QUEUE MANAGEMENT SYSTEM

OneSmart

Living space, Innovated.

DOCUMENTATION

Table of Contents

Introduction

The majority of the time, Queue Management for the Heath Division industry is a manual procedure. It might also be regarded as one of the largest supply chain networks, with a variety of stakeholders like patients, Doctors, laboratory staff and etc. The suggested solution aims to centralize the process and address frequent problems with the current method. Wherever possible, the system would be automated to cut down on the steps required to complete a given activity and to make the process considerably simpler. patients, Doctors, laboratory staff and nurses would be the major stakeholders, or rather, users, of the system. The management would be in charge of the system's fundamental but crucial data and policies.

Nowadays, many people are close with WhatsApp social media, so this system mainly manages queues in hospitals by providing token number users through WhatsApp API.

The system must be web-based and simple to use due to the nature of the Medical center. Because of this, Administrators are given access to the system via a mobile application just like patients, while management and Health staff would have access via a web application with space for expansion as needed. this application is mobile responsive web site.

The system attends to challenges that are experienced in the sector in addition to fulfilling customer requirements. Because the system is intended to eliminate unnecessary intermediaries, it has made it possible for many stakeholders to interact and communicate directly with key parties using a variety of features.

Assumption for Queue Management System

1. "OneSmartBee "Queue Management System is focused on providing services to Health Industry. These are some stake holders to this system.

System Administrator

- view token details
- Search User details
- view customer details
- view Branches details
- add new Branch
- Search Branches details
- Update Branches details
- Delete Branches details
- Generate QR
- manage Admins
- Update message Content
- Delete message Content
- add new message Content
- Search message Content
- view notifications
- update QR code
- Generate QR
- Login
- Create New Admins

Customer

- view token details
- view Branches details

Methodology

The Queue Management system named by “ OnesmartBee Queue ” that help us to get an idea about Patients things like Ticket management, QR Generate, Customer Handling, Message Handling what are the technologies that use to develop these things so on. System overview is displayed below.

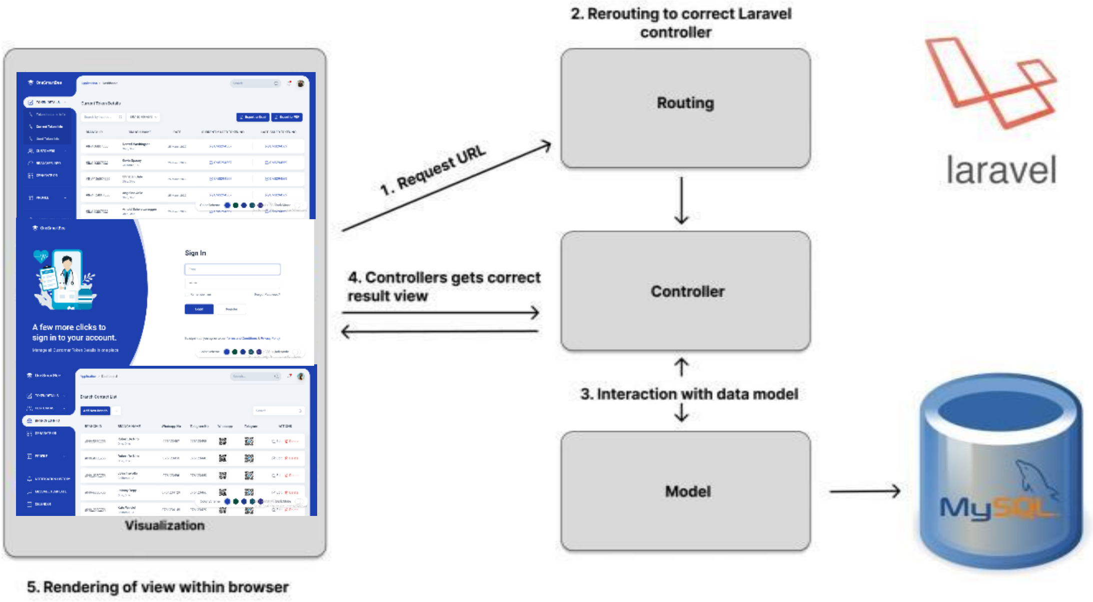

2. Rerouting to correct Laravel controller
Routing
laravel
1. Request URL
4. Controllers gets correct result view
Controller
3. Interaction with data model
Model
MySQL
Sign In
A few more clicks to sign in to your account.
Visualization
5. Rendering of view within browser

Fig.1.OverView Diagram_1 for Queue Management System

I use Laravel and MySQL DB Connection for implement this System. I implement System frontend using PHP Framework- (Laravel-9). Laravel is a web application framework with expressive, elegant syntax. A web framework provides a structure and starting point for creating your application, allowing you to focus on creating something amazing while we sweat the details.

Laravel strives to provide an amazing developer experience while providing powerful features such as thorough dependency injection, an expressive database abstraction layer, queues and scheduled jobs, unit and integration testing, and more.

Whether you are new to PHP web frameworks or have years of experience, Laravel is a framework that can grow with you. We'll help you take your first steps as a web developer or give you a boost as you take your expertise to the next level. We can't wait to see what you build [3]. And I used Façade design patterns in Laravel for implement crud functions that in OnesmartBee , Queue management system. Facades have many benefits. They provide a terse, memorable syntax that allows you to use Laravel 's features without remembering long class names that must be injected or configured manually. Furthermore, because of their unique usage of PHP's dynamic methods, they are easy to test.

However, some care must be taken when using facades. The primary danger of facades is class "scope creep". Since facades are so easy to use and do not require injection, it can be easy to let your classes continue to grow and use many facades in a single class. Using dependency injection, this potential is mitigated by the visual feedback a large constructor gives you that your class is growing too large. So, when using facades, pay special attention to the size of your class so that its

scope of responsibility stays narrow. If your class is getting too large, consider splitting it into multiple smaller classes [3].

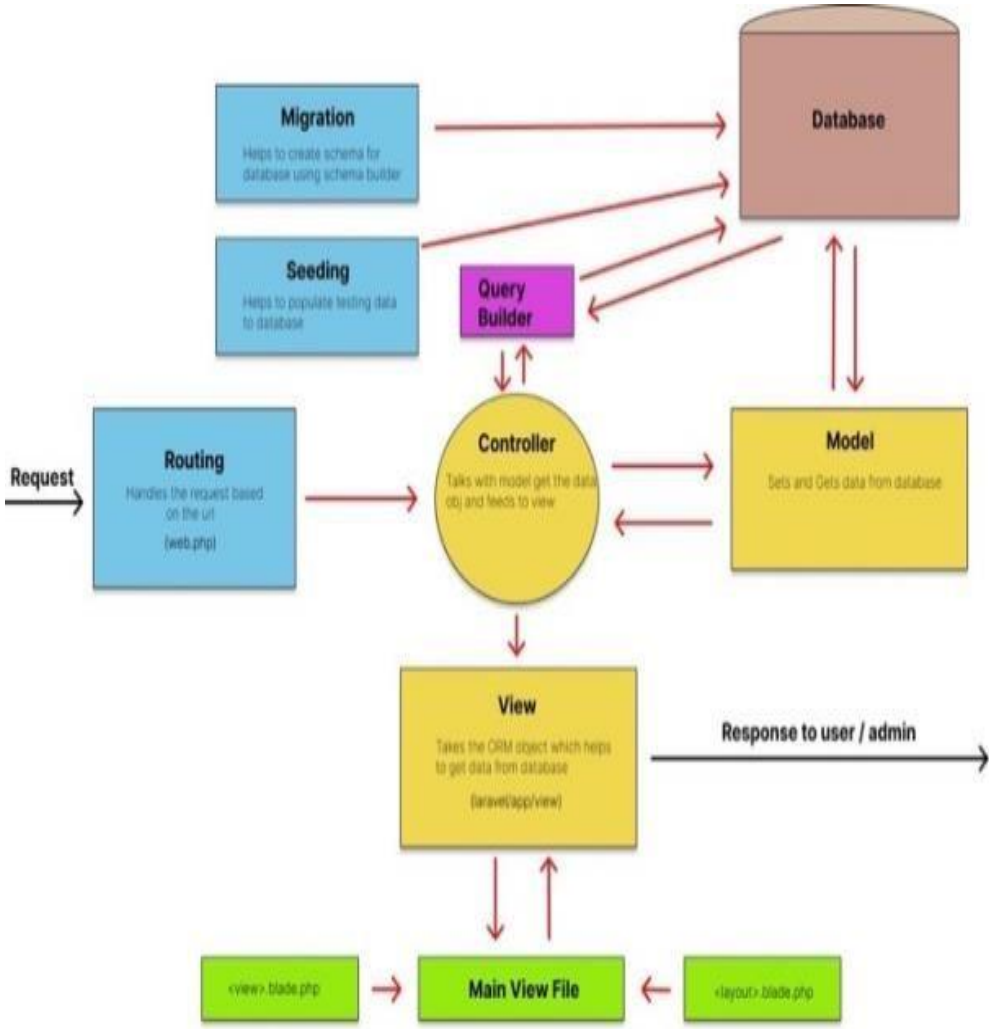

Migration
Helps to create schema for database using schema builder
Database
Seeding
Helps to populate testing data to database
Query Builder
Request
Routing
Handles the request based on the url
(web.php)
Controller
Talks with model get the data obj and feeds to view
Model
Sets and Gets data from database
View
Takes the ORM object which helps to get data from database
(laravel/app/view)
Response to user / admin
<view>.blade.php
Main View File
<layout>.blade.php

Fig.2.OverView Diagram_2 for Queue Management System

This is the flow of the Queue management system. These are the technologies that I use to implement this system.

