# Email Parsing and Sending System with IP Verification using Laravel

***Jayesh Raju Sonkusare [1], Rupa Ashutosh Fadnavis [2]***

[1]*Student, Department of Information Technology, Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India.*

[2]*Professor, Department of Information Technology, Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India.*

*Corresponding Author: iamincperson@gmail.com*

**Abstract: -** Sending emails is one of the most common feature that each and every web application offers. Many websites use their emailing feature to send newsletters. Many websites use their emailing feature to collect feedback responses. Some websites use their emailing feature for authentication like OTP and password. So these PHP powered websites use the Mail () php method to send emails. However, in this case, security is often compromised as it is not the most secure way to send/receive emails on the website. The emails can be hacked and the emails can get stuck in the 'spam' section of the email service provider. These problems can be minimized if these PHP powered websites use a better framework like Laravel, Symphony, etc. Laravel, the framework with which this project is done, provides several advanced features to work with. It has different packages available in the market that makes the integration of emailing functionality effortlessly.

**Key Words: —** ***Emails, Laravel, MVC.***

## I. INTRODUCTION

Sending an electronic mail is a prevalent requisite for any web application. Some general utilizations of sending emails include verifying user registration, getting user's feedback, providing options to contact the site admin, etc. Laravel is an open-source PHP framework, which is robust and facile to understand. Laravel reuses the subsisting components of different frameworks which avails in engendering a web application. The web application thus designed is more structured and pragmatic. Laravel offers an affluent set of functionalities which incorporates the fundamental features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very opulent set of features which will boost the celerity of web development. If you are acclimated with Core PHP and Advanced PHP, Laravel will make your task more facile. It preserves a lot time if you are orchestrating to develop a website from scratch. Moreover, a website built in Laravel is secure and averts several web attacks.

Manuscript revised August 08, 2021; accepted August 09, 2021. Date of publication August 11, 2021.
This paper available online at www.ijprse.com
ISSN (Online): 2582-7898

The Laravel framework contains several packages to send emails from the Laravel project, which increases the popularity of Laravel. SMTP, Mailgun, and Amazon SES are utilized in Laravel for sending simple, transactional, and bulk emails. Laravel has an electronic mail-sending library denominated SwiftMailer to send an electronic mail with an electronic mail template with which this project is implemented. The aim and objective is to create a system which sends bulk emails to various email ids. The IP verification is just like the login/registration authentication. You can use this authentication rather than the IP one. It's just an alternative.

## II. REQUIREMENTS

*XAMPP***:** XAMPP is a free and open-source software. It is used for web server solution stack package which is developed by Apache Friends. It consists mainly of the Apache HTTP Server, MariaDB database, and interpreters for the scripts written in PHP and Perl programming languages. In short, it creates a local server on a local computer.

*Composer***:** Composer is a software which helps in installing Laravel.

## III. WORK DONE USING LARAVEL

Laravel is a web-development framework which can be used for sending emails. Laravel uses MVC design pattern. MVC stands for 'Model View Controller'. It represents architecture which developers undertake whilst constructing packages. With the MVC architecture, we have a look at the application shape as regards to how the records drift of our application works. MVC is a software architecture…that separates area/utility/business…logic from the relaxation of the user interface. It does this by way of setting apart the utility into 3 parts: the model, the view, and the controller. The figure below shows these parts.

