1. **INTRODUCTION**

**Students News Portal** is a web-based application designed to streamline communication between educational institutions and students by providing a centralized platform for publishing and accessing news. Developed using PHP and MySQL, this system aims to eliminate the need for manual noticeboards or inconsistent communication methods, ensuring timely and efficient information dissemination.

The system features **two user roles**: **Staff** and **Students**. Staff members can log in through a secure portal to create, manage, and publish news articles related to academic updates, events, announcements, and other relevant information. The news posting module allows staff to categorize news (e.g., events, exam schedules, placements), schedule publication dates, and attach supporting documents or media files for enhanced communication.

On the other side, students access the portal through their dedicated login. They can view all the published news, filter content based on categories, and stay updated on important institutional developments. The student dashboard presents a clean, responsive, and user-friendly interface that highlights recent and important news updates, helping students stay informed at all times.

**The portal also includes additional features like:**

**Role-based Access Control**: Ensuring staff can manage content while students have view-only access.

**Search and Filter Options**: Allowing students to quickly find relevant news items based on keywords, categories, or dates.

**Notification System**: To alert students about newly posted news or important updates.

**News Archiving**: For storing past news items which students or staff can revisit anytime.

**Audit Log Module**: Tracks all important actions by staff for transparency and accountability.

**Report Generation**: Admins can generate reports on posted news, user activity, and engagement levels.

**Export Features**: News reports can be exported in PDF or Excel formats for record-keeping or analysis.

1. **SYSTEM CONFIGURATION**

# HARDWARE REQUIMENTS

Processor : Dual Core

RAM : 1GB

Hard Disk : 160GB

# SOFTWARE REQUIPMENTS

Front End : PHP-Laravel

Back End : PHP-Laravel

Database :Mysql

# ABOUT THE SOFTWARE

Windows 10 is an operating system developed by Microsoft and was released on July 29, 2015. It was designed to be a versatile platform that works across a wide range of devices, including desktops, laptops, tablets, and hybrid devices. It was the successor to Windows 8.1 and introduced several new features and improvements.

Here are some key features and aspects of Windows 10:

# Start Menu and Interface

**Return of the Start Menu**: Windows 10 brought back the familiar Start Menu from Windows 7, combining it with the live tiles from Windows 8. The Start Menu is customizable, allowing users to pin apps, documents, and other shortcuts.

**Taskbar and Action Center**: The taskbar includes icons for pinned apps, system notifications, and the Start button. The Action Center provides quick access to settings, notifications, and important alerts.

**Virtual Desktops**: Windows 10 introduced virtual desktops, allowing users to create multiple desktop environments for organizing tasks and separating workflows.

# Cortana

**Voice Assistant**: Cortana, Microsoft’s virtual assistant, is integrated into Windows 10 and can help with tasks such as setting reminders, launching apps, searching the web, and more. However, its functionality has been reduced in recent updates.

# Microsoft Edge Browser

**New Browser**: Windows 10 initially shipped with the Microsoft Edge browser (based on EdgeHTML) but later transitioned to a new version based on the Chromium engine. The updated Edge browser is faster and more feature-rich, supporting extensions from the Chrome Web Store.

# Windows Defender Security

**Built-in Antivirus**: Windows 10 includes Windows Defender, a built-in antivirus and security tool that provides real-time protection against malware, viruses, and other threats.

**Windows Security Center**: This central hub allows users to manage security settings, firewall settings, and antivirus protection.

# Windows Updates and Service Model

**Windows as a Service**: Windows 10 introduced a new model for updates, where the OS receives regular feature updates twice a year. This continuous update approach keeps the system secure and up to date with the latest features.

**Update Control**: The system gives users more control over when updates are installed, but updates cannot be fully disabled in most cases.

# DirectStorage and Gaming Features

**DirectStorage**: This feature helps improve gaming performance by speeding up data loading times from storage devices (particularly on NVMe SSDs).

**Xbox Integration**: Windows 10 has integrated Xbox features such as Game Mode, Xbox Live, and streaming, providing a smoother gaming experience.

# Windows Ink

**Pen Support**: Windows 10 provides features for users with digital pens (styluses), including drawing and annotating directly on the screen, making it more versatile for creative tasks.

# File Explorer Enhancements

**Improved Navigation**: File Explorer has seen improvements in Windows 10, with better integration of cloud storage, such as OneDrive, and a more user-friendly interface for browsing and managing files.

# Touchscreen and Tablet Mode

**Tablet Mode**: Windows 10 can automatically switch to a tablet-friendly interface when using devices with touchscreens or convertible devices, making it more versatile for various form factors.

# Privacy and Settings

**Privacy Controls**: Windows 10 offers more granular control over privacy settings, allowing users to manage data collection, location settings, camera and microphone access, and other features that may involve privacy concerns.

# System Requirements

**Hardware Compatibility**: Windows 10 has relatively low system requirements, but it runs best on modern hardware. The system needs at least 1 GHz or faster processor, 1 GB of RAM for 32-bit or 2 GB for 64-bit, and 16 GB of storage for 32-bit or 20 GB for 64-bit installations.

# Support and Lifecycle

**End of Support**: As of 2024, Microsoft has announced that Windows 10 will continue to receive support until October 14, 2025, after which it will no longer receive updates or security patches.

# FRONTEND & BACKEND:

**PHP -LARAVEL**

**PHP** (Hypertext Preprocessor) is a widely-used open-source server-side scripting language designed specifically for web development. It can also be embedded into HTML to add dynamic functionality to websites. PHP is one of the key technologies in the LAMP stack (Linux, Apache, MySQL, PHP/Python/Perl), which is commonly used to build dynamic websites and web applications.

PHP is a server- side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994,the PHP reference implementation is now produced by the PHP Group. PHP originally stood for personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template system, web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the wed server or as a common gateway interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command line interface (CLI) and can be used to implement standalone graphical applications. The Standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work had gone

on to create a formal PHP specification. PHP web development means developing websites and dynamic web pages using the versatile and capable server-side scripting language, PHP. It offers a plethora of benefits. Not only is it open-source but also feature-rich and has all the functionality that a proprietary or paid scripting language would offer. Here are a few benefits that have made PHP web development so sought after.

# Key Features and Concepts of PHP:

**Server-Side Scripting**:

PHP runs on the server and generates dynamic HTML content. When a user accesses a webpage, PHP scripts are processed on the server, and the resulting output (usually HTML) is sent to the user's browser.

# Embedded in HTML:

PHP is often embedded within HTML files. You can write PHP code between special PHP tags, and the code is executed on the server while the resulting HTML is sent to the browser.

# Dynamic Content Generation:

PHP can generate dynamic content based on user input or database queries. For example, a PHP script can display user-specific data (such as a profile page) or interact with a database to show the latest articles or posts.

# Database Interaction:

PHP is commonly used to interact with databases, especially **MySQL** or **MariaDB**, to store and retrieve data. It allows you to create, read, update, and delete (CRUD) database records dynamically, making it ideal for applications like content management systems (CMS) or e-commerce sites.

