How to Install or Deploy

**Eyelix Ticketing System**

in Local IIS(XAMPP)

**Table of Contents**

**Objectives**

A ticketing system serves various purposes depending on the context in which it's implemented. Here are some common objectives:

**Efficient Issue Tracking**: One primary objective is to efficiently track and manage customer or user-reported issues, inquiries, or requests. Each "ticket" represents a specific task or issue that needs attention, allowing teams to prioritize and address them in a structured manner.

**Improved Communication**: Ticketing systems facilitate better communication between users or customers and support teams. They provide a centralized platform where users can submit their queries or issues and receive updates or resolutions from support staff. This helps in managing expectations and ensuring transparency in the resolution process.

**Streamlined Workflow**: Implementing a ticketing system helps in streamlining workflow processes within an organization. It enables teams to assign, prioritize, and track tasks effectively, reducing the chances of tasks falling through the cracks or getting lost in email chains.

**Enhanced Accountability**: With a ticketing system in place, it's easier to track the status and progress of individual tasks or issues. This fosters accountability among team members, as they are responsible for updating and resolving tickets within agreed-upon timelines.

**Data Analysis and Reporting**: Ticketing systems generate valuable data and metrics that can be used for analysis and reporting purposes. This includes information such as ticket volumes, resolution times, common issues, and customer satisfaction ratings. Analyzing this data helps in identifying trends, areas for improvement, and making data-driven decisions.

**Customer Satisfaction**: Ultimately, the goal of a ticketing system is to enhance customer satisfaction by providing timely and effective support. By ensuring that customer issues are addressed promptly and efficiently, organizations can improve their overall customer experience and build long-term relationships with their users or customers.

These objectives collectively contribute to increased operational efficiency, better customer service, and improved organizational performance.

**Disclaimer**

We all known that ticketing system is the most effective tool in utilizing customers concern in using the company Software such as Request, Incident , Change Order. In-order to manage this we developed a Ticketing Tool that will help them to communicate with us if there is a problem arise while using our software. Our ticketing system is created in open-source program language (PHP) which you can also use. The source code was placed in Github Repository and You can fork it anytime you want, to help us also improve our system we allowed the community to help us enrich our ticketing system in-order to be advance and can help our users to use it in easiest way.

Our Ticketing System has same common function with other ticketing system. We are glad to have this and as long as we use this system we will enhance this and add some features so that it could be better one and can be use by other company also.

You may have our revision documents for your future reference. Since we are in Revision 1. We see to it that we will improve this. Your suggestion and comment is highly recommended.

If you have any question or inquiries you are free to contact us. We will make sure that we will serve your better.

All information enclosed on this documentation is property of EyeLix Company.

**Contact Information**

**Website:**

**Email:**

**Telephone:**

**Mobile:**

**Discretion Notes (DPA)**

Discretion notes are additional comments or annotations added to a ticketing system to provide context or instructions that may not be appropriate or necessary for public viewing. Here are some scenarios where discretion notes might be used:

**Sensitive Information**: Discretion notes may contain sensitive information related to a customer's issue or personal details that should not be shared publicly. For example, if a customer provides confidential information in their support request, such as account numbers or passwords, this information should be recorded in a discretion note rather than in the public ticket thread.

**Internal Communication**: Discretion notes can be used for internal communication between team members working on the ticket. This could include instructions, reminders, or updates that are relevant only to the support staff handling the ticket.

**Escalation Instructions**: In cases where a ticket needs to be escalated to a higher level of support or management, discretion notes can contain instructions for the appropriate escalation process. This ensures that the escalation is handled smoothly and without causing confusion for the customer.

**Special Handling Instructions**: Discretion notes may include special handling instructions for the support staff, such as specific steps to follow or additional resources to consult when addressing the customer's issue. These instructions may not be relevant to the customer but are necessary for ensuring the ticket is handled appropriately.

