

**Project Report**

**on**

**Influencer Website**

Submitted to

**LOVELY PROFESSIONAL UNIVERSITY**

for

**INT 221 MVC PROGRAMMING**

**Submitted By Submitted to**

**SARANSH DAS Navneet Kaur**

**11905423 Assistant Professor**

**LOVELY FACULTY OF TECHNOLOGY & SCIENCES**

**LOVELY PROFESSIONAL UNIVERSITY**

**PUNJAB**

**April 2022**

**Table of Contents**

**Chapter Particulars Page No.**

1. Introduction……………………………………………………………... 2
2. Technologies used………………………………………………………. 3
3. Modules………………………………………………………................. 5
4. Website snapshots………………………………………………………. 7
5. GitHub link………………………........................................................... 12

References…………………………………………………………………………… 13

**CHAPTER 1**

**INTRODUCTION**

* 1. **Introduction**

**1.1.1 What is MVC Programming?**

MVC (Model-View-Controller) is a pattern in software design commonly used to implement user interfaces, data, and controlling logic. It emphasizes a separation between the software's business logic and display. This "separation of concerns" provides for a better division of labor and improved maintenance. Some other design patterns are based on MVC, such as MVVM (Model-View-ViewModel), MVP (Model-View-Presenter), and MVW (Model-View-Whatever).

**1.2 What is Laravel?**

Laravel is a web application framework with expressive, elegant syntax. We’ve already laid the foundation — freeing you to create without sweating the small things. Laravel values beauty. It enables clean code just as much as we want. Simple, elegant syntax puts amazing functionality at our fingertips. Every feature has been thoughtfully considered to provide a wonderful developer experience.

**1.2.1 Features of Laravel**

Laravel's "Eloquent" ORM provides best-in-class database abstraction without the headaches. Querying and updating data without breaking a sweat. Eloquent pairs perfectly with MySQL, Postgres, SQLite, and SQL Server. It also has authentication. It provides scaffolding for secure, session-based authentication, while Laravel Sanctum provides painless authentication for APIs and mobile applications. It is committed to delivering the best testing experience. No more brittle tests that are a nightmare to maintain. Beautiful testing APIs, database seeding, and painless browser testing let you ship with confidence. It is for everyone whether you have been programming for 20 years or 20 minutes. It's for architecture astronauts and weekend hackers.

**CHAPTER 2**

**TECHNOLOGIES USED**

**2.1 Name of Technologies Used**

**2.1.1 Laravel Framework**

Laravel is a back-end PHP-based and open-source framework used for building a wide range of custom web applications. It's an entirely server-side framework that manages data with the help of Model-View-Controller (MVC) design which breaks an application back-end architecture into logical parts.

**2.1.2 PHP**

PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

**2.1.3 XAMPP**

XMPP is a short form for Extensible Messaging Presence Protocol. It's protocol for streaming XML elements over a network in order to exchange messages and presence information in close to real time. This protocol is mostly used by instant messaging applications like WhatsApp.

**2.1.4 HTML**

HTML (HyperText Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

**2.1.5 CSS**

CSS, or “Cascading Style Sheets,” is used for styling and laying out webpages. It can be used to adjust content size, spacing, color and font or add decorative features, such as animations or split content into columns.

**2.1.6 JavaScript**

JavaScript is a single-threaded programming language, which means it has a single Call Stack. Therefore, it can do one thing at a time. The Call Stack is a data structure which records basically where in the program we are. If we step into a function, we put it on the top of the stack. The way it works is interesting. Inside a normal Web page, you place some JavaScript code. When the browser loads the page, the browser has a built-in interpreter that reads the JavaScript code it finds in the page and runs it.

**2.1.7 SCSS**

In SCSS, you can save any CSS value (even with units) in variables. Variables are defined using the $ symbol. When Sass is converted to CSS, all the variables are replaced with their original values. SCSS variables are useful to keep fonts, colors, and other values consistent thought a website or web app.

**2.1.8 Bootstrap**

Bootstrap is a potent front-end framework used to create modern websites and web apps. It's open-source and free to use, yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms. Bootstrap also supports JavaScript extensions.

**2.2 Websites used for UI Templates/Images**

**2.2.1 Pexels**

Pexels is a free stock photo and video website and app that helps designers, bloggers, and everyone who is looking for visuals to find great photos and videos that can be downloaded and used for free. If you see a photo or video you like, simply download it for free

**2.2.2 BootstrapDASH**

BootstrapDash is a website that provides High-quality themes and templates designed by industry experts and developed with the highest coding standards. Loved and trusted by developers, freelancers, and companies worldwide.

**CHAPTER 3**

**MODULES**

**3.1 Directory Structure**

**3.1.1 Models**

The model manages fundamental behaviors and data of the application. It can respond to requests for information, respond to instructions to change the state of its information, and even notify observers in event-driven systems when information changes. This could be a database or any number of data structures or storage systems. In short, it is the data and data-management of the application. This project contains 4 Models namely Cart.php , Product.php, Order.php and User.php .

**3.1.2 Views**

The view effectively provides the user interface element of the application. It’ll render data from the model into a form that is suitable for the user interface.

The controller receives user input and makes calls to model objects and the view to perform appropriate actions. There are 2 sets of Views made into 2 separate folders for better readability. The 2 folders are for the User and Admin. Both View Folders have views made for the Dashboard, Navbar, Product etc.

