

# MyApp Template

---

## 1. Introduction

### About MyApp

MyApp is web and mobile application template that combine the following technologies: PHP (Laravel and Codeigniter), Bootstrap, Jquery, PHP Grid, Query Builder, Chart 4 PHP, Stimulsoft Report, API Server with Sanctum and Flutter for mobile development tool. It is designed to simplify application development for business processes.

MyApp provides common features needed for (web) developer as start package to boost and accelerate implementation. It already has authentication and authorization, crud (create read update delete), master detail, query builder, report, chart and api server.

MyApp template is built on 2 reputable PHP Frameworks : Codeigniter v.3.x and Laravel v.9.x. It uses Mysql as database management.

## 2. Features

### a. Web Application

#### 1) Authentication and Authorization

Authentication and authorization features are already available for both CI and Laravel. User will be authenticated with password and token (for API), then User will have roles based on its level.

On Codeigniter, developer use the following lines:

```
$this->auth->restrict();
$this->auth->is_valid_token();
$this->auth->check_allowed_level('menu_id', 'function_name');
```

On Laravel :

```
Route::get('/appuser', [AdmController::class, 'appuser'])->
    middleware('is_admin');
OR
Route::get('/appuser', [AdmController::class, 'appuser'])->
    middleware('auth');
```

On database, there are two tables related to authentication and authorization : users and roles/users\_level.

## 2) Dynamic Navigation

Developer can make application menu / navigation dynamically. Navigation is defined on table menu which contains configuration for navigation structure , allowed users, url etc.

Admin > Menu Management						
	Keyword	Type	Name	Parent	Url	Level Allowed
<input type="checkbox"/>	Search...	Search...	Search...	-	Search...	-
<input type="checkbox"/>	Dashboard	summary	Dashboard	1 (Dashboard)	index.php/dashboard/index	Administrator, Admin_md, User
<input type="checkbox"/>	Admin User Management	table	User Management	2 (Admin)	index.php/admin/user_management	Administrator
<input type="checkbox"/>	Admin Menu Management	table	Menu Management	2 (Admin)	index.php/adm/menu_management	Administrator
<input type="checkbox"/>	Admin Level Management	table	Level Management	2 (Admin)	index.php/adm/level_management	Administrator
<input type="checkbox"/>	User Table Privilage	table	User Table Privilage	2 (Admin)	index.php/adm/user_table_privilege	Administrator, Admin_md
<input type="checkbox"/>	Master Area	table	Area	3 (Master)	index.php/master/area	Administrator, Admin_md
<input type="checkbox"/>	Master Discipline	table	Discipline	3 (Master)	index.php/master/discipline	Administrator, Admin_md
<input type="checkbox"/>	Master Phase	table	Phase	3 (Master)	index.php/master/phase	Administrator, Admin_md
<input type="checkbox"/>	Master Customers	table	Customers	3 (Master)	index.php/master/customers	Administrator, Admin_md
<input type="checkbox"/>	Master Products	table	Products	3 (Master)	index.php/master/products	Administrator, Admin_md, User
<input type="checkbox"/>	Master Categories	table	Categories	3 (Master)	index.php/master/categories	Administrator, User

### 3) Dynamic Layout/Skin

There are 6 layouts option available on MyApp Template. Developer need to specify class on body tag for both Codeigniter and Laravel.

```
<body class="layout_skin"> //layout_skin : smart-style-0 - smart-style-5
```

### 4) Create Read Update Delete (CRUD)

MyApp template using phpgrid library to manage table/data. It already integrated user authorization with the grid access. Admin or developer just need to add privilege to certain level in order to define access for table/data. PHP Code required if a new table need to be added to the navigation/menu.

The screenshot shows a grid-based administrative interface titled "Admin > User Table Privilege". The grid has three columns: "Menu", "Level", and "Privilege". Each row represents a menu item with its ID, name, level, and the permissions it has been granted. The "Privilege" column lists actions such as Create, Read, Update, Delete, Bulk Update, and Export.

<input type="checkbox"/>	Menu	Level	Privilege
	-	-	-
<input type="checkbox"/>	8 (User Management)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update
<input type="checkbox"/>	9 (Menu Management)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update
<input type="checkbox"/>	11 (User Table Privilage)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update
<input type="checkbox"/>	12 (Area)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update
<input type="checkbox"/>	13 (Discipline)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update
<input type="checkbox"/>	10 (Level Management)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update
<input type="checkbox"/>	14 (Phase)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update
<input type="checkbox"/>	15 (Customers)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update, Export
<input type="checkbox"/>	16 (Products)	1 (Administrator)	Create, Read, Update, Delete, Bulk Update, Export
<input type="checkbox"/>	16 (Products)	3 (User)	Read, Update, Export

## 5) Master Detail

MyApp provided a master detail example for developer with form and several detail tables involve. Master detail feature is supported for licensed version of PHPGrid.