**Legal or Compliance Notes**: If there are legal or compliance considerations related to the ticket, discretion notes can be used to document these details. This ensures that the support team is aware of any relevant regulations or policies that need to be followed when resolving the issue.

Overall, discretion notes serve to enhance communication and collaboration within the support team while also ensuring that sensitive information remains confidential and secure.

**Revision Notes**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | POC | Notes |
| **0001** | **06/12/24** | **Christopher Biagtan** | **Initial Draft for this document** |

**Notes: Any changes on this document must be tracked. Please update this Revision Notes so that we can monitor the changes of this document.**

**GITHUB Repository**

Our source code can be found on our GITHUB Repository by clicking this URL/link

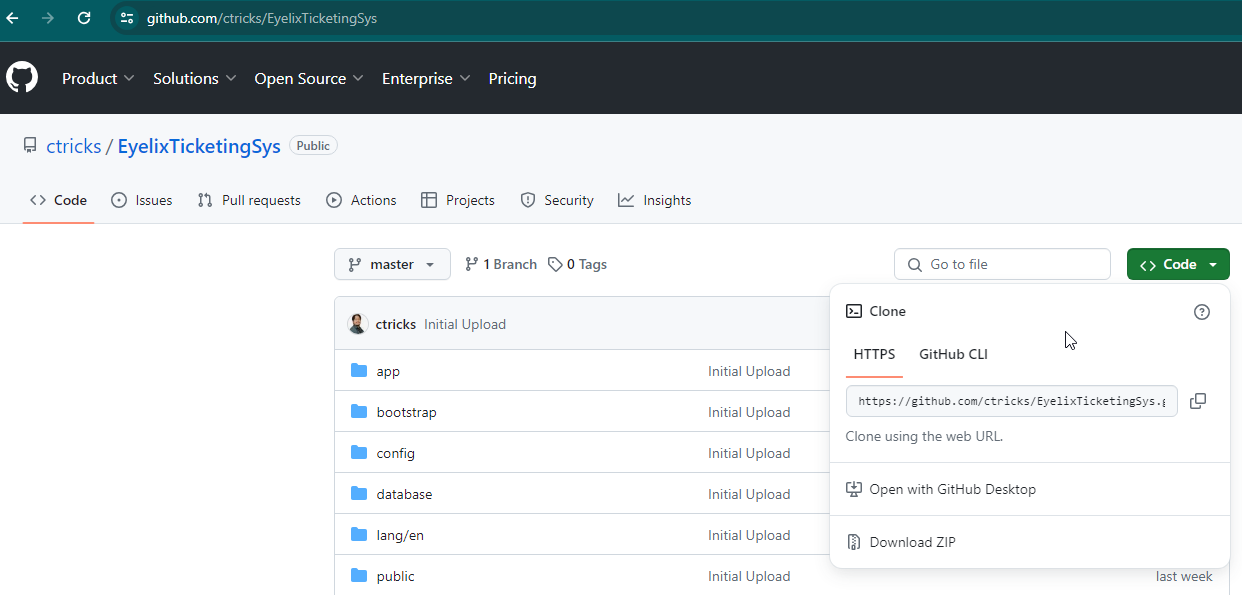
<https://github.com/ctricks/EyelixTicketingSys>

**About Ticketing System**

1. An Web Application which can be accessible around the globe using browser
2. Created in PHP Programming Language and with Laravel Framework. (Open-source)
3. Database to use is MySQL
4. Best to have it Chrome Browser

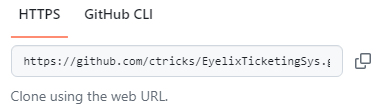
**Getting source code in GITHUB**

1. Open the link on browser : https://github.com/ctricks/EyelixTicketingSys



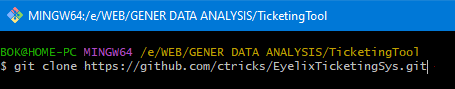
1. Click “Code” button and Copy the link in HTTPS.