**3.1.3 Controllers**

Controllers can group related request handling logic into a single class. For example, a UserController class might handle all incoming requests related to users, including showing, creating, updating, and deleting users. By default, controllers are stored in the app/Http/Controllers directory. In this project I have made 2 main controllers, namely Admin and Home.

**3.2 Name of Laravel Libraries Used**

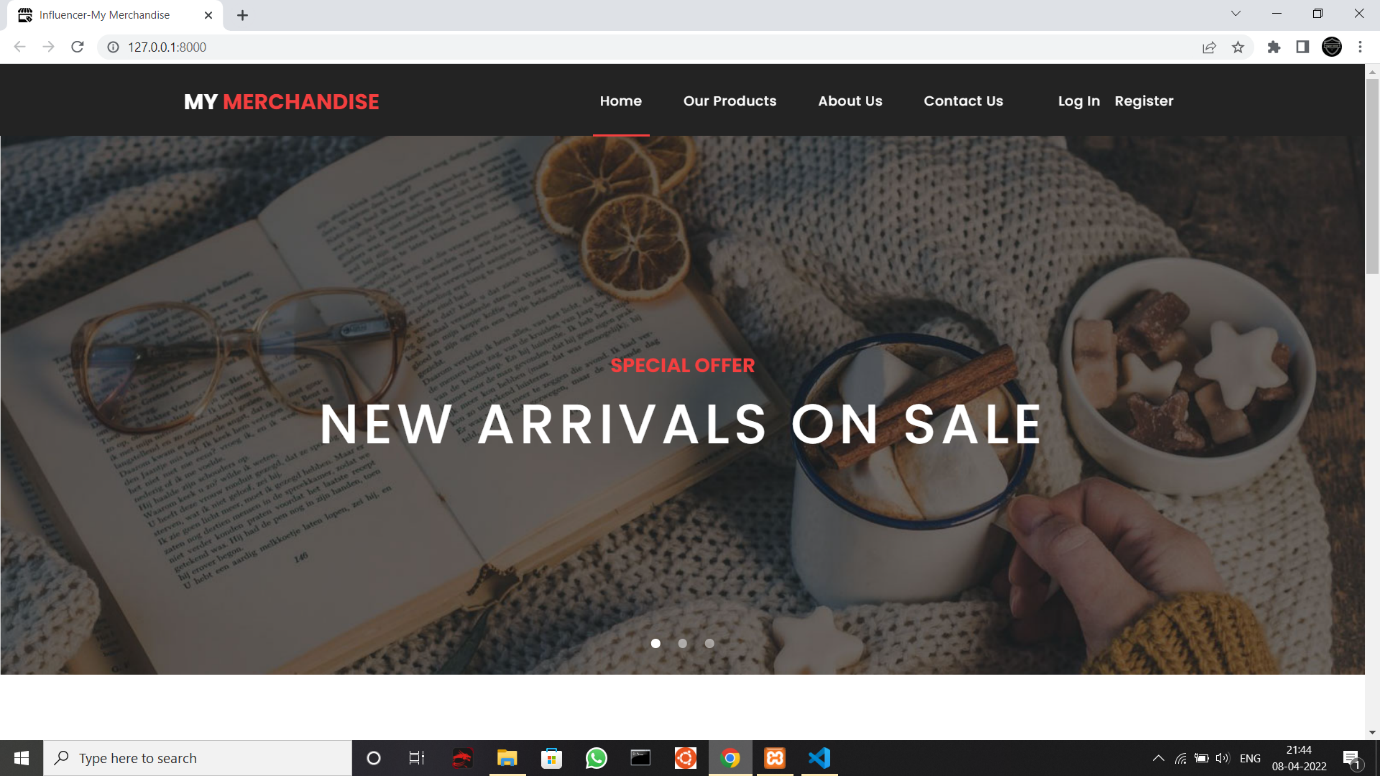
**3.2.1 Laravel Jetstream**

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.

