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

|  |  |  |
| --- | --- | --- |
| SRNO | DESCRIPTION | PAGE NO |
| 1. | Intoduction | 3 |
| 2. | Project profile | 4 |
| 3. | System Development Life Cycle | 5 |
| 4. | Feasibility study | 6 |
| 5. | Requirement Analysis | 7 |
| 6. | About Tool and technologies | 8 |
| 7. | Database Design | 15 |
| 8. | DFD | 18 |
| 9. | Use Case Diagram | 21 |
| 10. | Hipo Char | 22 |
| 11. | Complete Database Overview | 23 |
| 12. | Screen Shots | 24 |

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.

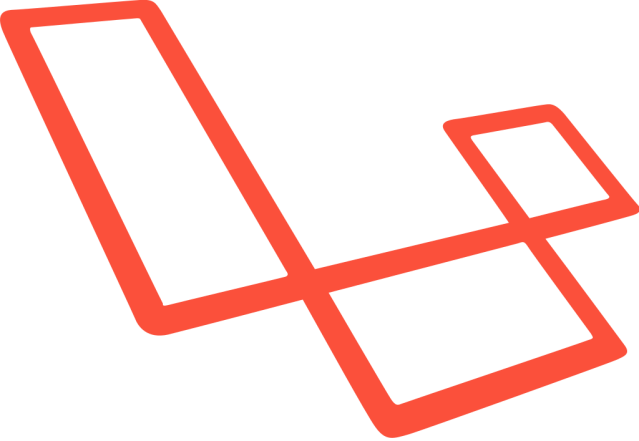
About Tools And Technology Used

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.

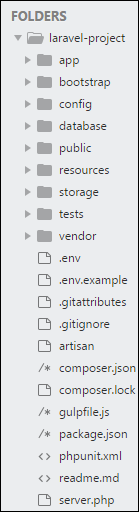
Laravel offers the following key features which make it an ideal choice for designing web applications −

* **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 −

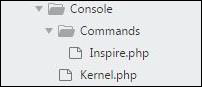


* **App**

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.



* **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

* **Exceptions**

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

* **Http**

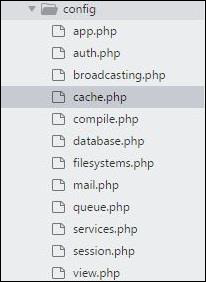
The **Http** folder has sub-folders for controllers, middleware and application requests. As Laravel follows the MVC design pattern, this folder includes model, controllers and views defined for the specific directories.

* **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.



* **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](https://en.wikipedia.org/wiki/Open-source_software) relational database management system (RDBMS).

The MySQL server software itself and the client libraries use [dual-licensing](https://en.wikipedia.org/wiki/Dual_license) distribution.

MySQL is offered under two different editions: the [open source](https://en.wikipedia.org/wiki/Open-source_software) MySQL Community Server and the proprietary [Enterprise Server](https://en.wikipedia.org/wiki/MySQL_Enterprise).

Major features as available in MySQL 5.6:

* Cross-platform support
* [Stored procedures](https://en.wikipedia.org/wiki/Stored_procedure), using a procedural language that closely adheres to [SQL/PSM](https://en.wikipedia.org/wiki/SQL/PSM)[[75]](https://en.wikipedia.org/wiki/MySQL" \l "cite_note-HarrisonFeuerstein2008-75)
* [Triggers](https://en.wikipedia.org/wiki/Database_trigger)
* [Cursors](https://en.wikipedia.org/wiki/Cursor_(databases))
* Updatable [views](https://en.wikipedia.org/wiki/View_(SQL))
* Online [Data Definition Language](https://en.wikipedia.org/wiki/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