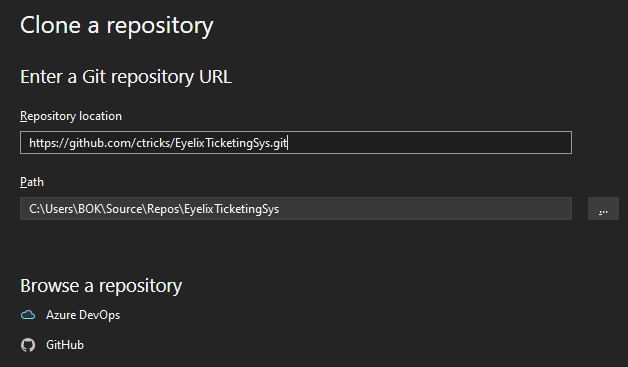


1. You may clone this code on your GitBash, Visual Studio , Github Desktop

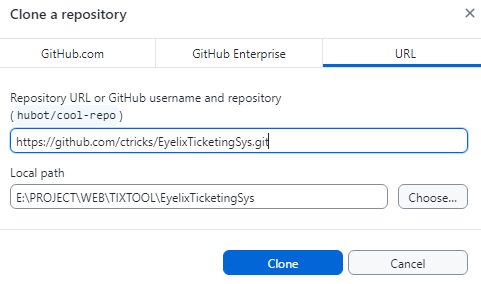
**For GitBash:**



**For Visual Studio**

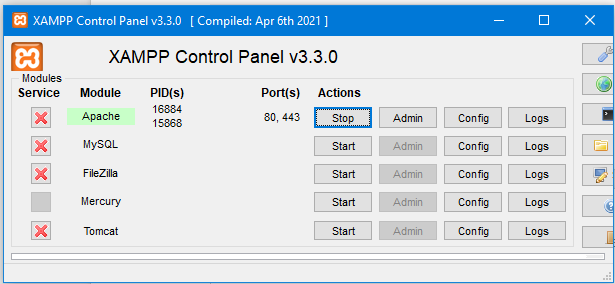


**For Github Desktop**



**Application to Use before Deployment**

1. Prepare your Localhost. (Make sure XAMPP is alread installed on your local)

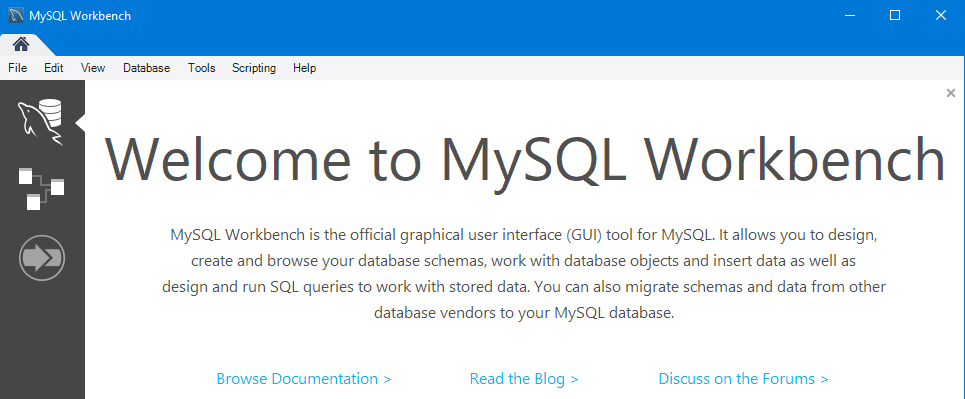


1. Make sure also you have MySQL Server(for this one we used 15.1), Type this on your command prompt (cmd):

Type: mysql -V

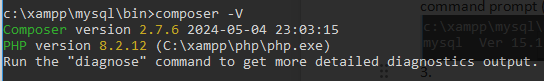


For IDE: We use MySQL WorkBench



1. Check also if you have Composer on your PHP(XAMPP),

Type: composer -V



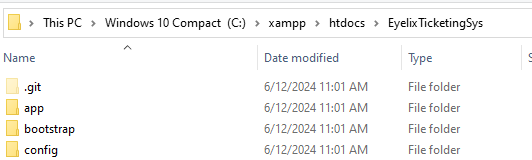
In case you don’t have this on your local you may download and install this on your local first.