**CHAPTER 4**

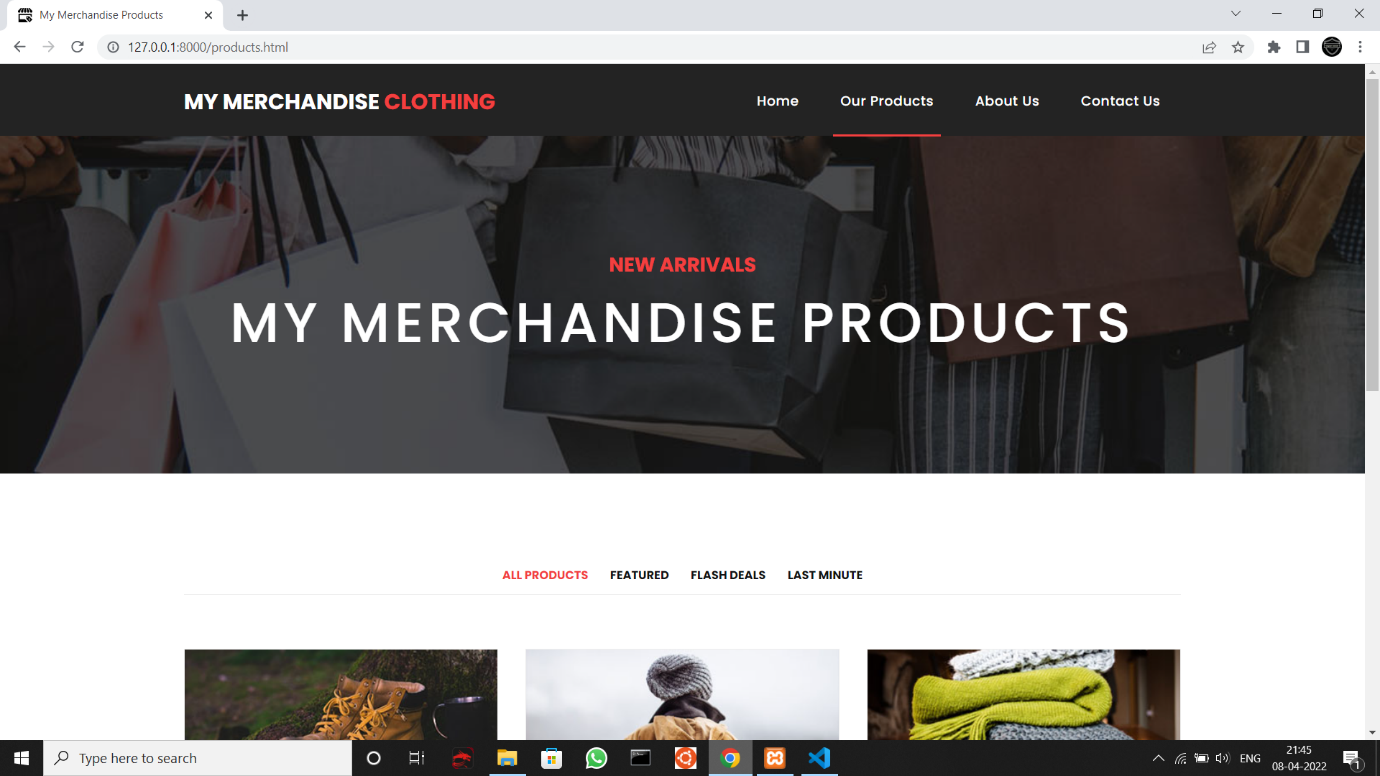
**WEBSITE SNAPSHOTS**

**4.1 Overview of Website**

**4.1.1 Home Page**

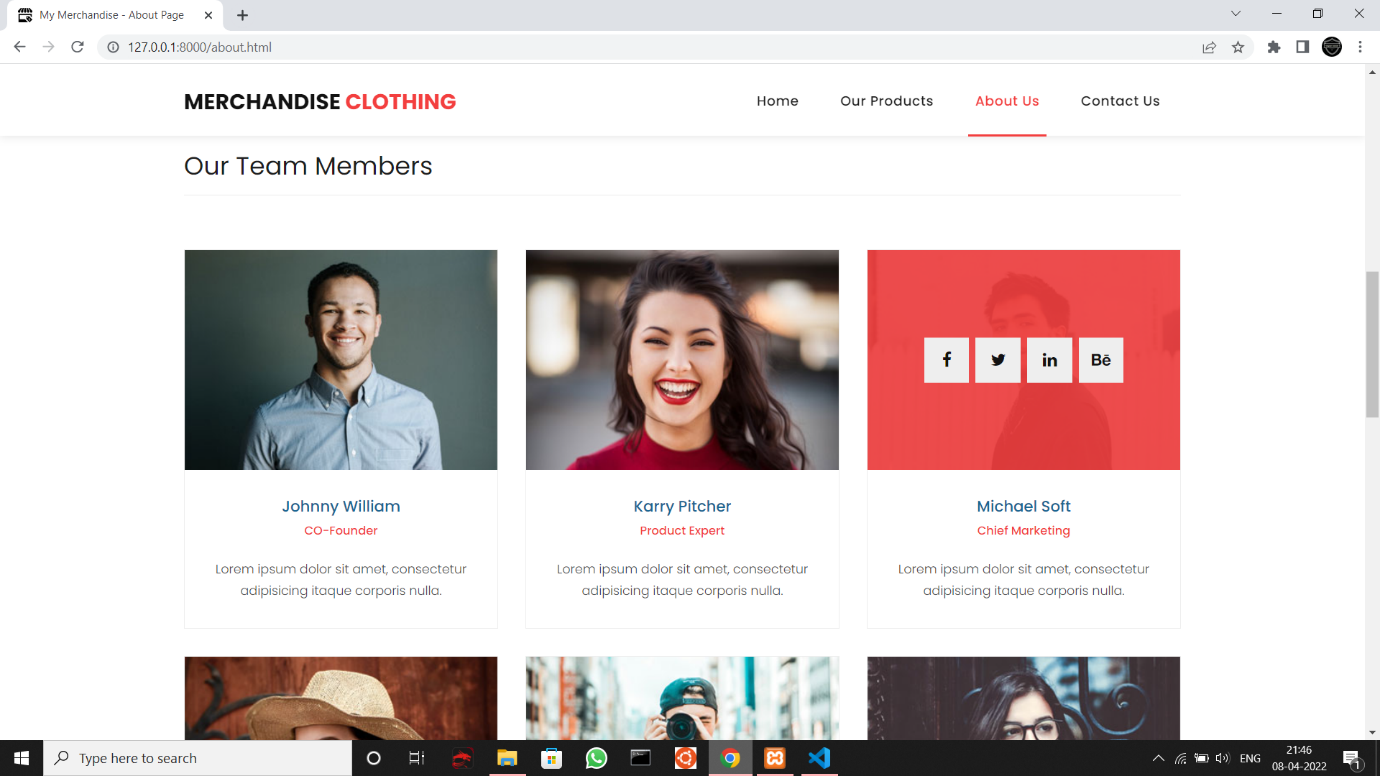
**Figure 4.1 Home Page**

**4.1.2 Product Page**

****

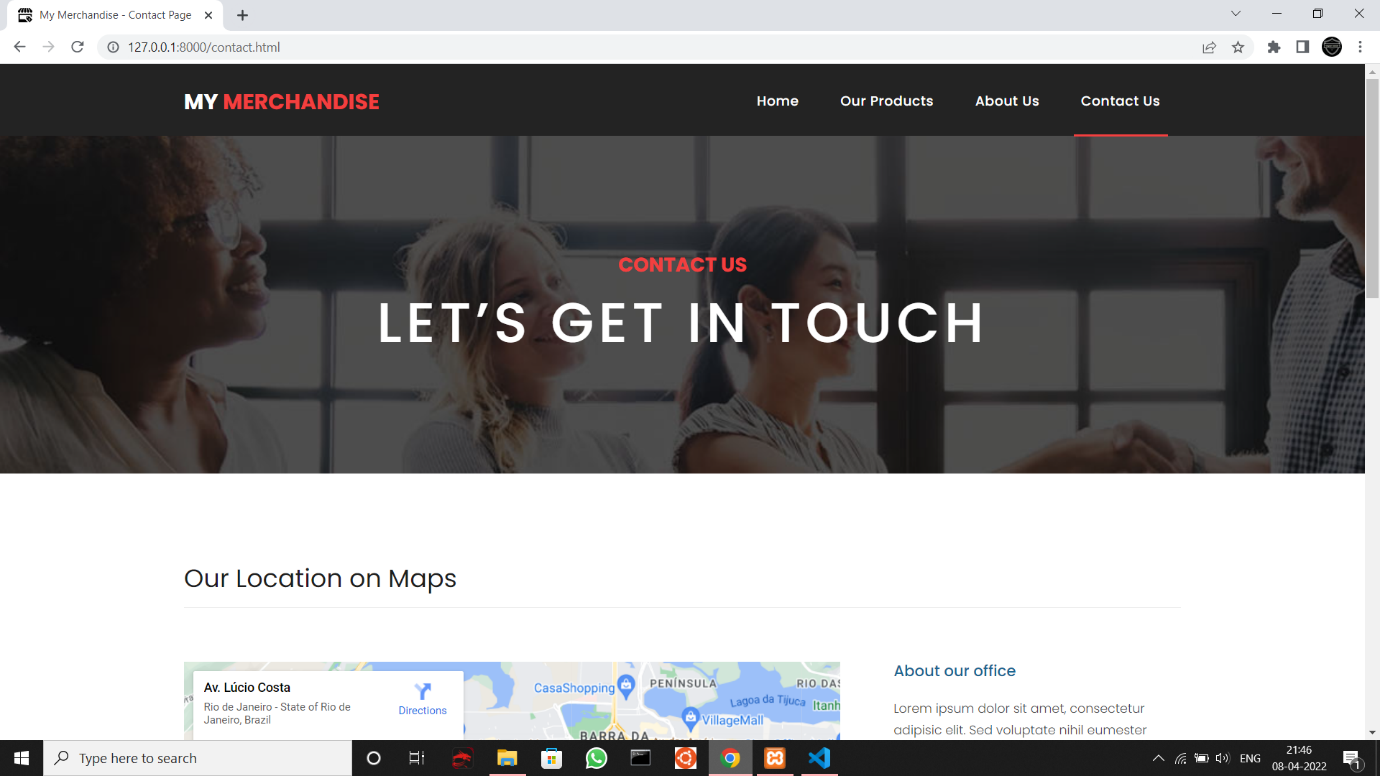