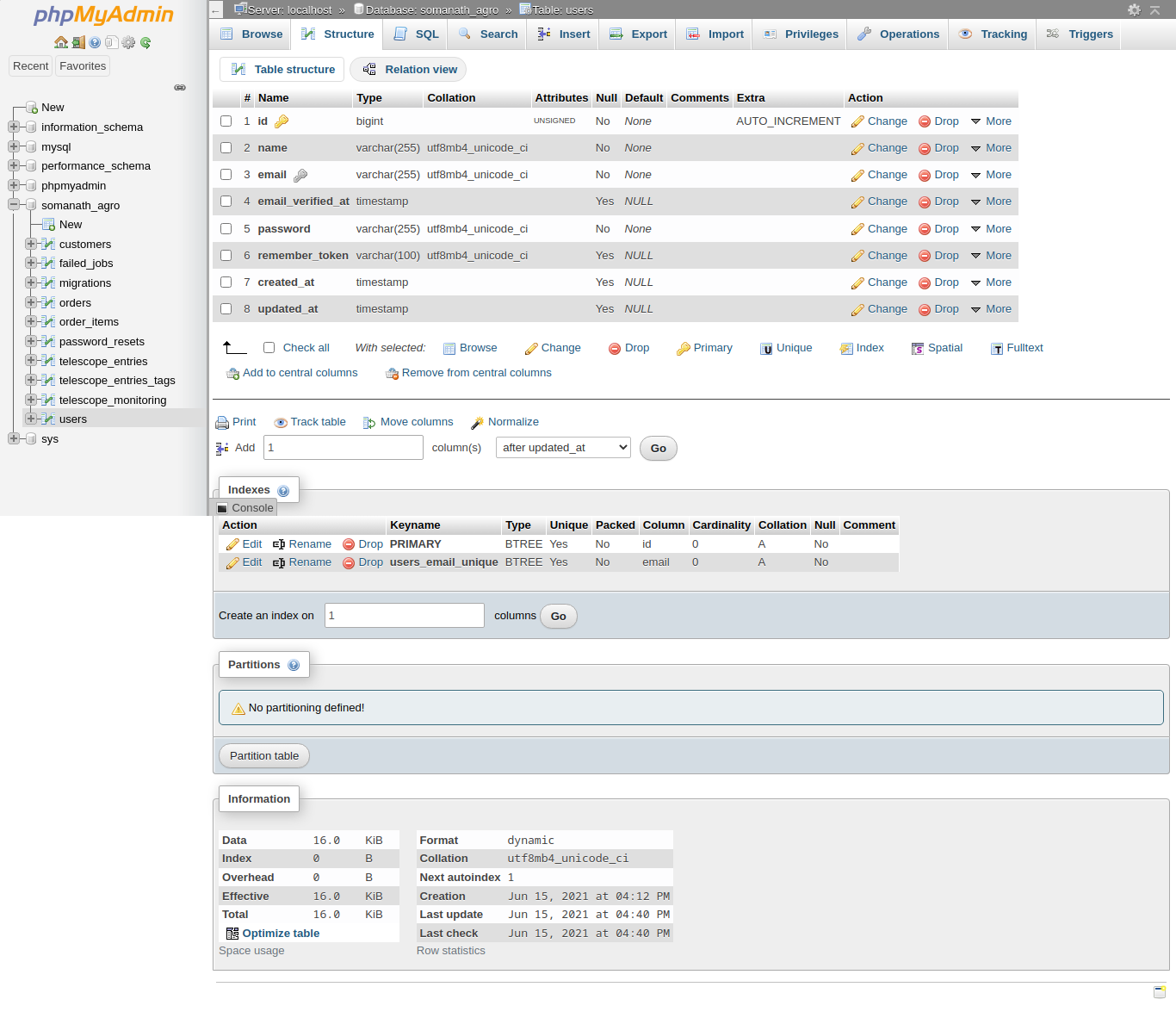


Table Name:- customers