Technologies:

Frontend Development

- html5
- css
- bootstrap
- javascript
- jquery
- ajax

Backend Development

- laravel9, php

Database

- MySQL

Tools

- VS Code
- XAMPP
- GitHub
- PhpMyAdmin

Resources

- Laravel Document
- Laravel Related Videos
- Google Charts

Prerequisites

- VS code should be installed
- XAMPP server should be installed
- composer has been installed to the machine
- node has been installed to the machine
- npm has been installed

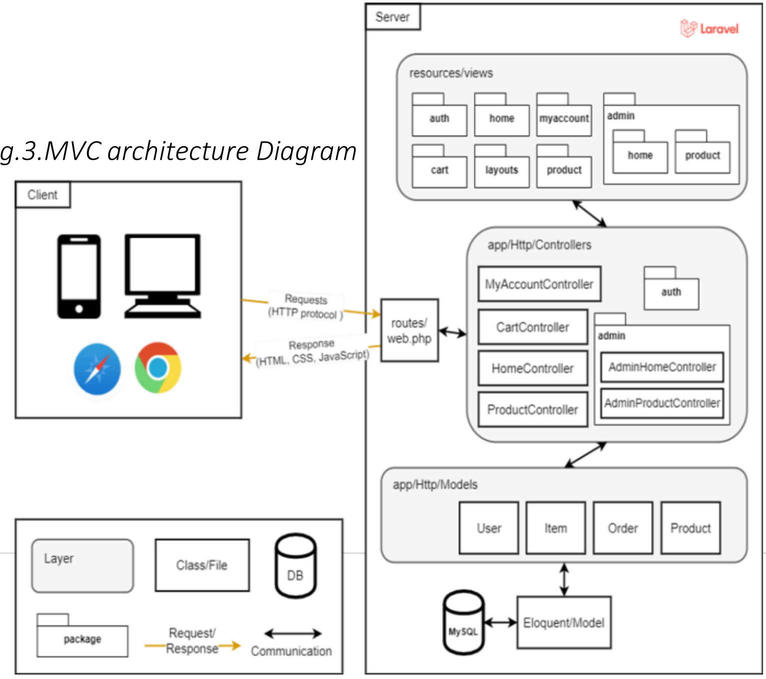
Procedure of setting up prerequisites and run the project

- Download a zip folder of this project and unzip or clone the repository.
- Open the project in VS code.
- Open a new terminal.
- Type 'composer create-project' and click enter.
- After the project has been created, start Laravel's local development server using the Laravel's Artisan CLI serve command: php artisan serve.
- Started the Artisan development server, application will be accessible web browser at http://localhost:8000

- Host server = apptest.onesmartapi.com

To develop the Queue Management System, I designed an MVC architecture. This architecture is presented next:

Fig.3.MVC architecture Diagram



You can see the MVC in gray. This diagram matches the real project code. Let’s have a quick analysis of this architecture:

- On the left, we have clients (users of our application e.g., browsers in mobile/desktop devices). Clients connect to the application through the Hypertext Transfer Protocol (HTTP). HTTP gives users a way to interact with our web application.
- On the right, we have the server where we place our application code.
- All client interactions first pass for a route file called web.php.
- The web.php file passes the interaction to the controllers.
- Controllers communicate with models and pass information to the views, which are finally delivered to the clients as HTML, CSS, and JavaScript code.

For the view layer I used Blade which is a Laravel templating system.

The previous diagram serves to understand how the application is connected, how each layer or file invokes other layers or files. You can even deduce some architectural rules from that diagram. For example, you cannot invoke views or models from the web.php route file. Or it is not allowed to invoke views from a model file. This is useful to keep our application code clean and flexible to evolving changes. Database integration, login, and web sessions

We used MySQL as our database server. And we used Laravel Eloquent to connect our Laravel application with

MySQL. This is our final database structure:[4]

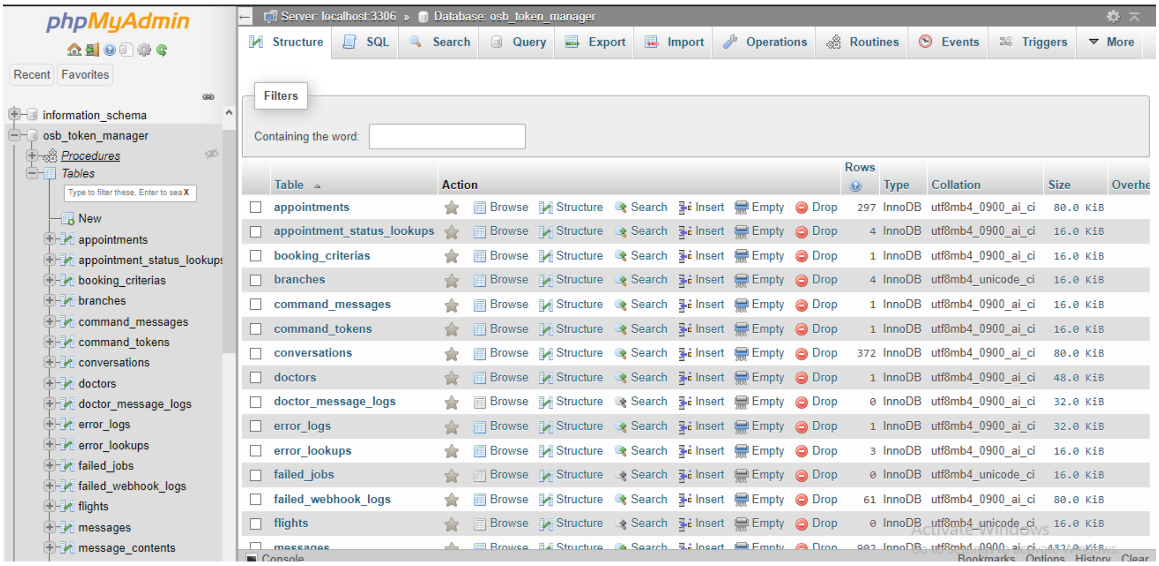


	Table	Action	Rows	Type	Collation	Size	Overhe
□	appointments	☆ Browse Structure Search Insert Empty Drop	297	InnoDB	utf8mb4_0900_ai_ci	80.0 KiB	
□	appointment_status_lookups	☆ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	
□	booking_criterias	☆ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	
□	branches	☆ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	
□	command_messages	☆ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	
□	command_tokens	☆ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	
□	conversations	☆ Browse Structure Search Insert Empty Drop	372	InnoDB	utf8mb4_0900_ai_ci	80.0 KiB	
□	doctors	☆ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_0900_ai_ci	48.0 KiB	
□	doctor_message_logs	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	
□	error_logs	☆ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	
□	error_lookups	☆ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	
□	failed_jobs	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	
□	failed_webhook_logs	☆ Browse Structure Search Insert Empty Drop	61	InnoDB	utf8mb4_0900_ai_ci	80.0 KiB	
□	flights	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	
□	messages	☆ Browse Structure Search Insert Empty Drop	902	InnoDB	utf8mb4_0900_ai_ci	432.0 KiB	

Fig.4. Database Connection Tables

System Login

Email: admin@gmail.com

Password: password

LARAVEL 9 .env File Settings

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:yQsRHIT8bV5Gb0M4XC/M42uPjPjXv2yOdwdMyMpd8rQ=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=apptest.onesmartapi.com

DB_PORT=3306
DB_DATABASE=osb_token_manager
DB_USERNAME=dasunika
DB_PASSWORD=FMFits@5544

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailhog
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME=https
PUSHER_APP_CLUSTER=mt1

VITE_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
VITE_PUSHER_HOST="${PUSHER_HOST}"
VITE_PUSHER_PORT="${PUSHER_PORT}"
VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

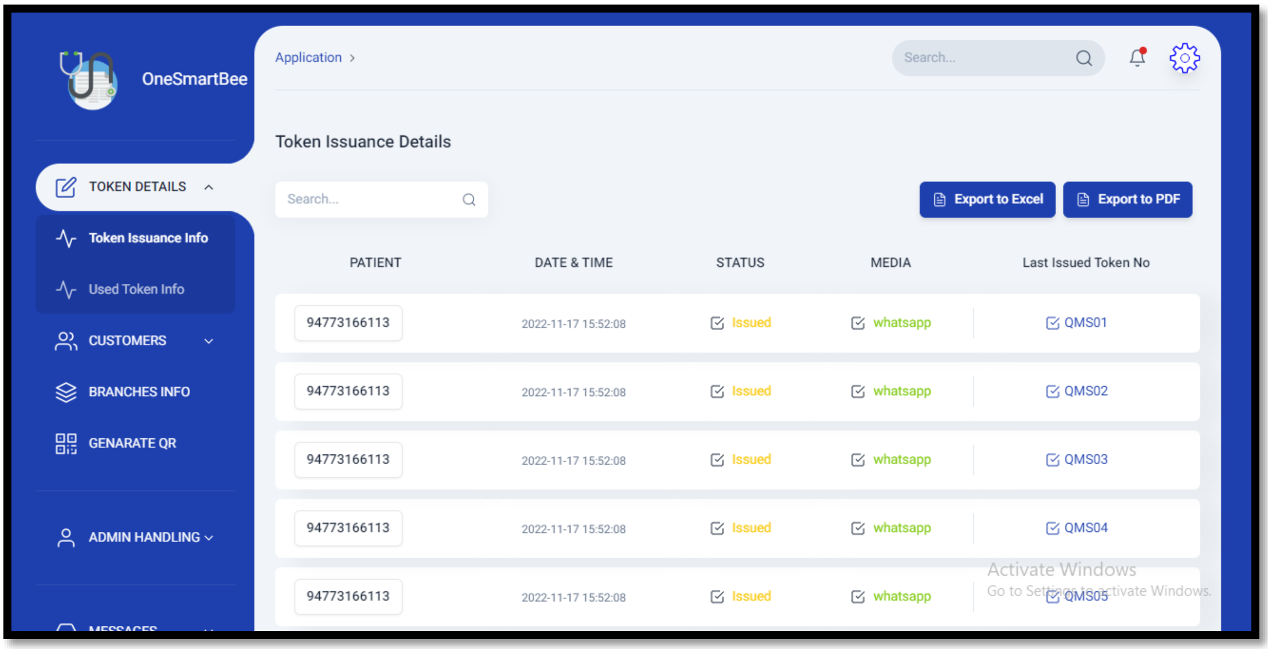
Proposed System Main Functions

Currently, our team has developed a web application for the Queue Management System called "OnesmartBee" When a user comes to our system , they can only view the login interface; after they login to the system, they can view each information separately in dashboard. There are few main management parts.

They are Token management, Customer management , Generate QR, Admin handling , Branch Information Management and Message Handling. Etc.

