PROJECT REPORT ON

SOMANATH AGRO BILLING SYSTEM

M.Sc. I.T. (2019-2021)

Prepared by

Vataliya Yatharth

Project Guide

Dr. CK Kumbharana

Report Submitted To

Department of Computer science
Saurashtra university

INDEX

Introduction

Somanath Agro Billing System is a idea is come from making traditional billin storing system to edge of technology.

MySql is database that I used to store data about user transactions and some partial information of documents they upload.

Application or project is completely build on laravel.

Some packages are used to support modern authentication and security like laravel/ui package is used for authentication scaffolding.

Most of project used VueJs, Angular or react for making things live or you can say reactive but in this project there is a power full and modern front end and that is laravel-livewire technology to directly interact with PHP or Laravel functions and variable from javascript.

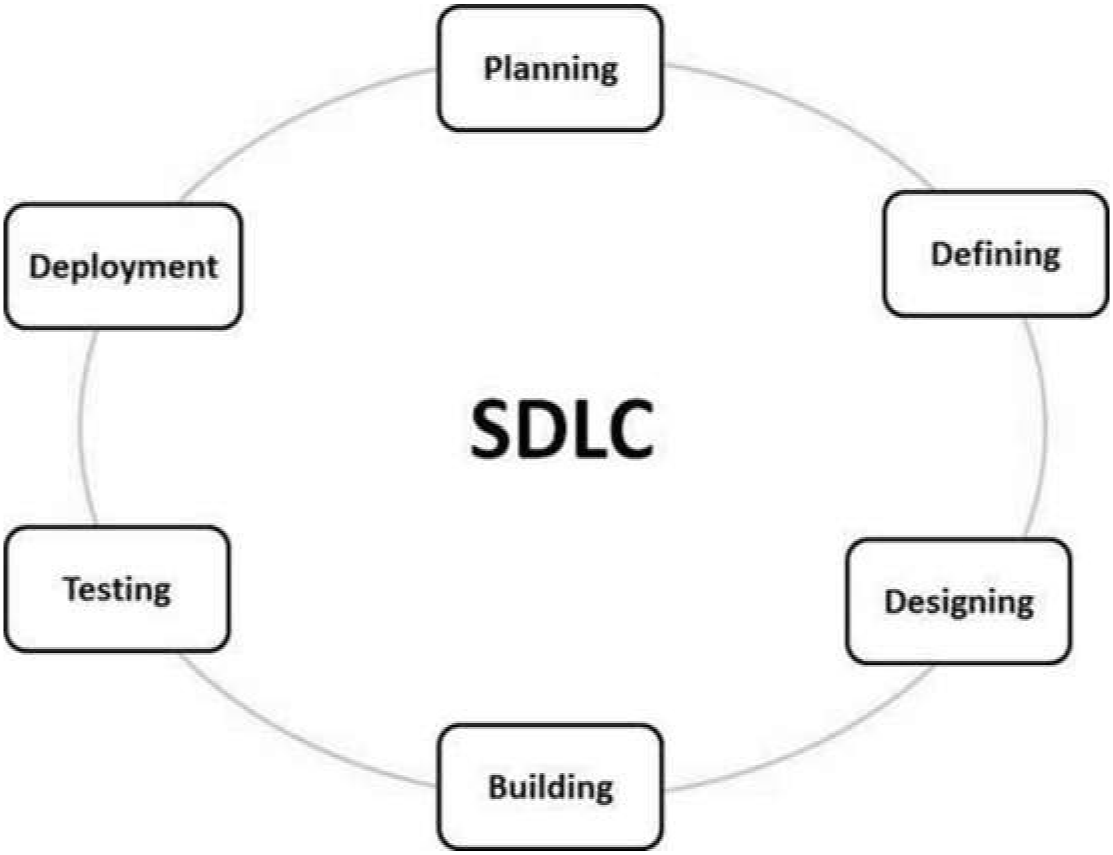
Everything in this project is using livewire technique to make project running faster and smoothly.

Project Profile

Project Title	: Somanath Agro Billing Systetm
Operating System	: windows 7,8,8.1,10,ubuntu.
Front End	: HTML,PHP,CSS,Livewire, Bootstrap, ,javascript.
Back End	: MySql
Documentation Tools	: Microsoft World
Submitted By	: Vataliya Yatharth
Submitted to	: Department of computer Science

System Development Life Cycle

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.



Feasibility Study

We try to make it more easy to use by end user or client who are going to use it after completion of project. Because of current globe that more people are irritate to use complex UI and UX we try to make it very normal to understand and use.

- It is feasible for most of the users as it only need basic knowledge of computer operations or mobile operations.
- It does not cost extra in terms of hardware to the users or institution.
- Basic instructions are enough to operate this web application.
- The system is technically feasible as the technical requirements are very minimal, that is it only needs basic hardware and software system to operate on.
- Also it has operation feasibility as software is easily configured with hardware because this system only needs browser which is built in or easily installed.
- The functionalities in this project are designed such that the novice users can also easily understand.

Requirement analysis

We meet the some professional at the college, at big industries and ask them about there work problem in traditional way like storing their files and data into registers or in books and manage each records like traditional way. So we define every need of document management to make it very easy to use using modern technology and make it better ui experience. We also meet some local and low standard shops to understand the small needs for low level usage of our project and for covering up the more market in current trend.

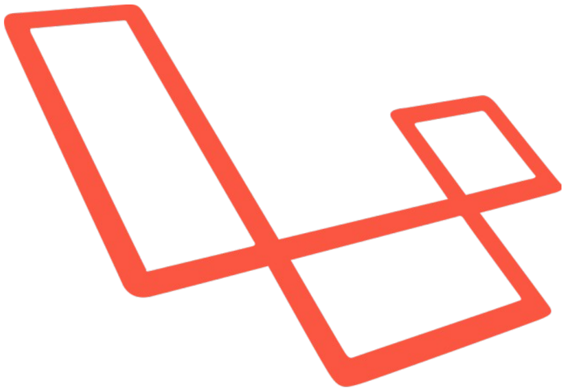
About Tools And Technology Used

Introduction to PHP:

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of **PHP** way back in 1994. **PHP** is a recursive acronym for "**PHP**: Hypertext Preprocessor". **PHP** is a server side scripting language that is embedded in HTML.

We use Laravel a popular framework of php so let's see about the laravel a little bit.

Laravel



Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.

Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development.

If you are familiar with Core PHP and Advanced PHP, Laravel will make your task easier. It saves a lot time if you are planning to develop a website from scratch. Moreover, a website built in Laravel is secure and prevents several web attacks.

❖ Advantages of Laravel

Laravel offers you the following advantages, when you are designing a web application based on it −

- The web application becomes more scalable, owing to the Laravel framework.
- Considerable time is saved in designing the web application, since Laravel reuses the components from other framework in developing web application.
- It includes namespaces and interfaces, thus helps to organize and manage resources.

Laravel offers the following key features which make it an ideal choice for designing web applications −

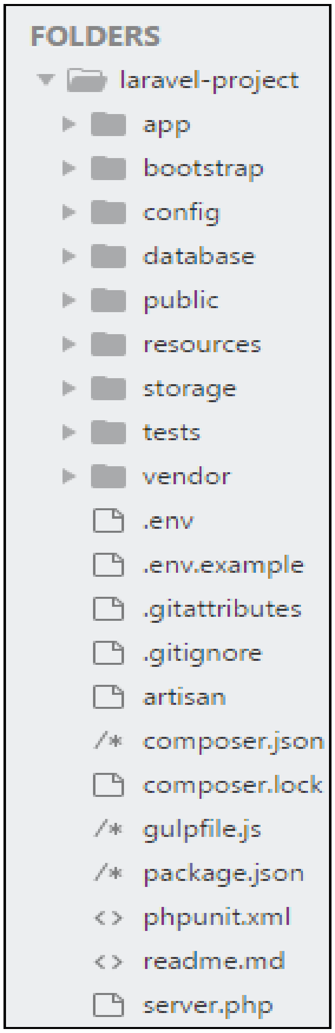
- **Modularity**
- **Testability**
- **Routing**
- **Configuration Management**
- **Query Builder and ORM**
- **Schema Builder**
- **Template Engine**
- **E-mail**

- **Authentication**
- **Redis**
- **Queues**
- **Event and Command Bus**

❖ Laravel - Application Structure

The application structure in Laravel is basically the structure of folders, sub-folders and files included in a project. Once we create a project in Laravel, we get an overview of the application structure as shown in the image here.

The snapshot shown here refers to the root folder of Laravel namely **laravel-project**. It includes various sub-folders and files. The analysis of folders and files, along with their functional aspects is given below −



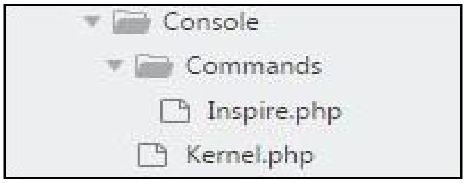
```
FOLDERS
▼ laravel-project
  ▶ app
  ▶ bootstrap
  ▶ config
  ▶ database
  ▶ public
  ▶ resources
  ▶ storage
  ▶ tests
  ▶ vendor
    .env
    .env.example
    .gitattributes
    .gitignore
    artisan
    composer.json
    composer.lock
    gulpfile.js
    package.json
    phpunit.xml
    readme.md
    server.php
```

➢ App

It is the application folder and includes the entire source code of the project. It contains events, exceptions and middleware declaration. The app folder comprises various sub folders as explained below −

- **Console**

Console includes the artisan commands necessary for Laravel.



Console
Commands
Inspire.php
Kernel.php

- **Events**

This folder includes all the events for the project. Events are used to trigger activities, raise errors or necessary validations and provide greater flexibility.

Events
Event.php

- **Exceptions**

This folder contains all the methods needed to handle exceptions. It also contains the file **handle.php** that handles all the exceptions.

- **Http**

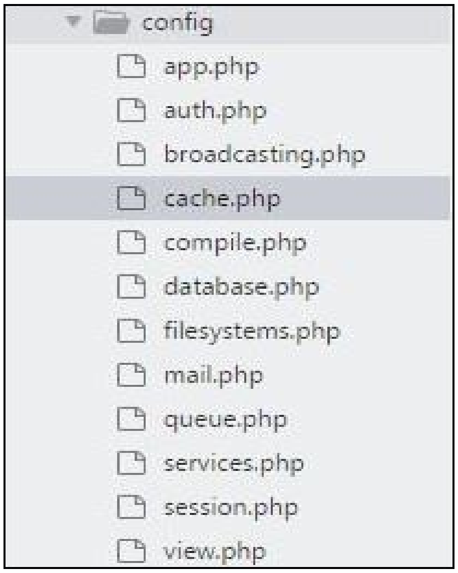
The **Http** folder has sub-folders for controllers, middleware and application requests. As Laravel follows the MVC design pattern, this folder includes model, controllers and views defined for the specific directories.

➢ Bootstrap

This folder encloses all the application bootstrap scripts. It contains a sub-folder namely **cache**, which includes all the files associated for caching a web application. You can also find the file **app.php**, which initializes the scripts necessary for bootstrap.

- **Config**

The **config** folder includes various configurations and associated parameters required for the smooth functioning of a Laravel application. Various files included within the config folder are as shown in the image here. The filenames work as per the functionality associated with them.



config
app.php
auth.php
broadcasting.php
cache.php
compile.php
database.php
filesystems.php
mail.php
queue.php
services.php
session.php
view.php

- **Database**

As the name suggests, this directory includes various parameters for database functionalities. It includes three sub-directories as given below −

- **Seeds** − This contains the classes used for unit testing database.
- **Migrations** − This folder helps in queries for migrating the database used in the web application.
- **Factories** − This folder is used to generate large number of data records.