# Form Handling:

PHP is often used to handle form submissions. It can collect form data, validate inputs, send emails, and save data to a database.

# Session and Cookie Management:

PHP allows the creation and management of sessions and cookies, enabling websites to remember users, track their activities, and personalize their experience across multiple pages or visits.

# Error Handling:

PHP offers various mechanisms for handling errors, such as **try-catch** blocks and custom error reporting. This allows developers to manage potential issues gracefully.

# Security:

PHP offers various built-in functions for sanitizing user inputs, preventing SQL injection, and other common security threats. However, it is still important for developers to follow security best practices.

# Extensive Libraries and Frameworks:

PHP has many libraries and frameworks (like **Laravel**, **Symfony**, and **CodeIgniter**) that make development easier by providing pre-built modules, routing, and templating engines, among other features.

# Cross-Platform

PHP can run on various platforms, including Windows, Linux, and macOS. This makes it highly versatile and suitable for different environments.

# How PHP Works

**Web Server Integration**

PHP works alongside web servers like **Apache**, **Nginx**, or **IIS**. When a user sends a request to a web server, the server processes the PHP script, executes the PHP code, and returns the generated HTML to the user's browser.

# Execution:

PHP code is executed on the server, and only the results (usually HTML or JSON) are sent to the client. This makes PHP more secure than client-side scripting languages like JavaScript, as sensitive information (e.g., database credentials) can be kept hidden from users.

# Database Connectivity:

PHP connects to databases using built-in extensions like **MySQLi** or **PDO (PHP Data Objects)**. With these extensions, PHP can interact with the database to perform CRUD operations, retrieve data, and more.

# Advantages of PHP:

**Ease of Use**:

PHP is easy to learn and use for beginners. Its syntax is simple, and there is extensive documentation available to help developers get started quickly.

# Open-Source:

PHP is free to use, and there are no licensing fees. As open-source software, it has a large, active community that contributes to its development and support.

# Cross-Platform Compatibility:

PHP works on multiple operating systems, including Windows, Linux, and macOS, making it versatile and adaptable to various environments.

# Integration with Databases:

PHP is excellent for database-driven applications, especially with its ability to interact seamlessly with MySQL and other relational databases. It makes managing data storage, retrieval, and manipulation efficient.

# Fast Development:

With many frameworks and libraries available, PHP allows developers to build applications faster. Frameworks like **Laravel** and **Symfony** offer reusable components and features, speeding up the development process.

# Large Ecosystem:

PHP has a vast ecosystem of libraries, frameworks, and tools that enhance productivity and enable developers to build scalable, robust applications quickly.

# Scalability:

PHP is scalable, meaning it can handle large applications and traffic when properly optimized. Major websites like Facebook and Wikipedia use PHP, demonstrating its capability to scale.

# Community Support:

Being one of the most popular languages for web development, PHP has a large community. This means there are plenty of resources, tutorials, forums, and support available for developers.

# Built-in Functions:

PHP has a rich set of built-in functions for handling file manipulation, data encryption, form validation, image processing, and more, which makes it highly powerful for a wide range of applications.

# Security Features:

PHP has several built-in security features to protect against common vulnerabilities, such as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF), though developers must still be cautious and follow best security practices.

# PHP Use Cases:

**Content Management Systems (CMS)**:

Many popular CMS platforms, such as **WordPress**, **Drupal**, and **Joomla**, are built using PHP. These platforms allow users to create, manage, and modify digital content without needing programming skills.

# E-commerce:

PHP powers many e-commerce platforms like **Magento**, **PrestaShop**, and **OpenCart**, which provide the tools to build online stores, manage products, and process payments.

# Social Networking:

Many social networking sites, including Facebook (in its early years), were built using PHP. PHP can handle complex data management and user interaction features that are common in social networking apps.

# Web Applications:

PHP is widely used to develop web applications like customer portals, online banking, CRM systems, and more. Its flexibility and ability to handle large amounts of data make it ideal for business applications.

# APIs:

PHP is also frequently used to build **RESTful APIs** that allow different applications or services to communicate with each other over the internet.

PHP is a versatile, server-side language widely used for web development. It is especially powerful for creating dynamic, data-driven websites and applications. With its ease of use, flexibility, and vast ecosystem, PHP remains one of the most popular choices for building websites and applications.

# Database : MySQL

MySQL is an open-source relational database management system (RDBMS) that is widely used for managing and organizing data in databases. It uses structured query language (SQL) for accessing, managing, and updating data, making it a powerful tool for building dynamic websites and applications. MySQL is a part of the LAMP stack (Linux, Apache, MySQL, PHP/Python/Perl) and is commonly used in combination with PHP, Python, and other server-side languages to create dynamic, data-driven websites.

# Key Features and Concepts of MySQL:

**Relational Database**:

MySQL is a relational database, meaning it stores data in tables, which are organized into rows and columns. Tables can be related to each other using keys (primary and foreign keys) to establish relationships between data.

# SQL (Structured Query Language):

MySQL uses SQL, a standard language for querying and managing databases. SQL allows you to perform operations like inserting, updating, deleting, and retrieving data. It also supports complex queries, filtering, sorting, and joining data from multiple tables.

# Data Types:

MySQL supports a wide variety of data types, including numeric (INT, FLOAT), string (VARCHAR, TEXT), date/time (DATE, DATETIME), and binary types (BLOB). These data types allow users to store different kinds of data in an organized way.

# Indexing:

MySQL supports indexing, which speeds up data retrieval operations. Indexes are created on one or more columns of a table to improve query performance, especially for large datasets.

# ACID Compliance:

MySQL ensures data integrity through the principles of **ACID** (Atomicity, Consistency, Isolation, Durability). These properties guarantee that database transactions are processed reliably and ensure that the data remains consistent and recoverable, even in the event of failures.

# Normalization:

MySQL encourages data normalization, the process of organizing data to reduce redundancy and improve data integrity. Data is often split into multiple related tables to eliminate unnecessary repetition of data.

# Transactions:

MySQL supports transactions, which allow you to group multiple SQL queries into a single operation. This ensures that all queries within the transaction are executed successfully before the changes are committed to the database, providing rollback capabilities in case of errors.

# Multi-User Support:

MySQL supports multiple users and allows for user permissions to be set at different levels. You can control who can access specific databases, tables, and operations based on user roles.

# Replication:

MySQL supports database replication, allowing data from one MySQL server to be copied to another. This is commonly used for backup, scaling, and high availability.

# Backup and Recovery

MySQL provides tools for backing up databases and recovering them in case of failure. You can perform full or incremental backups, and the recovery process can restore data to a specific point in time.

# How MySQL Works:

**Database Server**:

MySQL operates as a database server that accepts connections from applications or users via a client (e.g., MySQL Workbench, PHP, or command-line interface). The server processes SQL queries and returns the results to the client.