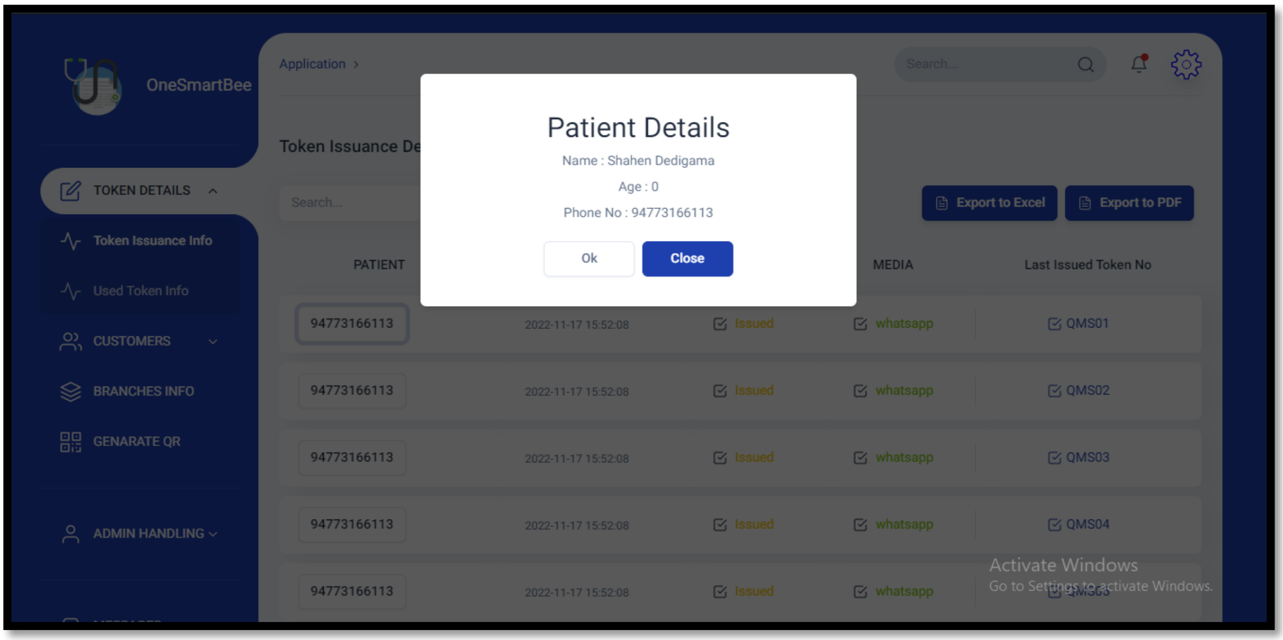
01.Token management

The Token management is the first one, handing mainly token issuance info and used token info. Token issuance info can view each issued token details(Patient ,Date & Time, Status, Media ,Last Issued Token No) search relevant issued token details providing customer phone number, show more information of customers ,etc.



Token Issuance Details

PATIENT	DATE & TIME	STATUS	MEDIA	Last Issued Token No
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS01
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS02
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS03
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS04
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS05



Token Issuance De

Patient Details

Name : Shahen Dedigama

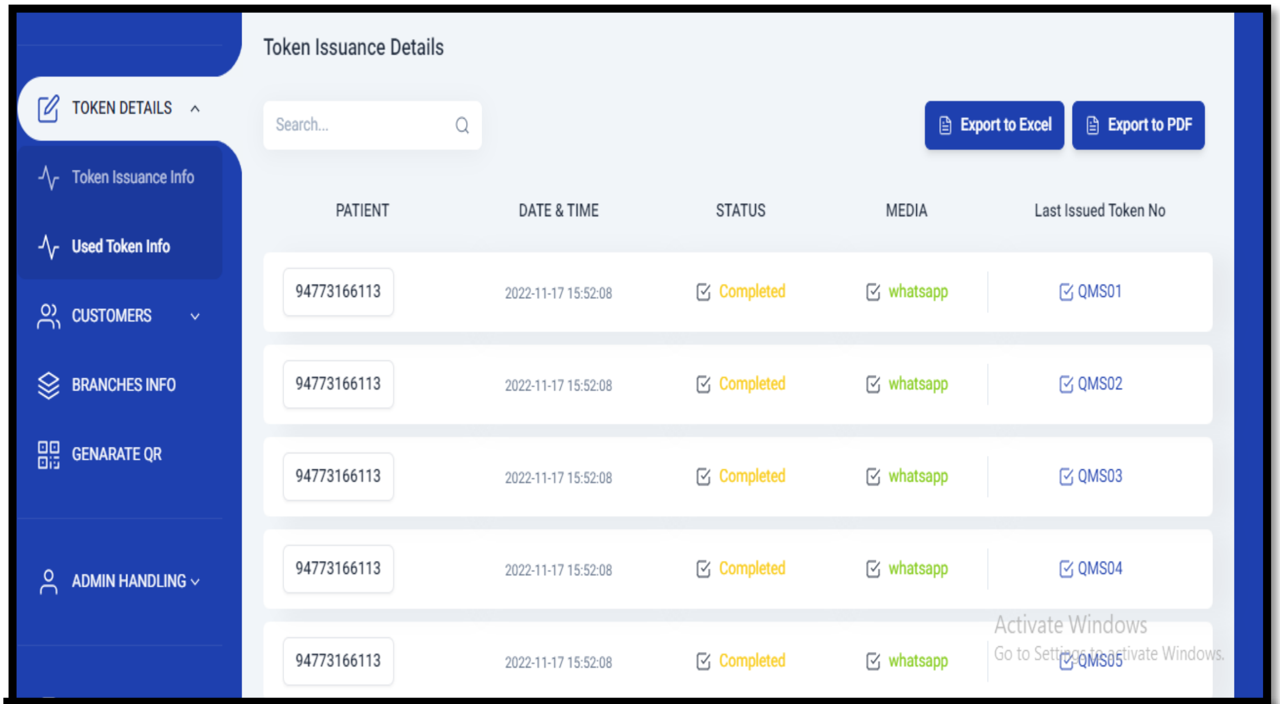
Age : 0

Phone No : 94773166113

Ok Close

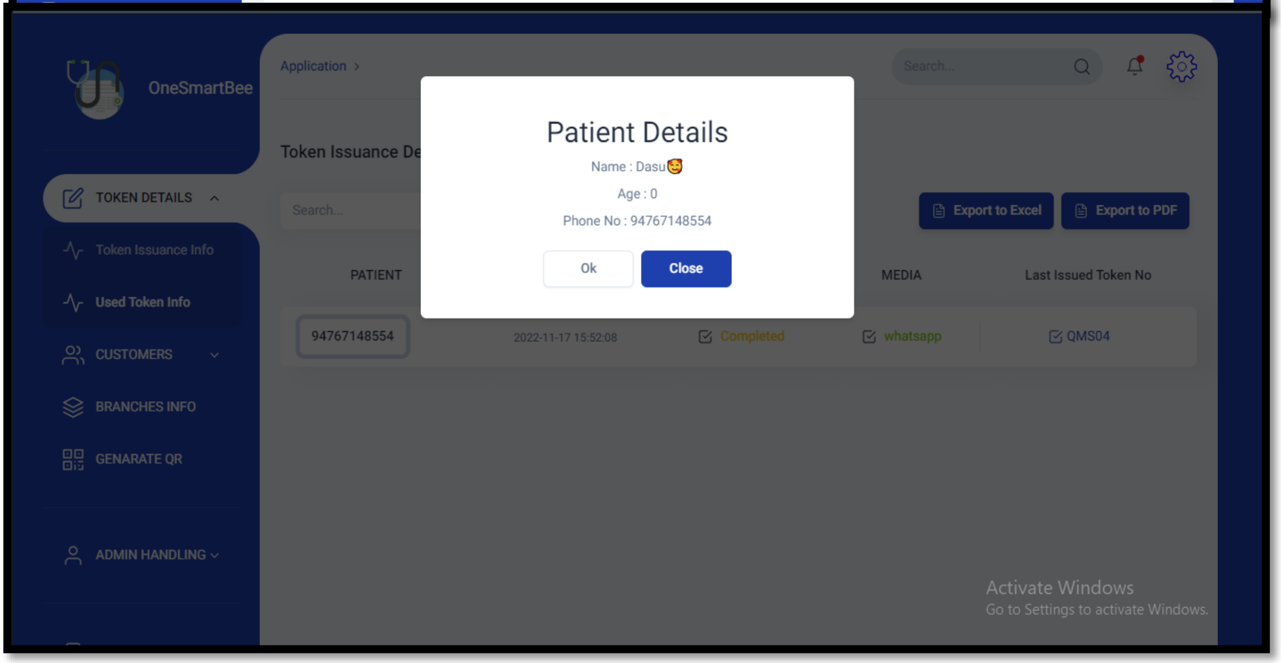
PATIENT			MEDIA	Last Issued Token No
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS01
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS02
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS03
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS04
94773166113	2022-11-17 15:52:08	Issued	whatsapp	QMS05

Used Token info can view each used token details(Patient ,Date & Time, Status, Media ,Last Issued Token No) search relevant used token details providing customer phone number, show more information of customers ,etc.



Token Issuance Details

PATIENT	DATE & TIME	STATUS	MEDIA	Last Issued Token No
94773166113	2022-11-17 15:52:08	Completed	whatsapp	QMS01
94773166113	2022-11-17 15:52:08	Completed	whatsapp	QMS02
94773166113	2022-11-17 15:52:08	Completed	whatsapp	QMS03
94773166113	2022-11-17 15:52:08	Completed	whatsapp	QMS04
94773166113	2022-11-17 15:52:08	Completed	whatsapp	QMS05