The screenshot shows a master-detail application. On the left, there is a form titled "Order" with fields for Customer (VINET), Handle By (5), Order Date (04-Jul-1996), Required Date (01-Aug-1996), Shipped Date (16-Jul-1996), Status (CREATED), and buttons for Save and Print Order. Below this form is a table titled "Order Details" with columns for Order Id, Product, Unit Price, Quantity, and Total. It lists four items: Queso Cabrales, Singaporean Hokkien Fried Mee, Mozzarella di Giovanni, and Chai, along with their respective details and a Grand Total of \$458.00. To the right of the master form is a grid titled "List of Order" with columns for Customer, Handle by, Date, and Status. The grid displays several rows of order data, with the first row highlighted in yellow.

## 6) Report and Query Builder

MyApp provides query builder for dynamic reports. It build sql query to be passed to the report.

The screenshot shows a query builder interface titled "Report Filter - Order Detail Report". It features a search bar at the top with placeholder text "Report Filter - Order Detail Report". Below the search bar is a filter configuration area with "AND" and "OR" buttons. The "AND" section contains a dropdown for "Order Id" with an "in" operator and a value selector showing "10248 (VINET) 10251 (VICTE)", with a "Delete" button next to it. The "OR" section contains a dropdown for "Customer" with an "equal" operator and a value selector showing "ANTON (Antonio Moreno Taquería) AROUT (Around the Horn) BERGS (Berglunds snabbköp) BLAUS (Blauer See Delikatessen)", with a "Delete" button next to it. At the bottom of the filter area are "Reset" and "Display Report" buttons. The background of the interface is light orange.

Developer can choose Report tools library based on necessity. He can use Stimulsoft Report if lot of reports need to be created. This library is premium version, need to buy license for legal usage. Developer has experience in crystal report will have advantage when using stimulsoft report due to its similar usage.

Other option is to use dompdf library. It's also available on MyApp for development when only small amount of report is needed.

## 7) Dashboard and Chart



Developer can choose to use several charts available on MyApp Template. Two usage examples are available on MyApp, PHPChart (licensed) and ChartJs.

## 8) API (Jason Web Token -JWT or Sanctum)

Codeigniter using JWT token for Web and API (Application Programming Interface).

```
$data = array('id'=>$user['id'], 'username'=>$user['username']);
$this->authorization_token->generateToken($data);
```

```
$this->authorization_token->validateTokenWeb($token); //for Web
$this->authorization_token->validateToken($headers['Authorization']); //for API
```

Laravel use Library Sanctum for this purpose.

Route :

```
Route::apiResource('/tasks', TasksController::class)->middleware('auth:sanctum');
```

Usage :

```
$user->createToken('API Token')->plainTextToken;
```

## 9) API Access using Postman

A Controller (ApiToken) is available in MyApp to test token generation and verification.

On Postman fill the body of api endpoint as follow :

```
POST http://localhost/myapp/index.php/apitoken/tokenGen

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL

 user_id 198
 fullname Azka Salman
 email azka198@gmail.com

Key Value Description
Body Cookies (1) Headers (10) Test Results Status: 200 OK Time: 65 ms Size: 623 B Save Response
Pretty Raw Preview Visualize JSON
1
2
3
4
5
{"status": true,
 "message": "success",
 "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJlc2VyX2lkIjoiMTk4IiwidXNlcm5hbWUiOiJBemthIFNhbg1hbisImVtYWlsIjoiYXpriYTE50EBnbWFpbC5jb20iLCJBUElfVElNRSI6M
TY3NDY5Mzk4Mn0.YSmddSsrDNUWbqKS3iNgUSd1awhg-bmd3Rv-A33s4tuU"
}

Send
```

When verifying user, fill the header with these informations :

The screenshot shows the Postman application interface. At the top, there's a header bar with 'POST' selected as the method and the URL 'http://localhost/myapp/index.php/apitoken/verify'. On the right side of the header is a blue 'Send' button. Below the header, there are tabs for 'Params', 'Authorization', 'Headers (11)', 'Body', 'Pre-request Script', 'Tests', and 'Settings', with 'Headers (11)' currently active. To the far right, there's a 'Cookies' tab. The main content area has a table for headers:

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Accept	application/json				
<input checked="" type="checkbox"/> Content-Type	application/json				
<input checked="" type="checkbox"/> Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJc...				

Below the headers, there's a table for key-value pairs:

Key	Value	Description

At the bottom left, there are tabs for 'Body', 'Cookies (1)', 'Headers (11)', and 'Test Results', with 'Body' currently active. To the right of these tabs, status information is displayed: 'Status: 200 OK', 'Time: 166 ms', 'Size: 616 B', and a 'Save Response' button. Below the status, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON' (with a dropdown arrow). The JSON response body is shown in a code editor-like area:

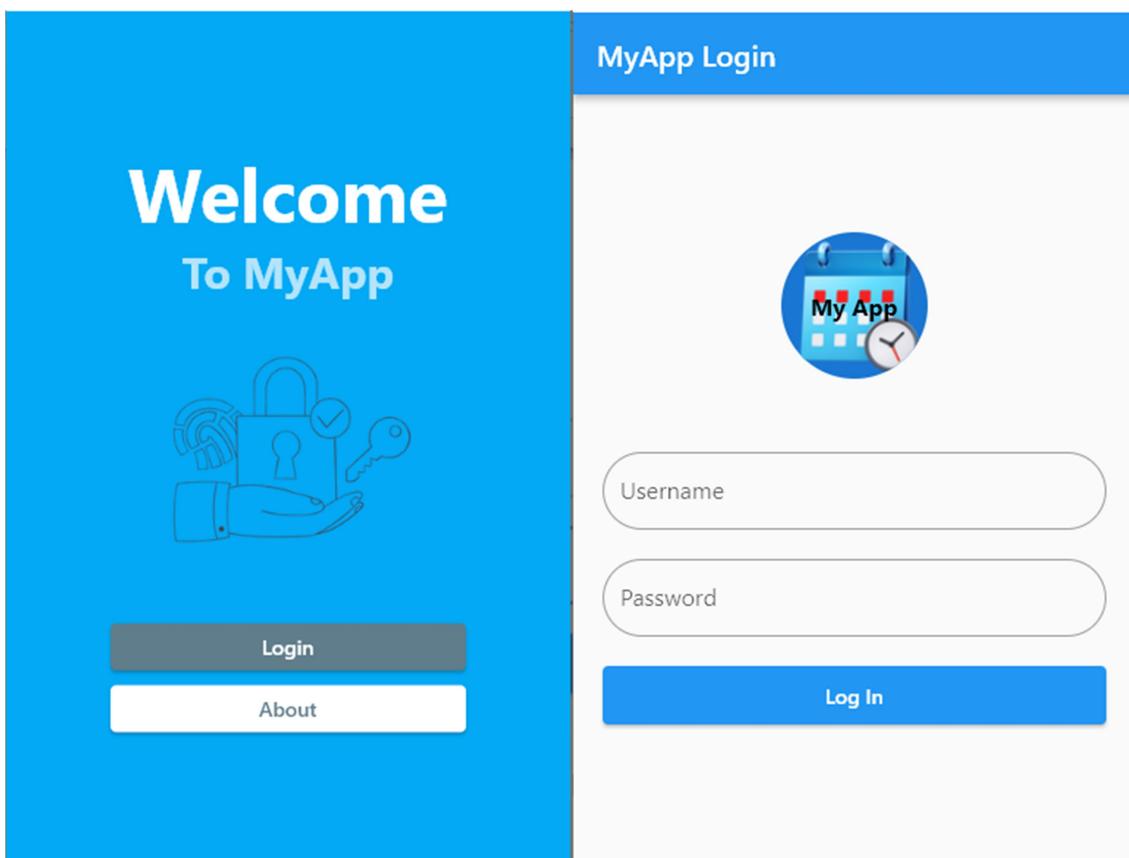
```
1
2     "status": true,
3     "data": {
4         "user_id": "198",
5         "fullname": "Azka Salman",
6         "email": "azka198@gmail.com",
7         "API_TIME": 1674693982
8     }
9 }
```

Authorization is filled with token value getting from login. List of API url is available on appendix.

## b. Mobile Application

MyApp has two mobile application templates build with Flutter. These mobile app templates are used for accessing API service on MyApp and simple CMS (News Management).

### 1) MyApp Mobile Template



Login Page

The image shows the Category screen of a mobile application named "MyApp". The left side displays a list of seven categories, each with a blue circular index number, a name, a description, and edit/delete icons. The categories are:

- 1 Beverages: Soft drinks, coffees, teas, beers, and ales
- 2 Condiments: Sweet and savory sauces, relishes, spreads, and seasonings
- 3 Confections: Desserts, candies, and sweet breads
- 4 Dairy products: Cheeses
- 5 Grains/Cereals: Breads, crackers, pasta, and cereal
- 6 Meat/Poultry: Prepared meats
- 7 Produce: Dried fruit and bean curd

A large blue circular button with a white plus sign is located at the bottom right of the category list. Below the list is a navigation bar with icons for Home, Category, Product, Customer, Supplier, and Order. The right side of the screen is a modal titled "Add Category" with fields for "Category Name" and "Description", and a "Save" button.

Category Screen

**Product Screen**

Category	Product Name	Supplier	Price
Beverages	Chai Original	New Orleans Cajun Delights	18.00
Beverages	Max Juice	Exotic Liquids	19.00
Condiments	Aniseed Syrup	Exotic Liquids	10.00
Condiments	Chef Anton's Cajun Seasoning	New Orleans Cajun Delights	22.00
Condiments	Chef Anton's Gumbo Mix		

### ← Chai Original



Chai Original  
New Orleans Cajun Delights  
Beverages  
18.00

### ← Edit Product

Product Name  
Chai Original

New Orleans Cajun Delights ▾

Beverages ▾

Unit Price  
18.00

Image 2 ▾

**Save**

Product Detail

### MyApp

Search  
be| 🔍

**BERGS**

Company : Berglunds snabbköp  
 Contact : Christina Berglund  
 Address : Berguvsvägen 8  
 Country : Sweden