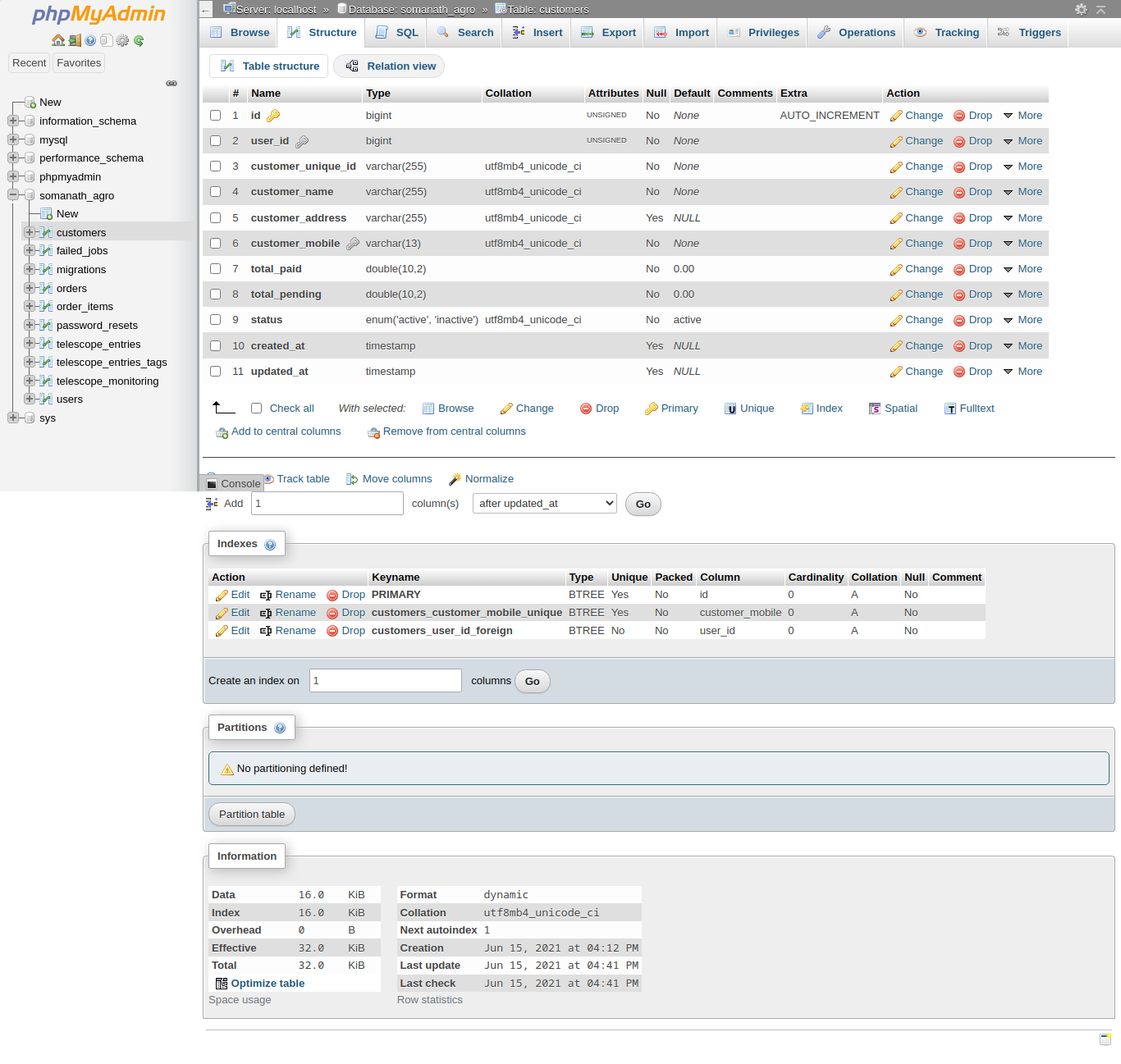


Table Name:- orders