Installer: https://getcomposer.org/Composer-Setup.exe

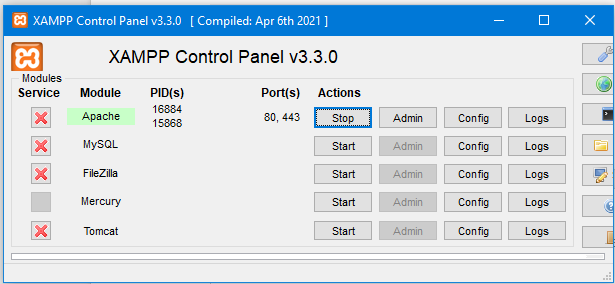
**Source-code Deployment on XAMPP (Localhost)**

1. Paste your source-code on your XAMPP Folder

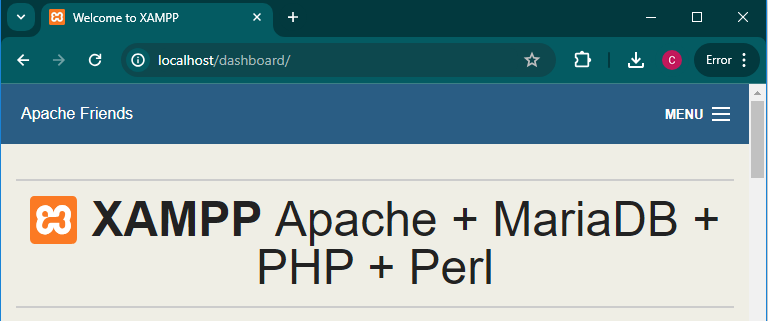
Path: C:\xampp\htdocs



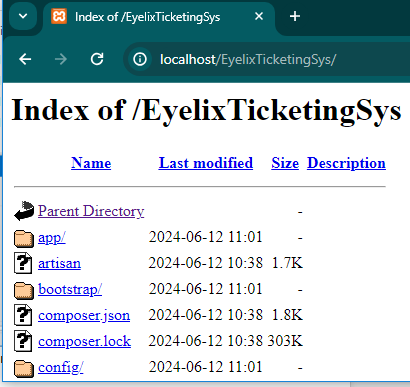
1. Start your XAMPP Server



1. Check if XAMPP is running: type in your browser

[http://localhost/dashboard/](http://localhost/dashboard/)

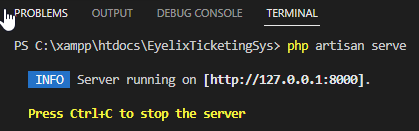
1. When you access your folder directly in Browser it will not load properly



