

Fonctionnalités du projet « Z'Burger »

1. La possibilité pour le client de s'inscrire

```
18 Route::post( uri: 'register', [RegisteredUserController::class, 'store']);
```

Figure 1.1 : extrait du code dans le fichier « auth.php »

```
34 public function store(Request $request)
35 {
36     $request->validate([
37         'first_name' => ['required', 'string', 'max:255'],
38         'name' => ['required', 'string', 'max:255'],
39         'email' => ['required', 'string', 'email', 'max:255', 'unique:'.User::class],
40         'password' => ['required', 'confirmed', Rules\Password::defaults()],
41     ]);
42
43     $user = User::create([
44         'first_name' => $request->first_name,
45         'name' => $request->name,
46         'email' => $request->email,
47         'password' => Hash::make($request->password),
48     ]);
49
50     event(new Registered($user));
51
52     Auth::login($user);
53
54     return redirect( to: RouteServiceProvider::HOME);
55 }
56
```

Figure 1.2 : extrait du code dans le fichier « RegisteredUserController.php »

2. La possibilité pour le client de s'authentifier

```
23 Route::post( uri: 'login', [AuthenticatedSessionController::class, 'store']);
```

Figure 2.1 : extrait du code dans le fichier « auth.php »

```
29 public function store(LoginRequest $request)
30 {
31     $request->authenticate();
32
33     $request->session()->regenerate();
34
35     return redirect()->intended( default: RouteServiceProvider::HOME);
36 }
```

Figure 2.2 : extrait du code dans le fichier « AuthenticatedSessionController.php »

3. Affichage du menu avec un design attractive

```
70 Route::get( uri: '/menu', function () {  
71     return view( view: 'main.menu.menu');  
72 }) -> name( name: 'menu');
```

Figure 3.1 : extrait du code dans le fichier « web.php »

4. Affichage des photos du Fast Food qui permet d'attirer l'attention du client et lui augmenter la crédibilité du Fast Food « Z'Burger »

```
37 Route::get( uri: '/gallery', function () {  
38     return view( view: 'main.gallery');  
39 }) -> name( name: 'gallery');
```

Figure 4.1 : extrait du code dans le fichier « web.php »

5. Affichage des tous les produits de Fast Food en une seule page pour faciliter l'accès au client

```
33 Route::get( uri: '/shop', [\App\Http\Controllers\ShopController::class, 'showShopPage'])  
34 ->name( name: 'shop');
```

Figure 5.1 : extrait du code dans le fichier « web.php »

```
8     public function showShopPage() {  
9         return view( view: 'main.shop', ['produitList' => Produit::all()]);  
10     }  
11 }
```

Figure 5.1 : extrait du code dans le fichier « ShopController.php »

6. Affichage des détails d'un produit sélectionné y compris les ingrédients de chaque produit

```
36 Route::get( uri: '/produits/{id}', action: 'App\Http\Controllers\ProduitController@show')
37     ->name( name: 'produits');
38
```

Figure 6.1 : extrait du code dans le fichier « web.php »

```
10 public function show($id) {
11     $produit = Produit::find($id);
12
13     $relatedProducts = Produit::where( column: 'category_id', $produit->category_id)
14         ->whereNot( column: 'id', $produit->id)
15         ->take( value: 4)
16         ->get();
17     return view( view: 'main.produit')
18         ->with('produit', $produit)
19         ->with('relatedProducts', $relatedProducts);
20 }
21 }
```

Figure 6.1 : extrait du code dans le fichier « ProduitController.php »

7. Affichage des produits relatives à celui sélectionné qui sont sous la même catégorie que ce dernier

```
10 public function show($id) {
11     $produit = Produit::find($id);
12
13     $relatedProducts = Produit::where( column: 'category_id', $produit->category_id)
14         ->whereNot( column: 'id', $produit->id)
15         ->take( value: 4)
16         ->get();
17     return view( view: 'main.produit')
18         ->with('produit', $produit)
19         ->with('relatedProducts', $relatedProducts);
20 }
21 }
```