# Client-Server Architecture:

MySQL follows a client-server architecture where the client (such as a web application or command-line tool) sends requests to the server, which processes the queries and sends the results back to the client.

# Tables and Relationships:

Data is stored in tables, and these tables can be related to one another using keys. **Primary keys** uniquely identify each record in a table, while **foreign keys** link records from one table to another.

# Query Processing:

SQL queries are parsed and executed by the MySQL query processor. The database engine processes the query, fetches the requested data, and returns it to the client. MySQL can execute complex queries with **joins**, **subqueries**, and **grouping**

# Advantages of MySQL:

**Open-Source**:

MySQL is free to use, with the source code being open and available to the public. It is widely supported by a large community, which contributes to its ongoing development and support.

# High Performance:

MySQL is known for its high performance, especially for read-heavy workloads. It offers optimization features such as indexing, caching, and query optimization to handle large volumes of data efficiently.

# Cross-Platform Compatibility:

MySQL works on various platforms, including Linux, Windows, macOS, and other UNIX-based systems. This makes it versatile for different environments and use cases.

# Scalability:

MySQL can handle databases of all sizes, from small websites to large-scale applications with millions of records. It offers features like clustering and replication to scale databases horizontally and increase performance.

# Security:

MySQL has built-in security features such as user authentication, encryption, and access control. You can set permissions for individual users and restrict access to sensitive data.

# Data Integrity:

MySQL ensures data integrity with features like foreign keys, constraints, and ACID compliance, which maintain the consistency and reliability of the data.

# Easy to Use:

MySQL is easy to install and use, with various tools (such as **phpMyAdmin** and **MySQL Workbench**) that provide user-friendly interfaces for database management. It also has extensive documentation and tutorials available.

# Support for Multiple Storage Engines:

MySQL supports multiple storage engines, such as **InnoDB** (the default), **MyISAM**, and **Memory**. This flexibility allows developers to choose the storage engine that best suits their needs.

# Community Support:

MySQL has a large and active user community, which makes it easier to find solutions to problems, get help with troubleshooting, and access community-driven resources.

# Common Use Cases of MySQL:

**Website and Web Application Databases**:

MySQL is commonly used in web development to manage databases for websites, user authentication, content management systems (CMS), and e-commerce sites.

# Content Management Systems (CMS):

Popular CMS platforms such as **WordPress**, **Drupal**, and **Joomla** use MySQL to store content, settings, and user data.

# E-Commerce Platforms:

MySQL is often used in e-commerce platforms (like **Magento**, **PrestaShop**, and

**WooCommerce**) to store product information, customer data, and transaction records.

# Data Warehousing and Reporting:

MySQL can be used for data warehousing and generating reports. It can store large amounts of structured data and provide powerful querying capabilities for data analysis.

# Logging and Tracking Systems:

MySQL is used to store logs and track user behavior on websites or applications. Its ability to handle large datasets makes it suitable for tracking events, user activities, and system logs.

# Real-Time Applications:

MySQL is also used in applications that require real-time data retrieval and analysis, such as social media platforms, gaming applications, and financial trading systems.

MySQL is a powerful, flexible, and widely-used relational database management system. It supports complex queries, transactions, and a variety of data types while offering excellent performance, security, and scalability. MySQL is commonly used in web development, e-commerce, content management systems, and data-driven applications, making it an essential tool for modern database management.

1. **SYSTEM ANALYSIS**
   1. **EXISTING SYSTEM**

In a typical tiles shop management system, traditional processes are often employed to handle business operations such as inventory management, sales, billing, and customer inquiries. These systems may include manual record-keeping, basic computer spreadsheets, or standalone software without integration capabilities. Here's an overview of the existing systems:

# Manual System:

* + - * Use of physical ledgers and handwritten records.
      * Manual calculation of bills and tracking sales.
      * Physical inventory counts without digital tools.

# Basic Computerized System:

* + - * Use of spreadsheets (e.g., Excel) for inventory and sales tracking.
      * Standalone accounting software for financial records.
      * Lack of integration between different aspects of the business.

# Generic POS Systems:

* + - * General-purpose Point of Sale (POS) systems with limited customization.
      * Inability to handle tiles-specific metrics like area (sq. ft., sq. m) and wastage.

# Disadvantages of the Existing System Manual System

1. **Time-Consuming**:
   * Manual processes are slow, especially for calculating bills or managing large inventories.
   * Frequent delays in providing quotations or finalizing transactions.

# Error-Prone:

* + High risk of calculation errors, particularly in measurements and billing.
  + Possibility of missing inventory or misplacing records.

# Inefficiency:

* + Difficulty in maintaining a clear and updated inventory.
  + Challenges in tracking customer history or product trends.
  + No tools for generating sales reports, inventory analysis, or business insights..

# Basic Computerized System

1. **Limited Customization**:
   * Generic tools like spreadsheets do not address specific needs like area calculation, tile patterns, or wastage.

# Data Redundancy:

* + Repetitive data entry across multiple spreadsheets or software.

# No Real-Time Updates:

* + Inventory and sales records are not updated in real-time, leading to discrepancies.

# No Integration:

* + Lack of synchronization between sales, inventory, and accounting.

# Generic POS Systems

1. **Inadequate for Tile Metrics**:
   * Cannot calculate based on area (e.g., square feet/meter) or handle unique requirements like tile design matching.

# Scalability Issues:

* + Not suitable for growing businesses with increasing inventory and customer base.

# Limited Reporting Capabilities:

* + No tailored reports for analyzing tile-specific sales trends or customer preferences.

# Customer Management Challenges:

* + Inability to maintain a proper customer database, impacting marketing and follow-up opportunities.

# PROPOSED SYSTEM

The proposed system is a modern, integrated, and user-friendly software solution designed specifically for the needs of a tiles shop. It automates and streamlines various operations, from inventory management to billing, while incorporating tile-specific metrics and enhancing customer service.

# Features of the Proposed System

* + 1. **Centralized Inventory Management**
       - Real-time inventory tracking.
       - Automated updates for stock levels after every sale or purchase.
       - Supports tile-specific attributes like size, color, design, and area.

# Advanced Billing and Sales Module

* + - * Quick and accurate billing based on tile area (square feet/meter) and wastage calculations.
      * Integration with barcode or QR code scanners for efficient item entry.
      * Digital invoices via email or SMS.

# Customer Management

* + - * Maintain a comprehensive customer database with purchase history.
      * Enable personalized marketing and loyalty programs.

# Vendor and Purchase Management

* + - * Track orders and deliveries from vendors.
      * Automate purchase reorders when inventory drops below predefined thresholds.

# Analytics and Reporting

* + - * Generate detailed sales and inventory reports.
      * Identify fast-moving and slow-moving products.
      * Dashboard for real-time insights into business performance.

# User-Friendly Interface

* + - * Simplified UI for ease of use by non-technical staff.
      * Mobile app support for on-the-go management.

# Integration Capabilities

* + - * Connect with accounting software (e.g., Tally, QuickBooks).
      * Online payment gateways for smoother transactions.

