatePagination(data);
    enableAddingToOrder();
}

function updatePagination(data) {
    const currentPage = data.food.current_page;
    const lastPage = data.food.last_page;

    const disabledBack = currentPage === 1 ? 'disabled' : '';
    const disabledNext = currentPage === lastPage ? 'disabled' : '';

    let html = `<ul class='pagination font-small'><li class='page-item
${disabledBack}'><a href='#' class='page-link pagination-back'><i class='fas fa-caret-
left'></i></a></li>`;

    for(let i = 0; i < lastPage; i++) {
        const activeClass = i + 1 === currentPage ? 'active' : '';
        html += `<li class="page-item ${activeClass}"><a href='#' class="page-link"
data-page='${i + 1}'>${i + 1}</a></li>`;
    }

    html += `<li class='page-item ${disabledNext}'><a href='#' class='page-link
pagination-next'><i class='fas fa-caret-right'></i></a></li></ul>`;

    $('#foodPagination').html(html);
    enablePagination();
}

function enablePagination() {
    $('.pagination a').click(function(e) {
        e.preventDefault();
    });

    $('.page-link').click(function() {
        const currentPage = $('.page-item.active .page-link').data('page');
        const page = $(this).data('page');

        if(page === currentPage)
            return;

        getFood(page);
    });

    $('.pagination-back').click(function() {
        const currentPage = $('.page-item.active .page-link').data('page');

        if(currentPage <= 1)
            return;

        getFood(currentPage - 1);
    });

    $('.pagination-next').click(function() {
        const currentPage = $('.page-item.active .page-link').data('page');
        const numberOfPages = $('.page-link').length;

```

```

        if(currentPage >= numberOfPages)
            return;

        getFood(currentPage + 1);
    });
}

function enableAddingToOrder() {
    // Adding to order
    $('.add-food').on('click', async function(e) {
        e.preventDefault();
        const id = $(this).data('id');

        const result = await ajaxCallback(base_url + '/cart/add', 'POST', {
            id: id
        });

        if(result)
            getCart();
    });
}

async function getCart() {
    const urlParts = window.location.href.split('/');
    const restaurantId = parseInt(urlParts[urlParts.length - 1]);

    const result = await ajaxCallback(base_url + "/cart/items", "POST", {
        restaurantId: restaurantId
    });

    printCart(result);
}

function printCart(data) {
    let html = '';
    let totalPrice = 0;

    if(data.length) {
        for(let el of data) {
            totalPrice += parseFloat(el.price) * parseInt(el.amount);
            html += `<li class="order-item font-small row d-flex align-items-center justify-content-between">
                <span class="col-8">${el.name} - <span class="fw-
bold">${el.price}</span></span>
                <span class="col-4 d-flex justify-content-end font-small">
                    <a href="#" data-id="${el.cart_id}" data-action="decrease"
class="btn-amount btn-decrease color-primary"><i class="fas fa-minus"></i></a>
                    <span class="mx-1">${el.amount}</span>
                    <a href="#" data-id="${el.cart_id}" data-action="increase"
class="btn-amount btn-increase color-primary"><i class="fas fa-plus"></i></a>
                    <a href="#" data-id="${el.cart_id}" data-action="remove"
class="ms-3 btn-amount btn-remove"><i class="fa fa-times" aria-hidden="true"></i></a>
                </span>
                </li>`;
        }

        html += `<li class="font-small d-flex align-items-center justify-content-
between mt-2"><span>Total:</span><span class="fw-
bold">${totalPrice.toFixed(2)}</span></li>`;
    } else {
        html = `<li class="font-small text-center">Your order is empty</li>`;
    }

    $('#orderItems ul').html(html);
}

```



```

        enableChangingCart();
    }

    function enableChangingCart() {
        $('#btn-amount').click(async function(e) {
            e.preventDefault();

            const id = $(this).data('id');
            const action = $(this).data('action');

            const result = await ajaxCallback(base_url + '/cart/change', 'POST', {
                id: id,
                action: action
            })

            if(result)
                getCart();
        });
    }

    // form submission
    const addressPattern = /^[A-ZŠĐČĆŽ][a-zšđčćž]{2,}(\s[A-ZŠĐČĆŽa-zšđčćž][a-zšđčćž]{2,})*\s\d+[A-Z]?(\s\d+)*$/;

    function checkAddress() {
        validateField($('#address'), addressPattern);
    }