**Figure 4.2 Product Page**

**4.1.3 About Us Page**

****

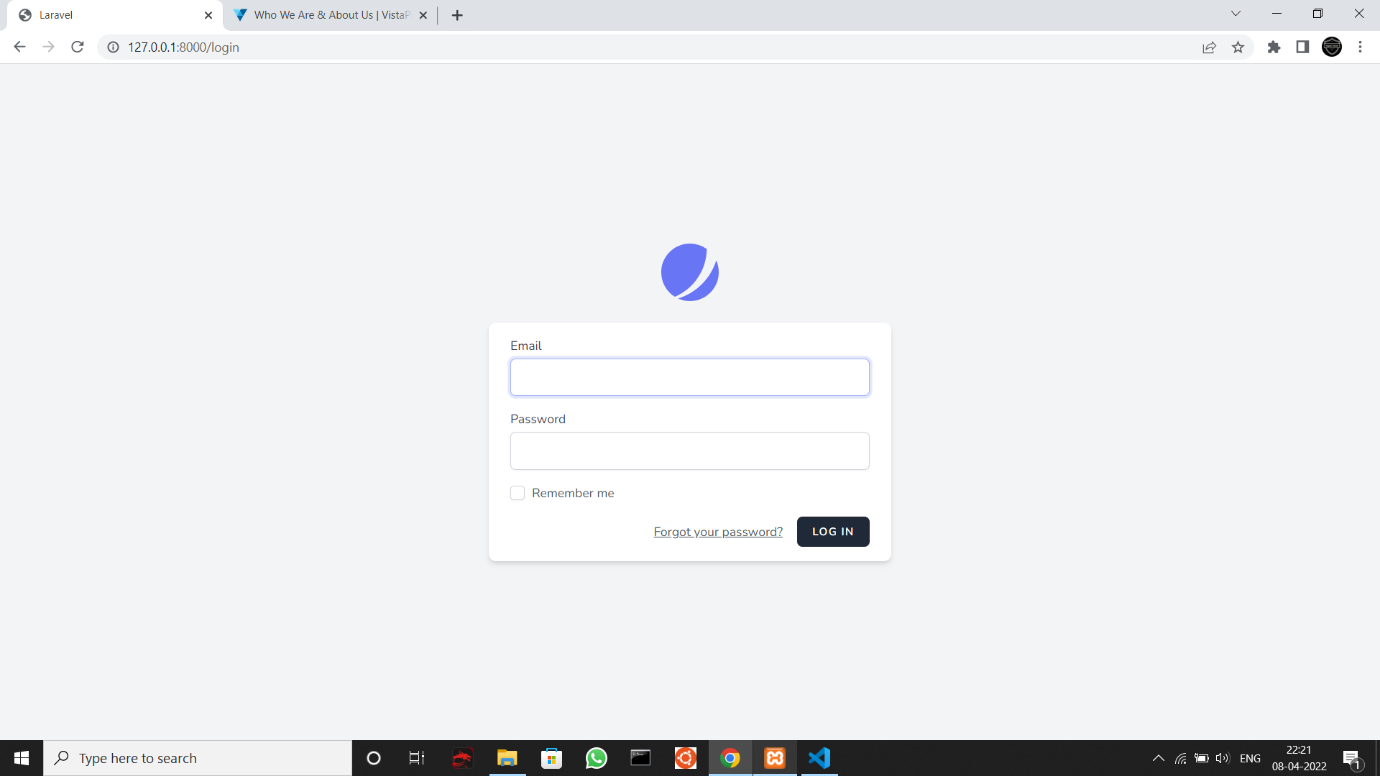
**Figure 4.3 About Us Page**

**4.1.4 Contact Us Page**

****

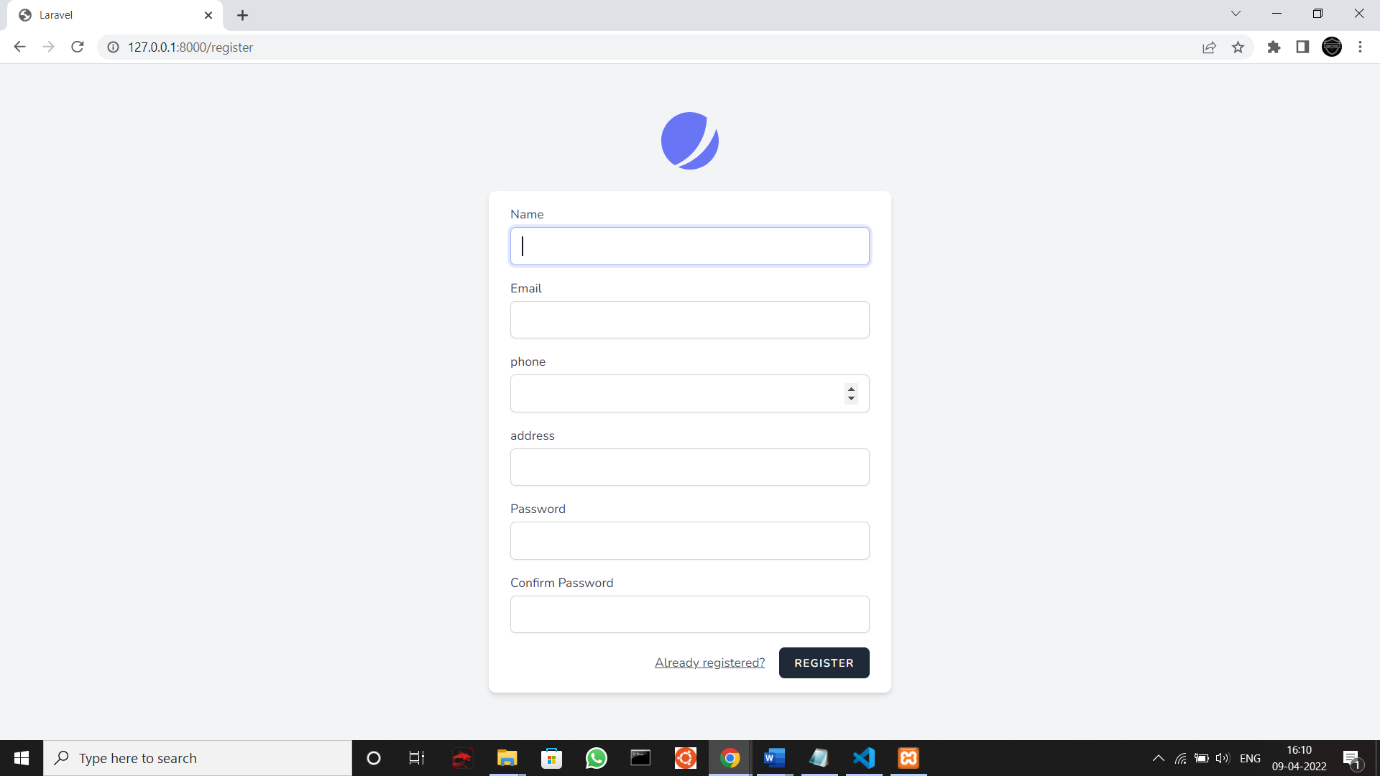