- **Public**

It is the root folder which helps in initializing the Laravel application. It includes the following files and folders −

- **.htaccess** − This file gives the server configuration.
- **javascript and css** − These files are considered as assets.
- **index.php** − This file is required for the initialization of a web application.

➢ Resources

Resources directory contains the files which enhances your web application. The sub-folders included in this directory and their purpose is explained below −

- **assets** − The assets folder include files such as LESS and SCSS, that are required for styling the web application.
- **lang** − This folder includes configuration for localization or internalization.
- **views** − Views are the HTML files or templates which interact with end users and play a primary role in MVC architecture.

➢ Storage

This is the folder that stores all the logs and necessary files which are needed frequently when a Laravel project is running. The sub-folders included in this directory and their purpose is given below −

- **app** − This folder contains the files that are called in succession.
- **framework** − It contains sessions, cache and views which are called frequently.
- **Logs** − All exceptions and error logs are tracked in this sub folder.

- **Tests**

All the unit test cases are included in this directory. The naming convention for naming test case classes is **camel_case** and follows the convention as per the functionality of the class.

- **Vendor**

Laravel is completely based on Composer dependencies, for example to install Laravel setup or to include third party libraries, etc. The Vendor folder includes all the composer dependencies.

MYSQL

MySQL is an open source relational database management system (RDBMS).

The MySQL server software itself and the client libraries use dual-licensing distribution.

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server.

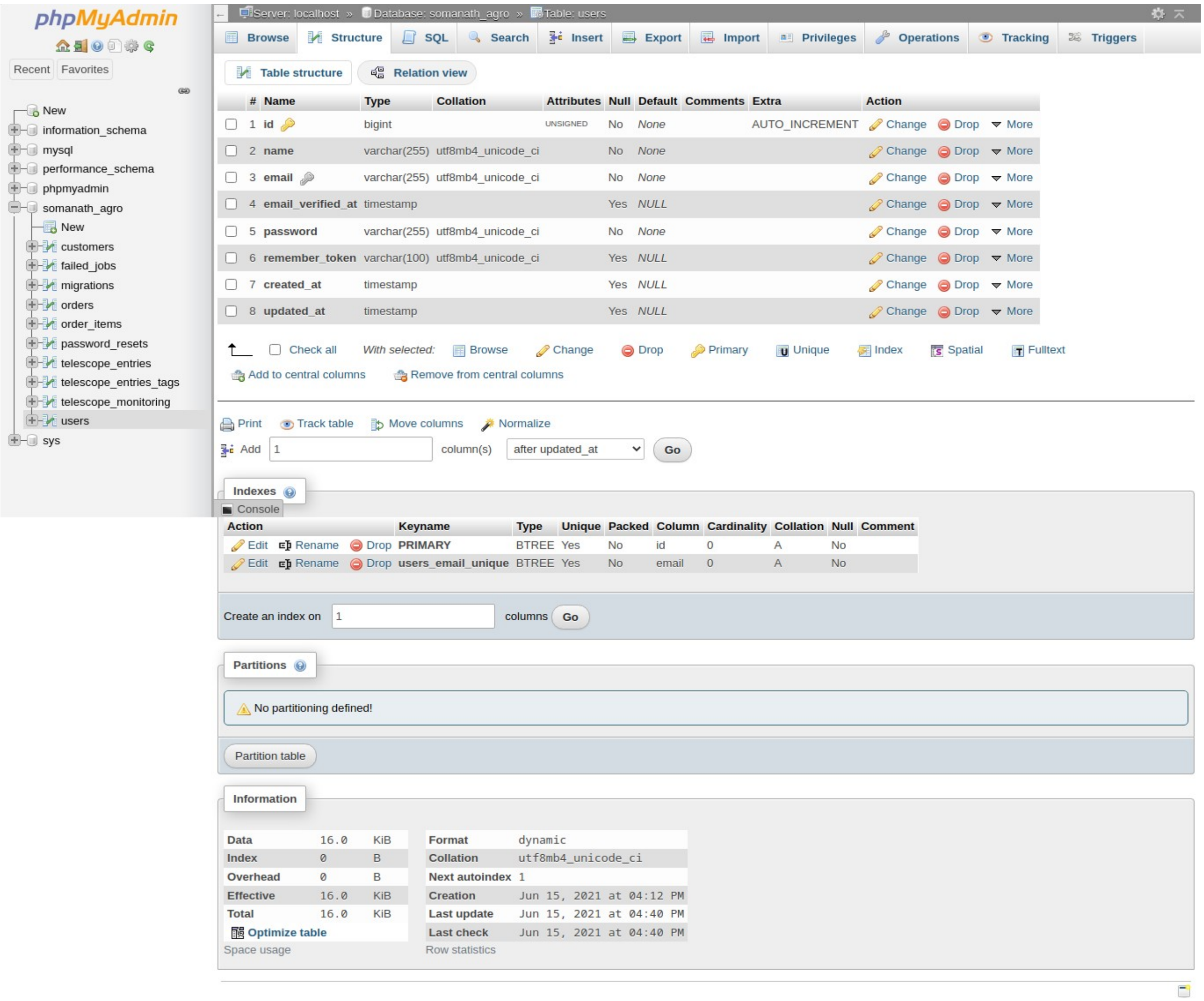
Major features as available in MySQL 5.6:

- Cross-platform support
- Stored procedures, using a procedural language that closely adheres to SQL/PSM[75]
- Triggers
- Cursors
- Updatable views
- Online Data Definition Language (DDL) when using the InnoDB Storage Engine.

Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.

DATA BASE DESIGN

Table Name:- users



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	bigint		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**name**	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	3	**email**	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	4	**email_verified_at**	timestamp			Yes	*NULL*			Change Drop More
□	5	**password**	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	6	**remember_token**	varchar(100)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
□	7	**created_at**	timestamp			Yes	*NULL*			Change Drop More
□	8	**updated_at**	timestamp			Yes	*NULL*			Change Drop More

Add 1 column(s) after updated_at Go

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	id	0	A	No	
Edit Rename Drop	**users_email_unique**	BTREE	Yes	No	email	0	A	No	

Create an index on 1 columns Go

No partitioning defined!

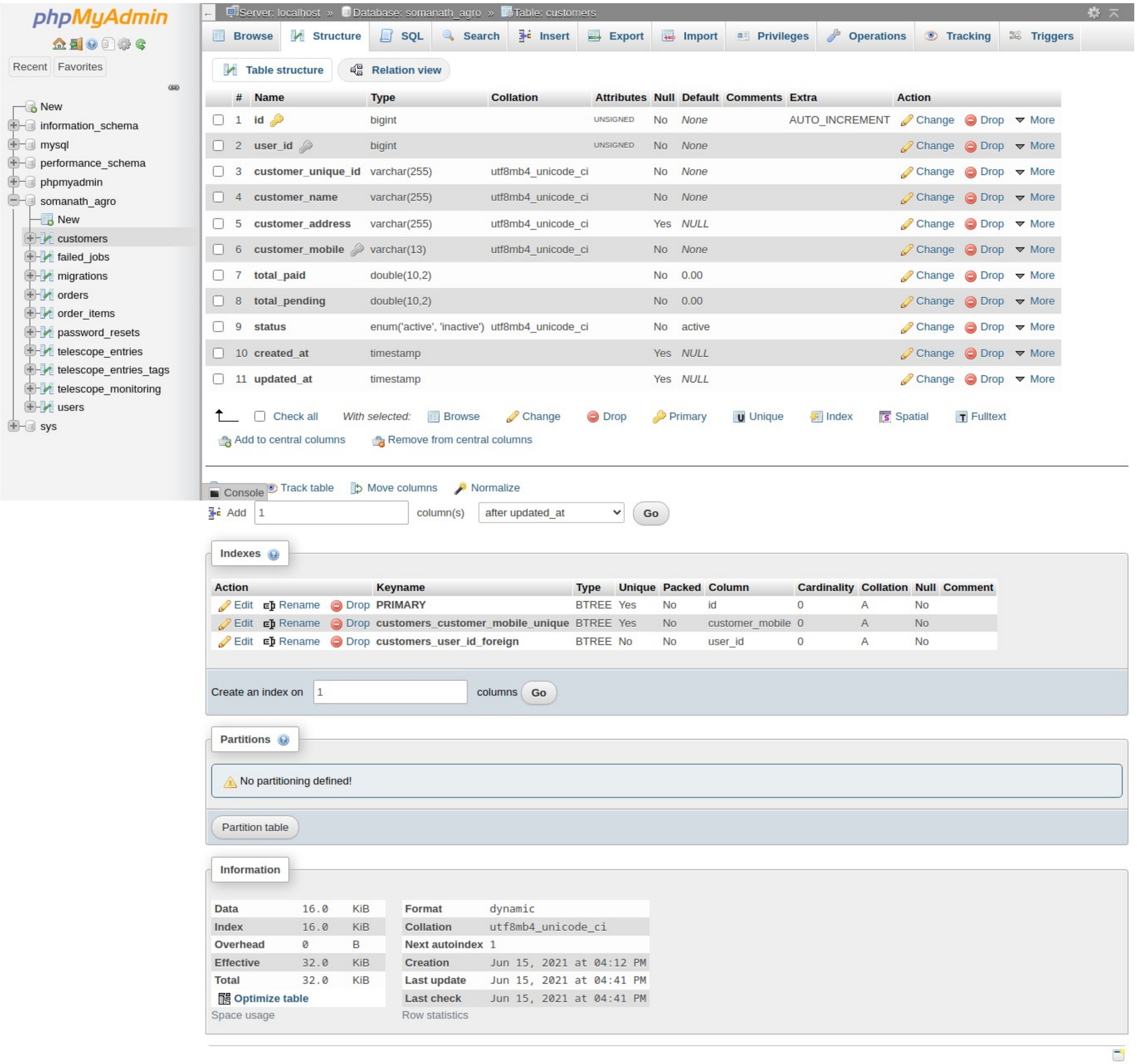
Data	16.0	KiB
Index	0	B
Overhead	0	B
Effective	16.0	KiB
Total	16.0	KiB
Optimize table		

Space usage

Format	dynamic
Collation	utf8mb4_unicode_ci
Next autoindex	1
Creation	Jun 15, 2021 at 04:12 PM
Last update	Jun 15, 2021 at 04:40 PM
Last check	Jun 15, 2021 at 04:40 PM

Row statistics

Table Name:- customers



☐	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id**	bigint		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**user_id**	bigint		UNSIGNED	No	*None*			Change Drop More
☐	3	**customer_unique_id**	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	4	**customer_name**	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	5	**customer_address**	varchar(255)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
☐	6	**customer_mobile**	varchar(13)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	7	**total_paid**	double(10,2)			No	0.00			Change Drop More
☐	8	**total_pending**	double(10,2)			No	0.00			Change Drop More
☐	9	**status**	enum('active', 'inactive')	utf8mb4_unicode_ci		No	active			Change Drop More
☐	10	**created_at**	timestamp			Yes	*NULL*			Change Drop More
☐	11	**updated_at**	timestamp			Yes	*NULL*			Change Drop More

Add 1 column(s) after updated_at Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	id	0	A	No	
Edit Rename Drop	**customers_customer_mobile_unique**	BTREE	Yes	No	customer_mobile	0	A	No	
Edit Rename Drop	**customers_user_id_foreign**	BTREE	No	No	user_id	0	A	No	

Create an index on 1 columns Go

Partitions

No partitioning defined!