    $('#address').blur(checkAddress);

    $('#btnOrder').click(function() {
        const result = submitForm([checkAddress], 'orderForm');

        if(result)
            $(this).addClass('disabled');

        return result;
    });
});

```

form-validation.js

```
$(document).ready(function() {
    $('.error-message').hide();
});

let noErrors;

function validateField(field, pattern) {
    if(field.val().match(pattern)) fieldCorrect(field);
    else fieldIncorrect(field);
}

function fieldCorrect(field) {
    $(field).addClass('border-correct');
    $(field).removeClass('border-incorrect');
    $(field).siblings('.error-message').slideUp();
}

function fieldIncorrect(field) {
    if($(field).attr('type') !== 'file') {
        $(field).addClass('border-incorrect');
        $(field).removeClass('border-correct');
    }
    $(field).siblings('.error-message').slideDown();
    noErrors = false;
}

function fieldNeutral(field) {
    $(field).removeClass('border-incorrect');
    $(field).removeClass('border-correct');
    $(field).siblings('.error-message').slideUp();
}

function submitForm(checkFunctions, formId = null, clear = false) {
    noErrors = true;
    for(let f of checkFunctions) {
        f();
    }
    if(noErrors) {
        if(formId && clear) clearForm(formId);
        return true;
    }
    return false;
}

function clearForm(formId) {
    document.getElementById(formId).reset();
    $('.error-message').hide();
    $('#${formId} .border-correct').removeClass('border-correct');
    $('#${formId} .border-incorrect').removeClass('border-incorrect');
}
```

home.js

```
$(document).ready(() => {
  $(window).scroll(setHeaderShadow);

  setHeaderShadow();

  function setHeaderShadow() {
    if($(window).scrollTop() > $("#header .container").outerHeight()) {
      $("#header").addClass("shadow");
    } else {
      $("#header").removeClass("shadow");
    }
  }

  // search
  $(window).click(function() {
    searchResultsList.slideUp('fast');
    //searchResultsList.html('');
  });

  $('#homepageForm').click(function(e) {
    e.stopPropagation();
  });

  const searchField = $('#tbSearch');

  searchField.focus(function() {
    getResults();
  });

  searchField.on('input', function() {
    getResults();
  });

  async function getResults() {
    const keyword = $('#tbSearch').val();

    if(keyword === '') {
      searchResultsList.html('<li class="p-3">Start typing...</li>');
      searchResultsList.slideDown('fast');
      return;
    }

    const restaurants = await ajaxCallback(base_url + "/restaurants/search",
'POST', {
    keyword: keyword
  });

  printResults(restaurants, keyword);
}

const searchResultsList = $('#searchResults ul');
searchResultsList.hide();

function printResults(results, keyword) {
  let html = '';

  if(results.data.length) {
    for(let r of results.data) {
      html += `<li><a class="d-block p-2 search-result" href="${base_url +
"/restaurants/" + r.id}">${r.name}</a></li>`;
    }

    if(results.total - results.to)
```

```
        html += `<li><a class="d-block p-2" href="${base_url +  
"/restaurants?keyword=" + keyword}">And ${results.total - results.to}  
More...</a></li>`;  
    } else {  
        html = '<li class="p-3">No matches</li>';  
    }  
  
    searchResultsList.html(html);  
    searchResultsList.slideDown('fast');  
}  
});
```

login.js

```
$(document).ready(() => {
    $('#email').focus();

    const emailPattern = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/;
    const passwordPattern = /^(?=.*\\d)(?=.*[a-z]).{8,}$/;

    function checkEmail() {
        validateField($('#email'), emailPattern);
    }

    function checkPassword() {
        validateField($('#password'), passwordPattern);
    }

    $('#email').blur(checkEmail);
    $('#password').blur(checkPassword);