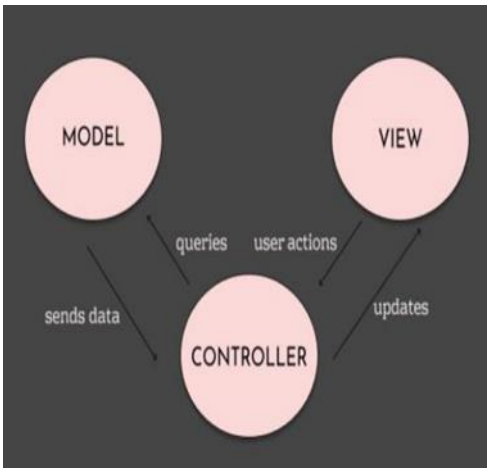


Fig.1. MVC

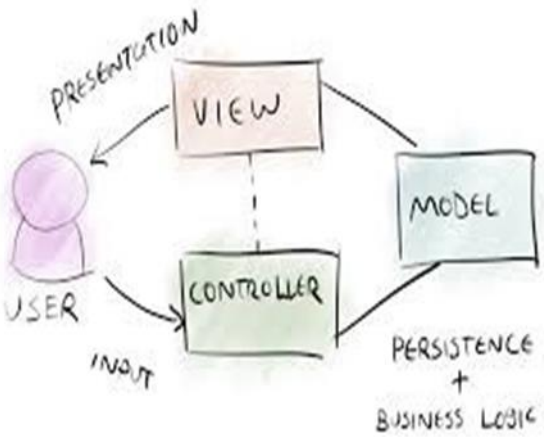


Fig.2. MVC

The Model is a representation of the actual appearance of the object in the code base. A View represents an interface in which the user uses to interact with the app. When the user performs the action, the Controller will be processed, and updates to the Model, if necessary. Let's take a look at a simple scenario. If a user has an e-commerce web site, the pages in the web site are nothing but provided by the View layer. When you click on a product to see more, the Controller layer handles the actions of the user. It can retrieve data from a data source with the help of the Model layer. The data collected is then arranged in the View layer, and is displayed to the user. In short, this process is 'Rinse and Repeat'. Ok so the first thing is to start Apache and MySql from XAMPP. This enables to have a local server and a database repository on the computer.

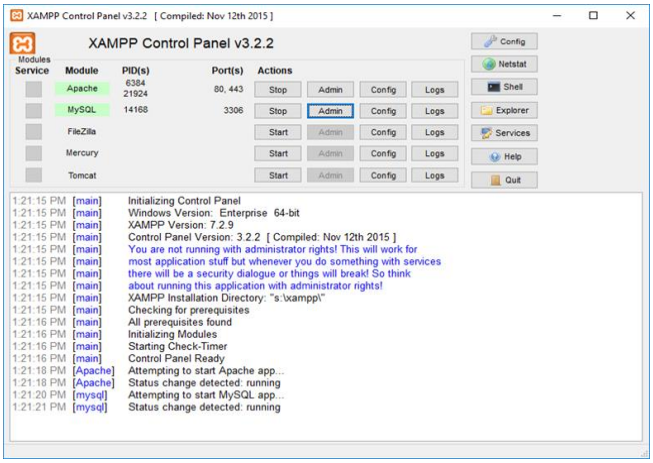


Fig.3. XAMPP Control Panel


Fig.4. Composer Command Line

Now after installing Laravel, the coding part starts. As told earlier, Laravel makes use of Model which contains the information to connect to the database, View which is nothing but the UI and the Controller processes the action and then it updates the Model. All this is coded behind-the-desk. But how does it work in a nutshell? If the IP which is stored in the database and the IP of the user through which he/she is using the website is different, the user is not allowed to send any emails otherwise its ok. It's the login/registration

authentication, just the alternative. Ok now the valid user gets into the system and then he/she needs to upload the CSV file which contains all the email ids to which the email template is to be sent. The UI has options to preview the mail template and to open the mail template in a new window. This webpage is responsive. Ok so when the user clicks the send all button, all the data is pushed to the 'jobs' table in the database. The queue then looks at the table and starts sending the emails one-by-one. It runs in the background so that the user is free to do anything at the frontend. Each email sending process is associated with its new job (task). Now there can be two possible scenarios: either is email is sent or not. In each case, the corresponding data is deleted from the 'jobs' table as it only stores the email process which is ongoing and which is reserved for the future. When the email is sent, it is ok but what if the email is not sent due to some error? We need to keep a track of this. So in this case, the corresponding data is stored in the 'failed_jobs' table along with the error. A cool thing about Laravel is that, you need not work on the 'jobs' and 'failed_jobs' table functionality. Laravel provides you with an option to automate this process once you schedule every job for every email sending process. The following figure will make you understand how this works.

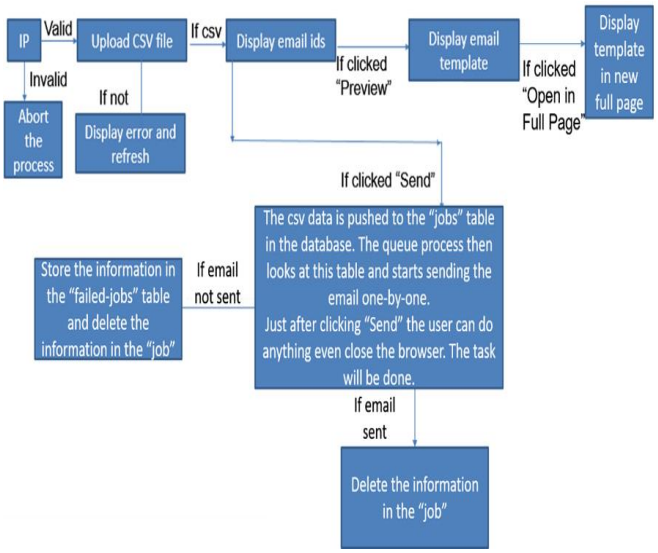


Fig.5. Front-End Work Chart

## IV. RESULTS

The following UI is shown when a registered client uses it. The UI is based on the above diagram and is kept as simple as possible so that the client understands what he/she needs to do.