**BSBEV**

Company : B's Beverages  
 Contact : Victoria Ashworth  
 Address : Fauntleroy Circus  
 Country : UK

### MyApp

- Exotic Liquids ▼
- New Orleans Cajun Delights ▼
- 3 Grandma Kelly's Homestead ▲
- Tokyo Traders ▼
- Cooperativa de Quesos 'Las Cabras' ▼
- 6 Mayumi's ▼
- Pavlova, Ltd. ▼
- 8 Specialty Biscuits, Ltd. ▼

Home  Category  Product  Customer  Supplier  Order

Customer and Supplier

☰
MyApp

Search Order by Customer

Order	Customer	Date	Handle
10269	White Clover Markets	1996-07-31	B
10358	La maison d'Asie	1996-11-20	B
10397	Princesa Isabel Vinhos	1996-12-27	B
10549	QUICK-Stop	1997-05-27	B
10569	Rattlesnake Canyon Grocery	1997-06-16	B
10575	Morgenstern Gesundkost	1997-06-20	B
10607	Save-a-lot Markets	1997-07-22	B
10654	Berglunds snabbköp	1997-09-02	B
10730	Bon app'	1997-11-05	B
10812	Reggiani Caseifici	1998-01-02	B

1-10 of 830

<
>

Home
 Category
 Product
 Customer
 Supplier
 Order

☰
MyApp

Search Order by Customer

Order	Customer	Date	Handle
10269	White Clover Markets	1996-07-31	Buchanani
10596	White Clover Markets	1997-07-11	Callahan
10469	White Clover Markets	1997-03-10	Davolio
10723	White Clover Markets	1997-10-30	Leverling
10483	White Clover Markets	1997-03-24	King
11066	White Clover Markets	1998-05-01	King
10344	White Clover Markets	1996-11-01	Peacock
10740	White Clover Markets	1997-11-13	Peacock
10861	White Clover Markets	1998-01-30	Peacock
10696	White Clover Markets	1997-10-08	Callahan

1-10 of 14

<
>

Home
 Category
 Product
 Customer
 Supplier
 Order

Order Screen

## 2) MyApp Template for CMS (News)

MyAppNews

### Breaking News

**Gorilla Glass Victus 2 dipastikan hadir perdana di Galaxy S23 series - ANTARA**  
Livia Kristianti

### Recent News - International

**Billy Packer, 82, longtime Final Four basketball analyst, dies - ESPN**  
Longtime college basketball ...

**La Nyalla Siap Kerja dengan Siapa Saja, Cuma Ogah Exco PSSI Lama - detikSport**  
Jakarta - La Nyalla Mahmud ...

**Jawaban Mikel Arteta soal Martin Odegaard Bocorkan Cara Hentikan Erling Haaland - Bolasport**

MyAppNews

### Sport News

**Asnawi Mangkualam Resmi Gabung Jeonnam Dragons - detikSport**  
Jakarta - Asnawi Mangkualam sud...

**La Nyalla Siap Kerja dengan Siapa Saja, Cuma Ogah Exco PSSI Lama - detikSport**  
Jakarta - La Nyalla Mahmud ...

**Jawaban Mikel Arteta soal Martin Odegaard Bocorkan Cara Hentikan Erling Haaland - Bolasport**



## Gorilla Glass Victus 2 dipastikan hadir perdana di Galaxy S23 series - ANTARA

Livia Kristanti



Jakarta (ANTARA) - Samsung memang sering kali menggunakan produk dari Corning yakni Gorilla Glass Victus untuk melindungi layar perangkat pintarnya seperti pada ponsel pintar Galaxy S22 series mereka... [+1710 chars]

Source: <https://www.antaranews.com/berita/3367641/gorilla-glass-victus-2-dipastikan-hadir-perdana-di-galaxy-s23-series>

MyAppNews



## Business News

**IUN 2022** Makin Murah Nih! Subsidi Rp 7 Juta Motor Listrik Rilis Minggu Depan - detikOto

Jakarta - Menteri Koordinator ...

**Harga Saham Emiten 'Kawalan' AA Gym Anjlok - CNN Indonesia**

Jakarta, CNN Indonesia -- ...

**Pernah Pakai Teknologi Populer Ini Artinya Kamu Sudah Tua - detikNet**



### 3. Code Structure

#### a. PHPGrid

##### 1) Introduction

PHPGrid is PHP Library focusing on database management. It has complete feature on how web application deal with table management. Main features available in this library : CRUD (Create Read Update Delete), foreign key management, master detail, roles, export import etc.

PHPGrid is licensed library. Developer can choose to use free version for basic table management or premium version for advance usage.