# Customization

* + - * Tailored modules for specific business needs, such as handling bulk orders or showcasing tile patterns visually.

# Advantages of the Proposed System Efficiency and Accuracy

1. **Automated Processes**:
   * Reduces manual efforts in billing, inventory updates, and sales tracking.
   * Eliminates errors in area calculation, pricing, and stock management.

# Faster Operations:

* + Streamlined billing and inventory checks save time for both staff and customers.
  + Enables quick access to product details and availability.

# Enhanced Customer Experience

1. **Improved Service**:
   * Accurate and quick quotations or invoices enhance customer satisfaction.
   * Personalized offers and follow-ups build customer loyalty.

# Visual Display Options:

* + Digital catalogs or AR/VR integration to help customers visualize tiles in their space.

# Better Decision-Making

1. **Data-Driven Insights**:
   * Sales and inventory analytics help in planning stock and marketing strategies.
   * Identify high-demand products or customer trends.

# Optimized Stock Levels:

* + Avoid overstocking or stockouts through real-time inventory monitoring.

# Scalability

1. **Supports Growth**:
   * Handles increasing inventory and customers as the business grows.
   * Customizable modules ensure future adaptability.

# Cost-Effective

1. **Reduced Operational Costs**:
   * Minimizes wastage due to accurate inventory management.
   * Lowers labor costs with automated processes.

# Improved Profitability:

* + Better tracking of sales trends and customer preferences boosts revenue.

# Enhanced Security

1. **Data Security**:
   * Secure database with regular backups to prevent data loss.
   * Role-based access control for staff.

# FEASIBILITY STUDY

A feasibility study evaluates whether developing and implementing the proposed tiles shop management system is viable, practical, and beneficial. It assesses the project from different perspectives: technical, economic, operational, and more.

# Technical Feasibility

This aspect determines if the required technology and expertise are available to develop and implement the system.

# Requirements:

* **Hardware**: Computers, barcode scanners, printers, and a reliable server (if using an on- premises system).
* **Software**: Modern programming platforms, database management systems (e.g., MySQL, MongoDB), and integration with accounting tools (e.g., Tally or QuickBooks).
* **Expertise**: Skilled developers for software design, user interface creation, and system maintenance.

# Assessment:

* The technology required is readily available.
* Cloud-based options can be explored for scalability and remote access.
* Development expertise can be sourced in-house or outsourced.

# Economic Feasibility

This assesses whether the system is cost-effective and will provide adequate returns on investment (ROI).

# Cost Analysis:

* **Initial Investment**: Hardware, software development or licensing, and implementation.
* **Operational Costs**: Maintenance, training, and potential subscription fees (if using cloud services).

# Benefits:

* Reduced operational costs through automation.
* Faster billing and inventory tracking, saving time and labor costs.
* Increased sales due to improved customer service and insights.
* Prevention of overstocking or stockouts saves money.

# Operational Feasibility

This aspect evaluates if the system aligns with business processes and can be effectively implemented.

# Key Considerations:

* Staff adaptability to the new system.
* Management's willingness to adopt and support the change.
* Minimal disruption during implementation.

# Mitigation:

* User-friendly interface ensures easy adoption by non-technical staff.
* Training sessions can enhance staff confidence in using the system.

# Schedule Feasibility

This evaluates if the system can be developed and deployed within a reasonable timeframe.

# Development Timeline:

* Requirements gathering: 2–3 weeks.
* System design: 3–4 weeks.
* Development and testing: 8–10 weeks.
* Deployment and training: 2–3 weeks.

# Legal and Regulatory Feasibility

This assesses if the system complies with legal and regulatory requirements.

# Considerations:

* Data protection and privacy laws (e.g., GDPR, if applicable).
* Tax regulations for billing and accounting modules.
* Compliance with local business laws.

# Market Feasibility

This evaluates if the system meets the needs of the market (tiles shops) and is likely to be adopted.

# Market Needs:

* Automation of repetitive tasks.
* Accurate billing and inventory tracking.
* Real-time insights into business performance.

1. **SYSTEM DESIGN**
   1. **DATA FLOW DIAGRAM**

View Tiles

ADMIN

About Us

Customer Details Employee Details

Dealer Details

Tiles Purchase & Supplier

# Customer Details

**Tiles Management**

**System**

ADMIN

Customer

Details

Report

Customer ID

Mobile Number

Address

Customer Name

Customer

Customer

**Employee Details**

Employee

Details

Report

Contact Number

Address

Employee ID

Employee Name

Employee

Employee

**Dealer Details**

Dealer Details

Report

Dealer ID

City

Address

Dealer Name

**Tiles Details**

Tiles Details

Tiles

Tiles

Tiles Name

Size

Report

**Supplier Details**

Tiles ID

Supplier

Details

Report

Kind of Tiles Name

Tiles Name

Name of the supplier

Supplier

Supplier

* 1. **DATABASE DESIGN**

**Database Name: Tiles**

**Table Name: Customer Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| Cid (Primary Key) | Text | Customer Id Generation |
| Cname | Text | Customer Name |
| Gender | Text | Customer Gender |
| Profession | Text | Customer Profession |
| Address | Text | Customer Address |
| Phone | Number | Customer Number |
| Email | Text | Customer Mail Id |

**Table Name: Sales Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data**  **Type** | **Description** |
| Date\_of\_purchase | Text | Date of Purchasing |
| SID (Primary Key) | Text | Sales ID |
| Items\_count | Number | Item Count |
| Total | Number | Purchasing item total  amount |

**Table Name: Dealer Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data**  **Type** | **Description** |
| Did (Primary Key) | Text | Dealer ID |
| Dname | Text | Dealer Name |
| Gender | Text | Dealer Gender name |
| Address | Text | Dealer address |

**Table Name: Employee Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data**  **Type** | **Description** |
| Eid (Primary Key) | Text | Employee Detail |
| Ephoto | Text | Employee Photo |
| Ename | Text | Employee Name |
| Address | Text | Employee address |
| Contact No | Number | Contact Number |
| Email | Text | Dealer Mail ID |
| Work offered | Text | Employee work  offered |
| Doj | Text | Date of Joining |
| Experience | Number | Employee Experience |

**Table Name: Purchase Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data**  **Type** | **Description** |
| Did (Primary Key) | Text | Dealer ID |
| Dname | Text | Dealer Name |
| Company | Text | Company Name |
| Producing | Text | Product Dealer |
| Price | Number | Price |

**Table Name: Tiles Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data**  **Type** | **Description** |
| Tid | Text | Tiles ID |
| Kind of tiles | Text | Tiles |
| Tname | Text | Tile Name |
| Size | Text | Tile Size |

**Table Name: Whole Sale Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data**  **Type** | **Description** |
| Dop | Text | Date of Product |
| Sid | Text | Sales ID |
| cid | Text | Customer ID |
| Cname | Text | Customer Name |
| Items | Number | Items |
| Total | Number | Total amount |
| Pay | Number | Payment |
| Balance | Number | Balance Amount |

* 1. **PROJECT DESCRIPTION**
* Admin Panel
* Category Management Module
* Product Management Module
* User Management Module
* Order Management Module
* Customer Registration & Login Module
* Product Viewing Module
* Order Placement & Tracking Module
* Payment & Invoice Module
* Customer Review & Feedback Module
* Customer Registration Module
* Customer Login Module
* Admin Login Module

# Admin Panel

The Admin Panel is the control center of the Tiles Shop Management System, allowing the administrator to manage categories, products, users, and orders efficiently. Through a secure login, the admin can access the dashboard, where they get an overview of sales, inventory, and customer activity. The admin has complete authority over the system to ensure smooth operations and manage product availability.