Token Issuance De

Patient Details

Name : Dasu🥰

Age : 0

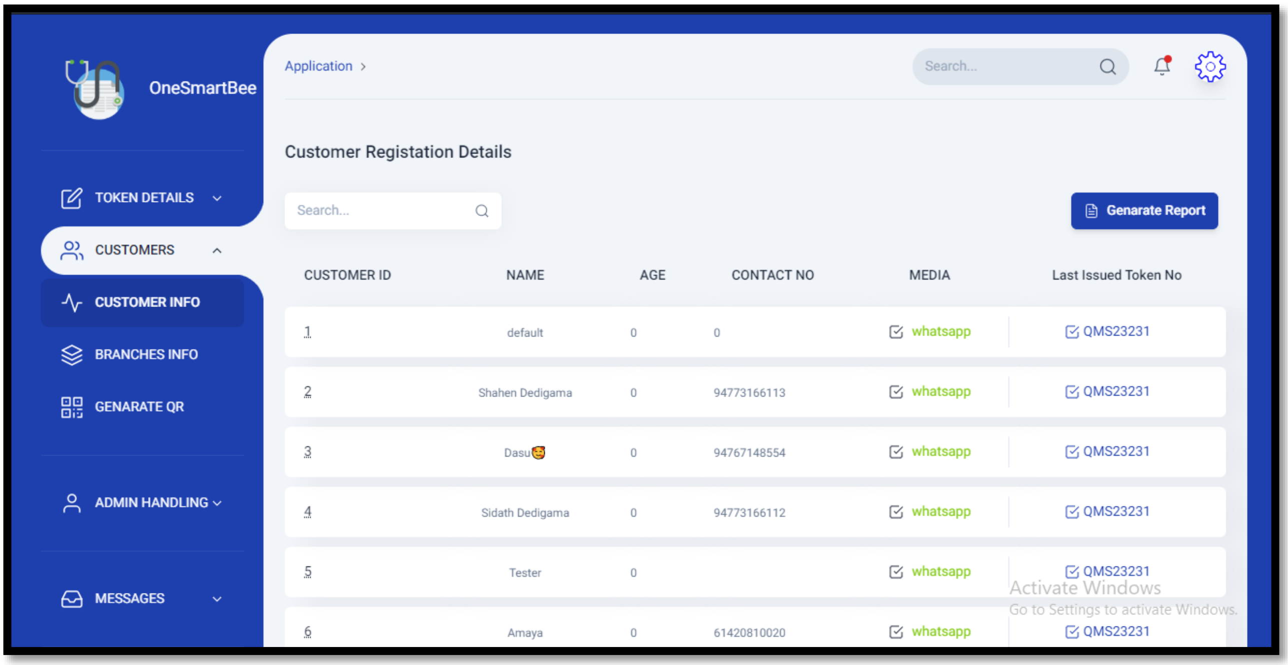
Phone No : 94767148554

Ok Close

PATIENT			MEDIA	Last Issued Token No
94767148554	2022-11-17 15:52:08	Completed	whatsapp	QMS04

02.Customer management

The Customer management is the Next one, handing mainly customers information it can view each WhatsApp customer details(customer id, Name, Age, contact No, Media ,Token No) search customer details providing customer phone number, etc..



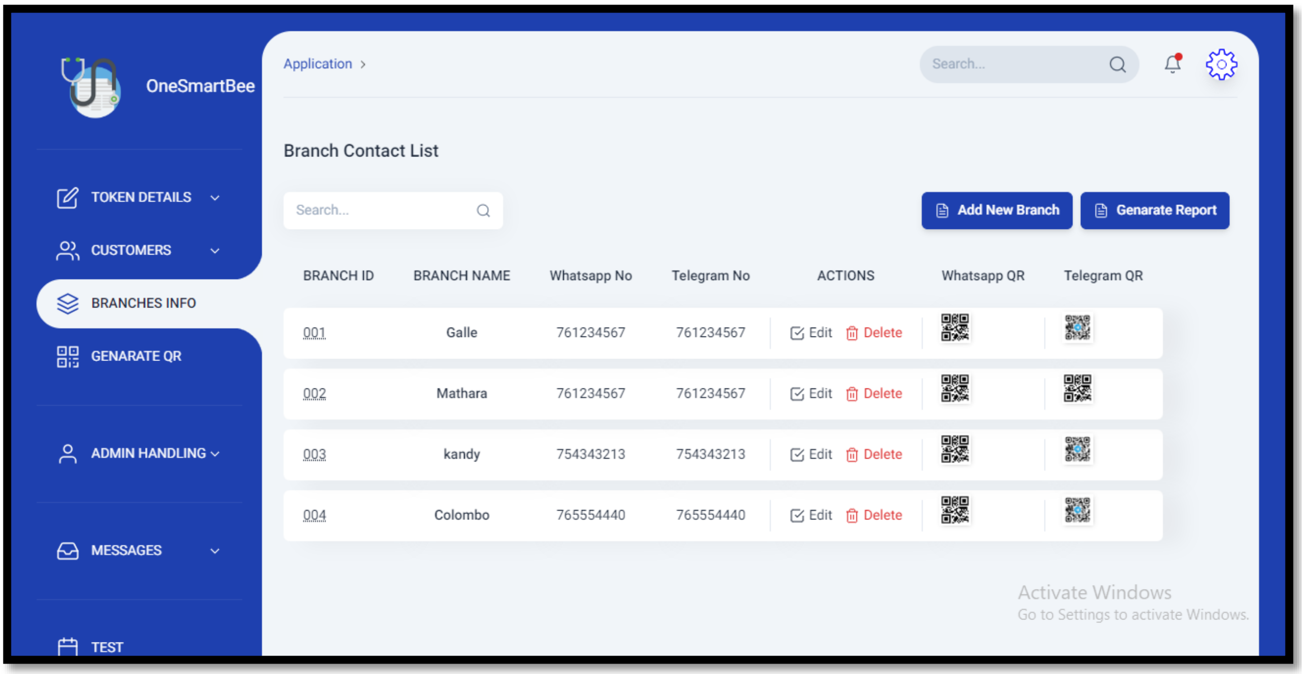
Customer Registation Details

CUSTOMER ID	NAME	AGE	CONTACT NO	MEDIA	Last Issued Token No
1	default	0	0	whatsapp	QMS23231
2	Shahen Dedigama	0	94773166113	whatsapp	QMS23231
3	Dasu🥰	0	94767148554	whatsapp	QMS23231
4	Sidath Dedigama	0	94773166112	whatsapp	QMS23231
5	Tester	0		whatsapp	QMS23231
6	Amaya	0	61420810020	whatsapp	QMS23231

03.Branch management

The Branch management is the Next one, handing mainly Branches information.
This page can view each WhatsApp customer details(Branch id, Branch Name, WhatsApp No, Telegram No ,WhatsApp QR and Telegram QR) .search Branch details providing Branch name. admin .can delete update and add new branch. And also update WhatsApp and telegram QR

03.1.Branch Details page

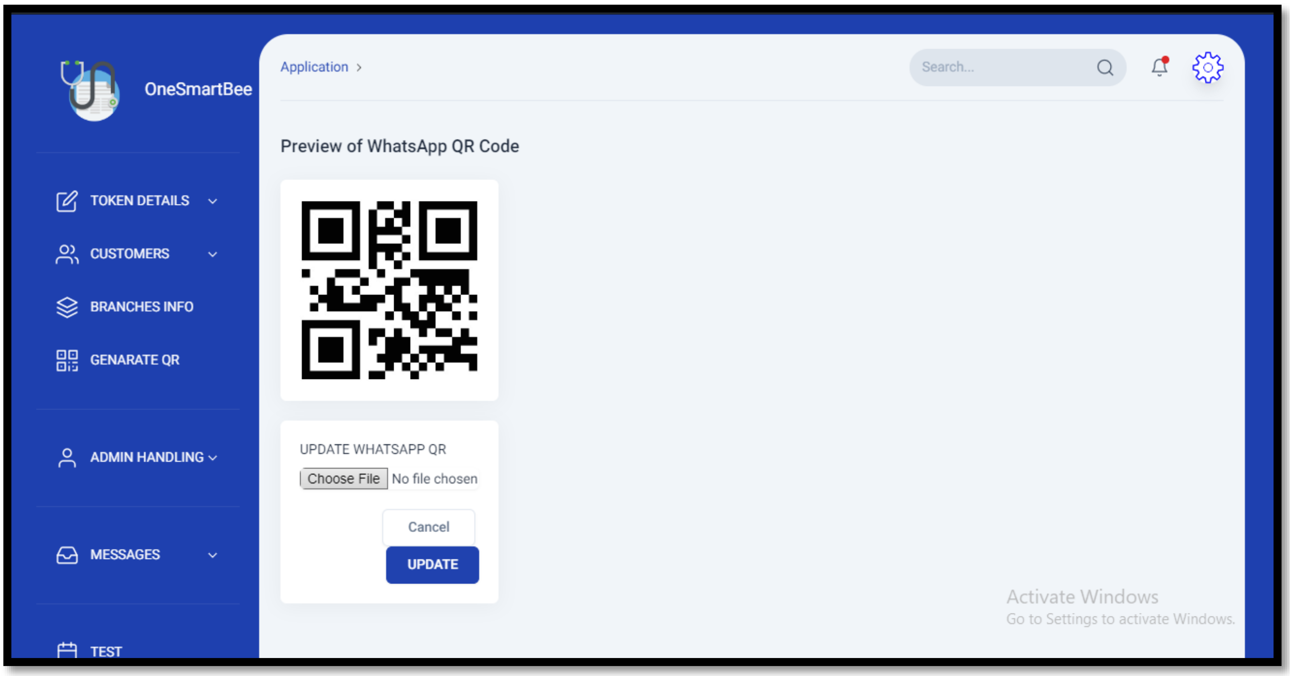


Branch Contact List

BRANCH ID	BRANCH NAME	Whatsapp No	Telegram No	ACTIONS	Whatsapp QR	Telegram QR
001	Galle	761234567	761234567	Edit Delete		
002	Mathara	761234567	761234567	Edit Delete		
003	kandy	754343213	754343213	Edit Delete		
004	Colombo	765554440	765554440	Edit Delete		

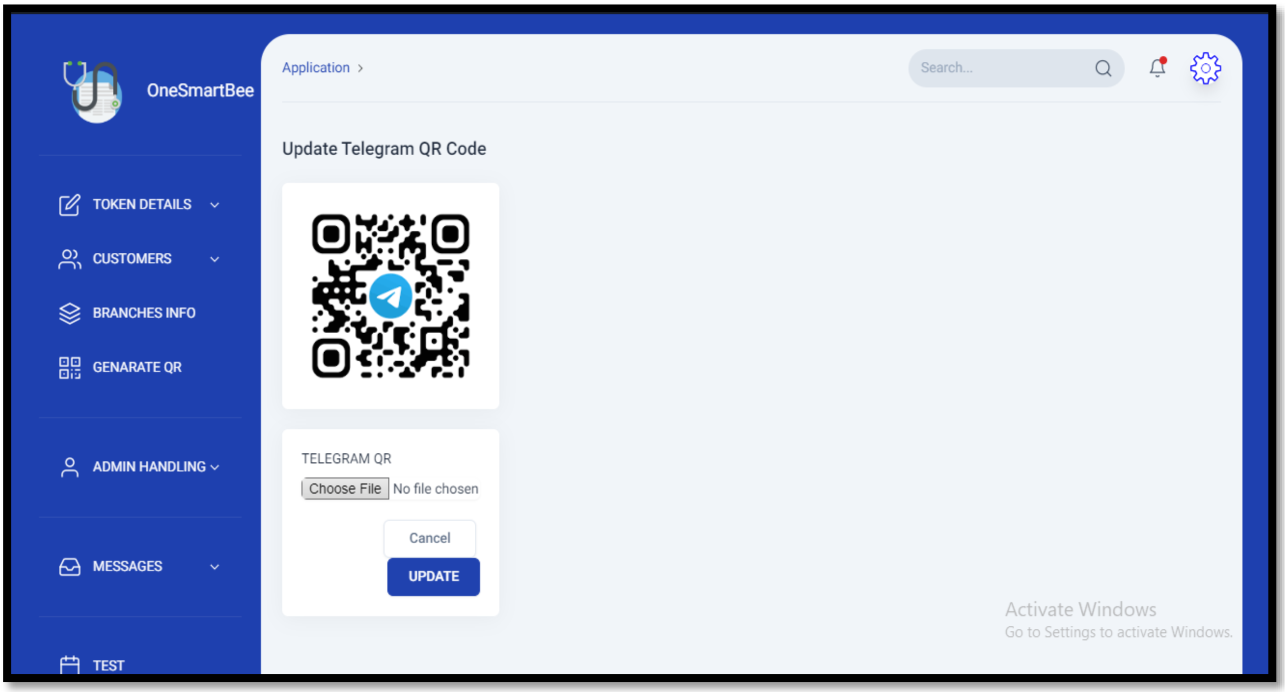
03.2.Update Whatsapp QR Code

Branches using this system have a specific whatsapp QR code. Sometimes they need to be updated and it can be done easily from this page.