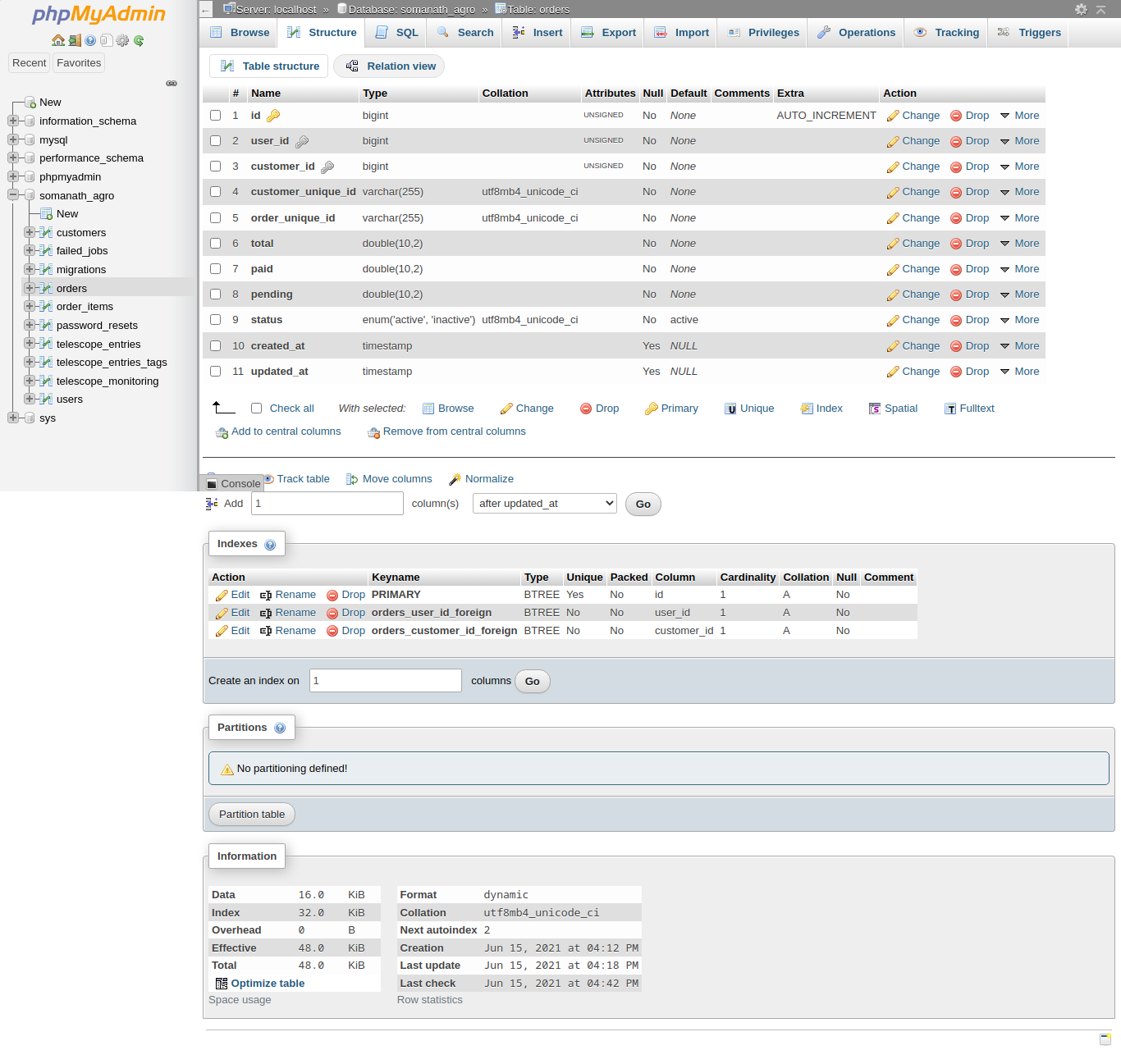
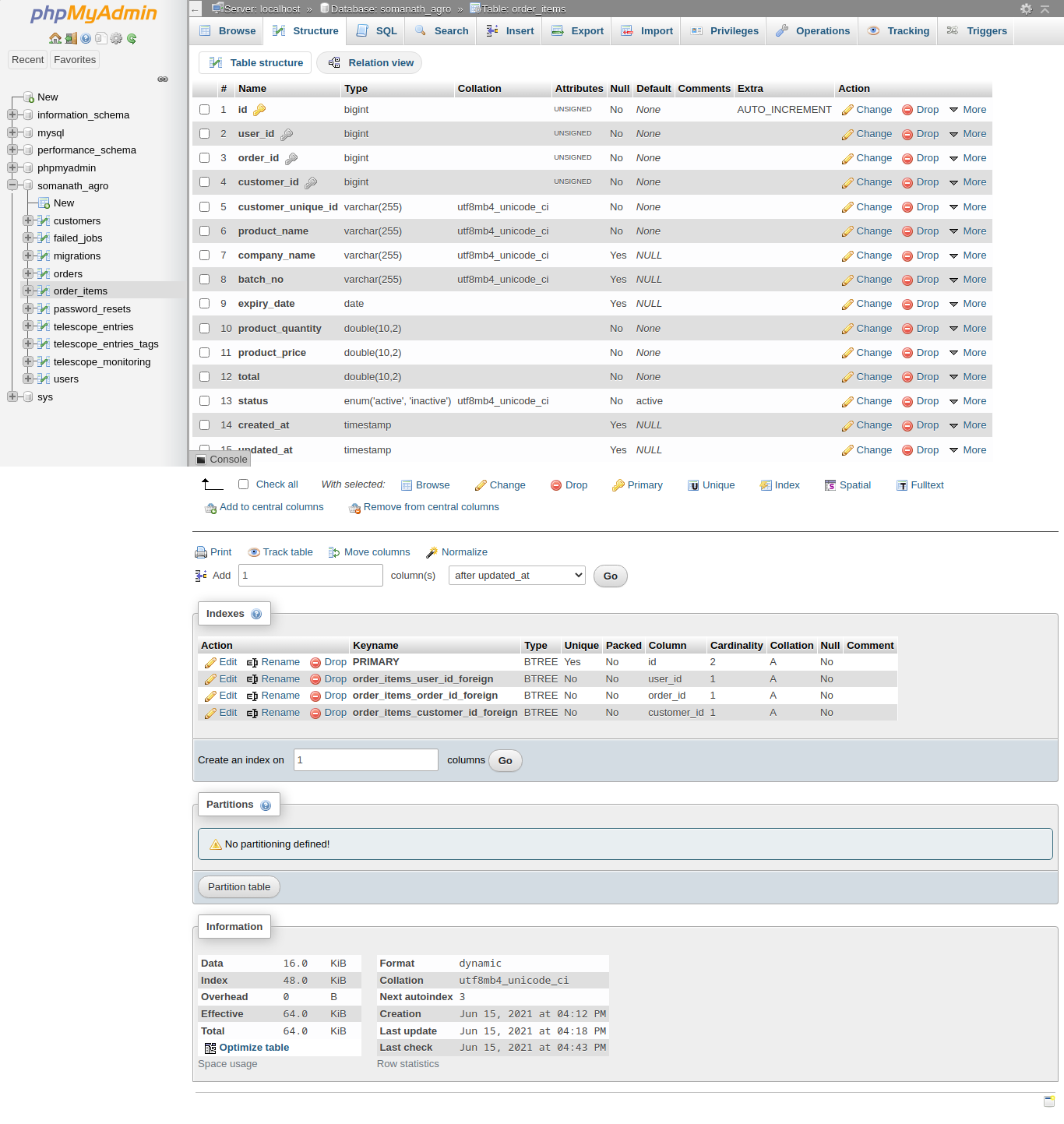


