



DeliverEats Project

Team 4

Phil Arceno

Dmitri Bانشchikov

Bahareh Jabarzadnouri

Background and objective



provide customers a way to place an order at a restaurant over the internet

Help owners obtain the transparency of the delivery management process



Discover New Restaurants by customers & New Customers by restaurants

Admin dashboard features

Create new user

Update Users

Delete Users

Read list of orders

Delete food items

Customer dashboard features

User can register

Users can browse among different categories/ restaurants

Users can add food to their cart/remove from their card/update their card

Users can transfer their card items to the orders

Users can make payments

Business Owner Dashboard Features

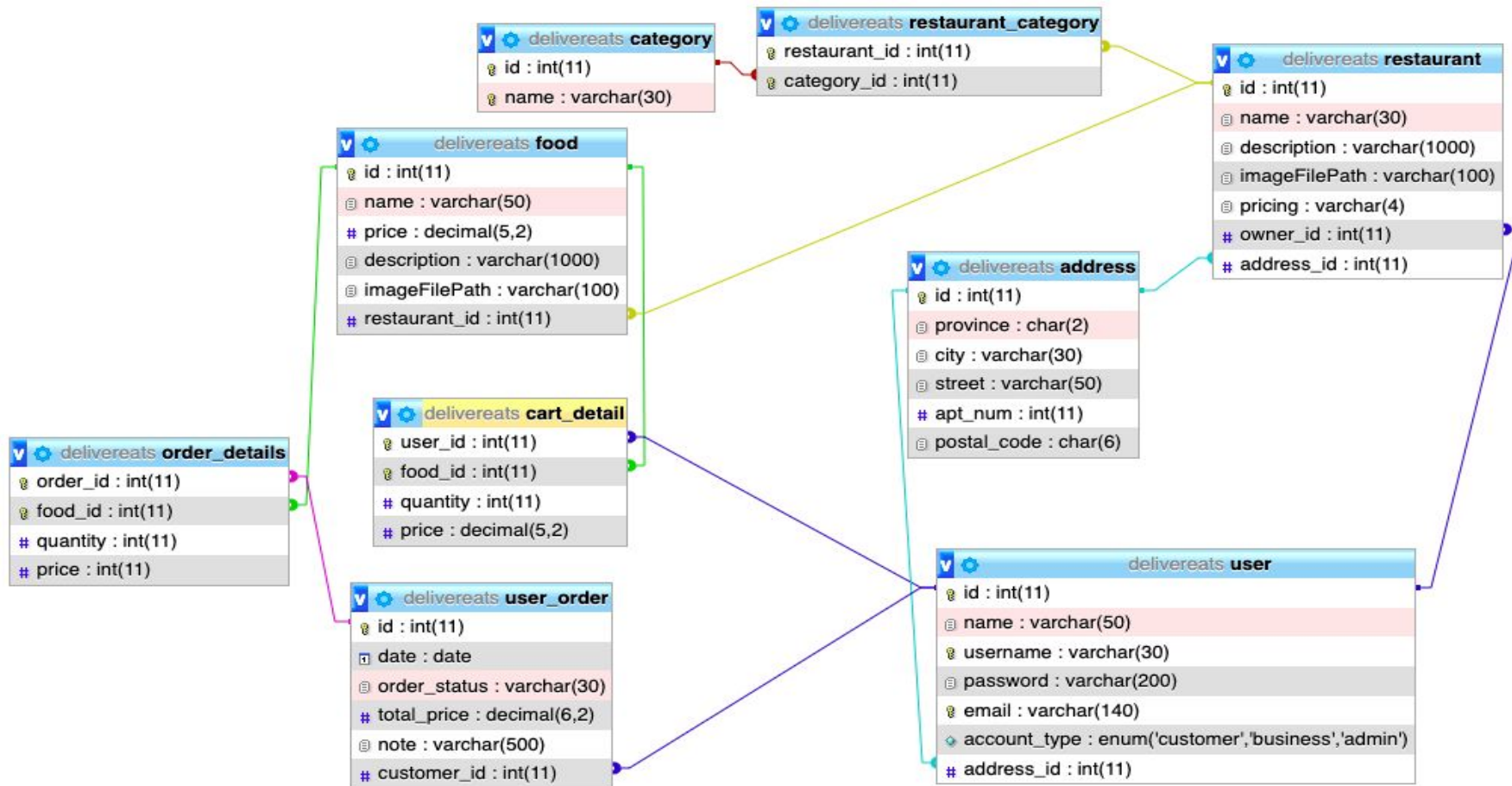
User can register/login as a business owner

CREATE: Users can add restaurants as well as food

UPDATE: Business owners can modify their restaurants/food

DELETE: Business owners can delete their restaurants/food

Database Structure



Registration

Login

Sign In

Hi there! Nice to see you again.