    $('#btnLogin').click(function() {
        const result = submitForm([checkEmail, checkPassword], 'loginForm');

        if(result)
            $(this).addClass('disabled');

        return result;
    });
});
```

register.js

```
$(document).ready(() => {
    $('#username').focus();

    const usernamePattern = /^(?=.*[a-z])[a-zA-Z0-9\_]{8,25}$/;
    const emailPattern = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/;
    const passwordPattern = /^(?=.*\\d)(?=.*[a-z]).{8,}$/;

    function checkUsername() {
        validateField($('#username'), usernamePattern);
    }

    function checkEmail() {
        validateField($('#email'), emailPattern);
    }

    function checkPassword() {
        validateField($('#password'), passwordPattern);
        checkConfirmPassword();
    }

    function checkConfirmPassword() {
        if($('#password').val() === '')
            return;

        if($('#confirmPassword').val() === $('#password').val() &&
!$('#password').hasClass('border-incorrect'))
            fieldCorrect($('#confirmPassword'));
        else
            fieldIncorrect($('#confirmPassword'));
    }

    $('#username').blur(checkUsername);
    $('#email').blur(checkEmail);
    $('#password').blur(checkPassword);
    $('#confirmPassword').blur(checkConfirmPassword);

    $('#btnRegister').click(function() {
        const result = submitForm([checkUsername, checkEmail, checkPassword,
checkConfirmPassword], 'registerForm');

        if(result)
            $(this).addClass('disabled');

        return result;
    });
});
```

restaurants.js

```
$(document).ready(() => {
  getRestaurants();

  $('#allCategories').click(function(e) {
    e.preventDefault();

    $('.food-category').removeClass("active");
    $(this).addClass("active");

    getRestaurants();
  });

  $('.food-category').click(function(e) {
    e.preventDefault();

    $('#allCategories').removeClass("active");
    if($(this).hasClass('active')) {
      $(this).removeClass('active')

      if(!$('.food-category.active').length) {
        $('#allCategories').addClass("active");
      }
    } else {
      $(this).addClass("active");
    }

    getRestaurants();
  });

  $('#tbName').on('input', () => {
    getRestaurants();
  });

  $('#ddlSort').on('change', () => {
    getRestaurants();
  });

  async function getRestaurants(page = 1) {
    const keyword = $('#tbName').val().trim();
    const sortOrder = $('#ddlSort').val();
    let categories = [];

    for(let el of $('.food-category.active'))
      categories.push($(el).data('id'));

    $('.restaurantDiv').removeClass('shadow');

    $('.loader').fadeIn('fast');
    const response = await ajaxCallback(base_url + "/restaurants/filter", "POST",
    {
      keyword: keyword,
      sortOrder: sortOrder,
      categories: categories,
      page: page
    });
    $('.loader').fadeOut('fast');

    printRestaurants(response);
  }

  function printRestaurants(data) {
    if(data.views.length) {
      let html = "";

```

```

        for(let r of data.views)
            html += r;

        $('#restaurantsWrapper').html(html);
    } else {
        $('#restaurantsWrapper').html("<span class='no-results col-12 font-medium'>No restaurants match your search criteria.</span>");
    }
    updatePagination(data);
}

function updatePagination(data) {
    const currentPage = data.restaurants.current_page;
    const lastPage = data.restaurants.last_page;

    const disabledBack = currentPage === 1 ? 'disabled' : '';
    const disabledNext = currentPage === lastPage ? 'disabled' : '';

    let html = `<ul class='pagination font-small'><li class='page-item ${disabledBack}'><a href='#' class='page-link pagination-back'><i class='fas fa-caret-left'></i></a></li>`;

    for(let i = 0; i < lastPage; i++) {
        const activeClass = i + 1 === currentPage ? 'active' : '';
        html += `<li class="page-item ${activeClass}"><a href="#" class="page-link" data-page='${i + 1}'>${i + 1}</a></li>`;
    }

    html += `<li class='page-item ${disabledNext}'><a href='#' class='page-link pagination-next'><i class='fas fa-caret-right'></i></a></li></ul>`;

    $('#restaurantsPagination').html(html);
    enablePagination();
}

function enablePagination() {
    $('.pagination a').click(function(e) {
        e.preventDefault();
    });

    $('.page-link').click(function() {
        const currentPage = $('.page-item.active .page-link').data('page');
        const page = $(this).data('page');

        if(page === currentPage)
            return;

        getRestaurants(page);
    });

    $('.pagination-back').click(function() {
        const currentPage = $('.page-item.active .page-link').data('page');

        if(currentPage <= 1)
            return;

        getRestaurants(currentPage - 1);
    });