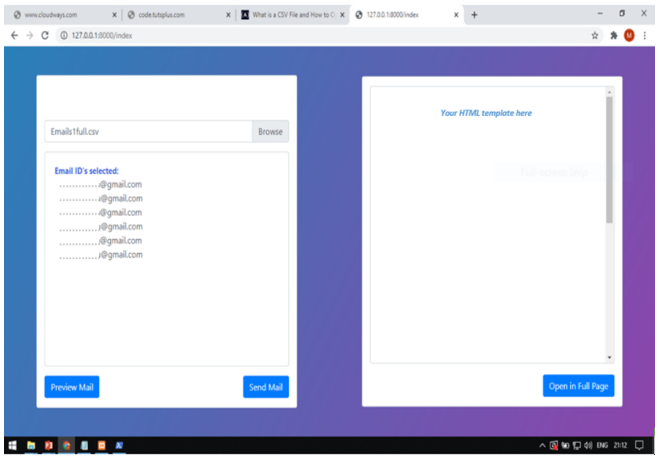


Fig.5. UI

The following is the background process which gets started as soon as the send mail button is clicked. Of course, the command for queue is to be entered to see this.

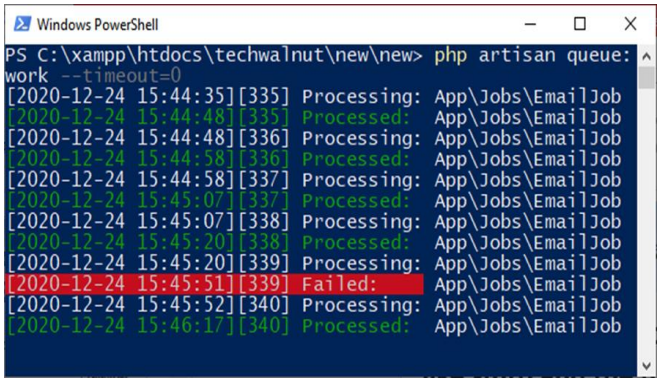


Fig.6. Background queue process

So all the ongoing and the reserved email processes are stored in the 'jobs' table. Once the ongoing task is completed, this data is deleted. However, if the ongoing task fails, this data is stored in the 'failed_jobs' table along with what caused the error and then this data in the 'jobs' table is deleted. These tables thus helps in tracking.

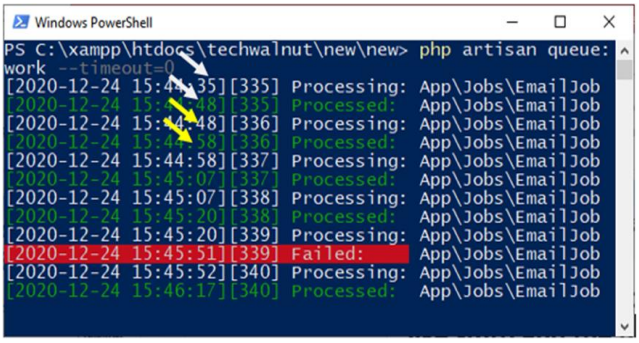


Fig.7. Background queue process time

The time taken for sending depends on the email template, the email driver like smtp and the network you use. It takes approximately 10 to 15 seconds for one email. For bulk mails like 500 – 1000 emails, this would be time-consuming and would bother the clients. Therefore, the best bet was to move this process to the background. The client can do anything once he/she pushes the send button and even close the browser. The job will be possibly done. The failed ones will be tracked down in the database.

## V. CONCLUSION

Taking into account the traditional web design methods that will result in major drawbacks, and the time-consuming techniques and other issues, the article presents the methods for the design and implementation of a web network based on the Laravel framework. Laravel allows the development of a description of an automatic process, certain relationships that are not related to the business logic. In this paper, we have designed and implemented a simple Laravel model,which allows us to automate the processing of this project i.e. automate the processing of emails in detail. The project proved that making Laravel framework work for automating the email process helps in scalability, and reliability, allowing in increase of the efficiency of the design. This project, in the end, turned out to be pretty satisfactory.

## REFERENCES

[1]. Utilization of Laravel Framework for Development of Web Based Recruitment Tool - Vishal V. Parkar, Prashant P. Shinde, Sanket C. Gadade, Prasad M. Shinde.
[2]. A comparative study of laravel and symfony PHP frameworks - Majida Laaziri1, Khaoula Benmoussa, Samira Khoulji, Kerkeb Mohamed Larbi, Abir El Yamami.
[3]. Design and implementation of web based on Laravel framework - He Ren Yu.
[4]. Analysis on one of the popular frameworks – Zend - Samuel Warnoff, S. Smitch, Harl Pent.
[5]. A Research Paper on Yii Framework - Varunala Bhagat.