**Figure 4.4 Contact Us Page**

**4.1.5 Log In Page**

****

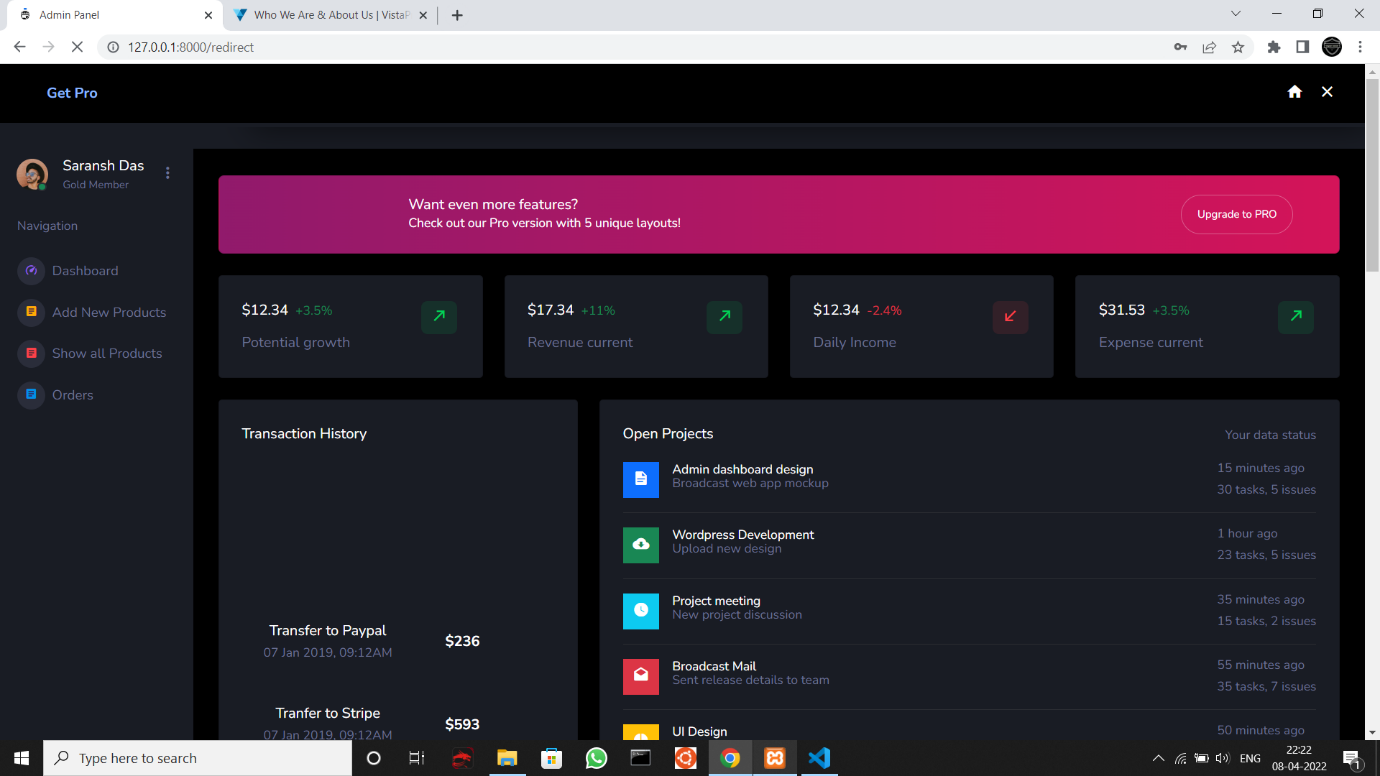
**Figure 4.5 Log In Page**

**4.1.6 Register Page**

****

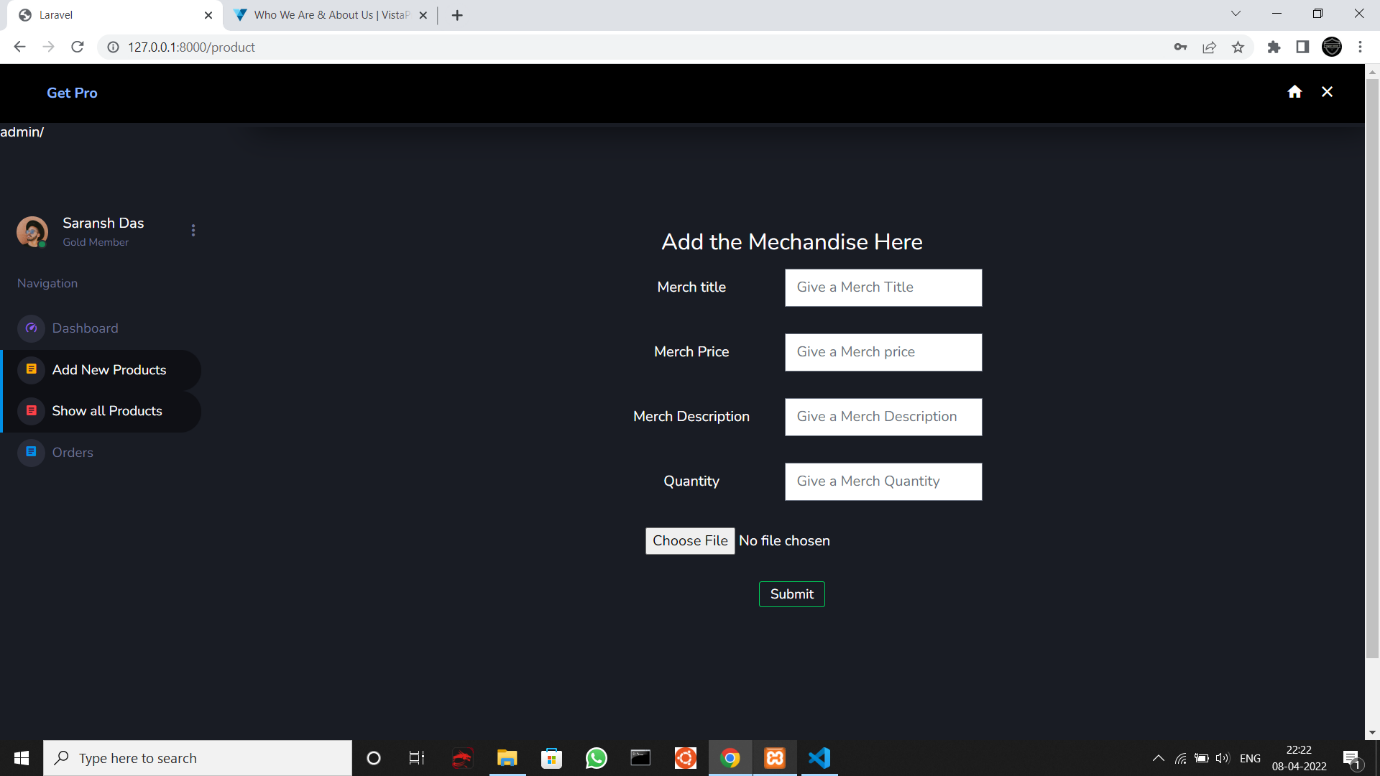
**Figure 4.6 Register Page**

**4.1.7 Admin Page**

****

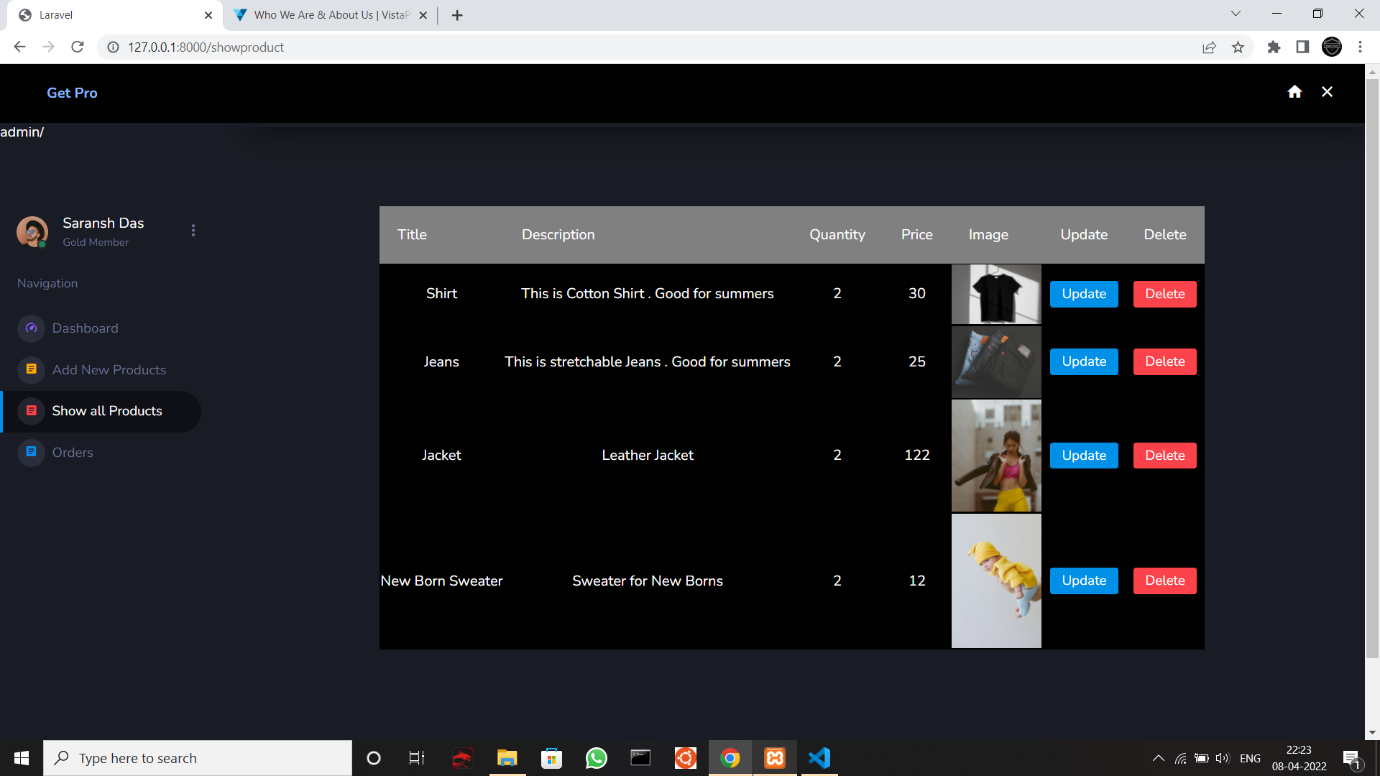