    $('.pagination-next').click(function() {
        const currentPage = $('.page-item.active .page-link').data('page');
        const numberOfPages = $('.page-link').length;

        if(currentPage >= numberOfPages)

```



```
        return;  
        getRestaurants(currentPage + 1);  
    });  
}  
});
```

admin/categories.js

```
$(document).ready(() => {  
  const namePattern = /^[A-Z][a-z]{2,}(\s[a-zA-Z][a-z]{2,})*$;/;  
  
  function checkName() {  
    validateField($('#name'), namePattern);  
  }  
  
  $('#name').blur(checkName);  
  
  $('#btnSubmit').click(function() {  
    const result = submitForm([checkName], 'categoryForm');  
  
    if(result)  
      $(this).addClass('disabled');  
  
    return result;  
  });  
});
```

admin/food.js

```
$(document).ready(() => {
  const pricePattern = /^(?=.+) (?:[1-9]\d*)?(?: (\.\d+) | (0\.\d*[1-9]+\d*))?$ /;

  const nameField = $('#name');
  function checkName() {
    if(nameField.val() === '') {
      fieldIncorrect(nameField)
    } else {
      fieldCorrect(nameField);
    }
  }

  const imageField = $('#image');
  function checkImage() {
    if(!edit) {
      if(imageField.val() === '') {
        fieldIncorrect(imageField);
      } else {
        fieldCorrect(imageField);
      }
    }
  }

  const priceField = $('#price');
  function checkPrice() {
    validateField(priceField, pricePattern);
  }

  nameField.blur(checkName);
  imageField.on('input', checkImage);
  priceField.blur(checkPrice);

  $('#btnSubmit').click(function() {
    const result = submitForm([checkName, checkImage, checkPrice], 'foodForm');

    if(result)
      $(this).addClass('disabled');

    return result;
  });
});
```

admin/restaurants.js

```
$(document).ready(() => {  
  const nameField = $('#name');  
  function checkName() {  
    if(nameField.val() === '') {  
      fieldIncorrect(nameField)  
    } else {  
      fieldCorrect(nameField);  
    }  
  }  
  
  const imageField = $('#image');  
  function checkImage() {  
    if(!edit) {  
      if(imageField.val() === '') {  
        fieldIncorrect(imageField);  
      } else {  
        fieldCorrect(imageField);  
      }  
    }  
  }  
  
  nameField.blur(checkName);  
  imageField.on('input', checkImage);  
  
  $('#btnSubmit').click(function() {  
    const result = submitForm([checkName, checkImage], 'restaurantForm');  
  
    if(result)  
      $(this).addClass('disabled');  
  
    return result;  
  });  
});
```

admin/users.js

```
$(document).ready(() => {
  const usernamePattern = /^(?=.*[a-z])[a-zA-Z0-9\_]{8,25}$/;
  const emailPattern = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/;
  let passwordPattern;

  if(edit)
    passwordPattern = /^$|^((?=.*\\d)(?=.*[a-z]).{8,})$/;
  else
    passwordPattern = /^(?=.*\\d)(?=.*[a-z]).{8,}$/;

  function checkUsername() {
    validateField($('#username'), usernamePattern);
  }

  function checkEmail() {
    validateField($('#email'), emailPattern);
  }

  function checkPassword() {
    validateField($('#password'), passwordPattern);
  }

  $('#username').blur(checkUsername);
  $('#email').blur(checkEmail);
  $('#password').blur(checkPassword);

  $('#btnSubmit').click(function() {
    const result = submitForm([checkUsername, checkEmail, checkPassword],
'userForm');

    if(result)
      $(this).addClass('disabled');

    return result;
  });
});
```

3.12 SCSS

admin.scss

```
$screen-xs: 575px;
$screen-sm: 767px;
$screen-md: 991px;
$screen-lg: 1199px;

.user-panel {
  a {
    cursor: default;
  }
}

td {
  vertical-align: middle !important;
}

.large-column {
  min-width: 170px;
}

.standard-form {
  width: 500px;
  @media screen and (max-width: $screen-sm) {
    width: 100%;
  }
}

.border-incorrect {
  border: 1px solid red;
}

.error-message {
  color: red;
}

// orders
#actionsForm {
  width: 600px;
  @media screen and (max-width: $screen-md) {
    width: 100%;
  }
}