Sign In

[Forgot Password?](#) [Sign Up](#)

Sign Up

Name

Username

Email

Password

Password (repeated)

Account Type

Phone



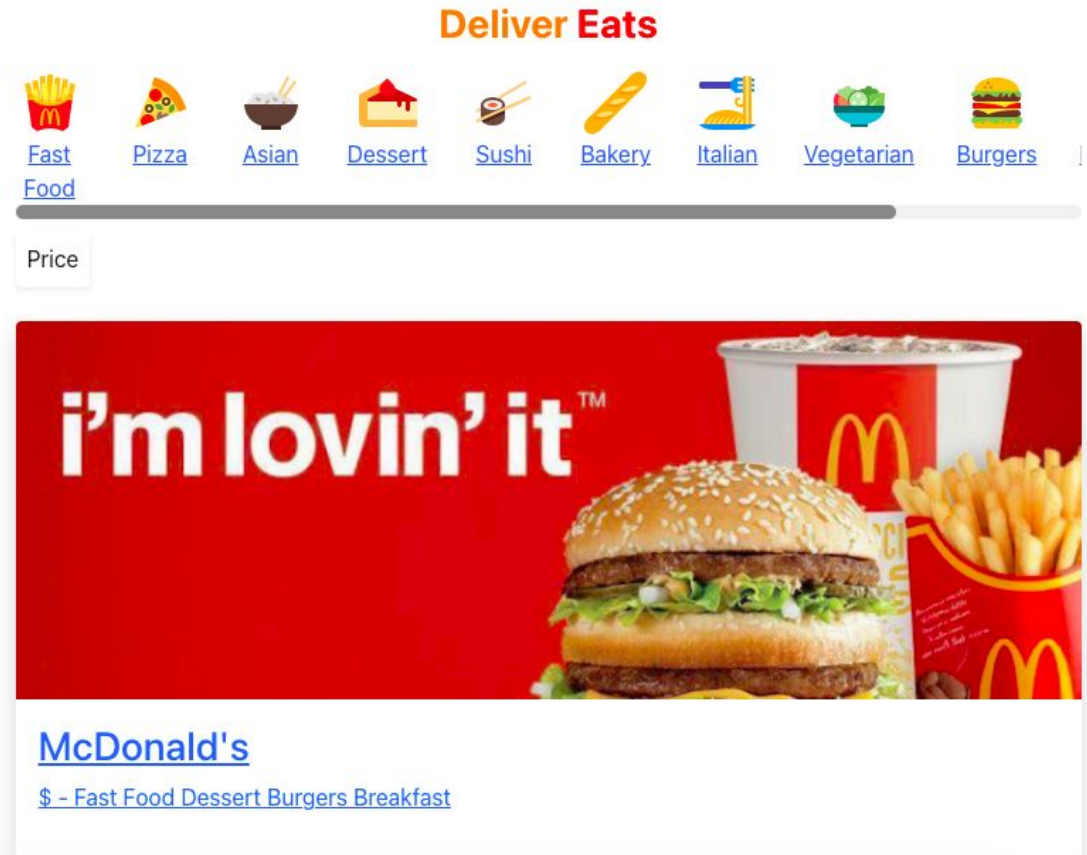
Address Selection

Have an Account? [Log in?](#)

Register

HOME PAGE

- User can browser in different food categories
- User can sort restaurants by their price range
- User can choose from different restaurants



Manage Account

- Update Profile Info
- Manage restaurants for business owners

Account

[My Profile](#)

[Manage Restaurants](#)

[Log out](#)

© Copyright 2021 by [FSD01 Team 4](#).


Home


Browse


Cart


Orders


Account

Manage Restaurants

Login successful

List of your restaurant(s)

Add new restaurant

Update existing record(s)

Name	Actions			
Bistro Italian	Add food	Menu	Change details	Delete

Manage Admin Account

You can manage:

[Users](#)[Restaurants](#)[Food items](#)[Check orders](#)[Log out](#)[Back to main page](#)[Create new user](#)






Update existing record(s)

#	name	username	email	account_type	actions
14	Ronald McDonald	RonMcDon	ronald@mcdonalds.com	business	Edit Delete
15	phil-jonathan	phila	philjonathan1996@gmail.com	business	Edit Delete
16	Anthony Barker	goodeater	goodeater@myspace.com	admin	Edit Delete
17	Richy G. Rich	restful	restful@api.ca	business	Edit Delete