Partition table

Information

Data	16.0	KiB
Index	16.0	KiB
Overhead	0	B
Effective	32.0	KiB
Total	32.0	KiB

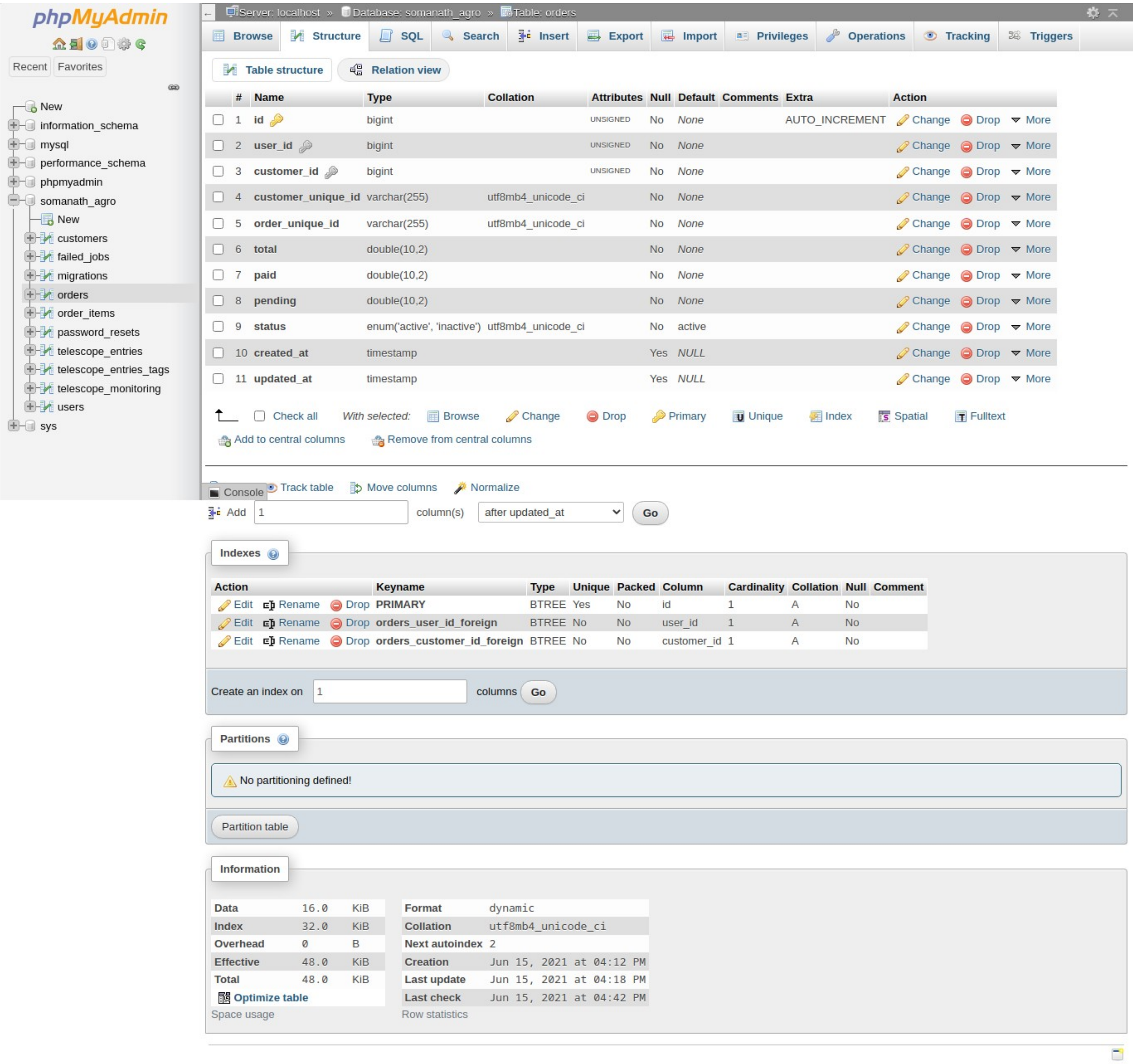
Optimize table

Space usage

Format	dynamic
Collation	utf8mb4_unicode_ci
Next autoindex	1
Creation	Jun 15, 2021 at 04:12 PM
Last update	Jun 15, 2021 at 04:41 PM
Last check	Jun 15, 2021 at 04:41 PM

Row statistics

Table Name:- orders



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
2	user_id	bigint		UNSIGNED	No	*None*			Change Drop More
3	customer_id	bigint		UNSIGNED	No	*None*			Change Drop More
4	customer_unique_id	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
5	order_unique_id	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
6	total	double(10,2)			No	*None*			Change Drop More
7	paid	double(10,2)			No	*None*			Change Drop More
8	pending	double(10,2)			No	*None*			Change Drop More
9	status	enum('active', 'inactive')	utf8mb4_unicode_ci		No	active			Change Drop More
10	created_at	timestamp			Yes	*NULL*			Change Drop More
11	updated_at	timestamp			Yes	*NULL*			Change Drop More

Add 1 column(s) after updated_at Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	1	A	No	
Edit Rename Drop	orders_user_id_foreign	BTREE	No	No	user_id	1	A	No	
Edit Rename Drop	orders_customer_id_foreign	BTREE	No	No	customer_id	1	A	No	

Create an index on 1 columns Go

Partitions

No partitioning defined!

Partition table

Information

Data	16.0	KiB
Index	32.0	KiB
Overhead	0	B
Effective	48.0	KiB
Total	48.0	KiB

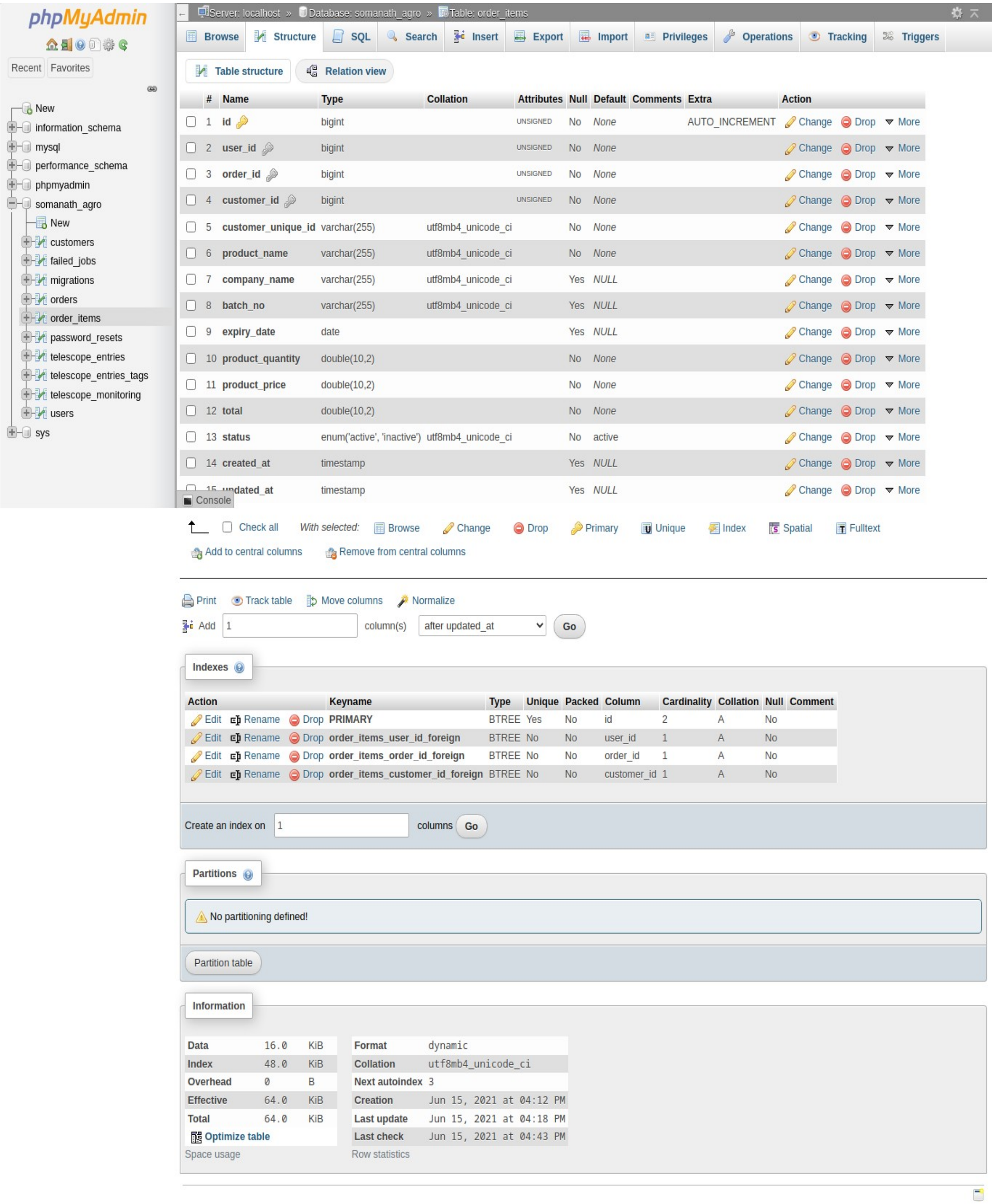
Optimize table

Space usage

Format	dynamic
Collation	utf8mb4_unicode_ci
Next autoindex	2
Creation	Jun 15, 2021 at 04:12 PM
Last update	Jun 15, 2021 at 04:18 PM
Last check	Jun 15, 2021 at 04:42 PM