// restaurants
.restaurant-image {
  width: 150px;
  margin-left: auto;
  margin-right: auto;
}

// food
.food-image {
  width: 70px;
  margin-left: auto;
  margin-right: auto;
}
```

app.scss

```
@font-face {
  src: url("../fonts/VarelaRound-Regular.ttf");
  font-family: "VarelaRound";
}

$color-primary: #ed2d2d;
$color-secondary: #ffc144; // #ed902d
$color-white: #fff;
$color-black: #141414;
$color-black-t: rgba(0, 0, 0, .7);
$color-black-t-lt: rgba(0, 0, 0, .4);
$color-error: red;

$screen-xs: 575px;
$screen-sm: 767px;
$screen-md: 991px;
$screen-lg: 1199px;

* {
  font-family: "VarelaRound";
  box-sizing: border-box;
}

body {
  color: $color-black;
}

ul {
  list-style: none;
  margin: 0;
  padding: 0;
}

a {
  text-decoration: none;
}

//text size
.font-xs {
  font-size: 12px;
  @media screen and (max-width: $screen-sm) {
    font-size: 10px;
  }
}

.font-small {
  font-size: 16px;
  @media screen and (max-width: $screen-sm) {
    font-size: 14px;
  }
}

.font-medium {
  font-size: 19px;
  @media screen and (max-width: $screen-sm) {
    font-size: 17px;
  }
}

.font-large {
  font-size: 30px;
  @media screen and (max-width: $screen-sm) {
    font-size: 25px;
  }
}
```

```

    }
}

.font-xl {
  font-size: 40px;
  @media screen and (max-width: $screen-sm) {
    font-size: 30px;
  }
}

.font-xxl {
  font-size: 60px;
  @media screen and (max-width: $screen-sm) {
    font-size: 40px;
  }
}
// /text size

.form-control, .form-select {
  &:focus {
    outline: none;
    box-shadow: 0 0 0 0 #000000;
    border-color: $color-secondary;
    //color: $color-black;
  }
}

.cover {
  width: 100%;
  height: 100%;
  background-color: $color-black-t;
}

.section-padding {
  padding-top: 150px;
  padding-bottom: 100px;

  @media screen and (max-width: $screen-md) {
    padding-top: 120px;
    padding-bottom: 70px;
  }
}

.btn {
  &:hover, &:active, &:focus, &:active:focus {
    box-shadow: 0 0 0 0 #000;
  }
}

.btn-primary, .btn-primary.disabled, .btn-primary.disabled {
  background-color: $color-primary;
  color: $color-white;
  transition: .1s opacity;
  border: none;
  &:hover, &:active, &:focus {
    opacity: .75;
    background-color: $color-primary;
  }
}

.btn-secondary, .btn-secondary.disabled, .btn-secondary.disabled {
  background-color: $color-secondary;
  color: $color-black;
  transition: .1s opacity;
  border: none;
}

```



```

    &:hover, &:active, &:focus {
      opacity: .75;
      background-color: $color-secondary;
    }
  }

  .color-primary {
    color: $color-primary !important;
  }

  .color-secondary {
    color: $color-secondary !important;
  }

  // header
  #header {
    background-color: $color-primary;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    transition: .2s box-shadow;
    z-index: 99;
  }

  #logo {
    width: 170px;
    @media screen and (max-width: $screen-md) {
      width: 150px;
    }
    @media screen and (max-width: $screen-sm) {
      width: 100px;
    }
  }

  #nav {
    & > ul {
      @media screen and (max-width: $screen-md) {
        display: none !important;
      }
      a {
        color: $color-white;
        transition: .1s color;
        &:hover {
          color: $color-secondary;
        }
      }
      .active {
        color: $color-secondary;
      }
    }
  }

  #btnSignIn {
    background-color: $color-secondary;
    color: $color-black;
    border: none;
  }

  #hamburgerMenu {
    display: none;
    @media screen and (max-width: $screen-md) {
      display: block !important;
    }
    a {

```

```

        color: $color-white;
    }
}

#sideNav {
    a {
        color: $color-white;
        transition: .1s background-color, .1s color;
        &:hover {
            background-color: $color-white;
            color: $color-primary;
        }
    }
    .active {
        background-color: $color-white;
        color: $color-primary;
    }
}