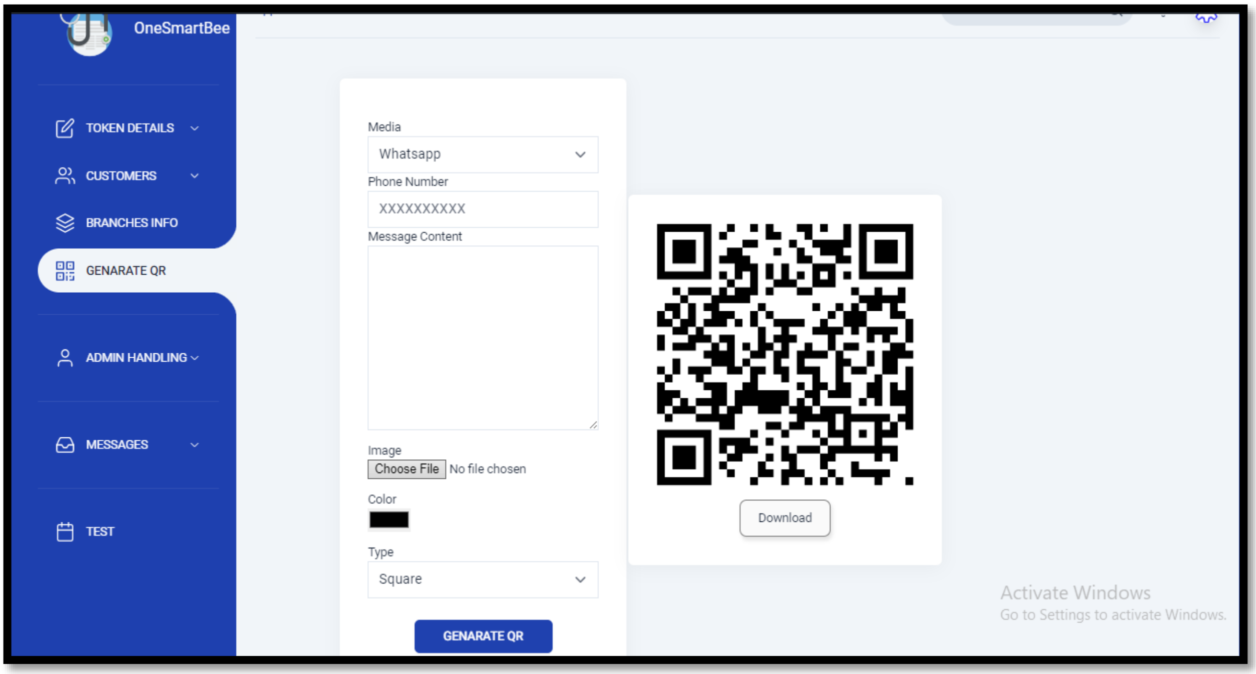
03.3 Update Telegram QR code

Branches using this system have a specific telegram QR code. Sometimes they need to be updated and it can be done easily from this page.



04. Generate QR

The branches that use this system have their own phone numbers. The QR generated by them is used for the communication of these branches. This facility has been included to create a new QR Code within this system. It can be done easily from this page. ..Many facilities are provided here to customize the QR code according to your needs.

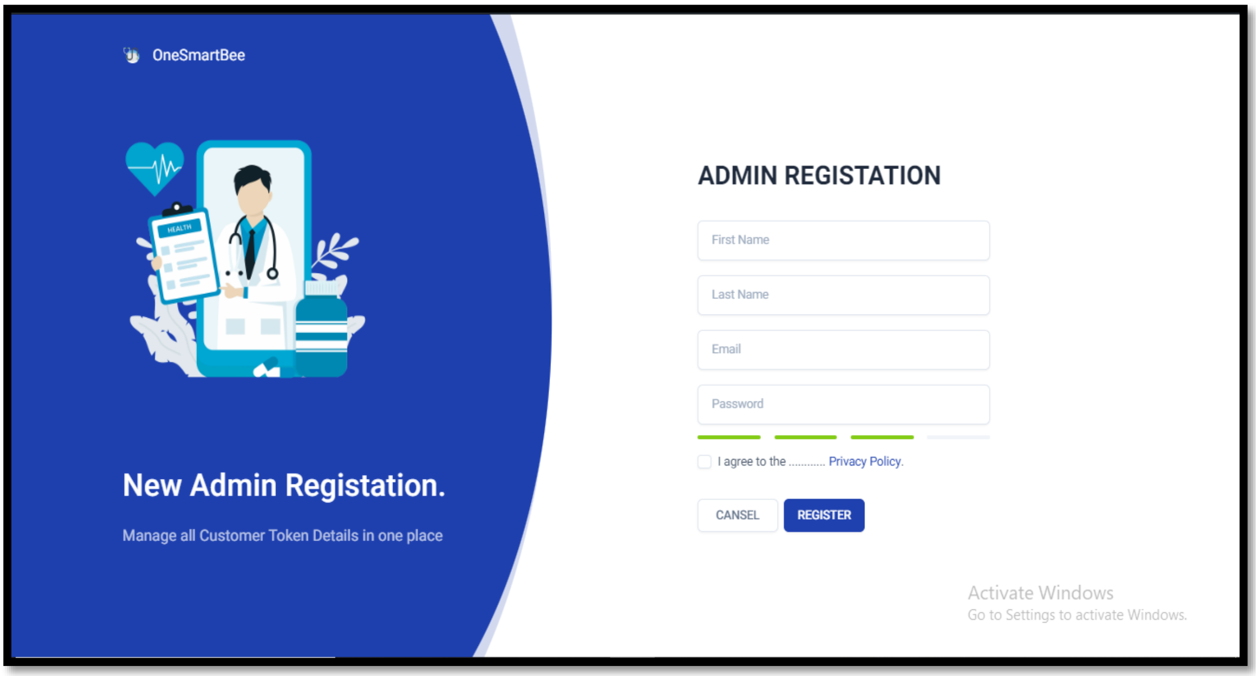


Media
Whatsapp
Phone Number
XXXXXXXXXX
Message Content
Image
Choose File No file chosen
Color
Type
Square

05. Admin Handling

<u>05.1.Admin Registation</u>

Creating new admins can only be done by super admins who are logged into the system. This page is used for that**.**



OneSmartBee
HEALTH
New Admin Registation.
Manage all Customer Token Details in one place
CANSEL
REGISTER
Activate Windows
Go to Settings to activate Windows.

ADMIN REGISTATION

First Name

Last Name

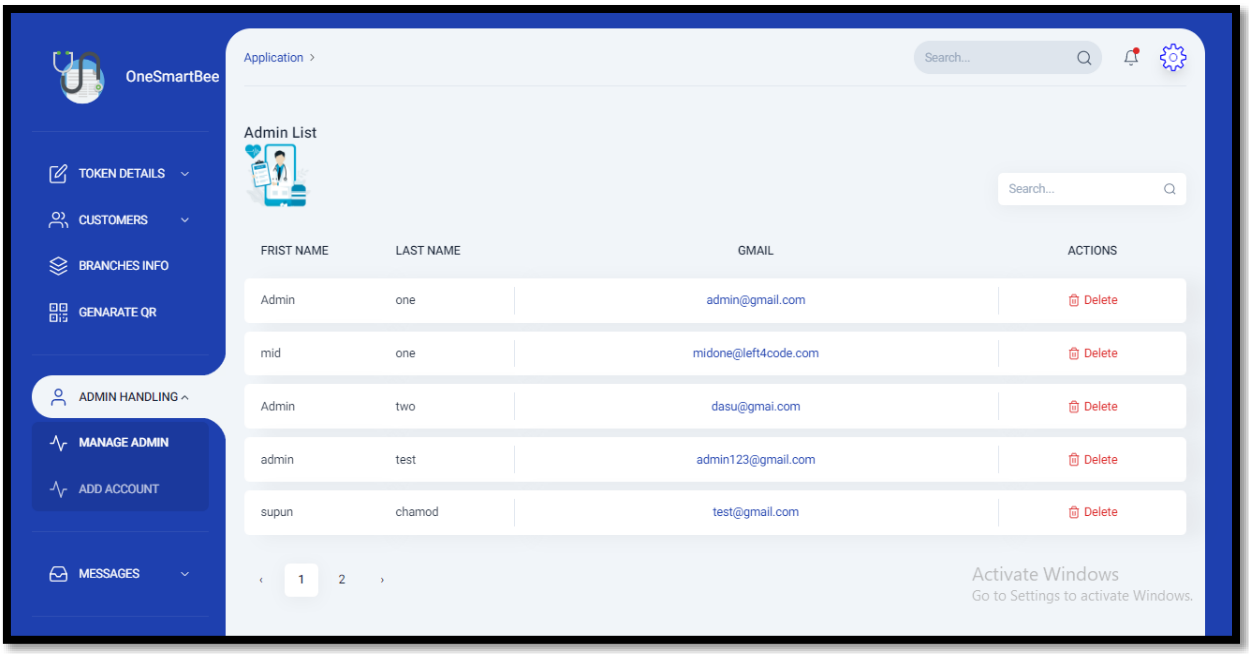
Email

Password

□ I agree to the .... Privacy Policy.