Row statistics

Table Name :- order_items



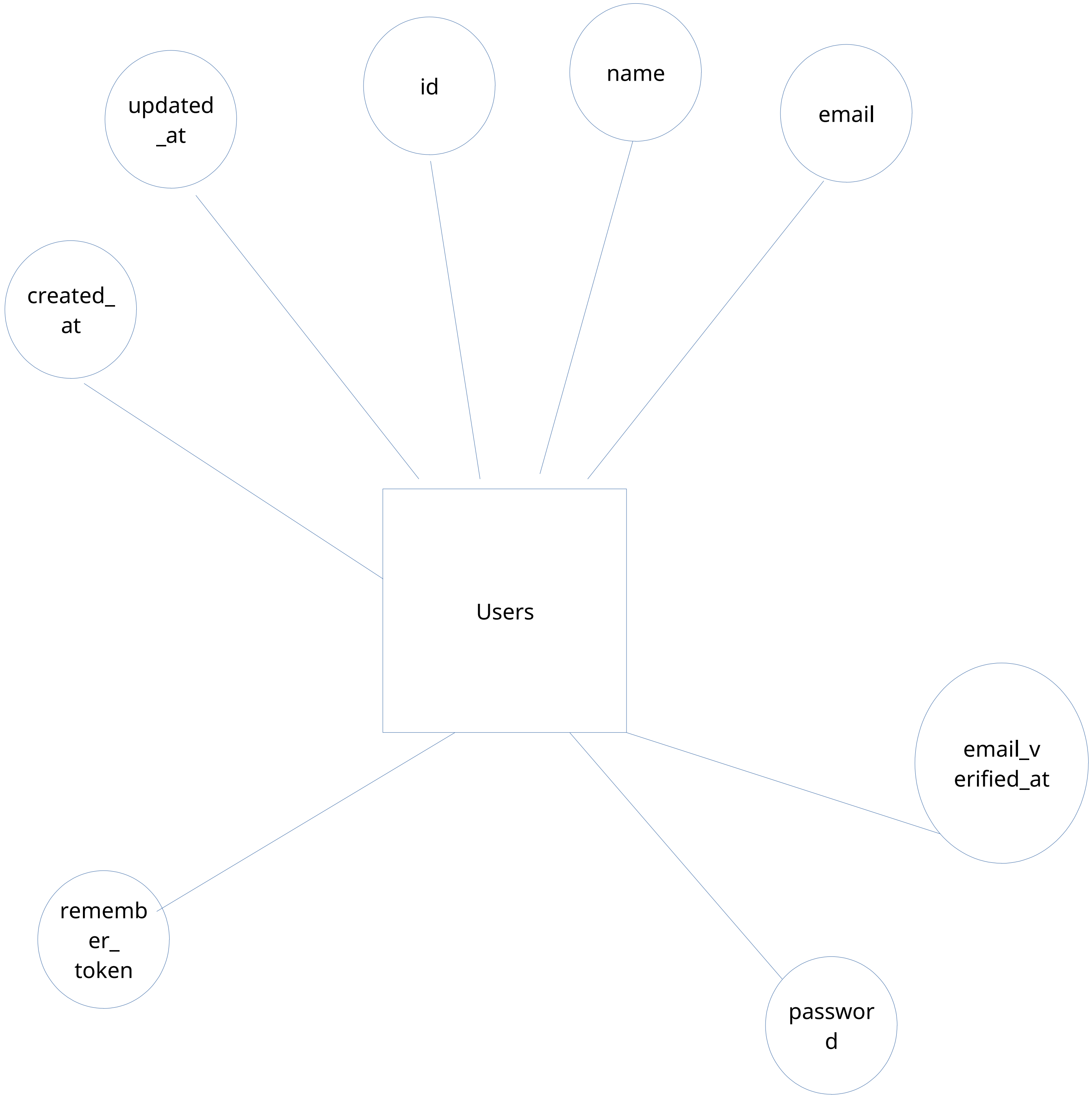
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
□	2	user_id	bigint		UNSIGNED	No	None			Change Drop More
□	3	order_id	bigint		UNSIGNED	No	None			Change Drop More
□	4	customer_id	bigint		UNSIGNED	No	None			Change Drop More
□	5	customer_unique_id	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
□	6	product_name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
□	7	company_name	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
□	8	batch_no	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
□	9	expiry_date	date			Yes	NULL			Change Drop More
□	10	product_quantity	double(10,2)			No	None			Change Drop More
□	11	product_price	double(10,2)			No	None			Change Drop More
□	12	total	double(10,2)			No	None			Change Drop More
□	13	status	enum('active', 'inactive')	utf8mb4_unicode_ci		No	active			Change Drop More
□	14	created_at	timestamp			Yes	NULL			Change Drop More
□	15	updated_at	timestamp			Yes	NULL			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	2	A	No	
Edit Rename Drop	order_items_user_id_foreign	BTREE	No	No	user_id	1	A	No	
Edit Rename Drop	order_items_order_id_foreign	BTREE	No	No	order_id	1	A	No	
Edit Rename Drop	order_items_customer_id_foreign	BTREE	No	No	customer_id	1	A	No	

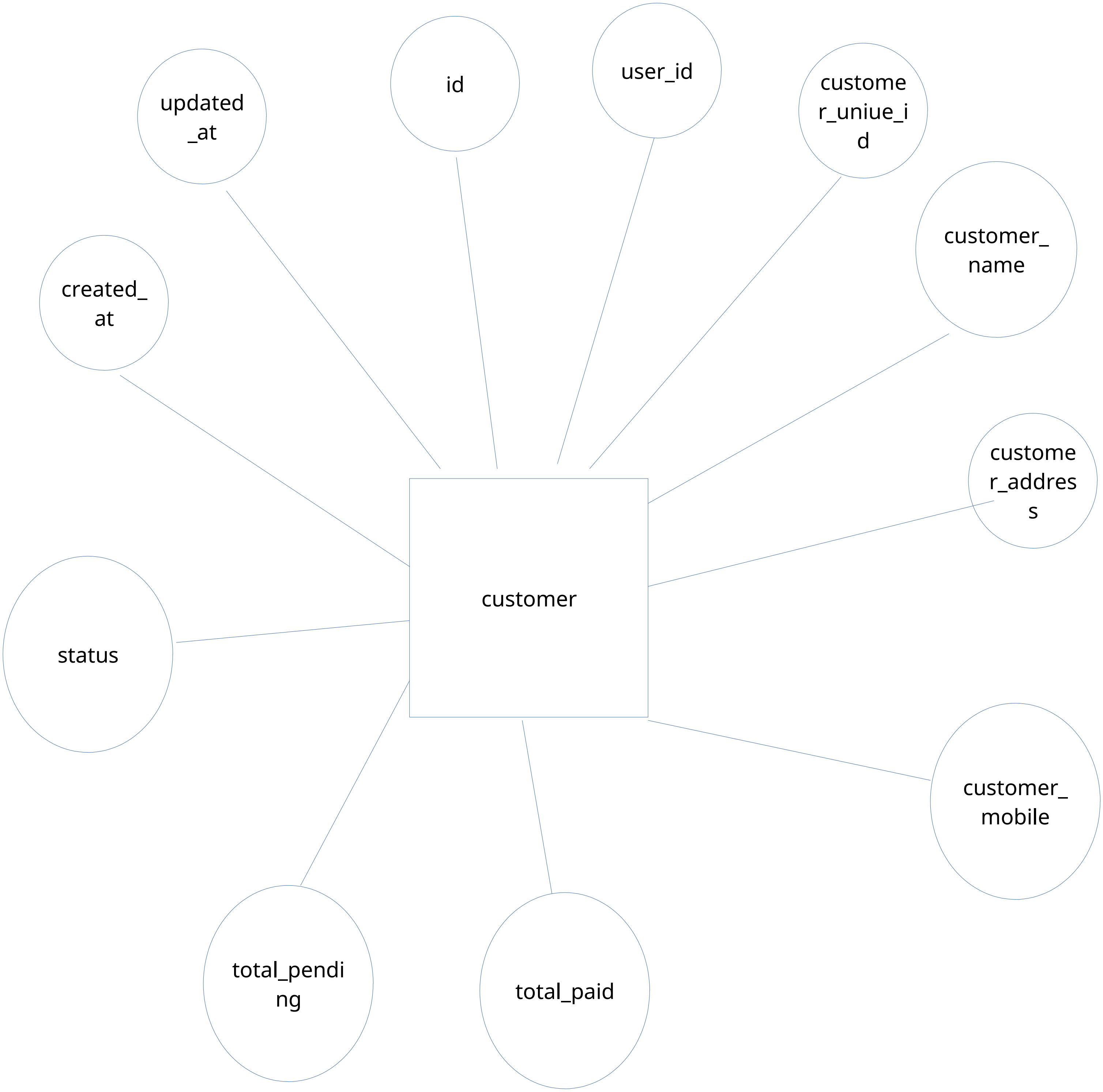
Data	16.0	KiB
Index	48.0	KiB
Overhead	0	B
Effective	64.0	KiB
Total	64.0	KiB

Format	dynamic
Collation	utf8mb4_unicode_ci
Next autoindex	3
Creation	Jun 15, 2021 at 04:12 PM
Last update	Jun 15, 2021 at 04:18 PM
Last check	Jun 15, 2021 at 04:43 PM