#sidenavCover {
    position: fixed;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    z-index: 98;
    backdrop-filter: blur(2px);
}

#userDivFront {
    color: $color-secondary;
    cursor: pointer;
}

#userHoverMenu {
    top: 100%;
    left: 0;
    min-width: 100%;
    background-color: $color-black-t;
    a, input[type=submit] {
        border: none;
        cursor: pointer;
        background: none;
        width: 100%;
        text-align: start;
        display: block;
        color: $color-white !important;
        transition: .1s color, .1s background-color;
        &:hover {
            background-color: $color-white;
            color: $color-black !important;
        }
    }
}

// home
#mainSlider {
    background-color: $color-primary;
    h1 {
        color: $color-white;
    }
}

#mainSliderWave {
    path {

```

```

        fill: $color-primary;
    }
}

#homepageImage {
    width: 350px;
}

#homepageForm {
    width: 500px;
    @media screen and (max-width: $screen-lg) {
        width: 100%;
    }
    @media screen and (max-width: $screen-md) {
        width: 500px;
    }
    @media screen and (max-width: $screen-sm) {
        width: 100%;
    }
    .input-group-text {
        color: $color-black;
        background-color: $color-secondary;
        border: none !important;
    }
    .form-control {
        border: none !important;
    }
}

#searchResults {
    background-color: $color-white;
    ul {
        li {
            color: $color-black;
            a {
                color: $color-black;
                &:hover {
                    color: $color-primary;
                }
            }
        }
    }
}

// restaurants
#restaurantsWrapper {
    min-height: 200px;
    padding-left: 0;
    padding-right: 0;
}

.restaurantDiv {
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
    min-height: 200px;
}

.restaurantText {
    h3, li {
        color: $color-white;
        background-color: $color-black-t;
    }
}

```

```

.food-categories {
  a {
    color: $color-black;
  }
  .active {
    color: $color-secondary;
    //font-weight: bold;
  }
}

.loader {
  position: absolute;
  top: 0;
  bottom: 0;
  left: 0;
  right: 0;
  display: flex;
  align-items: center;
  justify-content: center;
  background-color: $color-white;
}

.loader-image {
  width: 100px;
  height: 100px;
}

.pagination {
  li {
    &:hover, &:active, &:focus, &:active:focus {
      box-shadow: 0 0 0 0 #000;
    }
    &.active {
      color: $color-white;
      a {
        background-color: $color-primary !important;
        border-color: $color-primary !important;
      }
    }
    a {
      color: $color-primary;
      &:hover, &:active, &:focus, &:active:focus {
        box-shadow: 0 0 0 0 #000;
        color: $color-primary;
      }
      &:active:focus {
        color: $color-white !important;
      }
    }
  }
}

// food
#foodWrapper {
  min-height: 200px;
}

.food-image {
  width: 40%;
}

.food-text {
  width: 55%;
}

```

```

.btn-remove {
  color: $color-black-t;
  &:hover {
    color: $color-black-t;
  }
}

// author
#authorImage {
  width: 370px;
  @media screen and(max-width: $screen-xs){
    width: 100%;
  }
}

#authorSocials {
  a {
    width: 35px;
    height: 35px;
    color: $color-white;
    background-color: $color-black-t;
    transition: .1s opacity;
    &:hover {
      opacity: .75;
    }
  }
}

#portfolioLink {
  color: $color-primary;
}

// form validation
.border-correct {
  //
}

.border-incorrect {
  border: 1px solid $color-error;
}

.error-message {
  color: $color-error;
}

.server-message {
  width: 100%;
  // @media screen and(max-width: $screen-md) {
  //   width: 100%;
  // }
}

// contact
#contactForm, #loginForm, #registerForm {
  width: 500px;
  @media screen and(max-width: $screen-md) {
    width: 100%;
  }
}

// log in / register
.signInFormLink {
  color: $color-primary;
  &:hover {
    color: $color-primary;
  }
}

```

```
        filter: brightness(.5);
    }
}

// footer
#footer {
    background-color: $color-black;
    color: $color-white;
    ul {
        a {
            color: $color-white;
        }
    }
}

#scrollTop {
    position: fixed;
    right: 20px;
    bottom: 20px;
    a {
        width: 48px;
        height: 48px;
        color: $color-white;
        background-color: $color-black-t;
    }
}
```