##### 2) Adding Code

Codeigniter

```
require_once("inc/config.php");
include("inc/jqgrid_dist.php");
$g = new jqgrid();

$g->table = "mst_products";
$out = $g->render("list1");

$data['out'] = $out;
```

Laravel

```
include_once("phpgrid\config.php");
include('phpgrid\jqgrid_dist.php');

$g = new \jqgrid();
$g->table = "users";

$out = $g->render("list1");
```

### 3) Grid Configuration

You can setup grid configuration dynamically based on your business process. Example PHPGrid Configuration :

```
$option["multiselect"] = true;
$option["autowidth"] = true;
$option["autoresize"] = true;
$option["toolbar"] = "bottom";
$option["altRows"] = true;
$option["altclass"] = "myAltRowClass";
$option["rowNum"] = 20;

$option["add_options"] = array('width'=>'500');
$option["edit_options"] = array('width'=>'500');
$option["view_options"] = array('width'=>'500');
$option["export"]["sheetname"] = "My Data";
$option["shrinkToFit"] = false;

$action["add"] = FALSE;
$action["edit"] = TRUE;
$action["delete"] = TRUE;
$action["view"] = TRUE;
$action["search"] = TRUE;
$action["clone"] = TRUE;
$action["showhidecolumns"] = TRUE;
$action["bulkedit"] = FALSE;
$action["export_excel"] = TRUE;
$action['rowactions'] = TRUE;
```

## b. CodeIgniter

Detailed documentation can be found on <https://codeigniter.com/userguide3/general/welcome.html>

In MyApp template, there are several additional segment (package, libraries, helper, vendor etc) to help developer accelerates development process, including

- Smartadmin package for user interface (Bootstrap, css, chart etc)  
`base_url() \ assets\smartadmin`
- PHPGrid for database management (licensed [www.phpgrid.org](http://www.phpgrid.org))  
`base_url() \ media\js`  
`base_url() \ application\controllers\inc`
- Rest Server  
`base_url() \ vendor\chriskacerguis\codeigniter-restserver\src`
- QueryBuilder  
`base_url() \ media\querybuilder`
- Stimulsoft for reporting tool (licensed [www.stimulsoft.com/en](http://www.stimulsoft.com/en))  
`base_url() \ report`
- Dompdf for reporting  
`base_url() \ vendor\dompdf`
- Jason Web Token (JWT)  
`base_url() \ application\third_party\php-jwt`

Some important configurations on MyApp template in Codeigniter :

- Helper  
`base_url() \ application\helpers\my_helper.php`
- Rest Server  
`base_url() \ vendor\chriskacerguis\codeigniter-restserver\src\ rest.php`
- Jason Web Token  
`base_url() \ application\config\ jwt.php`

### c. Laravel

Detailed information can be found on <https://laravel.com/docs/9.x>

Additional segment (package, libraries, helper, vendor etc) to help developer accelerates development process, including :

- Smartadmin package for user interface (Bootstrap, css, chart etc)  
`base_url() \ public\smartadmin`
- PHPGrid for database management (licensed [www.phpgrid.org](http://www.phpgrid.org))  
`base_url() \ public\phpgrid`  
`base_url() \ app\Http\Controllers\phpgrid`
- Rest Server with sanctum
- QueryBuilder  
`base_url() \ public\querybuilder`

Some important configurations on MyApp Template in laravel :

- Middleware IsAdmin.php  
`base_url() \ app\Http\Middleware`
- Gate AppServiceProvider.php  
`base_url() \ app\Providers`
- Helper for generating dynamic navigation  
`base_url() \ app\Http\Controllers\myLib`