# Category Management Module

The Category Management Module enables the admin to create, update, and delete product categories. Tiles are categorized based on material (ceramic, porcelain, vitrified, etc.), size, texture, and purpose (floor tiles, wall tiles, kitchen tiles, etc.). Organizing tiles into categories helps customers find products quickly and enhances the shopping experience.

# Product Management Module

The Product Management Module allows the admin to add, edit, and remove tile products from the system. Each product entry includes images, descriptions, prices, stock availability, and category selection.

# User Management Module

The User Management Module enables the admin to view and manage registered customers. The admin can approve, block, or delete accounts if necessary. This module stores customer details, including names, email addresses, phone numbers, and order history, ensuring a secure and well-maintained database of users.

# Order Management Module

The Or der Management Module allows the admin to view, process, and update customer orders. When a customer places an order, it is logged into the system, and the admin can update the order status (pending, confirmed, dispatched, delivered, or canceled). The admin can also generate invoices and manage refunds or replacements in case of customer complaints.

# Customer Registration & Login Module

The Customer Registration & Login Module allows customers to sign up and log in securely. During registration, customers provide basic details such as name, contact information, and address. A secure login system ensures that only registered users can place orders. Password recovery options are available for customers who forget their login credentials.

# Product Viewing Module

The Product Viewing Module allows customers to browse through the available tiles. Customers can filter products based on category, size, color, material, and price range. Each product page includes detailed descriptions, high-resolution images, and pricing information, helping customers make informed purchasing decisions.

# Order Placement & Tracking Module

The Order Placement & Tracking Module enables customers to add products to their cart, place orders, and track their order status. Customers receive real-time updates on their order, including processing, dispatch, and delivery tracking. The system generates a unique order ID, which customers can use to check the status of their purchase.

# Payment & Invoice Module

The Payment & Invoice Module supports multiple payment methods, including credit/debit cards, net banking, UPI, and cash on delivery. Once a payment is processed, the

system automatically generates an invoice that customers can download. The admin can also track payment histories and process refunds if needs.

# Customer Review & Feedback Module

The Customer Review & Feedback Module allows customers to leave ratings and reviews for purchased tiles. Customers can share their experience regarding product quality, pricing, and service. Admins can monitor reviews to ensure authenticity and respond to customer queries.

# Customer Registration Module

The Customer Registration Module allows new users to create an account to access the tiles shop's features. During registration, customers need to provide essential details such as name, email, phone number, address, and a secure password. The system performs email validation to ensure authenticity. A verification email or OTP-based verification can be implemented for added security. Once registered, customers can log in and start exploring the available products, placing orders, and tracking their purchases.

# Customer Login Module

The Customer Login Module provides a secure way for users to access their accounts. Customers enter their email/phone number and password to sign in. If the credentials match the database records, they are redirected to their dashboard, where they can browse products, place orders, and manage their account settings. If incorrect credentials are entered, the system provides an error message, preventing unauthorized access

# Admin Login Module

The Admin Login Module ensures that only authorized personnel can manage the tiles shop system. Admins log in using their registered email and password. The system may include security features like account lockout after multiple failed login attempts and IP-based access restrictions to prevent unauthorized logins. Upon successful login, the admin gains access to the dashboard, user management, order processing, and inventory management.

1. **SYSTEM TESTING**

System testing ensures that the Tiles Shop Management System functions as expected, covering all modules and verifying the system's performance, security, and usability. Below are the primary system testing methods with detailed descriptions and key features.

A Strategy for software testing integrates software test case design methods into a well- planned series of steps that result in successful construction of software. Testing is a set activity that can be planned in advance and conducted systematically.

There are different types of system testing

* Functional Testing
* Usability Testing
* Performance Testing
* Security Testing
* Database Testing
* Compatibility Testing
* Integration Testing
* Regression Testing
* Load Testing
* User Acceptance Testing (UAT)

# Functional Testing

Functional testing verifies that each module of the system operates according to the specified requirements. It ensures that all system functionalities, such as inventory management, customer orders, and billing, work correctly.

* + Tests all user interactions (admin, staff, and customer).
  + Ensures correct data entry and retrieval from the database.
  + Verifies form validations, such as required fields and input constraints.
  + Ensures correct calculations for pricing, discounts, and tax.
  + Checks if product updates (add, edit, delete) are functioning properly.

# Usability Testing

This testing assesses the user-friendliness of the system. It ensures that users can navigate easily and perform tasks efficiently.

* + Evaluates the simplicity of the user interface.
  + Ensures that menus, buttons, and options are well-organized.
  + Tests response times for different actions.
  + Checks if users can complete tasks without confusion.
  + Gathers feedback from real users for improvements.

# Performance Testing

Performance testing ensures that the system operates smoothly under different conditions, including peak loads. It includes speed, scalability, and stability tests.

* + Measures page load times for key modules (inventory, billing, etc.).
  + Tests response times when processing multiple requests.
  + Simulates multiple users accessing the system simultaneously.
  + Checks the system's behavior with large amounts of data (e.g., thousands of product entries).
  + Identifies bottlenecks in queries and system processes.

# Security Testing

Security testing ensures that the system is protected against threats such as unauthorized access, SQL injection, and data breaches.

* + Tests authentication mechanisms (login, role-based access control).
  + Ensures that passwords are encrypted and securely stored.
  + Checks input validation to prevent SQL injection and cross-site scripting (XSS).
  + Validates session management to prevent unauthorized access.
  + Ensures secure data storage and retrieval from the MySQL database.

# Database Testing

Database testing checks the integrity, accuracy, and efficiency of the MySQL database.

It ensures data is stored and retrieved correctly without corruption.

* + Verifies data consistency after CRUD operations (Create, Read, Update, Delete).
  + Ensures relationships between tables (e.g., orders linked to customers and inventory).
  + Checks database performance by optimizing queries and indexes.
  + Tests backup and recovery processes.
  + Ensures data validation rules are applied properly.

# Compatibility Testing

This test ensures that the system functions correctly on different web browsers and devices.

* + Tests system functionality on browsers such as Chrome, Firefox, Edge, and Safari.
  + Ensures proper display and responsiveness on desktops, tablets, and mobile devices.
  + Checks for UI consistency across different screen sizes.
  + Verifies that all features work regardless of the operating system.