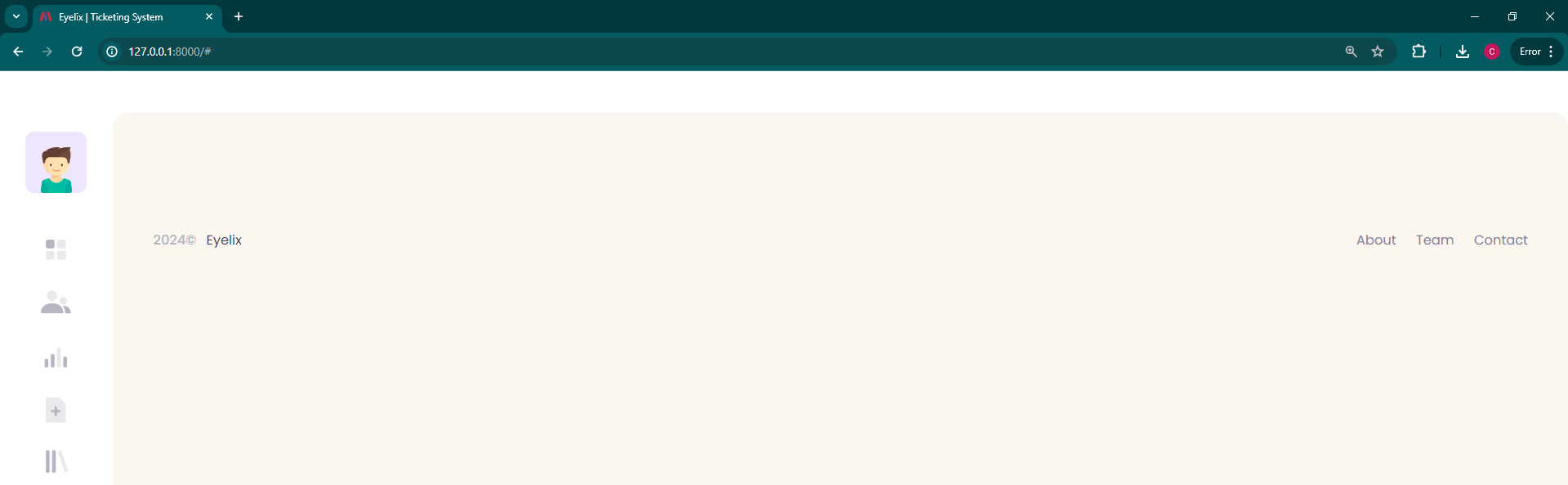
1. You need to start your Laravel - Composer in order to run the system.

**RUNNING THE APPLICATION (VS Code/Terminal)**

1. Open your source code on Visual Studio Code
2. Open your Terminal
3. Run this code: “php artisan serve”

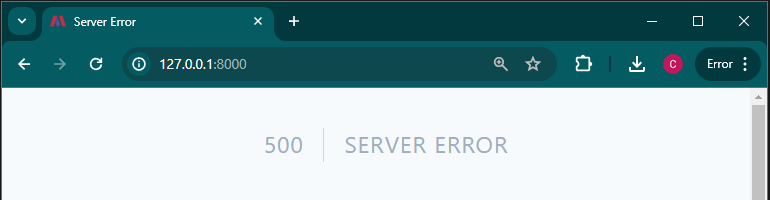


1. Run on your browser: <http://127.0.0.1:8000/>



Frequent Error Found on Installation:

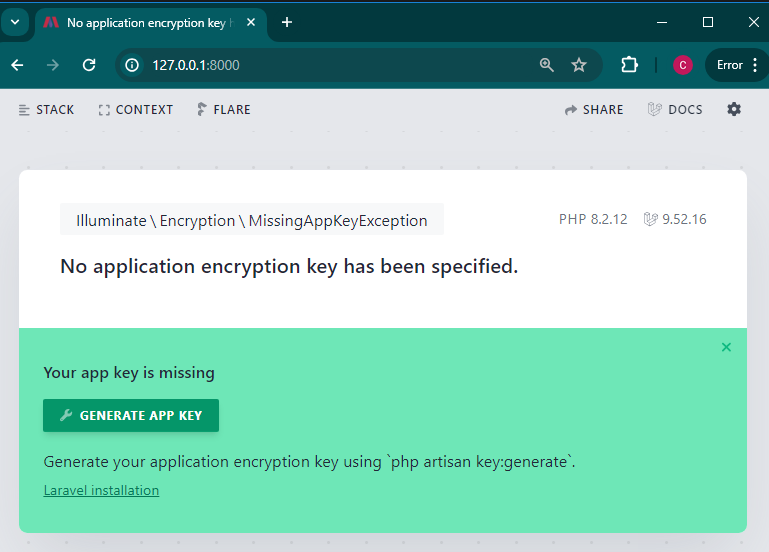
1. SERVER ERROR (500)