Figure 7.1 : extrait du code dans le fichier « ProduitController.php »

8. Ajout des produits aux panier personnel du client

```
62 Route::post( uri: '/panier/produit', [\App\Http\Controllers\PanierController::class, 'addProduct'])
63     ->name( name: 'panier.add-product');
```

Figure 8.1 : extrait du code dans le fichier « web.php »

```

36     public function addProduct(Request $request) {
37         $user = Auth::user();
38         $exists = false;
39
40         foreach ($user->panier->produits as $produit) {
41             if ($request['id'] == $produit->id){
42                 $exists = true;
43                 $user->panier->produits()->updateExistingPivot($produit->id,
44                     ['quantity' => $produit->pivot->quantity + $request['quantity']]);
45                 break;
46             }
47         }
48         if (!$exists) {
49             $produit = Produit::find($request['id']);
50             $user->panier->produits()->attach($produit, ['quantity'=> $request['quantity']]);
51         }
52         return redirect()->action( action: 'App\Http\Controllers\ProduitController@show',
53             ['id' => $request['id']]);
54     }
55 }
56

```

Figure 8.2 : extrait du code dans le fichier « PanierController.php »

9. Possibilité de visualiser son panier à n'importe quel moment n'importe où il se situe dans le site web

```

43     Route::get( uri: '/panier', [\App\Http\Controllers\PanierController::class, 'showPanierPage'])
44         ->middleware(['auth', 'verified'])
45         ->name( name: 'panier');

```

Figure 9.1 : extrait du code dans le fichier « web.php »

```

13      public function showPanierPage() {
14          $user = Auth::user();
15          $total = 0;
16
17          foreach ($user->panier->produits as $produit) {
18              $total += $produit->prix * $produit->pivot->quantity;
19          }
20
21          return view( view: 'main.panier')
22              ->with('panier-produits', $user->panier->produits)
23              ->with('total', $total)
24              ->with('total_livraison', ($total + 4));
25
26      }

```

Figure 9.2 : extrait du code dans le fichier « PanierController.php »

10. Possibilité de supprimer n'importe quel produit dans son panier

```

47      Route::delete( uri: '/panier/{id}', [\App\Http\Controllers\PanierController::class, 'destroy'])
48          ->name( name: 'panier.destroy');

```

Figure 10.1 : extrait du code dans le fichier « web.php »

```

28      public function destroy( $id)
29      {
30          $user = Auth::user();
31          $user->panier->produits()->detach($id);
32          return Redirect::back();

```

Figure 10.2 : extrait du code dans le fichier « PanierController.php »

11. Valider son panier et passer la commande

```

60      Route::post( uri: 'payement', [\App\Http\Controllers\CheckoutController::class, 'pay'])
61          ->name( name: 'checkout.pay');

```

Figure 10.1 : extrait du code dans le fichier « web.php »

```

31 public function pay(Request $request) {
32
33     $user = Auth::user();
34     $user->first_name = $request['first_name'];
35     $user->name = $request['name'];
36     $user->email = $request['email'];
37     $user->coordonee()->update(
38         ['adresse' => $request['adresse']]);
39     $user->coordonee()->update(
40         ['ville' => $request['ville']]);
41     $user->coordonee()->update(
42         ['code_zip' => $request['code_zip']]);
43     $user->coordonee()->update(
44         ['num_tel' => $request['num_tel']]);
45
46     $user->save();
47

```

Figure 11.1 : extrait du code dans le fichier « CheckoutController.php » (1/3)

```

48     $total = 0;
49
50     foreach ($user->panier->produits as $produit) {
51         $total += $produit->prix * $produit->pivot->quantity;
52     }
53
54     $commande = Commande::create([
55         'reference' => \Str::uuid()->toString(),
56         'user_id' => $user->id,
57         'date' => Carbon::now(),
58         'etat' => 'En attente de livraison',
59         'prix' => $total + 4,
60         'extraInfo' => $request['extraInfo']
61     ]);
62