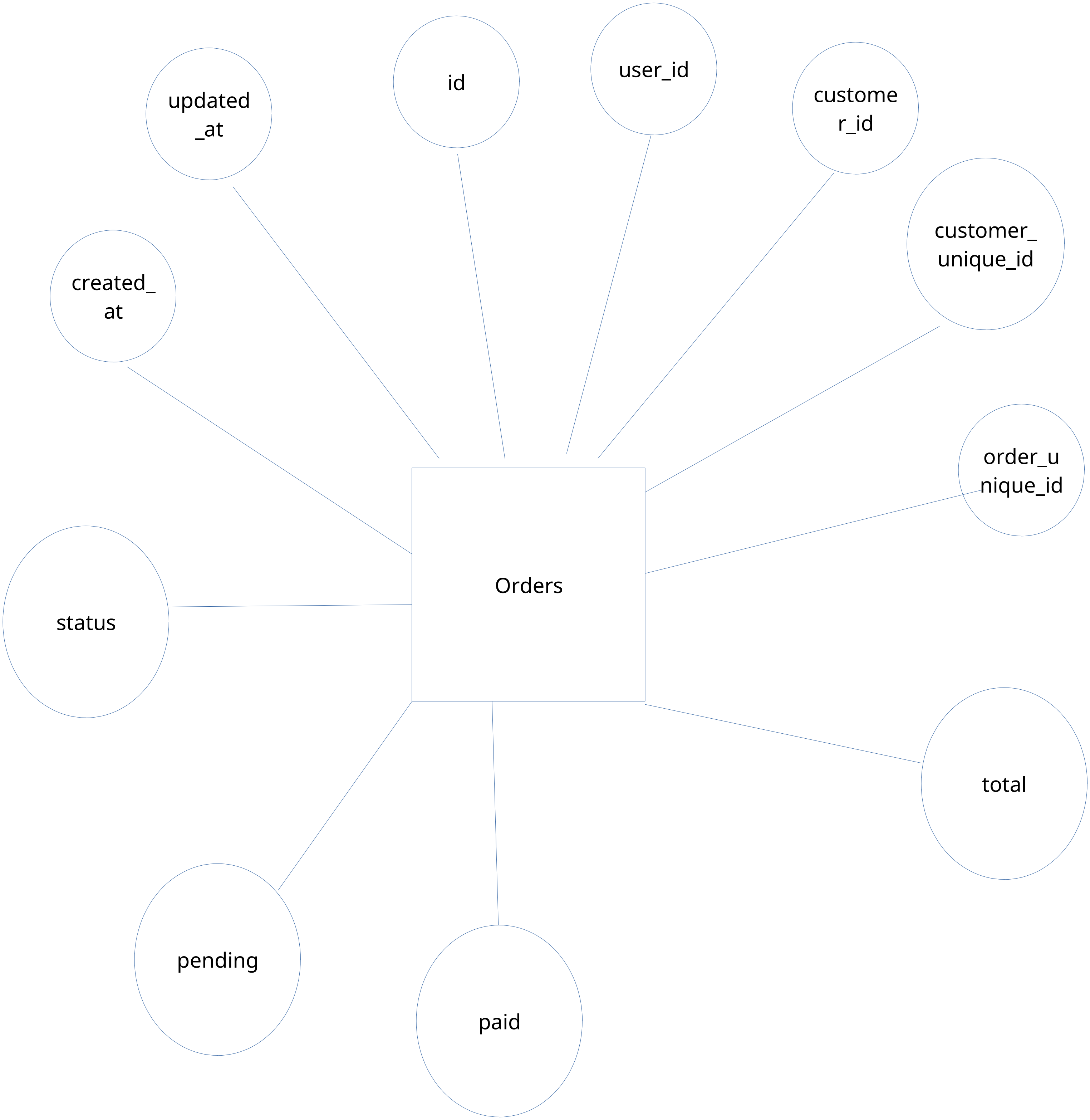
DATA FLOW DIGRAM



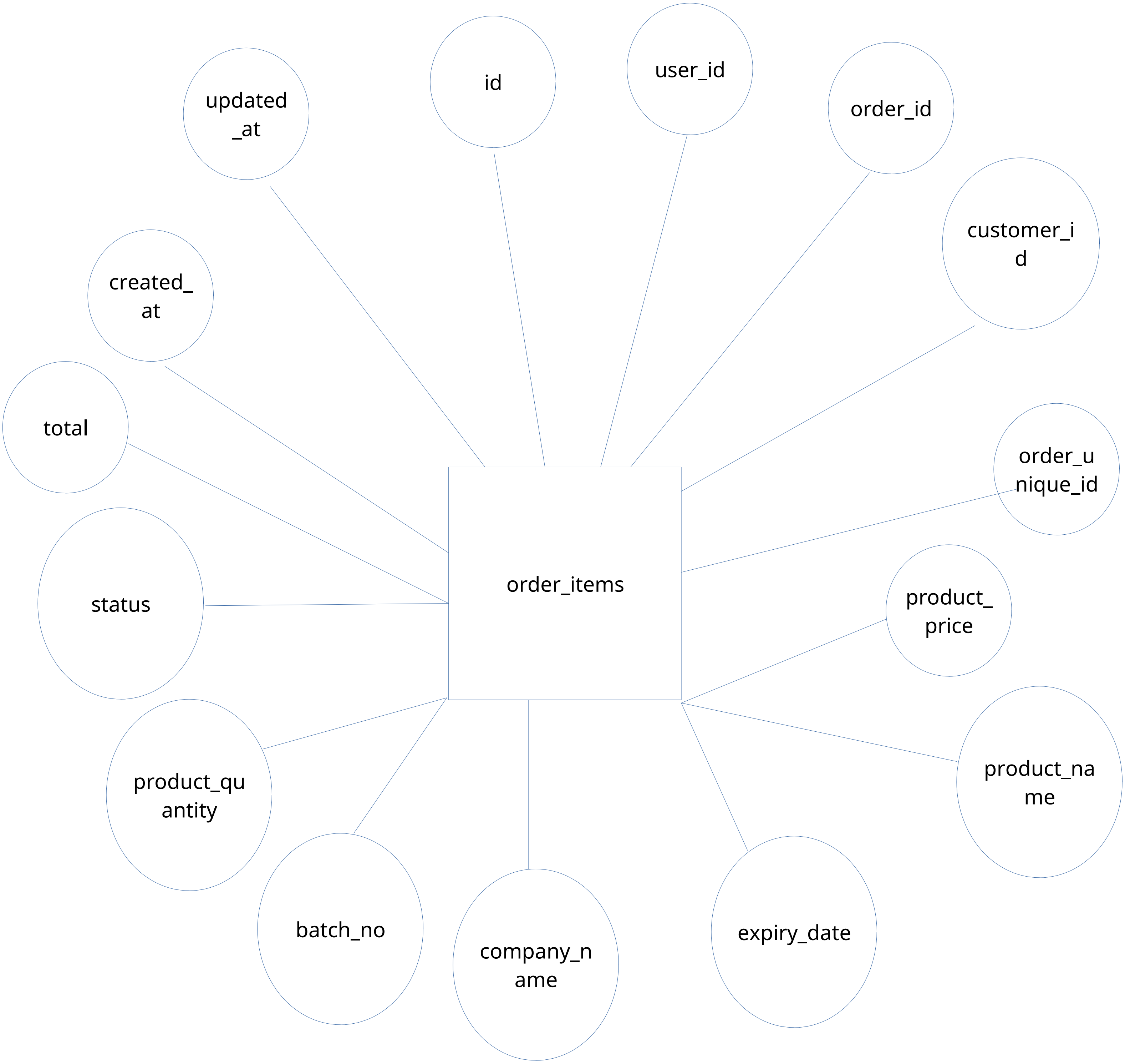
DATA FLOW DIGRAM



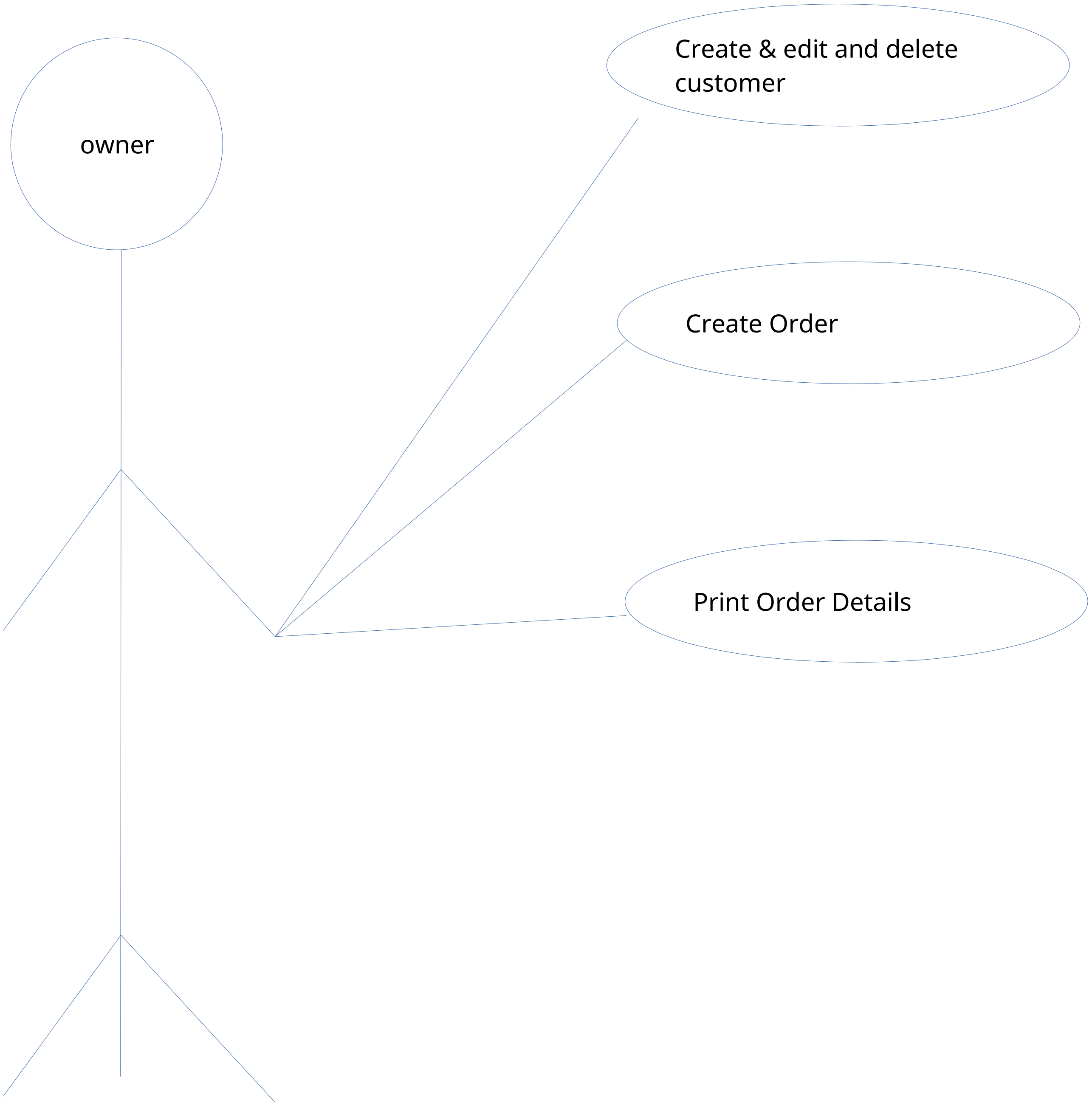
DATA FLOW DIGRAM



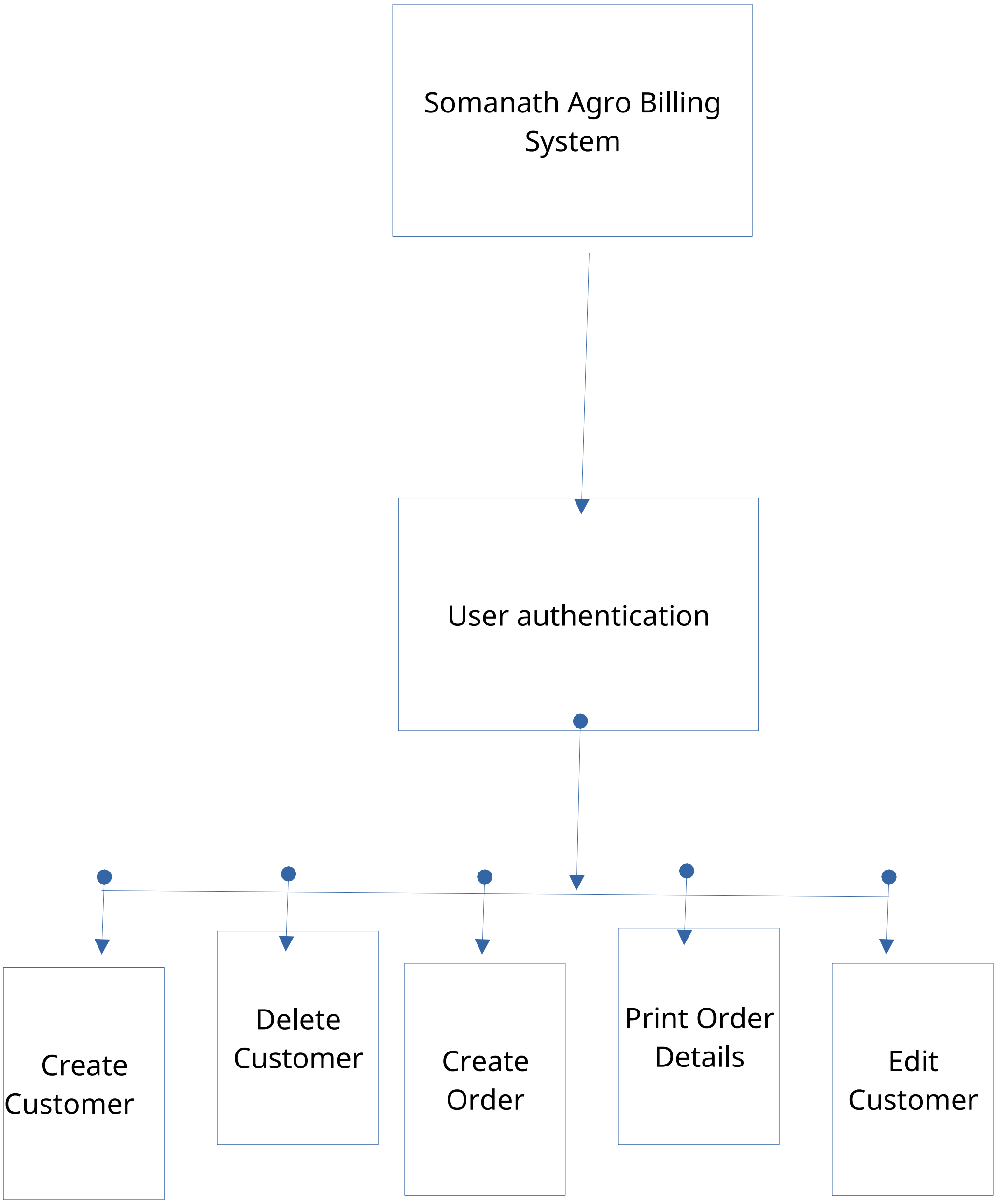
DATA FLOW DIGRAM



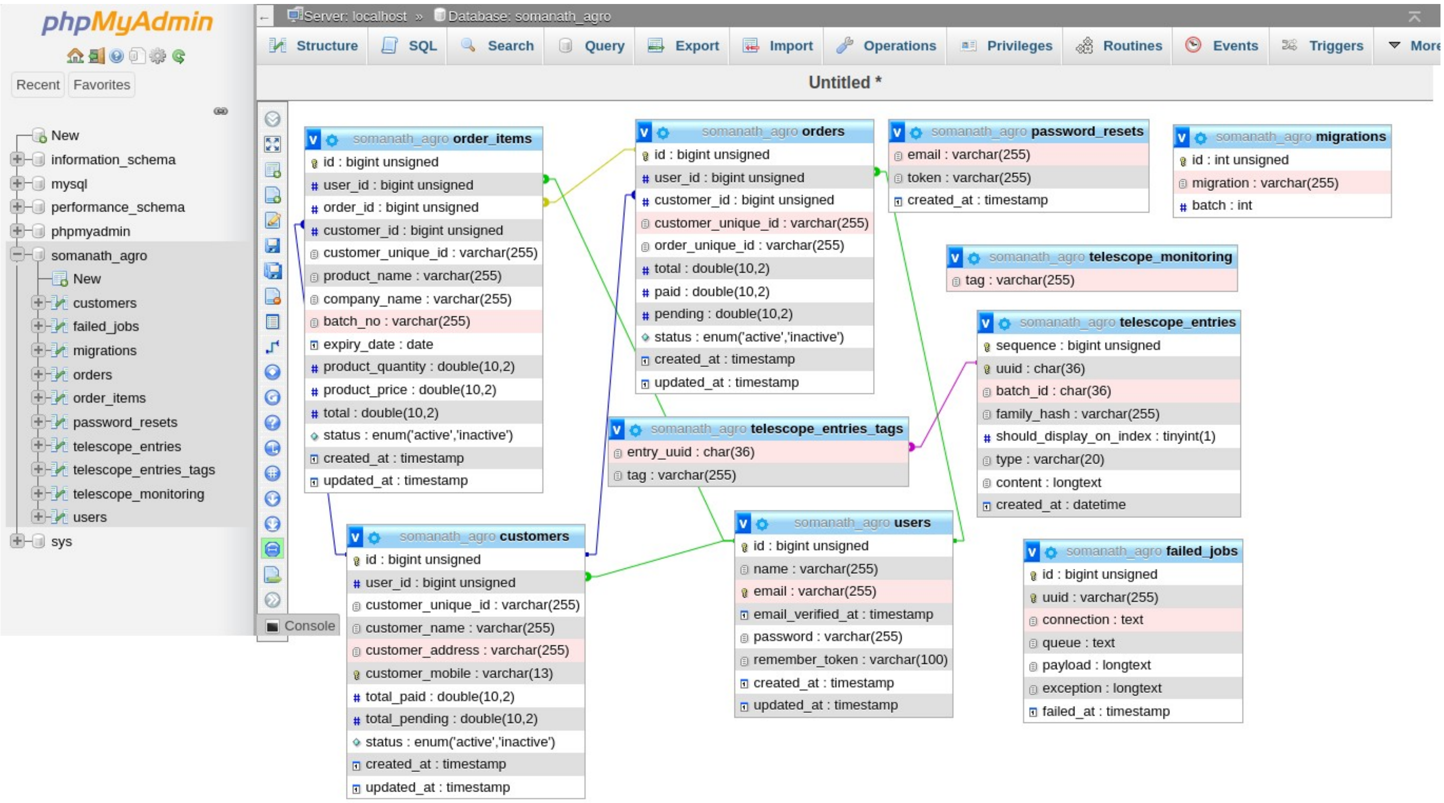
Use Case Diagram



Hipo Chart


Somanath Agro Billing System
User authentication
Create Customer
Delete Customer
Create Order
Print Order Details
Edit Customer

Database complete View



somanath_agro **order_items**
- id : bigint unsigned
- user_id : bigint unsigned
- order_id : bigint unsigned
- customer_id : bigint unsigned
- customer_unique_id : varchar(255)
- product_name : varchar(255)
- company_name : varchar(255)
- batch_no : varchar(255)
- expiry_date : date
- product_quantity : double(10,2)
- product_price : double(10,2)
- total : double(10,2)
- status : enum('active','inactive')
- created_at : timestamp
- updated_at : timestamp

somanath_agro **orders**
- id : bigint unsigned
- user_id : bigint unsigned
- customer_id : bigint unsigned
- customer_unique_id : varchar(255)
- order_unique_id : varchar(255)
- total : double(10,2)
- paid : double(10,2)
- pending : double(10,2)
- status : enum('active','inactive')
- created_at : timestamp
- updated_at : timestamp

somanath_agro **password_resets**
- email : varchar(255)