Manage Admin Account

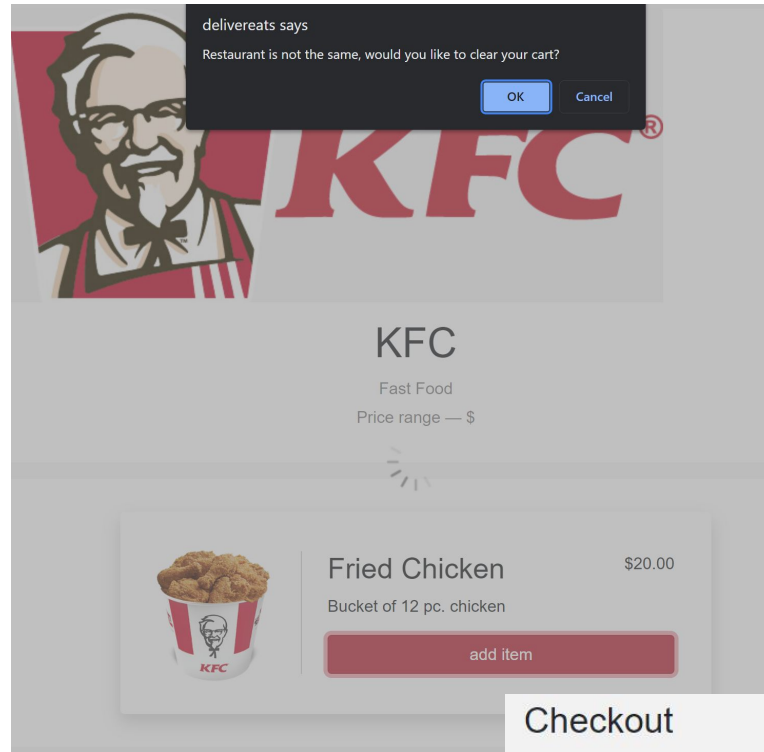
[Back to main page](#)

List of food items

#	name	price	description	image	actions
9	Cheeseburger	3.50	Hamburger with cheese, pickles, onions, lettuce, and ketchup.		Delete
10	Fries	1.25	Crispy french fries		Delete
11	Fried Chicken	20.00	Bucket of 12 pc. chicken		Delete
12	Brunch Combo #1	14.75	English muffin Eggs bacon sandwiches, potatoes, fruits.		Delete
13	Brunch Combo #2	16.00	2 Eggs, bacon, sausages, potatoes, waffles topped with apples and caramel sauce.		Delete
14	Pizza Perfecto	3.55	Fresh crust pizza with high-quality ingredients		Delete
15	pizza	12.75	Margarita wit cheese tomato and vegetables		Delete
16	cafe	3.00	cafe with milk		Delete