```

Figure 11.2 : extrait du code dans le fichier « CheckoutController.php » (2/3)

```

64         foreach ($user->panier->produits as $produit) {
65
66             $commande->produits()->attach($produit, [
67                 'quantity' => $produit->pivot->quantity,
68                 'prixTotal' => $produit->prix * $produit->pivot->quantity,
69             ]);
70         }
71
72         $user->panier->produits()->detach();
73
74         return Redirect::to( path: '/commandes' );
75     }

```

Figure 11.3 : extrait du code dans le fichier « CheckoutController.php » (3/3)

12. La mise à jour de ses données personnelles au moment du paiement

```

31     public function pay(Request $request) {
32
33         $user = Auth::user();
34         $user->first_name = $request['first_name'];
35         $user->name = $request['name'];
36         $user->email = $request['email'];
37         $user->coordonee()->update(
38             ['adresse' => $request['adresse']]);
39         $user->coordonee()->update(
40             ['ville' => $request['ville']]);
41         $user->coordonee()->update(
42             ['code_zip' => $request['code_zip']]);
43         $user->coordonee()->update(
44             ['num_tel' => $request['num_tel']]);
45
46         $user->save();

```

Figure 12.1 : extrait du code dans le fichier « CheckoutController.php »

13. La possibilité de voir l'historique de toutes ses commandes

```

66     Route::get( uri: '/commandes', [\App\Http\Controllers\ClientController::class, 'showCommandes'])
67         ->middleware(['auth', 'verified'])
68         ->name( name: 'commandes-show');

```

Figure 13.1 : extrait du code dans le fichier « web.php »

```
11      public function showCommandes() {
12          $user = Auth::user();
13
14          return view( view: 'main.commandes')
15              ->with('commandes', $user->commandes);
16      }
```

Figure 13.2 : extrait du code dans le fichier « ClientController.php »

14. La possibilité de voir les détails d'une commande sélectionnée

```
70      Route::get( uri: '/commandes/{id}', [\App\Http\Controllers\ClientController::class, 'showCommande'])
71          ->middleware(['auth', 'verified'])
72          ->name( name: 'commande-show');
```

Figure 14.1 : extrait du code dans le fichier « web.php »

```
18      public function showCommande($id) {
19          $user = Auth::user();
20
21          return view( view: 'main.commande-detail')
22              ->with('commande_prouduits', $user->commandes()->where('id', $id)->first()->produits)
23              ->with('commande', $user->commandes()->where('id', $id)->first());
24      }
```

Figure 14.2 : extrait du code dans le fichier « ClientController.php »

15. La possibilité de se déconnecter à tout moment

```
51      Route::group(['middleware' => ['auth']], function() {
52          /**
53           * Logout Route
54           */
55          Route::get( uri: '/logout', action: 'App\Http\Controllers\LogoutController@perform')
56              ->name( name: 'logout.perform');
57      });
```

Figure 15.1 : extrait du code dans le fichier « web.php »


```

16      public function perform()
17      {
18          Session::flush();
19
20          Auth::logout();
21
22          return redirect(to: 'login');
23      }

```

Figure 15.2 : extrait du code dans le fichier « LogoutController.php »

16. La mise en place des contraintes de sécurité sous formes des middlewares afin d'assurer le bon flow de l'application

```

55      protected $routeMiddleware = [
56          'auth' => \App\Http\Middleware\Authenticate::class,
57          'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
58          'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
59          'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
60          'can' => \Illuminate\Auth\Middleware\Authorize::class,
61          'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
62          'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
63          'signed' => \App\Http\Middleware\ValidateSignature::class,
64          'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
65          'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
66      ];

```

Figure 16.1 : extrait du code dans le fichier « Kernel.php »