05.2.Admin information

The details of the admins who currently have access to the system are mentioned here.



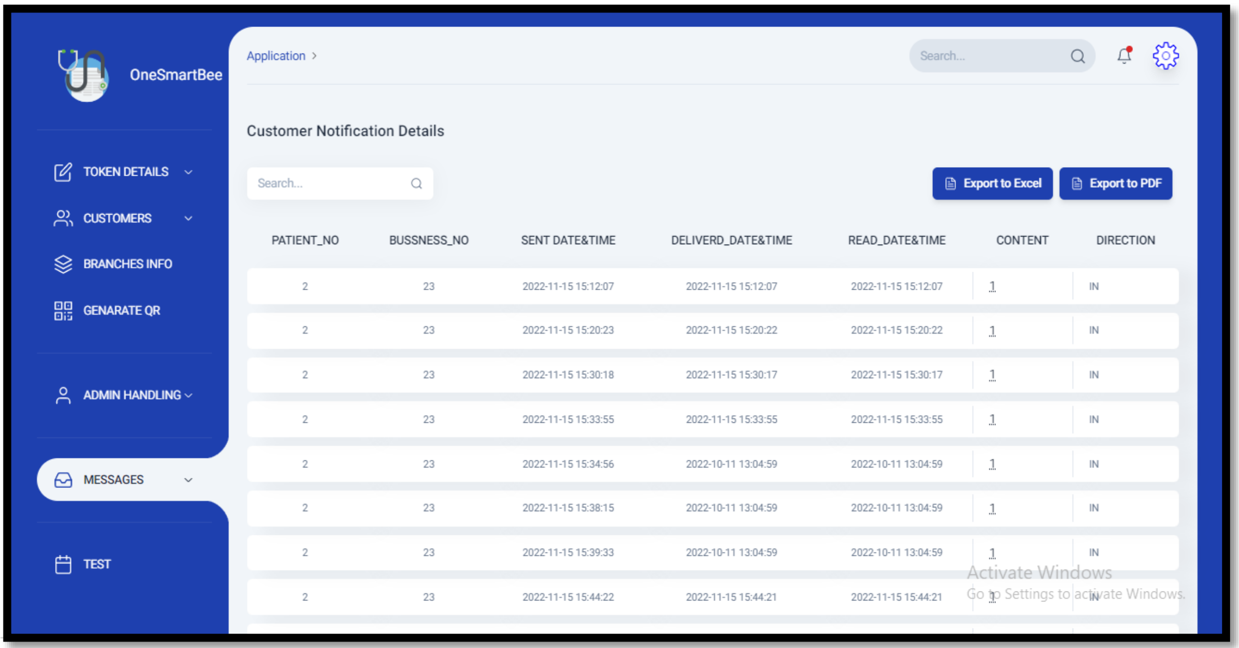
Admin List

FRIST NAME	LAST NAME	GMAIL	ACTIONS
Admin	one	admin@gmail.com	Delete
mid	one	midone@left4code.com	Delete
Admin	two	dasu@gmai.com	Delete
admin	test	admin123@gmail.com	Delete
supun	chamod	test@gmail.com	Delete

06. Message Handling

06.2.Admin information

This page displays messages and notifications communicated to customers through the system.

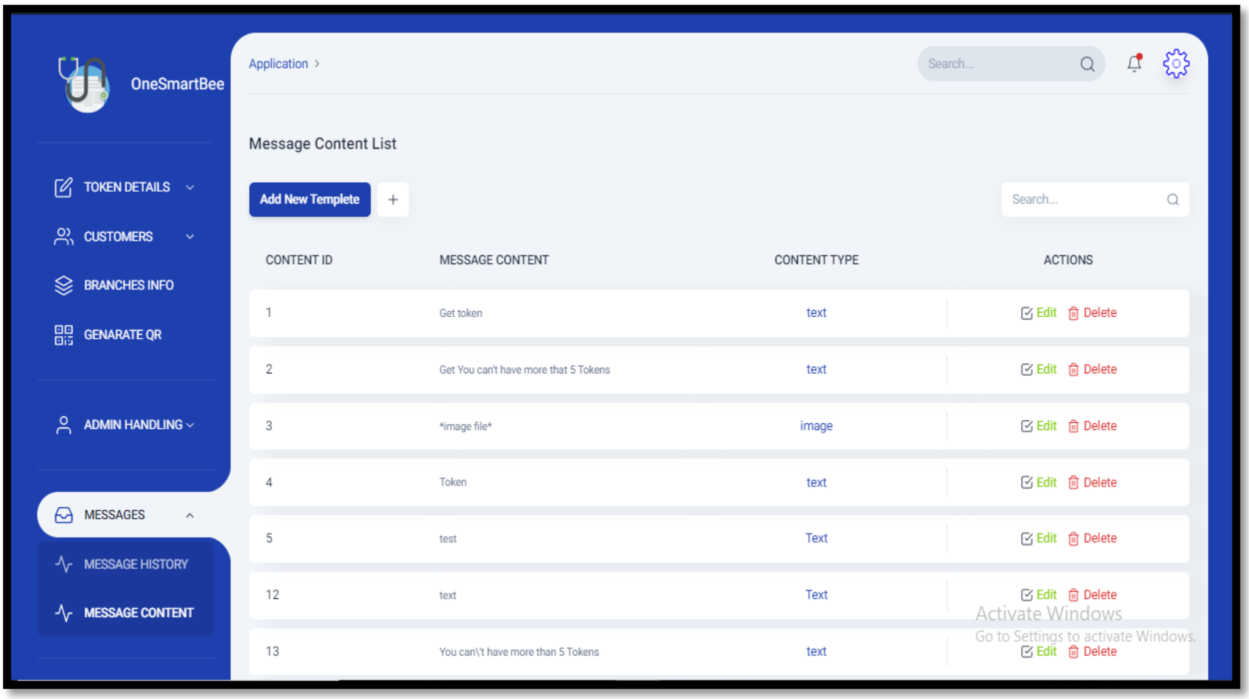


Customer Notification Details

PATIENT_NO	BUSSNESS_NO	SENT DATE&TIME	DELIVERD_DATE&TIME	READ_DATE&TIME	CONTENT	DIRECTION
2	23	2022-11-15 15:12:07	2022-11-15 15:12:07	2022-11-15 15:12:07	1	IN
2	23	2022-11-15 15:20:23	2022-11-15 15:20:22	2022-11-15 15:20:22	1	IN
2	23	2022-11-15 15:30:18	2022-11-15 15:30:17	2022-11-15 15:30:17	1	IN
2	23	2022-11-15 15:33:55	2022-11-15 15:33:55	2022-11-15 15:33:55	1	IN
2	23	2022-11-15 15:34:56	2022-10-11 13:04:59	2022-10-11 13:04:59	1	IN
2	23	2022-11-15 15:38:15	2022-10-11 13:04:59	2022-10-11 13:04:59	1	IN
2	23	2022-11-15 15:39:33	2022-10-11 13:04:59	2022-10-11 13:04:59	1	IN
2	23	2022-11-15 15:44:22	2022-11-15 15:44:21	2022-11-15 15:44:21	1	IN

06.2.Message content

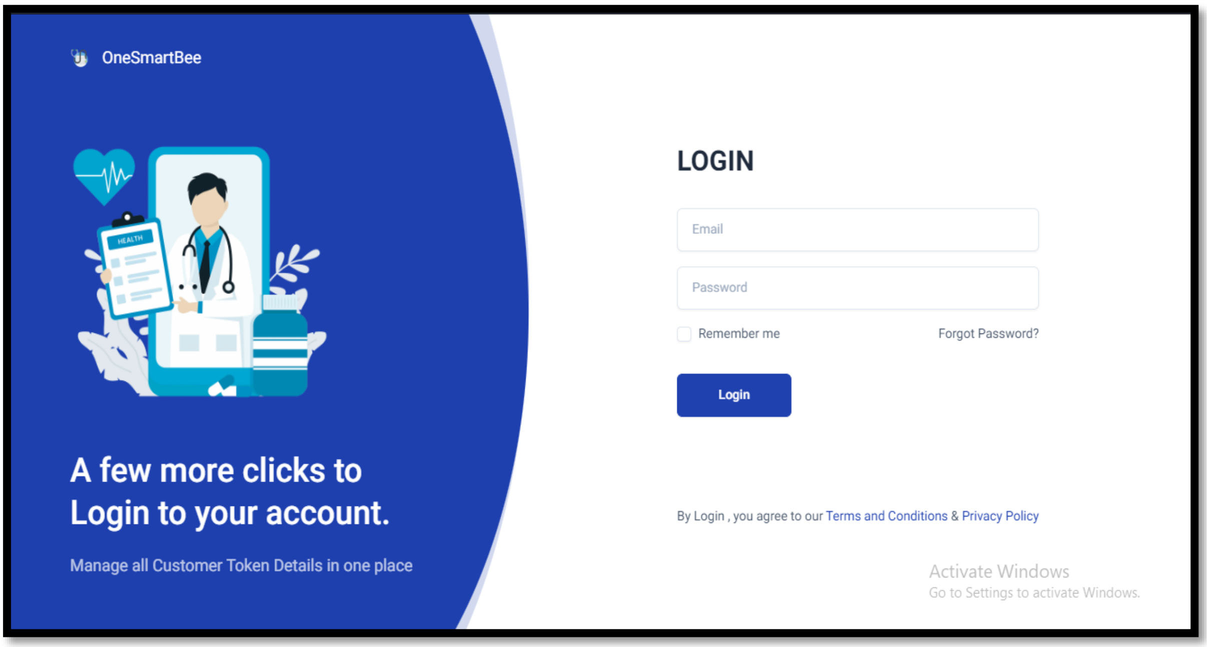
This page stores a list of common messages sent to users by the system. It can be edited and deleted by the admin. New message content can be added.



Message Content List

CONTENT ID	MESSAGE CONTENT	CONTENT TYPE	ACTIONS
1	Get token	text	Edit Delete
2	Get You can't have more that 5 Tokens	text	Edit Delete
3	*image file*	image	Edit Delete
4	Token	text	Edit Delete
5	test	Text	Edit Delete
12	text	Text	Edit Delete
13	You can\'t have more than 5 Tokens	text	Edit Delete

06. Admin Login



A few more clicks to Login to your account.