Table Name :- order\_items



DATA FLOW DIGRAM

DATA FLOW DIGRAM

DATA FLOW DIGRAM

DATA FLOW DIGRAM

Use Case Diagram

Hipo Chart

Somanath Agro Billing System

User authentication

Delete Customer

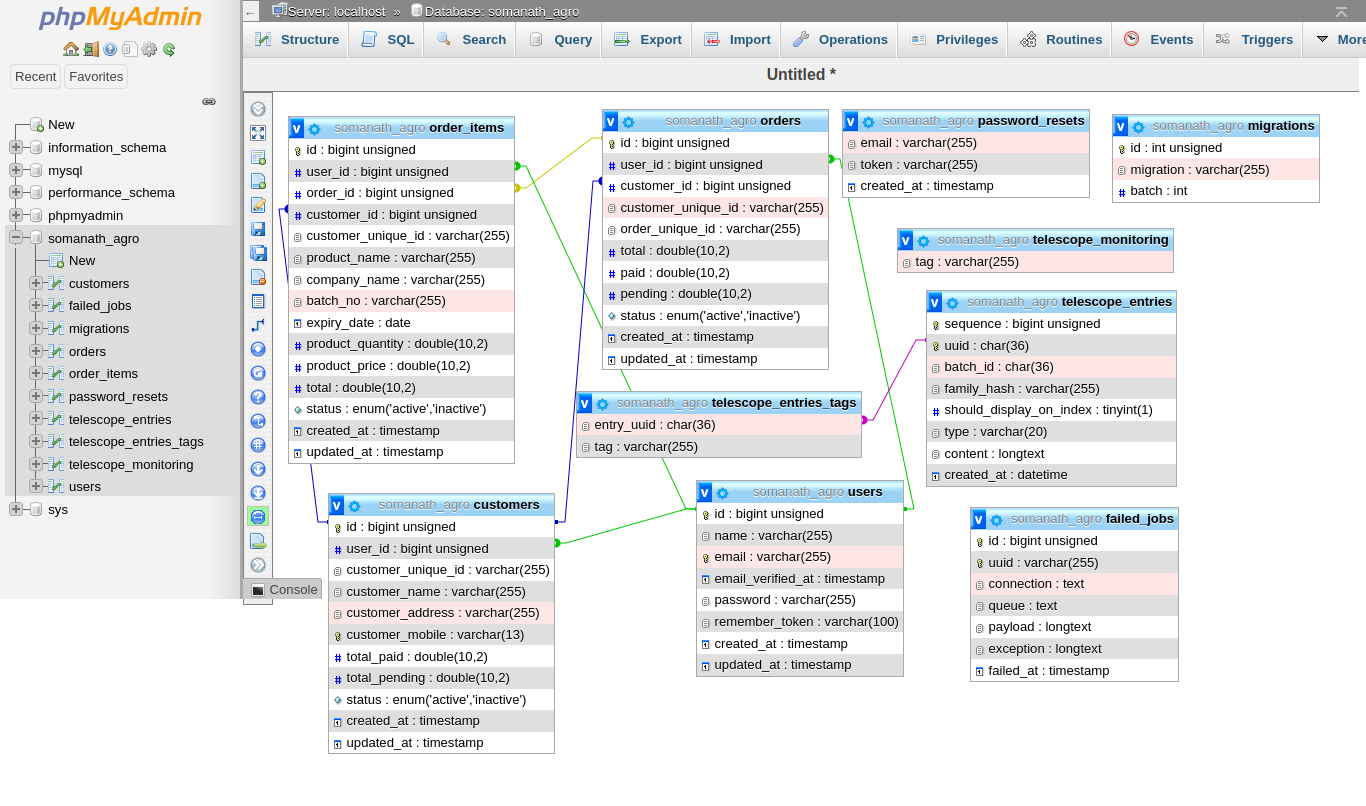