## 4. Appendix

### a. Database Structure (CI)

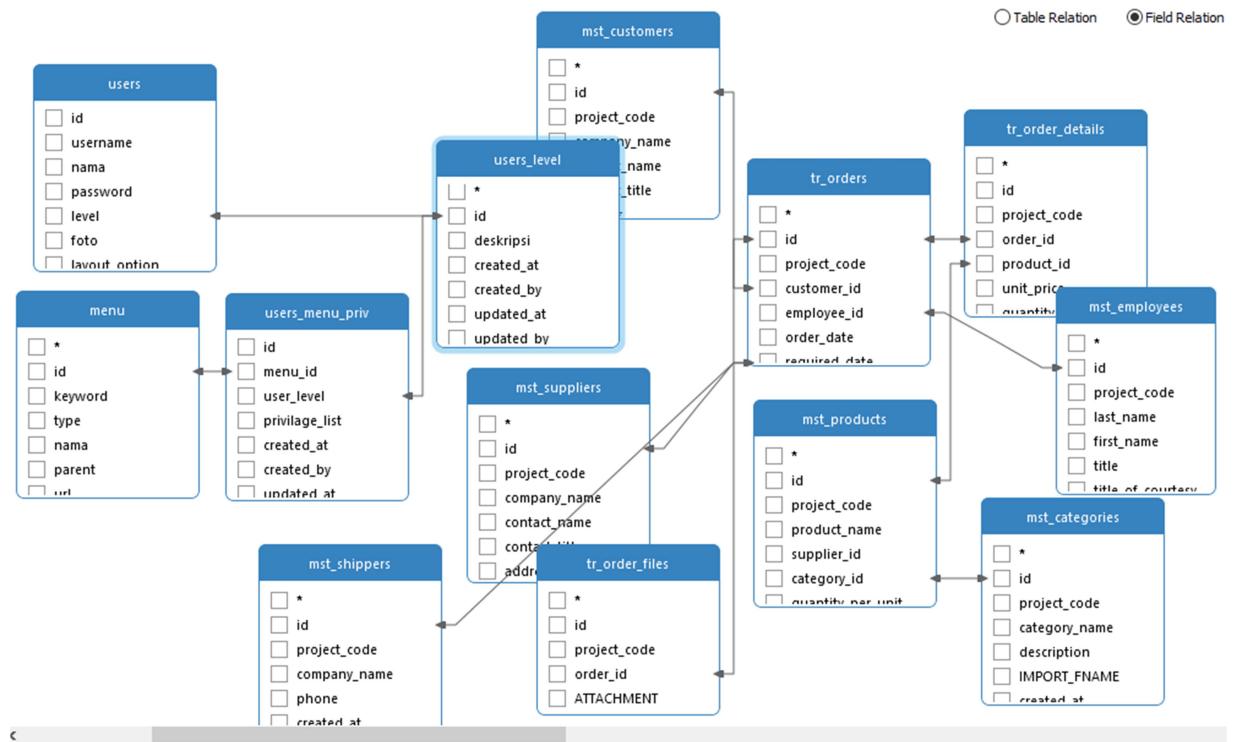
#### Tables

Name	Auto Increment Value	Modified Date	Data Length	Engine	Rows	Comment
apikeys	2		16 KB	InnoDB	0	
apikeys_limits	3		16 KB	InnoDB	2	
import_history	4		16 KB	InnoDB	2	
menu	25	2023-01-16 13:21:04	4 KB	MyISAM	24	
mst_area	14		16 KB	InnoDB	4	
mst_categories	9		16 KB	InnoDB	9	
mst_customers	0		48 KB	InnoDB	92	
mst_discipline	13		16 KB	InnoDB	12	
mst_employees	10		16 KB	InnoDB	9	
mst_module	4		16 KB	InnoDB	3	
mst_phase	4		16 KB	InnoDB	3	
mst_products	78		16 KB	InnoDB	77	
mst_shippers	4		16 KB	InnoDB	3	
mst_suppliers	30		16 KB	InnoDB	29	
tr_order_details	2159		160 KB	InnoDB	2157	
tr_order_files	6		16 KB	InnoDB	3	
tr_orders	11078		144 KB	InnoDB	830	
users	9	2023-01-13 06:53:14	2 KB	MyISAM	4	
users_level	6	2022-11-07 19:56:24	1 KB	MyISAM	3	
users_menu_priv	15	2022-11-09 10:16:42	1 KB	MyISAM	12	
users_privilege	10		16 KB	InnoDB	9	

#### Views

Name	Is Updatable
v_main_order_details	Yes
v_main_orders	Yes
v_order_summary_by_country	No
v_order_summary_by_employee	No
v_order_summary_by_shipper	No
v_products	Yes

## ER Diagram



## b. API Urls List (CI)

For testing and simplicity, JWT token is not activate for these API endpoints. Only two controllers using JWT Token : ApiToken and Authentication. In Order to activate the token, adding these steps :

1. Login to MyApp using this url : <http://localhost/myapp/index.php/authentication/login>

The screenshot shows a Postman request configuration for a POST method to the URL `http://localhost/myapp/index.php/authentication/login`. The **Body** tab is selected, showing the following form-data:

KEY	VALUE	DESCRIPTION	...	Bulk Ed
username	administrator			
password	****			
Key	Value	Description		

The response status is 200 OK, and the JSON response body is:

```
1 "status": true,
2 "message": "success",
3 "data": {
4     "id": "1",
5     "username": "administrator",
6     "nama": "Administrator",
7     "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjEiLCJ1c2VybmFtZSI6ImFkbWluaXN0cmF0b3IiLCJuYW1hIjoiQWRtaW5pc3RyYXRvciiSIkFQSV9USU1FIjoxNjc0NzAxMDExfQ.-j1eqEGMl1iDuC32FilAP_MqEykfL5hAT8L0jCK7p08"
9 }
10 }
```

2. Add Authorization on Header and assign its value with the token generated when login

The screenshot shows a Postman request configuration with the **Headers** tab selected. The headers listed are:

KEY	VALUE
Accept	application/json
Content-Type	application/json
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjEiLCJ1c2VybmFtZSI6ImFkbWluaXN0cmF0b3IiLCJuYW1hIjoiQWRtaW5pc3RyYXRvciiSIkFQSV9USU1FIjoxNjc0NzAxMDExfQ.-j1eqEGMl1iDuC32FilAP_MqEykfL5hAT8L0jCK7p08

3. Add this code on API function to verify user

```
$this->authorization_token>validateToken($headers['Authorization']);
```

MyApp API end point list :

(Developer should replace this <http://localhost/myapp/index.php> with your base url)