Manage all Customer Token Details in one place

□ Remember me

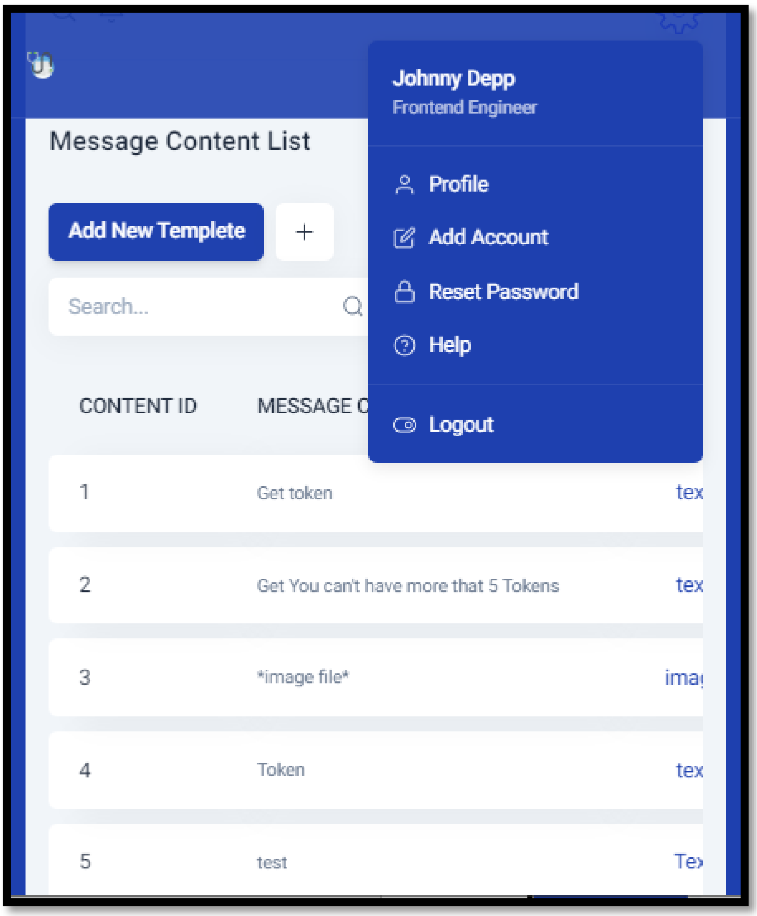
By Login , you agree to our Terms and Conditions & Privacy Policy

System Login

Email: admin@gmail.com

Password: password

06. Admin Logout



CONTENT ID	MESSAGE C	
1	Get token	tex
2	Get You can't have more that 5 Tokens	tex
3	*image file*	ima
4	Token	tex
5	test	Tex

In addition, I provide the best user-friendly user interfaces (Fig. UI) and high accuracy about the information.

Basically, this web application can identify two main types of users: admins (the system administrative team) and normal users. As an admin, they can manage each relevant Management when they login as an admin. Then admin can view the admin dashboard, select the relevant Management , and go ahead. On the Branch and Message template management side, they can insert newly created details , update, delete, and search some details.

When Basically , in this system (Fig.4.) can identify Eight classes as patient, user, admin, Branch, Message , Message Content, Ticket, QR, report, and pdf. Class patient is an ancestor of class user and admin if person is above user and admin in the inheritance hierarchy. User and Admin can access Token and branch classes. Additionally, only the administrator can access the QR, Patients ,Message, Message report class. That way he or she can generate a PDF for the entire system.

Admin Handling With Admin Role in laravel Jetstream

Laravel Jetstream is a beautifully designed application starter kit for Laravel and provides the perfect starting point for your next Laravel application. Jetstream provides the implementation for your application's login, registration, email verification, two-factor authentication, session management, API via Laravel Sanctum, and optional team management features.

Jetstream is designed using Tailwind CSS and offers your choice of Livewire or Inertia scaffolding.

Available Stacks

Laravel Jetstream offers your choice of two frontend stacks: Livewire and Inertia.js. Each stack provides a productive, powerful starting point for building your application; however, the stack you choose will depend on your preferred templating language.

#Livewire + Blade

Laravel Livewire is a library that makes it simple to build modern, reactive, dynamic interfaces using Laravel Blade as your templating language. This is a great stack to choose if you want to build an application that is dynamic and reactive, and is a great alternative to a full JavaScript framework like Vue.js.

When using Livewire, you may pick and choose which portions of your application will be a Livewire component, while the remainder of your application can be rendered as the traditional Blade templates you are used to.

Livewire Screencasts

If you're new to Livewire, check out the **screencasts available on the Livewire website**.

#Inertia + Vue

The Inertia stack provided by Jetstream uses Vue.js as its templating language. Building an Inertia application is a lot like building a typical Vue application; however, you will use Laravel's router instead of Vue router. Inertia is a small library that allows you to render single-file Vue components from your Laravel backend by providing the name of the component and the data that should be hydrated into that component's "props".

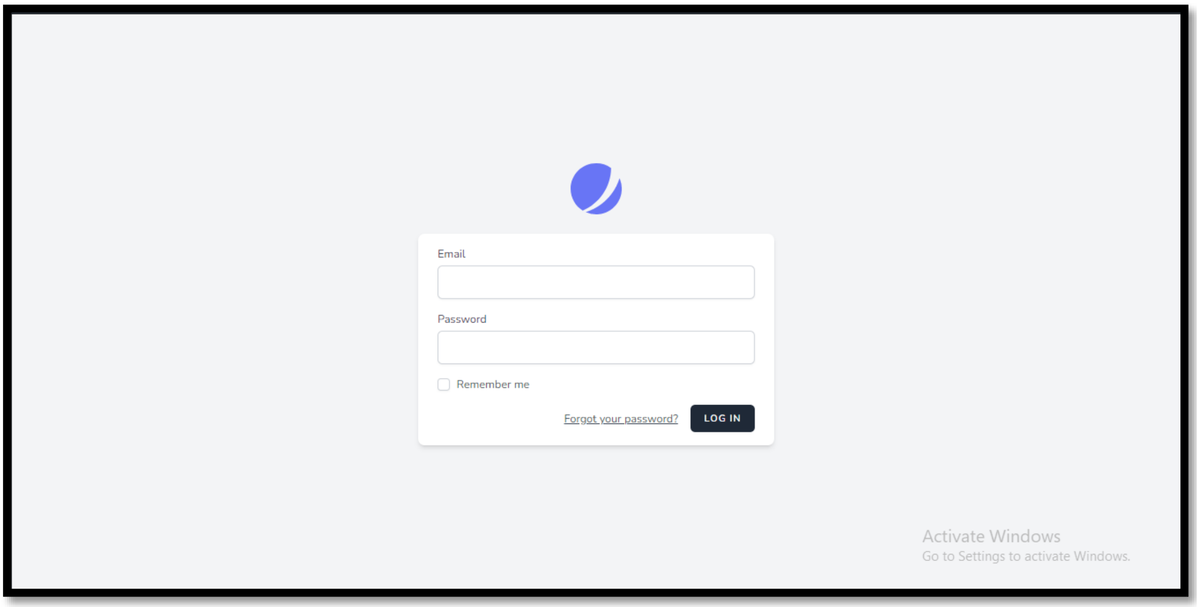
In other words, this stack gives you the full power of Vue.js without the complexity of client-side routing. You get to use the standard Laravel routing and view data hydration approaches that you are used to.

The Inertia stack is a great choice if you are comfortable with and enjoy using Vue.js as your templating language.

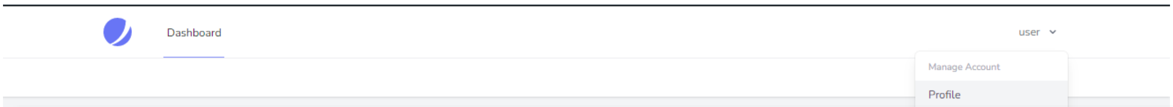
Inertia Screencasts

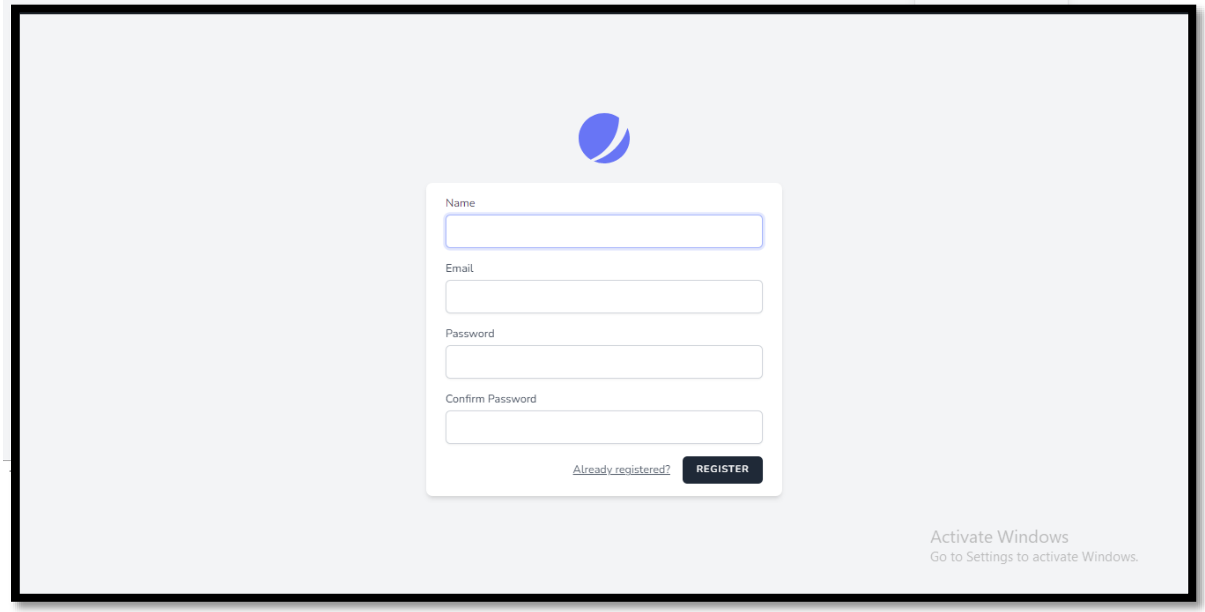
If you're new to Inertia, check out the **screencasts available on the Laracasts website**.