Print Order Details

Create Customer

Create Order

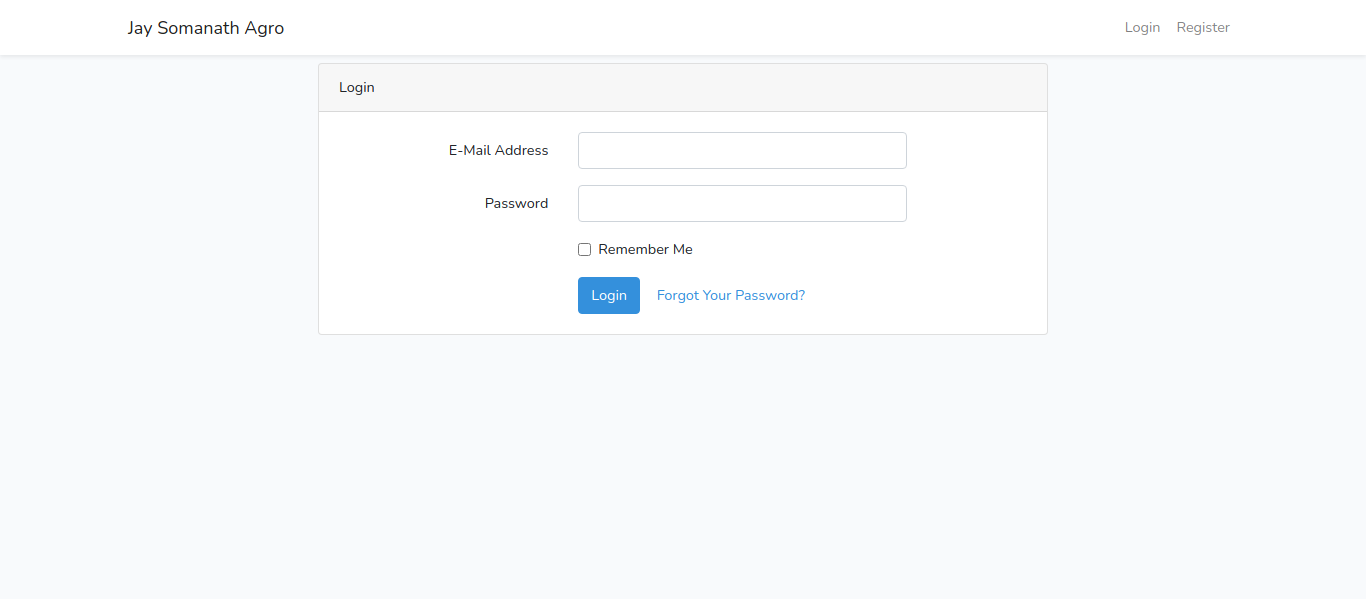
Edit Customer

Database complete View

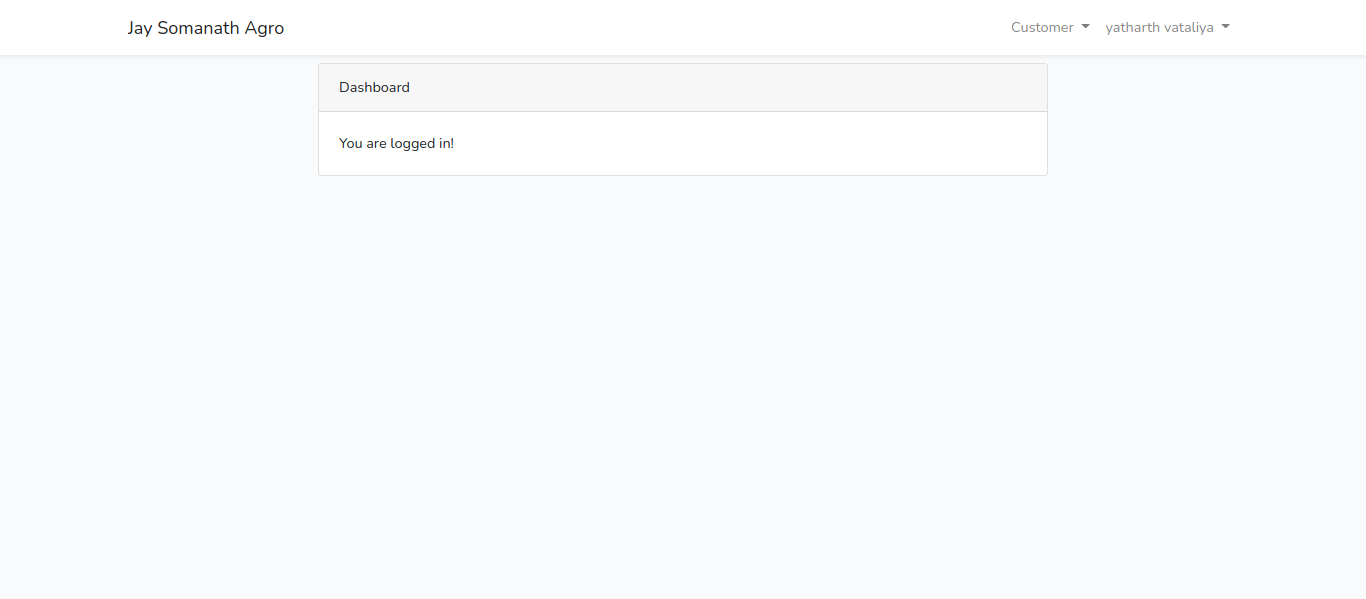


Screen Shots

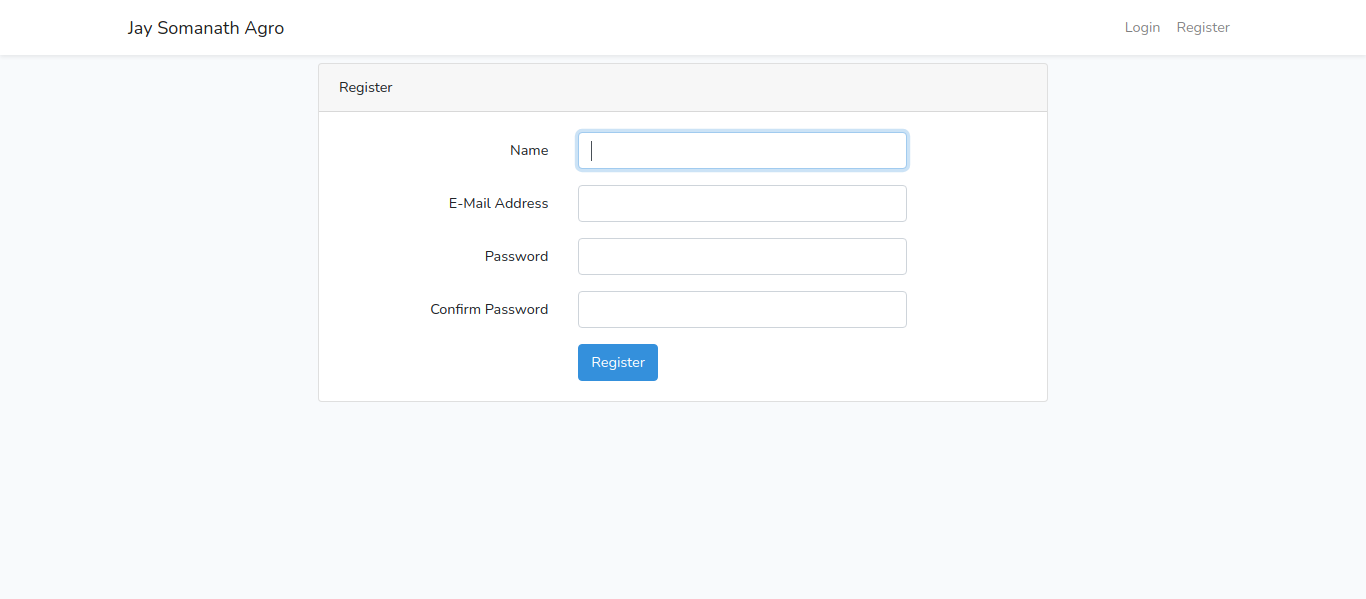