- token : varchar(255)
- created_at : timestamp

somanath_agro **migrations**
- id : int unsigned
- migration : varchar(255)
- batch : int

somanath_agro **telescope_monitoring**
- tag : varchar(255)

somanath_agro **telescope_entries**
- sequence : bigint unsigned
- uuid : char(36)
- batch_id : char(36)
- family_hash : varchar(255)
- should_display_on_index : tinyint(1)
- type : varchar(20)
- content : longtext
- created_at : datetime

somanath_agro **telescope_entries_tags**
- entry_uuid : char(36)
- tag : varchar(255)

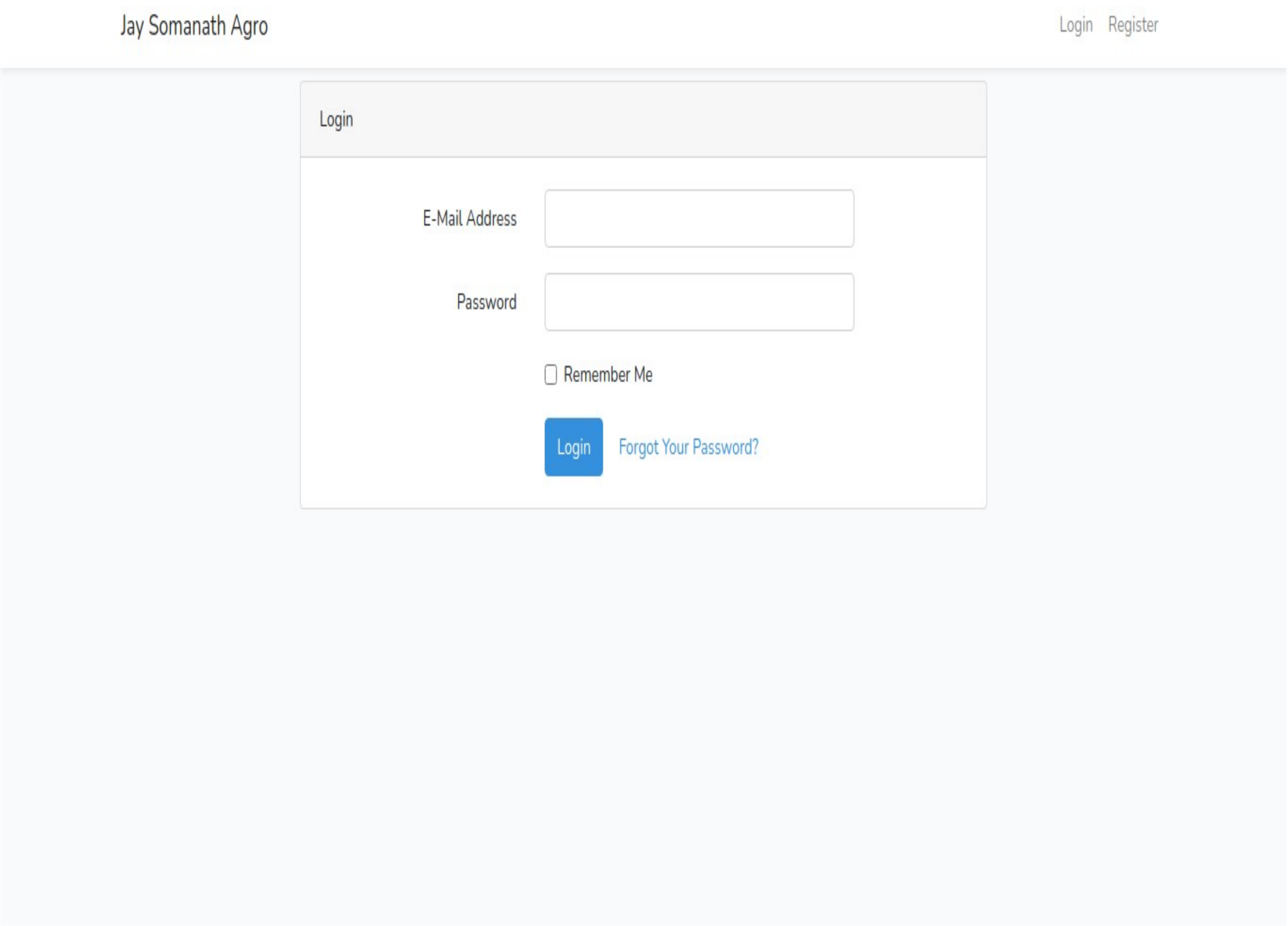
somanath_agro **customers**
- id : bigint unsigned
- user_id : bigint unsigned
- customer_unique_id : varchar(255)
- customer_name : varchar(255)
- customer_address : varchar(255)
- customer_mobile : varchar(13)
- total_paid : double(10,2)
- total_pending : double(10,2)
- status : enum('active','inactive')
- created_at : timestamp
- updated_at : timestamp

somanath_agro **users**
- id : bigint unsigned
- name : varchar(255)
- email : varchar(255)
- email_verified_at : timestamp
- password : varchar(255)
- remember_token : varchar(100)
- created_at : timestamp
- updated_at : timestamp

somanath_agro **failed_jobs**
- id : bigint unsigned
- uuid : varchar(255)
- connection : text
- queue : text
- payload : longtext
- exception : longtext
- failed_at : timestamp