# Integration Testing

Integration testing ensures that all modules (inventory, orders, payments, reports) work together seamlessly.

* + Tests the interaction between modules like inventory and billing.
  + Ensures that data is correctly passed between system components.
  + Validates third-party integrations, such as payment gateways.
  + Identifies errors that occur when different modules interact.

# Regression Testing

Regression testing ensures that new changes or updates do not break existing functionality.

* + Runs test cases on previously working functionalities.
  + Ensures bug fixes do not introduce new issues.
  + Checks for unintended changes in calculations, UI, or workflows.
  + Automates testing for frequent updates.

# Load Testing

Load testing evaluates how the system performs under heavy usage, ensuring it remains

stable.

* + Simulates multiple users placing orders and managing inventory.
  + Monitors system response under high traffic conditions.
  + Identifies system slowdowns and failures.
  + Ensures the database can handle large data transactions efficiently.

# User Acceptance Testing (UAT)

User acceptance testing ensures that the system meets business requirements and is ready for deployment.

* + Tests the system from an end-user perspective.
  + Conducts real-world usage scenarios.
  + Validates whether the system meets customer expectations.
  + Identifies usability concerns before deployment.

1. **SYSTEM IMPLEMENTATION**

The system implementation for a Tiles Shop Management System involves a structured approach to developing a robust and efficient software solution tailored to meet the needs of a tile business. The process begins with an in-depth requirement analysis to understand business operations, including inventory control, sales tracking, supplier and customer management, financial transactions, and reporting. The system is then designed with an intuitive user interface that allows easy navigation and quick access to essential features. A well-structured database is created to store product details such as tile types, sizes, colors, pricing, stock levels, and supplier information, ensuring efficient inventory management and reducing the risk of overstocking or stock outs. During the development phase, the system is built using suitable programming languages and frameworks, ensuring a scalable and responsive design. Advanced functionalities such as barcode scanning, automated invoicing, and integration with payment gateways are incorporated to streamline sales and billing processes. Additionally, customer relationship management (CRM) features are included to maintain customer records, track purchase history, and provide personalized services. The system also supports multi-user access with role-based permissions, ensuring secure handling of sensitive business data. Once the development is completed, rigorous testing is conducted, including unit testing, integration testing, and user acceptance testing, to identify and resolve any bugs or system inefficiencies. After successful testing, the system is deployed either on a local server or a cloud-based platform, depending on the business requirements. Comprehensive training is provided to employees to ensure they can efficiently operate the system and maximize its benefits. Regular system maintenance, security updates, and performance monitoring are carried out to keep the system reliable and secure. Additionally, data backup mechanisms are implemented to prevent data loss and ensure business continuity. By automating and optimizing various processes, the Tiles Shop Management System enhances productivity, improves customer service, and enables data-driven decision-making, ultimately contributing to the overall growth and success of the business.

1. **CONCLUSION**

The given project **“Tiles Management System”** was successfully completed and the required reports are generated. The project has been designed to fulfill the needs of the tiles customer and tiles Company. The system is extremely user friendly. The system is tested with real data. The results or the live data are compacted with the actual system verified for its accuracy. The system is flexible so that there is a lot of scope to update the system. As the system is flexible the system can be changed if any changes come in future. The developed system is portable. The developed system has been completed which is customized for the satisfaction of the userThe system has been analyzed, designed and developed with meticulous care and can be executed without any faults or errors.

1. **APPENDIX**
   1. **Codings**

Admin

<?php

session\_start();

include 'includes/index.inc.php'; include 'includes/search.inc.php';

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.7.2/css/all.min.css" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/index.css">

<link rel="stylesheet" href="[https://unpkg.com/aos@next/dist/aos.css"](https://unpkg.com/aos%40next/dist/aos.css)/>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/6.7.2/css/all.min.css" integrity="sha512- Evv84Mr4kqVGRNSgIGL/F/aIDqQb7xQ2vcrdIwxfjThSH8CSR7PBEakCr51Ck+w+/U6sw U2Im1vVX0SVk9ABhg==" crossorigin="anonymous" referrerpolicy="no-referrer" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/6.7.2/css/brands.min.css" integrity="sha512- 58P9Hy7II0YeXLv+iFiLCv1rtLW47xmiRpC1oFafeKNShp8V5bKV/ciVtYqbk2YfxXQMt58 DjNfkXFOn62xE+g==" crossorigin="anonymous" referrerpolicy="no-referrer" />

<title> Tiles | home</title>

</head>

<body>

<section class="head">

<header>

<?php

if(isset($\_SESSION['username'])){ echo '<div class="profile">';

if($\_SESSION['admin'] == 1){

echo '<a href="admin.php"><i class="fas fa-door-open"></i></a>';

}

echo '<a href="cart.php"><i class="fas fa-shopping-

cart"></i><p class="cart">'; echo $\_SESSION['cart']; echo'</p></a>

<form action="index.php" method="post">

<button class="logout-btn" type="submit" name="logout-submit"><i class="fas fa-sign-out-alt"></i></button>

</form>

</div>';

}

?>

<h1>Tiles Shop</h1>

<nav>

THOUGHTS</li>

</nav>

</header>

<ul>

</ul>

<li class="to-home">HOME</li>

<li class="to-login">LOGIN</li>

<li class="to-stories">STORIES</li>

<li class="to-cars">TILES</li>

<li class="to-my-thoughts">MY

<li class="to-gallary">GALLARY</li>

</section>

<section id="login" class="showcase">

<div class="showcase-text" data-aos="slide-up">

<p class="error">

<?php if(isset($\_SESSION['log\_message'])){

echo '<span style="background-color: rgba(200, 0, 0, 0.4); padding: 10px;

font-size: 12px;"

>'.$\_SESSION['log\_message'].'</span>';

} else if(isset($\_SESSION['sign\_message'])){ if($\_SESSION['sign\_message'] == "Account created successfully"){

echo '<span style="background-color: rgba(0, 200, 0, 0.4); padding: 10px;

font-size: 12px; border-radius: 3px;

margin-bottom: 10px;"

>'.$\_SESSION['sign\_message'].'</span>';

} else {

echo '<span style="background-color: rgba(200, 0, 0, 0.4); padding: 10px;

font-size: 12px;"

>'.$\_SESSION['sign\_message'].'</span>';

}

}

?>

</p>

<div class="login-signup">

<h1>Tiles</h1>

<h4 class="login active-form">Login</h4>

<h4 class="signup">Signup</h4>

</div>

<div class="forms">

method="post">

placeholder="Email"> <br>

<form class="login-form" action="index.php"

<input type="email" name="email"

<input type="password" name="password" placeholder="Password"> <br>

<button type="submit" name="login-submit"> Log in </button>

</form>

<form class="signup-form" action="index.php"

method="post">

placeholder="Username"> <br>

<input type="text" name="username"