Login Screen



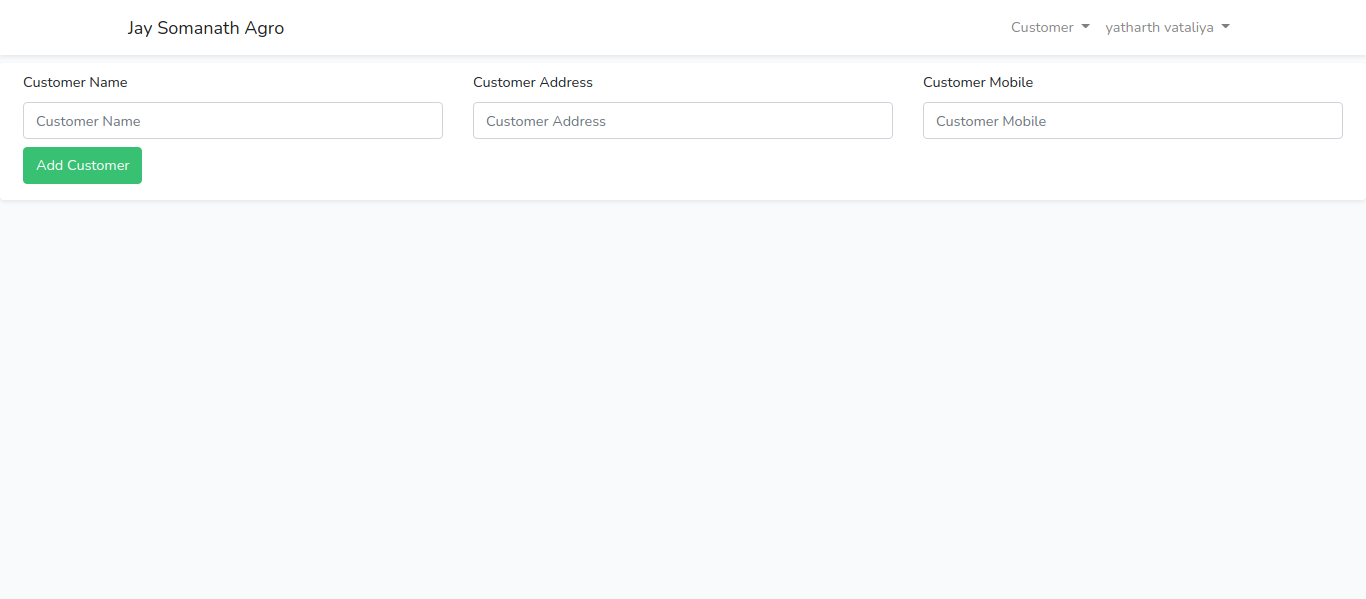
Dashboard



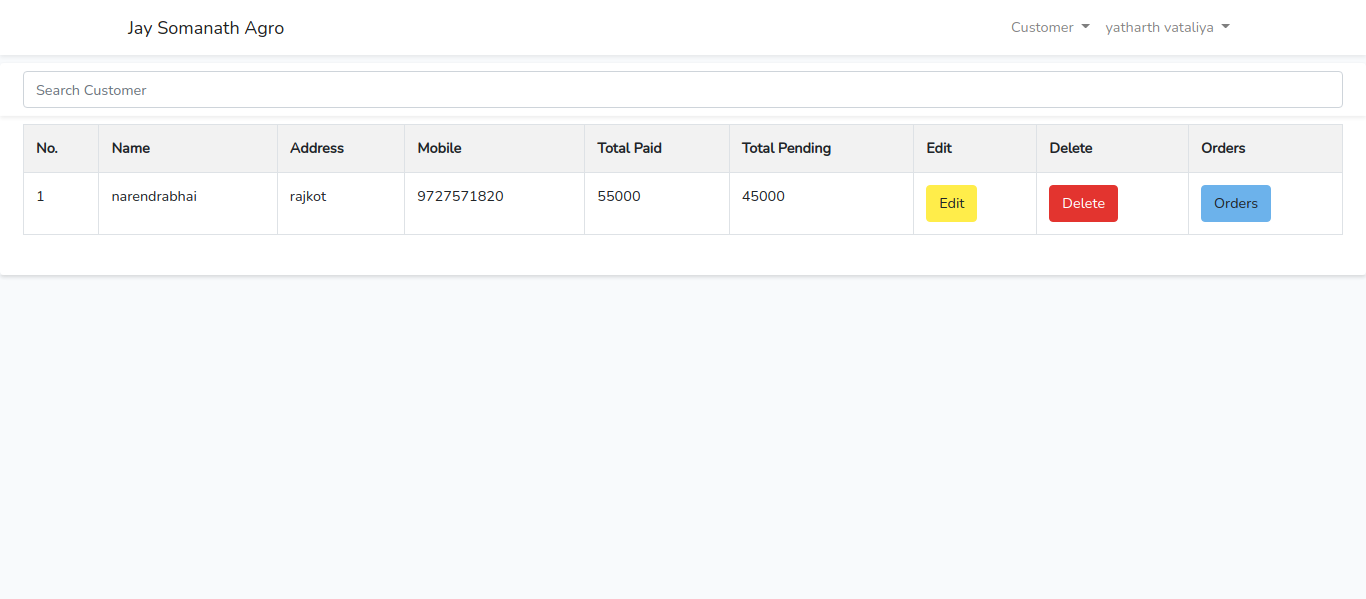
**Register**

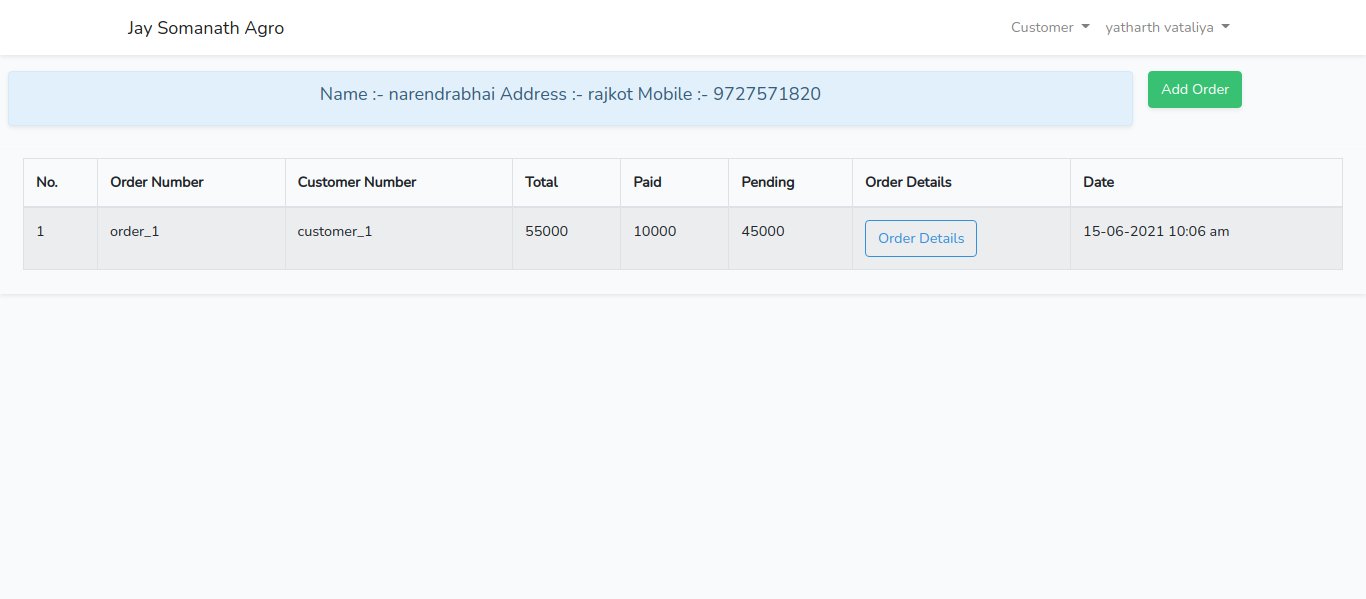


Create Customer

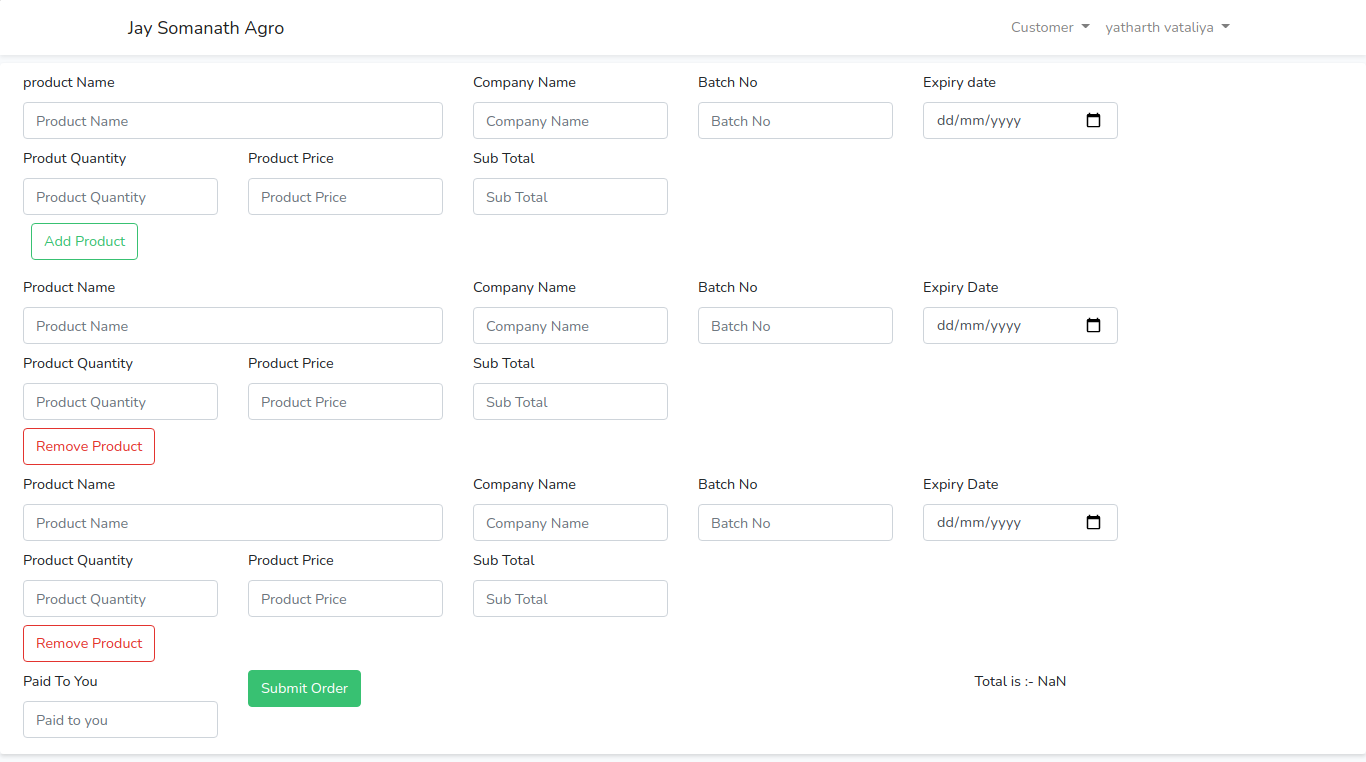


List Customers