- Login (POST) :  
input : body (username, password)  
<http://localhost/myapp/index.php/authentication/login>
- Order list (GET) :  
<http://localhost/myapp/index.php/api/orders>
- Order with pages (GET)  
Input : body (limit : number of data/page, offset : starting point)  
<http://localhost/myapp/index.php/api/orders?limit=10&offset=30>
- An Order (GET)  
<http://localhost/myapp/index.php/api/orders/id/10269>
- Category list (GET)  
<http://localhost/myapp/index.php/api/categories>

A Category (GET)  
• <http://localhost/myapp/index.php/api/categories/id/1>
- Create Category (POST)  
Input : body (category\_name, description)  
<http://localhost/myapp/index.php/api/categories>
- Update Category (PUT)  
Input : body (category\_name, description)  
<http://localhost/myapp/index.php/api/categories/id/10>
- Delete Category (DELETE)  
<http://localhost/myapp/index.php/api/categories/id/12>
- Product list (GET)  
<http://localhost/myapp/index.php/api/products>
- Product with pages (GET)  
<http://localhost/myapp/index.php/api/products?limit=10&offset=20>

- A Product (GET)  
<http://localhost/myapp/index.php/api/products/id/1>
- Create Product (POST)  
Input : body (product\_name, supplier\_id, category\_id, unit\_price, img\_path)  
<http://localhost/myapp/index.php/api/products>
- Update Product (PUT)  
Input : body (product\_name, supplier\_id, category\_id, unit\_price, img\_path)  
<http://localhost/myapp/index.php/api/products/id/79>
- Delete Product (DELETE)  
<http://localhost/myapp/index.php/api/products/id/1>
- Customer  
<http://localhost/myapp/index.php/api/customers>  
<http://localhost/myapp/index.php/api/customers?limit=10&offset=5>  
<http://localhost/myapp/index.php/api/customers/id/ANTON>
- Supplier  
<http://localhost/myapp/index.php/api/suppliers>  
<http://localhost/myapp/index.php/api/suppliers?limit=10&offset=5>  
<http://localhost/myapp/index.php/api/suppliers/id/1>