<input type="email" name="email" placeholder="Email"> <br>

<input type="password" name="password" placeholder="Password"> <br>

<input type="password" name="re-password" placeholder="Retype

password"> <br>

<button type="submit" name="signup-submit"> Sign up </button>

</form>

</div>

<p></p>

</div>

<div class="slideshow-container">

<div class="mySlides fade" style="background-image: linear- gradient(rgba(0, 0, 0, .5), rgba(0, 0, 0, .5)), url('assets/Display\_images/1.jpg');

background-position: bottom;

background-size: cover;">

</div>

<div class="mySlides fade" style="background-image: linear- gradient(rgba(0, 0, 0, .5), rgba(0, 0, 0, .5)), url('assets/Display\_images/2.jpg');

background-position: bottom; background-size: cover;">

</div>

<div class="mySlides fade" style="background-image: linear- gradient(rgba(0, 0, 0, .5), rgba(0, 0, 0, .5)), url('assets/Display\_images/3.jpg');

background-position: bottom; background-size: cover;">

</div>

</div>

</section>

<section class="stories">

<h1 data-aos="slide-down">STORIES</h1>

<div class="stories-container">

<?php

while($row = $stories->fetch\_assoc()){ if($row['showing'] == 1){

echo '<div class="item" data-aos="slide-left">

<div class="item-image" style="background- image:linear-gradient(rgba(0, 0, 0, .3), rgba(0, 0, 0, .3)), url(assets/Story\_images/story'.str\_replace(" ", "\_", $row['title']).'.jpg);

background-size: cover;

position:center;"> item"></div>

</div>

background-

<div class="inner-

$row['title']).'</p>

}

<h2>'.$row['title'].'</h2>

<p class="story-body" style="display: none;">'.$row['body'].'</p>

<p class="story-image" style="display: none;">story'.str\_replace(" ", "\_",

<p class="story-id" style="display: none;">'.$row['id'].'</p>

</div>';

}

?>

</div>

<div class="stories-controls">

</div>

</section>

<section class="cars">

<h1>Tiles</h1>

<h4 class="regular active-cars">GRANITE</h4>

<h4 class="sports">MARBLE</h4>

<h4 class="off-road">CERAMIC</h4>

<div class="search-car">

<!-- <form class="search-regular-car-form" action="more.php" method="post" onsubmit="add()">

<input class="search" type="text" name="regular-search" placeholder="Search">

<button type="submit" name="regular-search-submit"><i class="fas fa- search"></i></button>

<button class="voice-regular-search" type="button"><i class="fas fa- microphone"></i></button>

</form> -->

</div>

<div class="cars-container1">

<div class="cars-container">

<?php

$i = 0;

while($row = $cars->fetch\_assoc()){ if($row['car\_type'] == 'Regular' & $i <= 8){

echo '<div class="box" data-aos="slide-up">

<div class="box-image" style="background- image: url(assets/Car\_images/'.$row['image'].');

background-size: cover; background-position:center;">.

</div>

<p>'.$row['manufacturer'].'</p>

<p>'.$row['model'].'</p>

<p>₹'.$row['price'].'</p>

<form class="" action="product.php?id='.$row['id'].'" method="post">

<button type="submit" name="add-to-

cart-index"> Add to Cart</button>

$i++;

}

}

?>

</div>';

</form>

</div>

<div class="more">

<a class="more-btn" href="more.php?type=Regular"><i class="fas fa-angle- double-right"><span>More</span> </i></a>

</div>

</div>

<div class="cars-container2">

<div class="cars-container">

<?php

$j = 0;

while($row = $carsx->fetch\_assoc()){ if($row['car\_type'] == 'Sports' & $j <= 8){

echo '<div class="box" data-aos="slide-up">

<div class="box-image" style="background- image: url(assets/Car\_images/'.$row['image'].');

background-size: cover; background-position:center;">.

</div>

<p>'.$row['manufacturer'].'</p>

<p>'.$row['model'].'</p>

<p>₹'.$row['price'].'</p>

<form class="" action="product.php?id='.$row['id'].'" method="post">

<button type="submit" name="add-to-

cart-index"> Add to Cart</button>

$j++;

}

</div>';

</form>

}

?>

</div>

<div class="more">

<a class="more-btn" href="more.php?type=Sports"><i class="fas fa-angle- double-right"><span>More</span> </i></a>

</div>

</div>

<div class="cars-container3">

<div class="cars-container">

<?php

$k = 0;

while($row = $carsxx->fetch\_assoc()){ if($row['car\_type'] == 'Off road' & $k <= 8){

echo '<div class="box" data-aos="slide-up">

<div class="box-image" style="background- image: url(assets/Car\_images/'.$row['image'].');

background-size: cover; background-position:center;">.

</div>

<p>'.$row['manufacturer'].'</p>

<p>'.$row['model'].'</p>

<p>₹'.$row['price'].'</p>

<form class="" action="product.php?id='.$row['id'].'" method="post">

<button type="submit" name="add-to-

cart-index"> Add to Cart</button>

</form>

</div>';

$k++;

}

}

?>

</div>

<div class="more">

<a class="more-btn" href="more.php?type=all\_cars"><i class="fas fa-angle- double-right"><span>More</span> </i></a>

</div>

</section>

<div class="products-heading">

<h1>CERAMICS</h1>

</div>

<section class="products">

<div class="product-container">

<?php

while($row = $products->fetch\_assoc()){

echo '<div class="item" data-aos="slide-up">

<img src="assets/Product\_images/'.$row['image'].'">

<p>'.$row['manufacturer'].'</p>

<p>'.$row['model'].'</p>

<p>₹'.$row['price'].'</p>

<form class="" action="product.php?product\_id='.$row['id'].'"

method="post">

<button type="submit" name="add-to-cart-index"> Add to Cart</button>

</form>

</div>';

}

?>

</div>

</section>

<div class="product-controls">

<i id="pre-product" class="fas fa-chevron-circle-left"></i>

<i id="next-product" class="fas fa-chevron-circle-right"></i>

</div>

<section class="my-thoughts">

<div class="my-thoughts-container">

<h1>MY THOUGHT</h1>

<?php

$i = 0;

foreach($thoughts as $row){ if($i > 4) break;

echo '<div class="post" data-aos="slide-up">

<div class="post-mini1" style="background- image:url(assets/Thought\_images/'.$row['image'].');

background-size: cover; background-position:center;">

</div>

<div class="post-mini2">

<h2><a href="post.php?id='.$row['id'].'">'.$row['title'].'</a></h2>

<i class="far fa-user"> <p style="display: inline;"> Mahdi Hasan Tarunno</p></i>

5px 0px 5px;"> | </p>

<p style="display: inline; padding:0px

<i class="far fa-calendar-alt"> <p

style="display:inline;"> '.$row['create\_time'].'</p></i>

<p style="display: inline; padding:0px

5px 0px 5px;"> | </p>

<i class="fas fa-paperclip"

style="display:inline;"></i> <p style="display:inline;"> '.$row['tag'].' </p></i>