**Figure 4.7 Admin Page**

**4.1.8 Add New Products Tab**

****

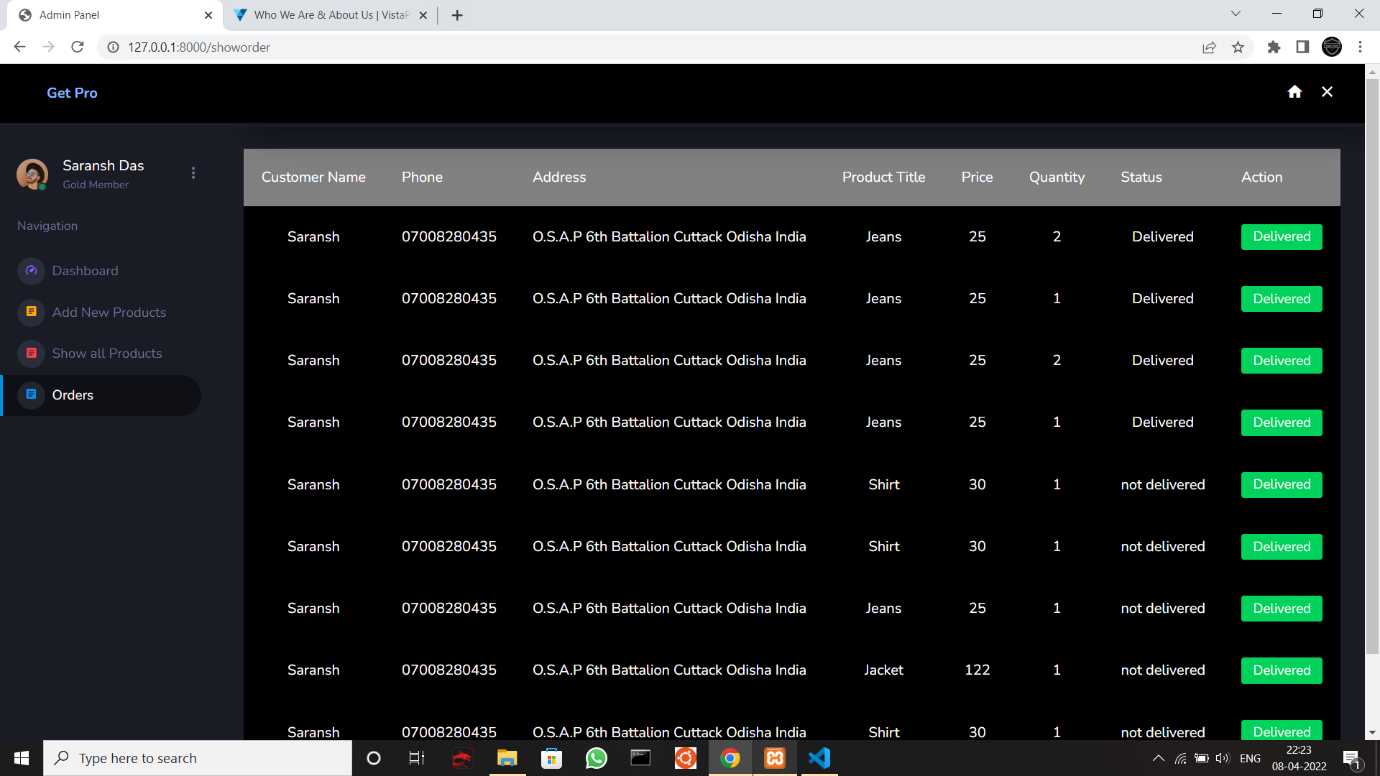
**Figure 4.8 Add New Product Tab**

**4.1.9 Show All Products Tab**

****

**Figure 4.9 Show All Products Tab**

**4.1.10 Orders Tab**

****

**Figure 4.10 Orders Tab**

**CHAPTER 5**

**GitHub Link**

* **https://github.com/Saransh-Das/Laravel-221-Project**

**REFERENCES**

* [**https://laravel.com/docs/9.x**](https://laravel.com/docs/9.x)
* [**https://jetstream.laravel.com/2.x/introduction.html**](https://jetstream.laravel.com/2.x/introduction.html)
* [**https://www.pexels.com/**](https://www.pexels.com/)
* [**https://www.bootstrapdash.com/**](https://www.bootstrapdash.com/)

|  |
| --- |
|  |