Resolution: Make sure that you have your Environment (.env) file

If you don’t have just make .env file on your TicketingSytem Folder

<https://github.com/laravel/laravel/blob/master/.env.example> and rename it “.env”

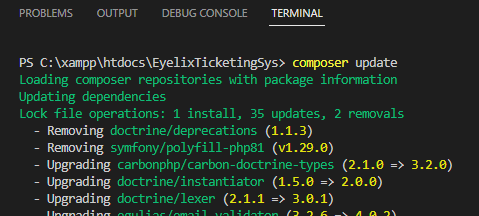


“Generate Key if needed”

And update your composer by

Type this on your terminal:

“composer update”



**Environment Set-up (sample .env)**

**APP\_NAME=Laravel**

**APP\_ENV=local**

**APP\_KEY=**

**APP\_DEBUG=true**

**APP\_TIMEZONE=UTC**

**APP\_URL=http://localhost**

**APP\_LOCALE=en**

**APP\_FALLBACK\_LOCALE=en**

**APP\_FAKER\_LOCALE=en\_US**

**APP\_MAINTENANCE\_DRIVER=file**

**APP\_MAINTENANCE\_STORE=database**

**BCRYPT\_ROUNDS=12**

**LOG\_CHANNEL=stack**

**LOG\_STACK=single**

**LOG\_DEPRECATIONS\_CHANNEL=null**

**LOG\_LEVEL=debug**

**DB\_CONNECTION=sqlite**

**# DB\_HOST=127.0.0.1**

**# DB\_PORT=3306**

**# DB\_DATABASE=laravel**

**# DB\_USERNAME=root**

**# DB\_PASSWORD=**

**SESSION\_DRIVER=database**

**SESSION\_LIFETIME=120**

**SESSION\_ENCRYPT=false**

**SESSION\_PATH=/**

**SESSION\_DOMAIN=null**

**BROADCAST\_CONNECTION=log**

**FILESYSTEM\_DISK=local**

**QUEUE\_CONNECTION=database**

**CACHE\_STORE=database**

**CACHE\_PREFIX=**

**MEMCACHED\_HOST=127.0.0.1**

**REDIS\_CLIENT=phpredis**

**REDIS\_HOST=127.0.0.1**

**REDIS\_PASSWORD=null**

**REDIS\_PORT=6379**

**MAIL\_MAILER=log**

**MAIL\_HOST=127.0.0.1**

**MAIL\_PORT=2525**

**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**

**VITE\_APP\_NAME="${APP\_NAME}"**

**Set-up your DATABASE**

1. Open your MySQL (Workbench)
2. Create schema “laravel”



1. Go to your Visual Studio Code and run the following in Terminal
2. Type:

php artisan migrate

php artisan migrate --seed

For migrate : This will only create table structure (First run)

For seed : This will include data on the table upon creation (First Run/Second Run)

For Error Found



Please check your .env File under this setting

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

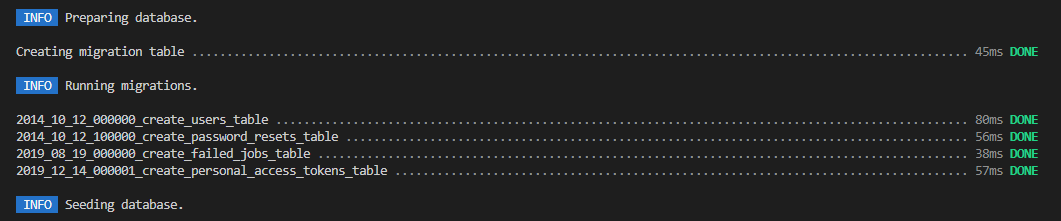
DB\_PORT=3306

DB\_DATABASE=laravel

DB\_USERNAME=root

DB\_PASSWORD= [your password here]

1. To know if success



1. Check your Schema