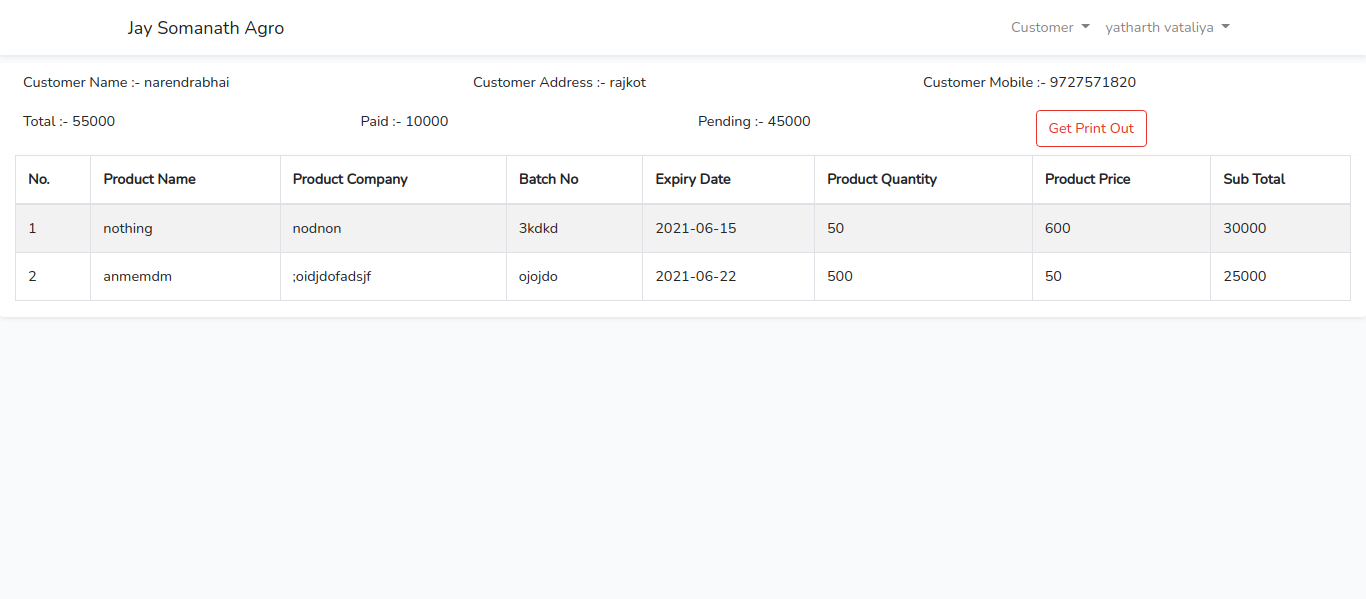
Customer Orders List



Customer Order Create



Customer Order Details



BIBLIOGRAPHY

REFERENCE BOOK

* PHP manual

On Web site Help

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