Login

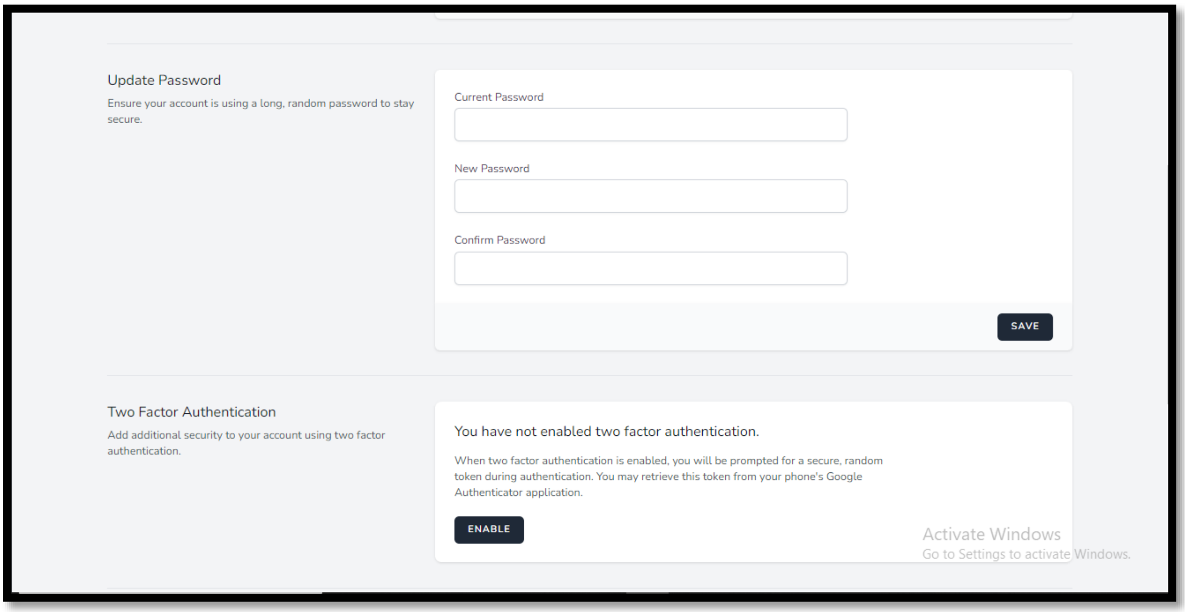


Register





Update password



Update Password

Ensure your account is using a long, random password to stay secure.

Current Password

New Password

Confirm Password

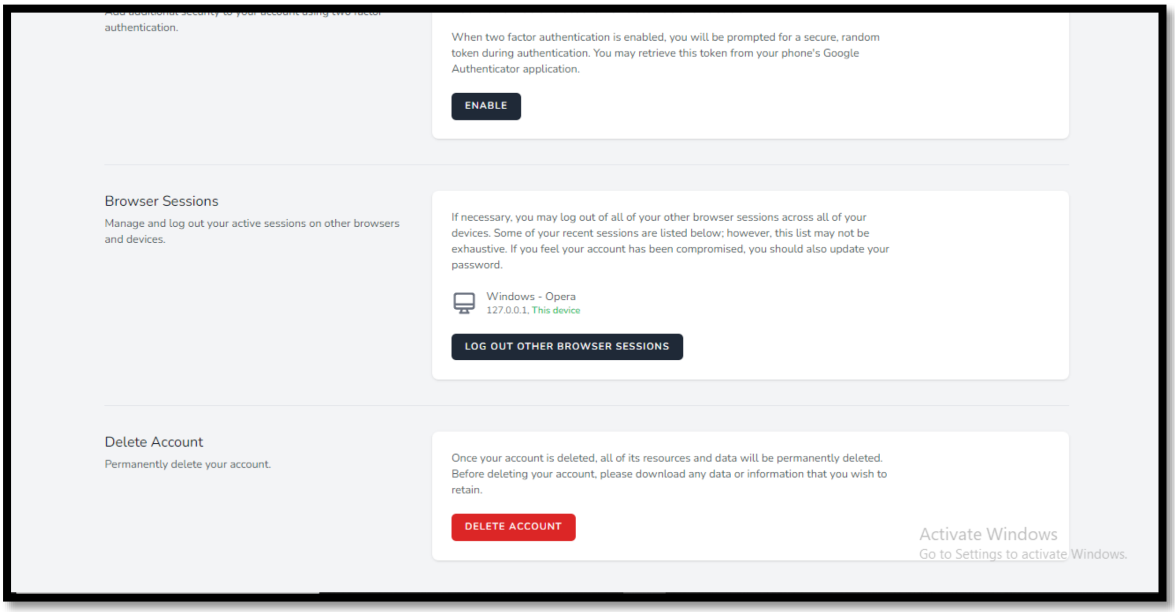
Two Factor Authentication

Add additional security to your account using two factor authentication.

You have not enabled two factor authentication.

When two factor authentication is enabled, you will be prompted for a secure, random token during authentication. You may retrieve this token from your phone's Google Authenticator application.

Delete Account



Add additional security to your account using two factor authentication.

When two factor authentication is enabled, you will be prompted for a secure, random token during authentication. You may retrieve this token from your phone's Google Authenticator application.

Browser Sessions

Manage and log out your active sessions on other browsers and devices.

If necessary, you may log out of all of your other browser sessions across all of your devices. Some of your recent sessions are listed below; however, this list may not be exhaustive. If you feel your account has been compromised, you should also update your password.

Windows - Opera
127.0.0.1, This device

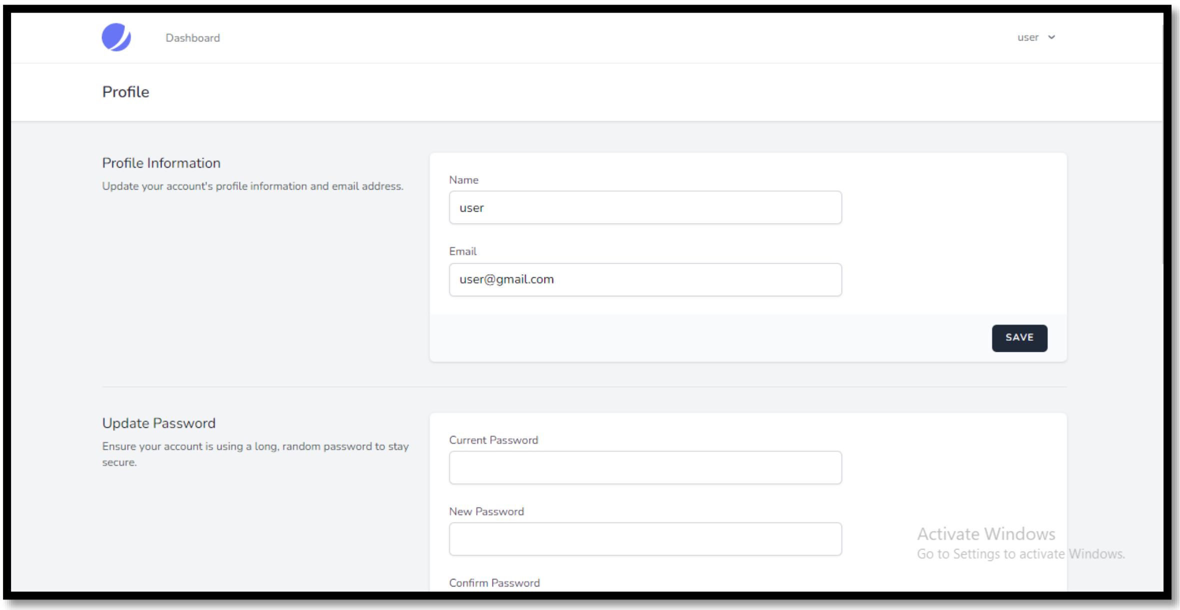
Delete Account

Permanently delete your account.

Once your account is deleted, all of its resources and data will be permanently deleted. Before deleting your account, please download any data or information that you wish to retain.

ENABLE
LOG OUT OTHER BROWSER SESSIONS
DELETE ACCOUNT
Activate Windows
Go to Settings to activate Windows.

Profile



Profile Information

Update your account's profile information and email address.

Name: user

Email: user@gmail.com

Update Password

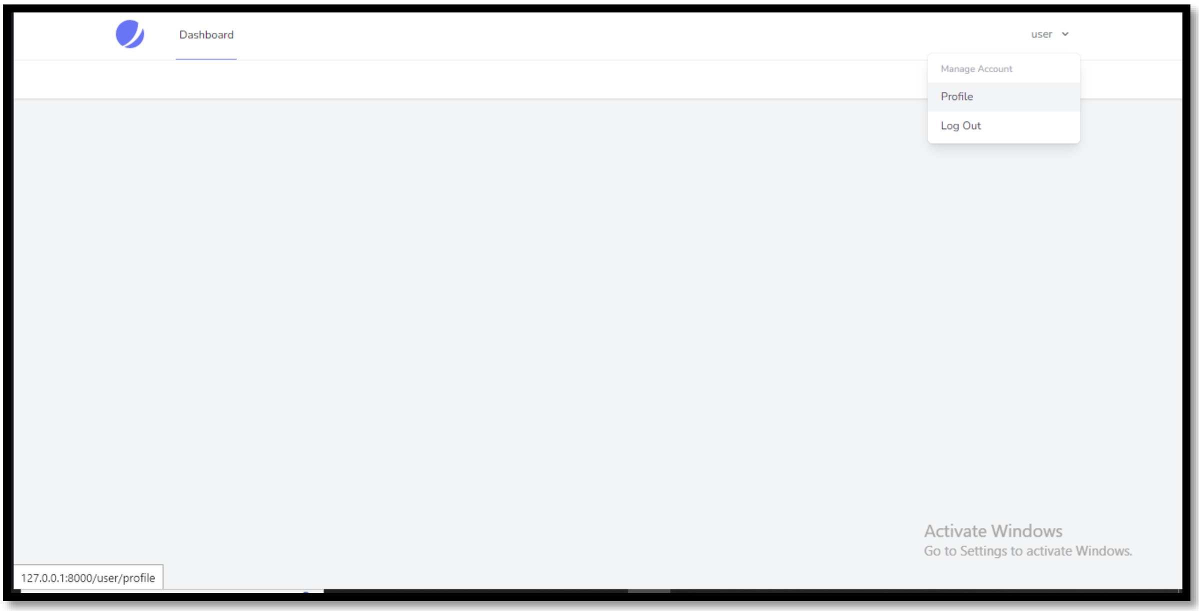
Ensure your account is using a long, random password to stay secure.

Current Password

New Password

Confirm Password

Dashboard



Dashboard
user
Manage Account
Profile
Log Out
Activate Windows
Go to Settings to activate Windows.
127.0.0.1:8000/user/profile