SHOPPING

- Add to cart
- Change quantity
- Remove items



Checkout

Img	Name	Description	Quantity	Price	Remove
	Fries	Crispy french fries	1	1.25	

Proceed to checkout

SHOPPING

- Checkout
- Enter Payment Info
- View past orders

Shopping Cart

Img	Name Description	Quantity	Price
	Fries Crispy french fries	1	1.25

Subtotal: \$1.25
Delivery Fee: \$10
Tax: 15%
Total: \$12.94

Card number

1234 1234 1234 1234



Expiry

MM / YY

CVC

CVC



Country

Canada



Postal code

M5T 1T4

Pay now

Your Orders



McDonald's

\$19.54

2021-11-14 — delivered



Cheeseburger

Qty: 2

Total: \$7

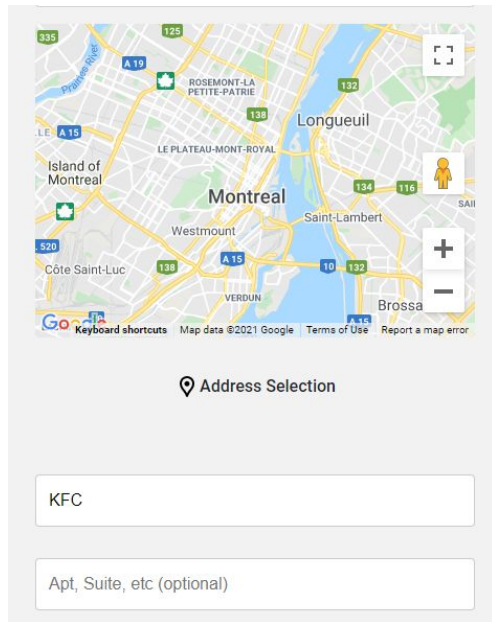


McDonald's

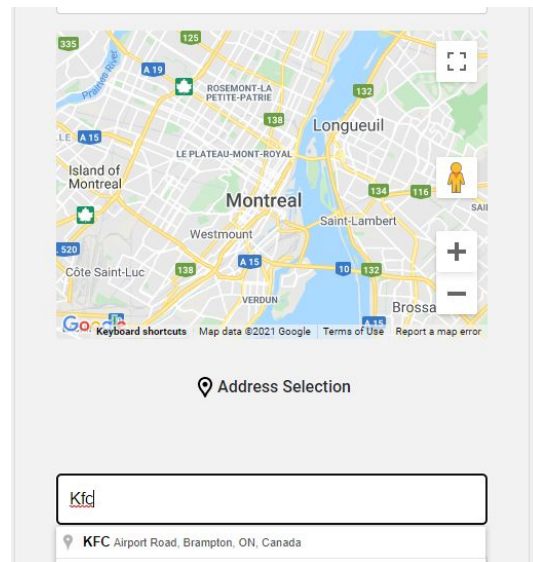
\$15.52

2021-11-13 — delivered

Google Maps API



Register



Restaurant

Google Maps API

Including the address form api creation

```
{% include 'address-js.html.twig' with {apiKey, addressType: "address"} %}
```

Autocomplete form creation

```
const autocomplete = new google.maps.places.Autocomplete(autocompleteInput, {  
  fields: ["address_components", "geometry", "name"],  
  types: ["{{addressType}}"],  
  componentRestrictions: {  
    'country': ['CA']  
  },  
});
```

Stripe API

```
$paymentIntent = \Stripe\PaymentIntent::create([  
    'amount' => calculateOrderAmount($jsonObj['items'], $log),  
    'currency' => 'cad',  
    'payment_method_types' => [  
        'card'  
    ],  
    'metadata' => [  
        'user_id' => $userId  
    ]  
]);
```

```
$cart = DB::query("SELECT * FROM cart_detail WHERE user_id=%i", $paymentIntent['metadata']['user_id']);
```

Sending specific data between api
and webhook

```
./stripe.exe listen --forward-to deliverats/api/order
```

```
// Handle the event  
switch ($event->type) {  
    case 'charge.succeeded':
```

```
./stripe.exe listen --forward-to deliverats/api/order -e charge.succeeded
```

Testing events locally

Javascript auto redirect with countdown

```
<script>
var secsLeft = 5;
setInterval(() => {
    document.getElementById("countdown").innerHTML = secsLeft + "";
    secsLeft--;
    if (secsLeft < 0) {
        window.location.href = "/";
    }
}, 1000);
</script>
{% endblock %}
```

You can login in now!

We are glad that you have become a member of our family. You will be automatically redirected in 3 seconds.

Flash message allowed to reduce the number of templates

```
$container['view']->getEnvironment()->addGlobal('flashMessage', getAndClearFlashMessage());

function setFlashMessage($message) {
    $_SESSION['flashMessage'] = $message;
}

// returns empty string if no message, otherwise returns string with message and clears it
function getAndClearFlashMessage() {
    if (isset($_SESSION['flashMessage'])) {
        $message = $_SESSION['flashMessage'];
        unset($_SESSION['flashMessage']);
        return $message;
    }
    return "";
}
```

Middleware

```
5
6 $app->add(function (ServerRequestInterface $request, ResponseInterface $response, callable $next) {
7     $url = $request->getUri()->getPath();
8     if (startsWith($url, "/admin")) {
9         if (!isset($_SESSION['user']) || $_SESSION['user']['account_type'] != "admin") { // refuse if u
10             $response = $response->withStatus(403);
11             return $this->view->render($response, 'admin/error-access-denied.html.twig');
12         }
13     }
14 }
```


Protection from access to other business owner's restaurants

```
if (!isset($_SESSION['user']) || $_SESSION['user']['account_type] != "business") {
    $response = $response->withStatus(403);
    return $this->view->render($response, 'businessOwner/not-owner.html.twig');
}
$item = DB::queryFirstRow("SELECT * FROM restaurant WHERE id=%d", $args['id']);
if (!$item || $item['owner_id] != $_SESSION['user']['id']) {
    $response = $response->withStatus(404);
    return $this->view->render($response, 'error-page-not-found.html.twig');
}
```

Future Work

```
$app->get('/foodsearch', function($request, $response, $args) {  
  
    $result = $_GET['result'];  
    $items = DB::query("SELECT * FROM food WHERE name LIKE '%$result%'");  
    return $this->view->render($response, 'result-search.html.twig', ['results' => $items]);  
});  
for user.php
```

- Implement enhanced search
- GPS/location services to show the nearest restaurants to the user
- Implement email password recovery
- Implement multiple languages support (multiple templates in subdirectories 'en' 'fr')

Summary

We successfully combined different technologies and have created a full-fledged web application.