Screen Shots

Login Screen



Jay Somanath Agro

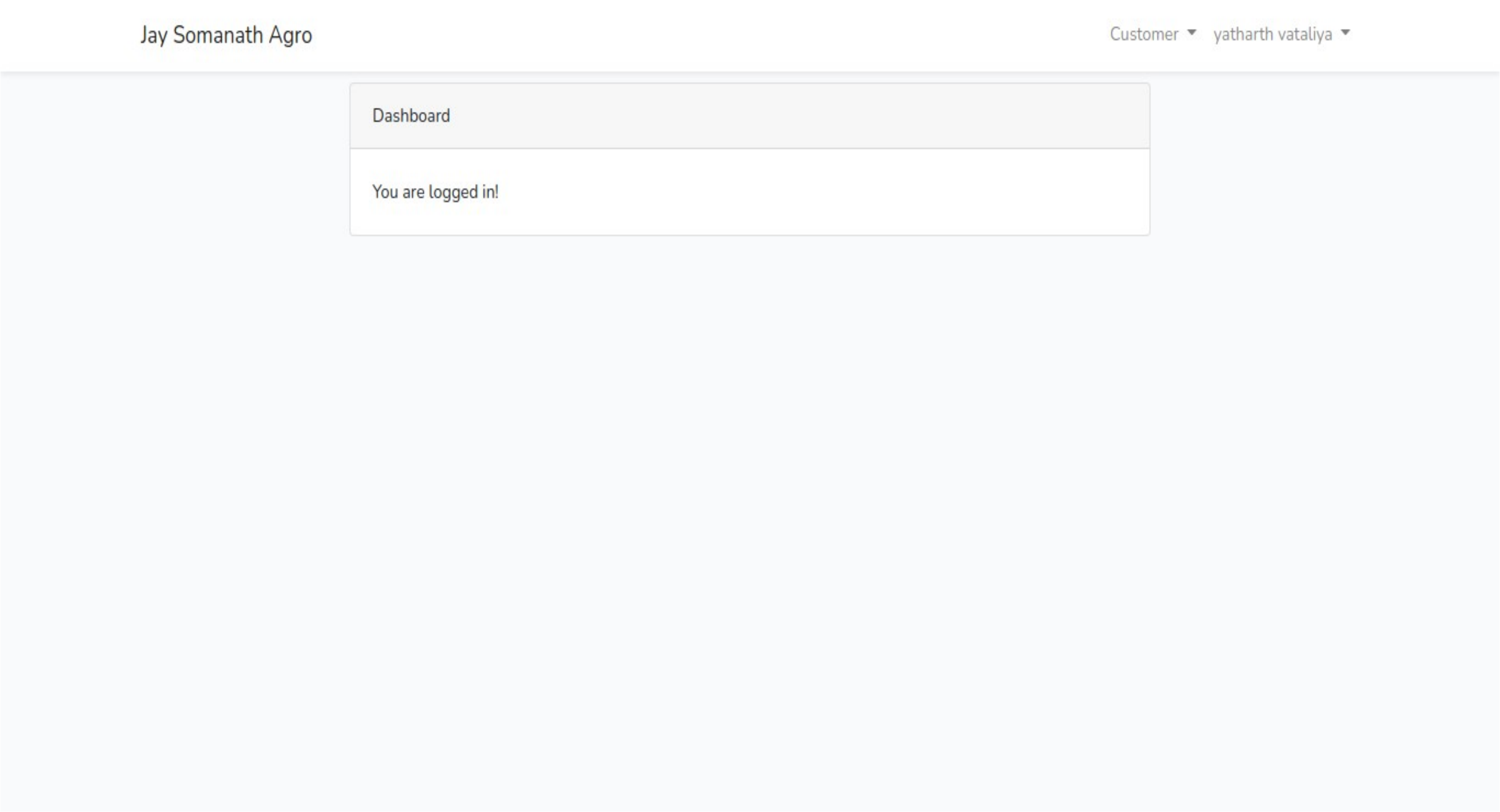
Login

E-Mail Address

Password

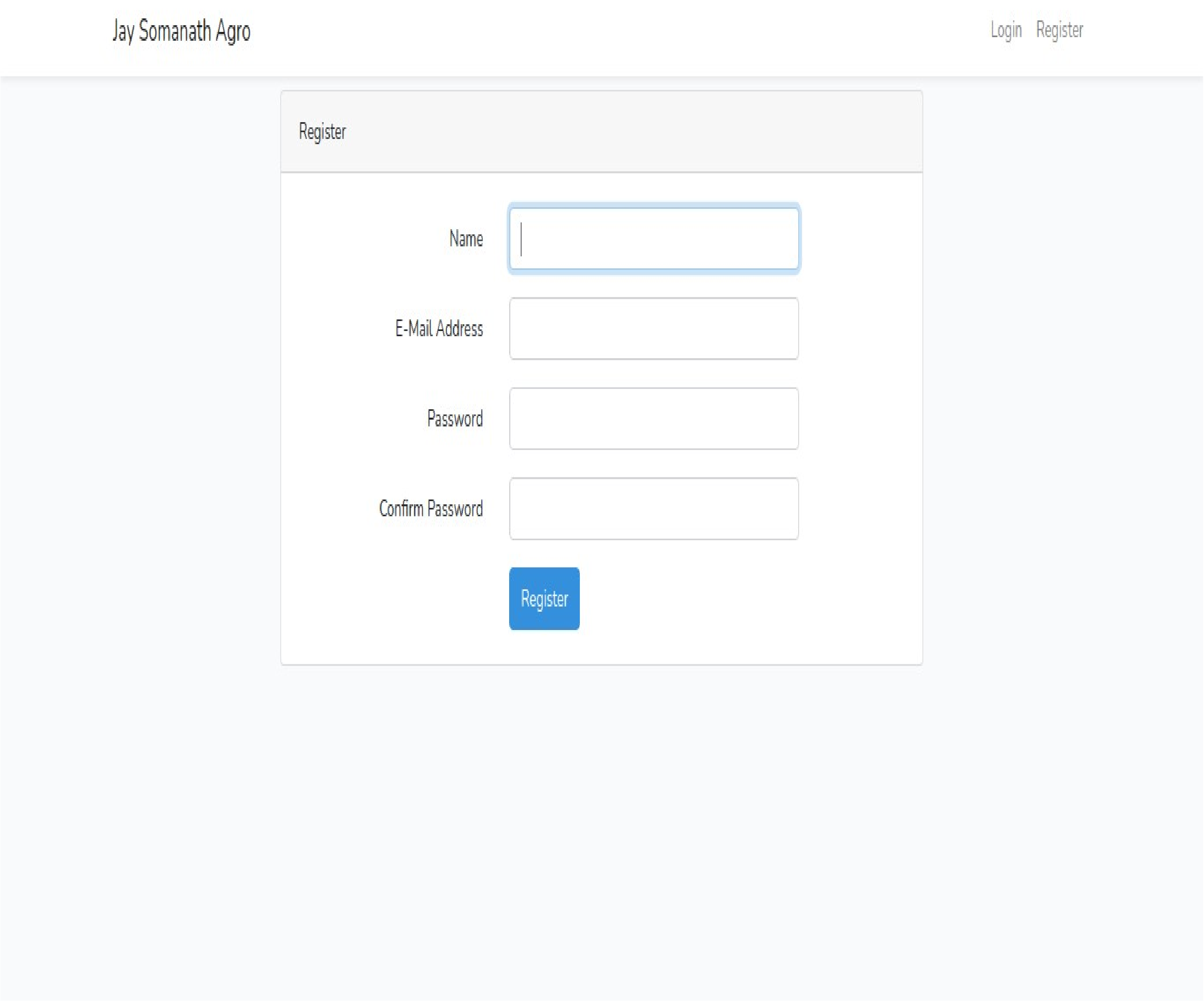
□ Remember Me

Dashboard



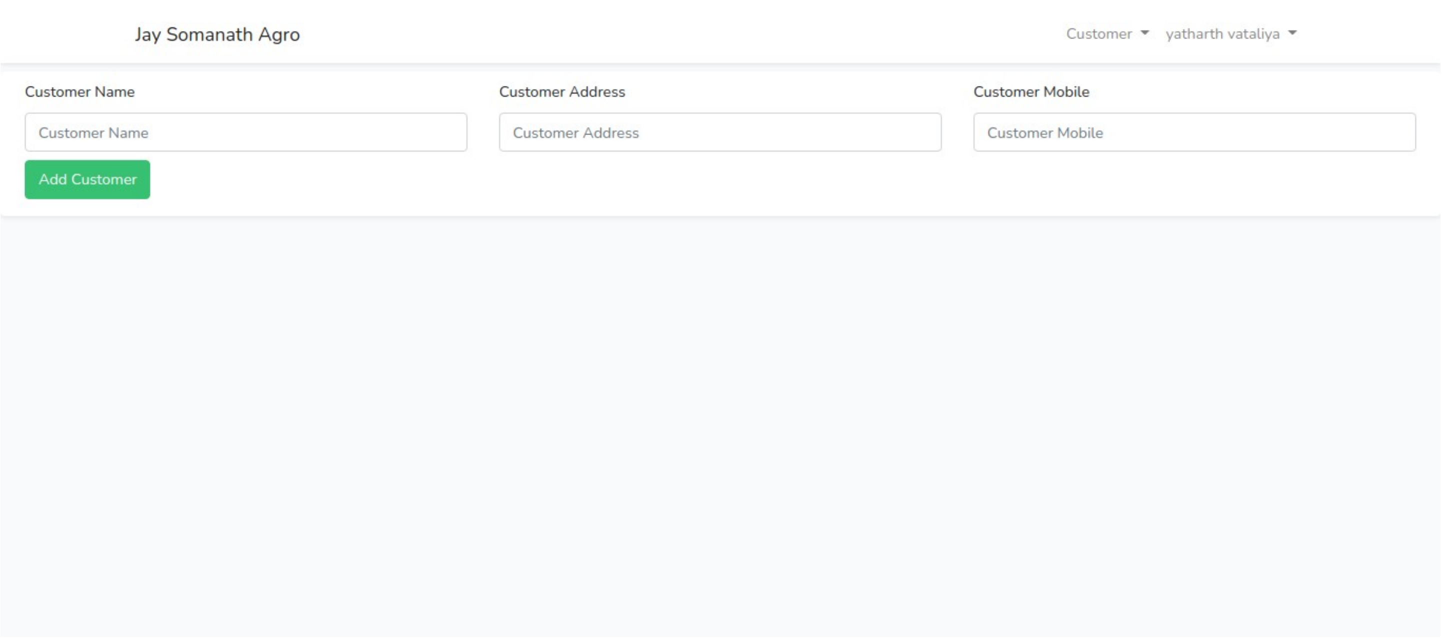
You are logged in!

Register

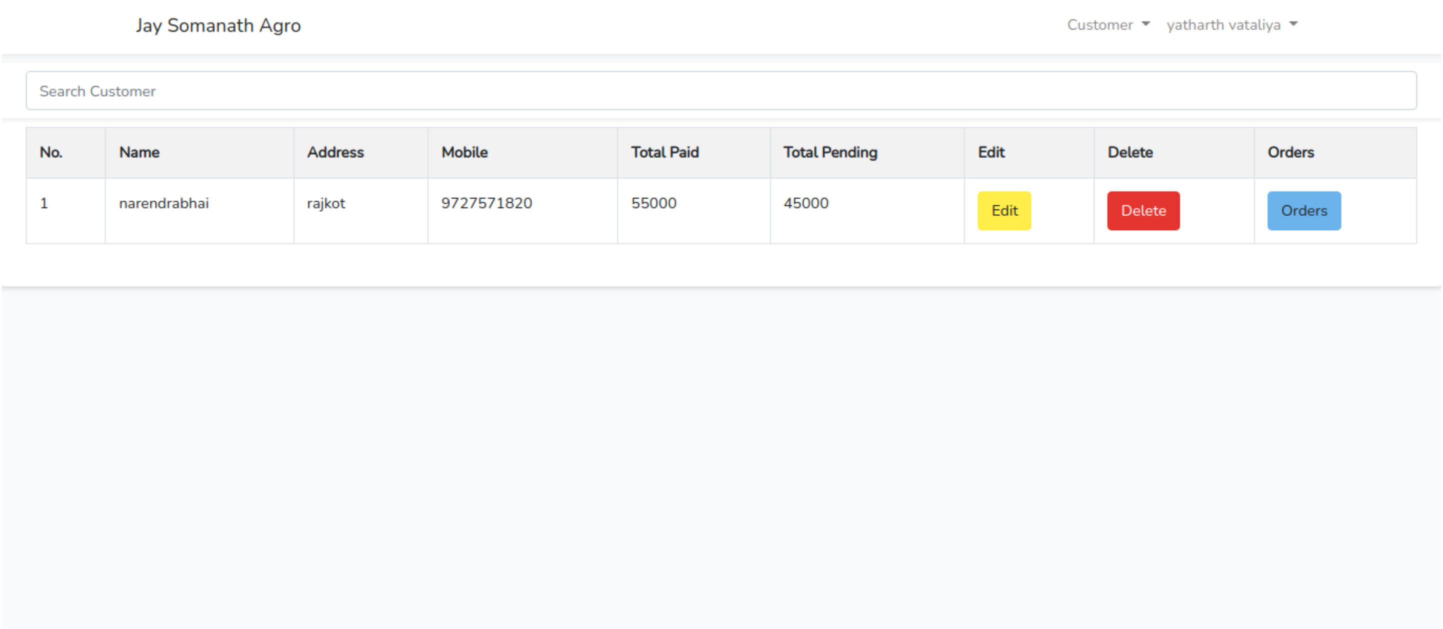


Jay Somanath Agro
Login
Register
Register
Name
E-Mail Address
Password
Confirm Password
Register