### c. Database Structure (Laravel)

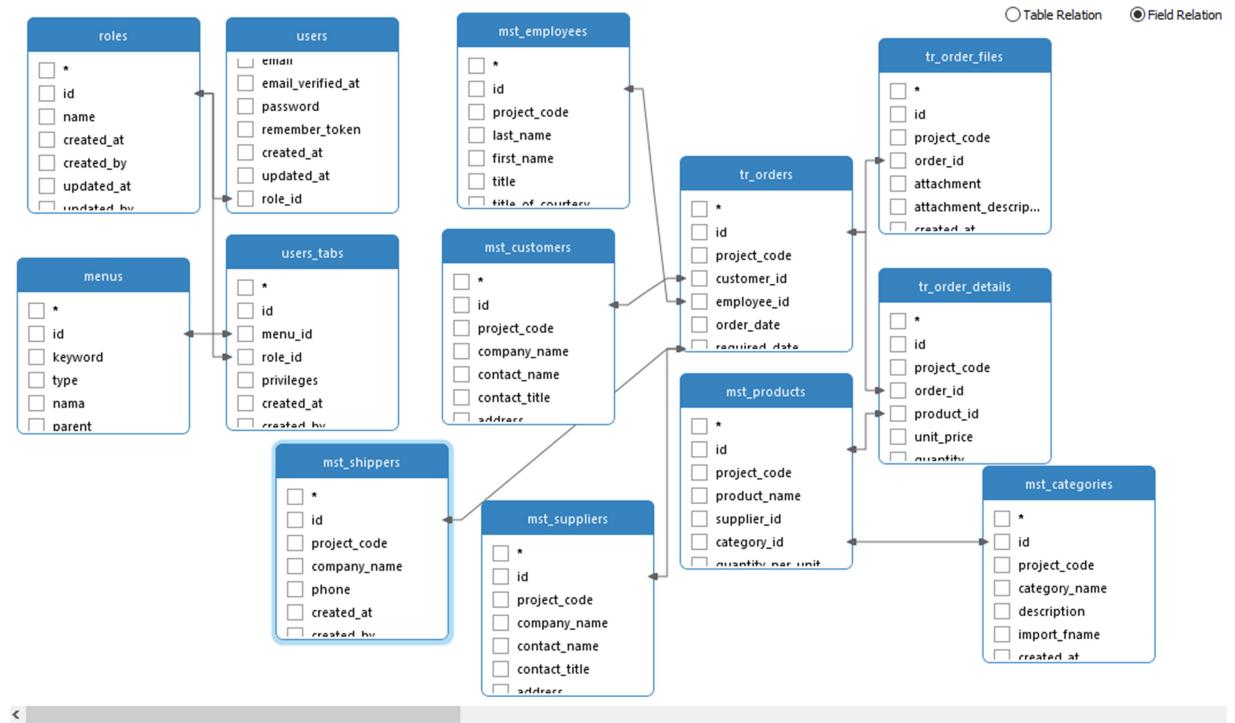
#### Tables

Name	Auto Increment Value	Modified Date	Data Length	Engine	Rows
failed_jobs	1		16 KB	InnoDB	0
imports_histories	3		32 KB	InnoDB	2
menus	18	2022-12-18 12:35:52	2 KB	MyISAM	17
migrations	8		16 KB	InnoDB	4
mst_categories	19		16 KB	InnoDB	8
mst_customers	0		48 KB	InnoDB	92
mst_employees	10		16 KB	InnoDB	9
mst_products	80		16 KB	InnoDB	77
mst_shippers	4		16 KB	InnoDB	3
mst_suppliers	30		16 KB	InnoDB	29
password_resets	0		16 KB	InnoDB	0
personal_access_tokens	5		16 KB	InnoDB	3
roles	4		16 KB	InnoDB	2
tasks	254		96 KB	InnoDB	250
tr_order_details	2159		160 KB	InnoDB	2157
tr_order_files	3		16 KB	InnoDB	2
tr_orders	11078		144 KB	InnoDB	830
users	5		32 KB	InnoDB	4
users_privileges	10		16 KB	InnoDB	9
users_tabs	11	2022-12-19 09:19:17	1 KB	MyISAM	10

#### Views

Name	Is Updatable
v_main_order_details	Yes
v_main_orders	Yes
v_order_summary_by_country	No
v_order_summary_by_employee	No
v_order_summary_by_shipper	No
v_products	Yes

## ER Diagram



## d. API Urls List (Laravel)

In Laravel, MyApp provide two controllers (AuthController and TasksController) that implements token with sanctum.

In case you need to add token to your API, do these steps :

1. On Your API Route, add this line

```
Route::apiResource('/endPoint', AController::class)->middleware('auth:sanctum');
```

2. Login on Postman using login endpoint (<http://localhost/myapp/public/api/login>)

The screenshot shows a Postman request configuration for a POST request to <http://localhost/myapp/public/api/login>. The 'Body' tab is selected, showing form-data fields: 'email' (azka198@gmail.com) and 'password' (\*\*\*\*). The response status is 200 OK, and the JSON response body is displayed in a pretty-printed format:

```
3   "message": null,
4   "data": [
5     "user": {
6       "id": 4,
7       "name": "Azka Salman",
8       "email": "azka198@gmail.com",
9       "email_verified_at": null,
10      "created_at": "2022-12-01T06:18:17.000000Z",
11      "updated_at": "2022-12-01T06:18:17.000000Z",
12      "role_id": 2
13    },
14    "token": "4|53gNZXtX3jvIhrsKFeg2o5FposPs5tGraEQu7bk"
15  ]
16 }
```

### 3. Add Bearer Token on Authorization with the token from login

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#).

Body Status: 200 OK

```
1
2   "data": [
3     {
4       "id": "2",
5       "attributes": {
6         "name": "Omnis autem autem sit suscipit neque.",
7         "description": "Natus qui possimus error eius et. Cupiditate harum excepturi omnis. Tempore tenetur non dolore Reprehenderit qui et omnis modi magnam.",
8         "priority": "high",
9         "created_at": "2022-12-01T07:21:36.00000Z",
10        "updated_at": "2022-12-01T07:21:36.00000Z"
11      },
12      "relationships": {
```

MyApp API end point list (for testing on live server, use this base url :

You should replace this part of api url <http://localhost/myapp/public> with your valid url.

- Category list (GET)  
<http://localhost/myapp/public/api/categories>
- A Category (GET)  
<http://localhost/myapp/public/api/categories/2>
- Create Category (POST)  
Input : body (category\_name, description)  
<http://localhost/myapp/public/api/categories>
- Update Category (PUT)  
Input : body (category\_name, description)  
<http://localhost/myapp/public/api/categories>
- Delete Category (DELETE)  
<http://localhost/myapp/public/api/categories/18>

- Product list (GET)  
<http://localhost/myapp/public/api/products>
- Product list with page (GET)  
<http://localhost/myapp/public/api/products?page=2>  
Note : to use paging in laravel API, add this code to controller  
`$products = Product::paginate(10);`
- A Product (GET)  
<http://localhost/myapp/public/api/products/2>
- Create Product (POST)  
Input : body (product\_name, supplier\_id, category\_id, unit\_price, img\_path)  
<http://localhost/myapp/public/api/products>
- Update Product (PUT)  
Input : body (product\_name, supplier\_id, category\_id, unit\_price, img\_path)  
<http://localhost/myapp/public/api/products>
- Delete Product (DELETE)  
<http://localhost/myapp/public/api/products/18>
- Customer
  - <http://localhost/myapp/public/api/customers>
  - <http://localhost/myapp/public/api/customers/ALFKI>
- Supplier
  - <http://localhost/myapp/public/api/suppliers>
  - <http://localhost/myapp/public/api/suppliers/1>
- Order
  - <http://localhost/myapp/public/api/orders>
  - <http://localhost/myapp/public/api/orders/10248>

For live testing, use this url as base url <https://myapp.procims.com/index.php>  
Currently, live testing on live server only for Codeigniter.