Create Customer

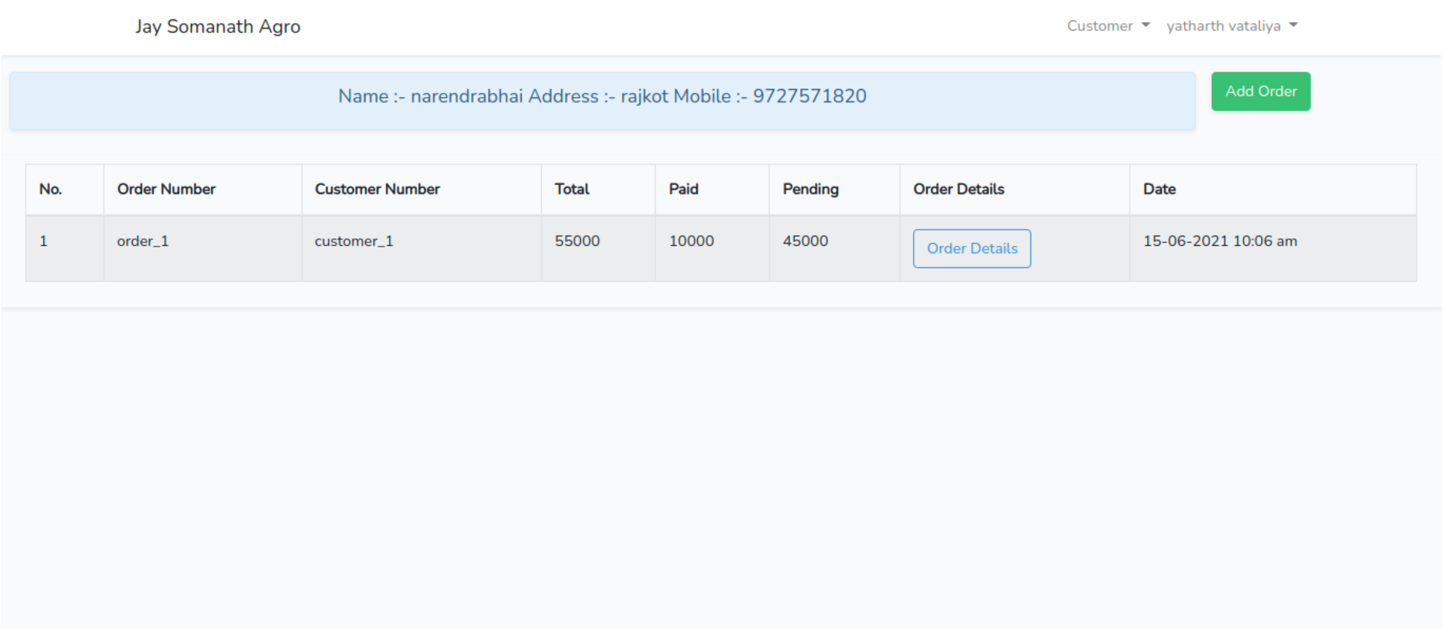


List Customers



No.	Name	Address	Mobile	Total Paid	Total Pending	Edit	Delete	Orders
1	narendrabhai	rajkot	9727571820	55000	45000	Edit	Delete	Orders

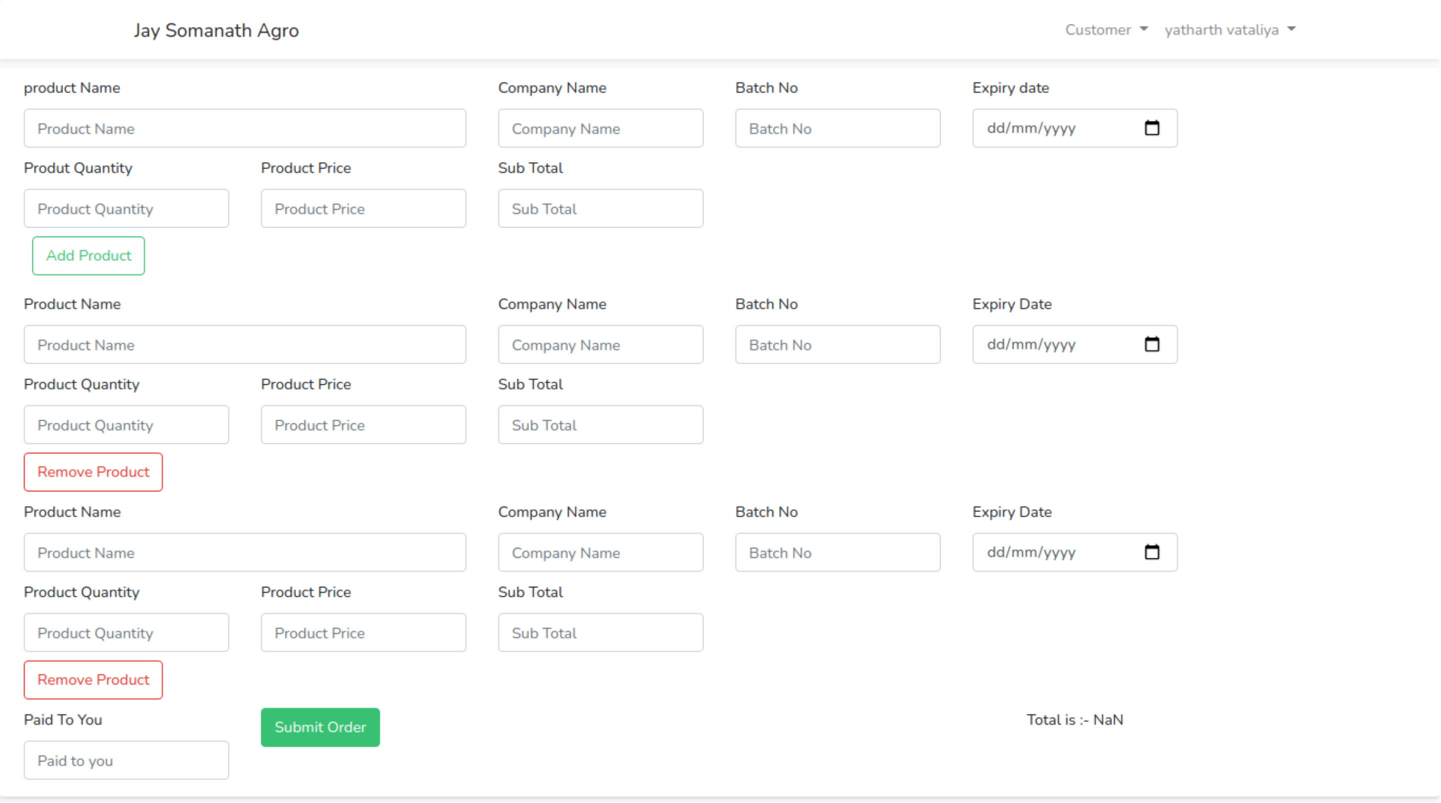
Customer Orders List



Name :- narendrabhai Address :- rajkot Mobile :- 9727571820

No.	Order Number	Customer Number	Total	Paid	Pending	Order Details	Date
1	order_1	customer_1	55000	10000	45000	Order Details	15-06-2021 10:06 am

Customer Order Create



Jay Somanath Agro

product Name: Product Name

Company Name: Company Name

Batch No: Batch No

Expiry date: dd/mm/yyyy

Produt Quantity: Product Quantity

Product Price: Product Price

Sub Total: Sub Total

Product Name: Product Name

Company Name: Company Name

Batch No: Batch No

Expiry Date: dd/mm/yyyy

Product Quantity: Product Quantity

Product Price: Product Price

Sub Total: Sub Total

Product Name: Product Name

Company Name: Company Name

Batch No: Batch No

Expiry Date: dd/mm/yyyy

Product Quantity: Product Quantity

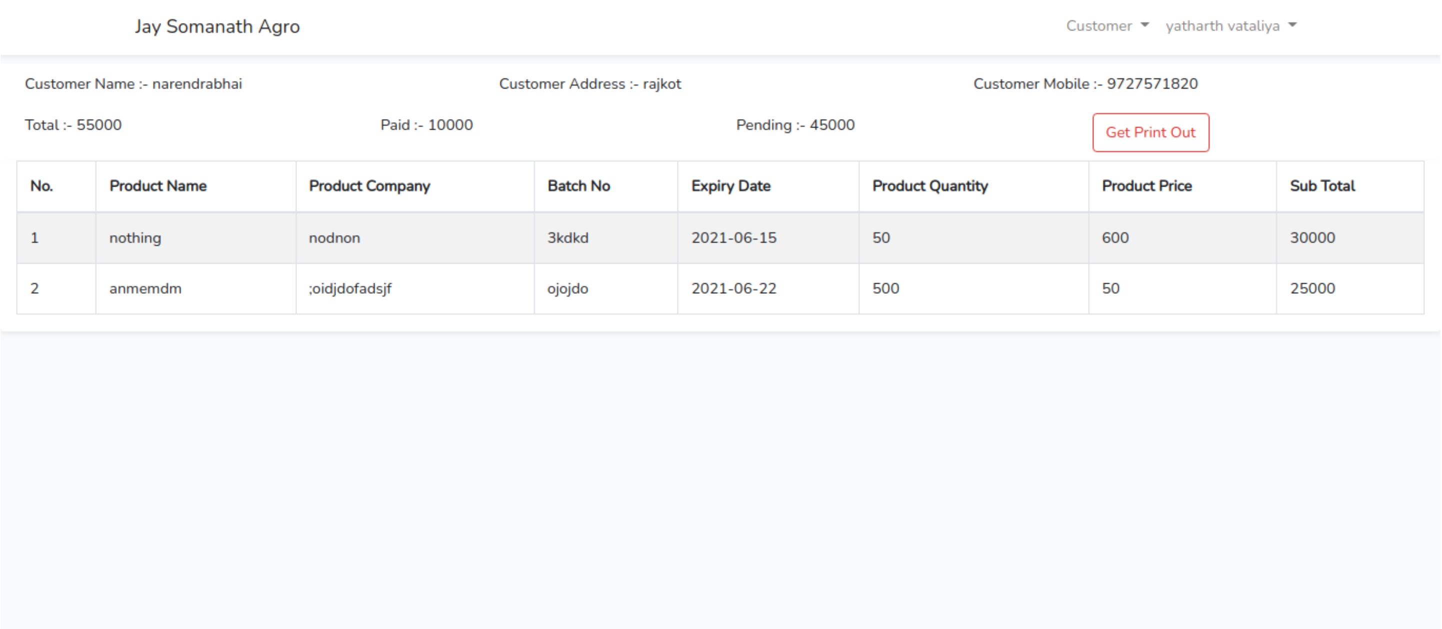
Product Price: Product Price

Sub Total: Sub Total

Paid To You: Paid to you

Total is :- NaN

Customer Order Details



Customer Name :- narendrabhai Customer Address :- rajkot Customer Mobile :- 9727571820

Total :- 55000 Paid :- 10000 Pending :- 45000

No.	Product Name	Product Company	Batch No	Expiry Date	Product Quantity	Product Price	Sub Total
1	nothing	nodnon	3kdkd	2021-06-15	50	600	30000
2	anmemdm	;oidjdofadsjf	ojojdo	2021-06-22	500	50	25000

BIBLIOGRAPHY

REFERENCE BOOK

- PHP manual

On Web site Help

- php.net
- laravel-livewire.com
- google.com
- laravel.com
- stackoverflow.com