<br><br>

<p>'.substr($row['body'], 0,

min(strlen($row['body']), 260)).'</p>

<br>

</div>

</div>';

$i++;

}

?>

<div class="more" style="text-align: center;">

<a class="more-btn" href="more.php?post=all\_post"><i class="fas fa-angle- double-right"><span>More</span> </i></a>

</div>

</div>

</section>

<section class="gallary">

<h1>MY GALLARY</h1>

<div class="gallary-img">

<?php

$row = $gallaries->fetch\_assoc(); echo '<div class="row">

<div class="column">

<div class="overlay">

<h4>Click for fullscreen</h4>

<img src="assets/Gallary\_images/'.$row['image1'].'" style="width:100%; height:200px;">

</div>

<div class="overlay">

<h4>Click for fullscreen</h4>

<img src="assets/Gallary\_images/'.$row['image2'].'"

style="width:100%; height:200px;">

</div>

</div>

<div class="column">

<div class="overlay-port">

<h4>Click for fullscreen</h4>

<img src="assets/Gallary\_images/'.$row['image3'].'"

style="width:100%; height:407px;">

</div>

</div>

<div class="column">

<div class="overlay">

<h4>Click for fullscreen</h4>

<img src="assets/Gallary\_images/'.$row['image4'].'"

style="width:100%; height:200px;">

style="width:100%; height:200px;">

</div>

</div>

<div class="overlay">

<h4>Click for fullscreen</h4>

<img src="assets/Gallary\_images/'.$row['image5'].'"

</div>

<div class="column">

style="width:100%;height:200px;">

style="width:100%; height:200px;">

</div>

</div>';

<div class="overlay">

<h4>Click for fullscreen</h4>

<img src="assets/Gallary\_images/'.$row['image6'].'"

</div>

<div class="overlay">

<h4>Click for fullscreen</h4>

<img src="assets/Gallary\_images/'.$row['image7'].'"

</div>

?>

</div>

</section>

<footer>

<div class="social">

<h2>FLLOW US</h2>

<a href="#"> <i class="fab fa-facebook"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-instagram"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-twitter"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-youtube"> <span></span> </i> </a>

</div>

<div class="credit">

<h1>AI | Anjana Infotech</h1>

</div>

</footer>

<script src="[https://unpkg.com/aos@next/dist/aos.js"](https://unpkg.com/aos%40next/dist/aos.js)></script>

<script type="text/javascript" src="javaScript/app.js"></script>

<script type="text/javascript" src="javaScript/voice\_search.js"></script>

<script type="text/javascript" src="javaScript/story.js"></script>

</body>

</html>

ORDER

<?php

session\_start();

include 'includes/admin.inc.php';

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.7.2/css/all.min.css" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/admin.css">

<title>TARUNNO DRIVES | product</title>

</head>

<body>

<div class="order-all">

<div class="heading">

<h1><a href="javascript:history.back();"><i class="fas fa- angle-double-left"></i></a>TARUNNO DRIVES <span> Admin </span> </h1>

</div>

<div class="user-info">

<h3>Client details</h3>

<?php

echo '

'.$\_SESSION['username'].'</h4> '.$\_SESSION['email'].'</h4>

?>

<?php

<h4><i class="fas fa-user"></i>

<h4><i class="fas fa-envelope"></i> ';

$user\_order = new Order();

$user\_order = $user\_order->get\_order($order\_id);

$user\_order = $user\_order->fetch\_assoc(); echo '

<h4><i class="fas fa-map-marker-alt"></i>

'.$user\_order['address'].' | '.$user\_order['city'].'</h4>

<h4><i class="fas fa-mobile-alt"></i>

'.$user\_order['phone'].'</h4> '.$user\_order['postal\_code'].'</h4>

?>

</div>

<h4><i class="fas fa-mail-bulk"></i> ';

<div class="order-container">

<h3>Order details</h3>

<?php

echo '<div class="items">';

while($row = $order\_items->fetch\_assoc()){

$product\_1 = $row['product\_id'];

$product\_2 = $row['product\_id2']; if($product\_1 == -1){

$item = new Product();

$item = $item-

>get\_product($product\_2);

$item = $item->fetch\_assoc();

src="assets/Product\_images/'.$item['image'].'"> name">'.$item['model'].'</h5> name">'.$item['manufacturer'].'</h5>

price">$'.$item['price'].'</h6>

echo '

<div class="item">

<img

<h5 class="item-

<h5 class="item-

<h6 class="item-

</div> ';

} else {

$item = new Car();

$item = $item->get\_car($product\_1);

$item = $item->fetch\_assoc();

src="assets/Car\_images/'.$item['image'].'"> name">'.$item['model'].'</h5> name">'.$item['manufacturer'].'</h5>

price">$'.$item['price'].'</h6>

echo '

<div class="item">

<img

<h5 class="item-

<h5 class="item-

<h6 class="item-

</div> ';

}

</div>

</div>

}

echo '</div>'

?>

<footer>

<div class="social">

<h2>FLLOW US</h2>

<a href="#"> <i class="fab fa-facebook"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-instagram"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-twitter"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-youtube"> <span></span> </i> </a>

</div>

<div class="credit">

<h1>TARUNNO DRIVES | Developed by mahadi hasan tarunno</h1>

</div>

</footer>

</body>

</html> Product

<?php session\_start();

include 'includes/product.inc.php';

if (!isset($\_SESSION['username'])) { header("Location: index.php"); exit();

}

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.7.2/css/all.min.css" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/product.css">

<title>TARUNNO DRIVES | product</title>

</head>

<body>

<div class="container">

<div class="nav">

<a href="javascript:history.back();"><i class="fas fa-angle-double-

left"></i></a>

<h1>TILES SHOP</h1>

</div>

<?php

'</p></div>';

if (isset($\_SESSION['message'])) {

echo '<div class="message"><p>' . $\_SESSION['message'] .

unset($\_SESSION['message']);

}

?>

<?php

if (isset($\_GET['product\_id'])) { echo '<div class="product">

<div class="info">

<div class="img">

<img src="assets/Product\_images/' .

$row['image'] . '" height="400" width="420" style="display: block;">

</div>

. '</h3>

</div>

<div class="image">

<div class="pro-info">

<h2>Tiles Shop </h2>

<h3>' . $row['manufacturer'] . ' ' . $row['model']

<h5>₹' . $row['price'] . '</h5>

<br>

<h3>Condition:</h3>

<p>' . $row['condition'] . '</p>

<form action="product.php?product\_id=' .

$row['id'] . '" method="post">

<button class="add-to-cart" type="submit"

name="add-to-cart-product"> Add to Cart</button>

<button type="button" class="call"><i

class="fas fa-phone"></i></button> class="fas fa-envelope"></i></button>

</div>

</div>

</div>

</div>';

} else {

<button type="button" class="mail"><i

</form>

echo '<div class="product">

<div class="info">

<div class="img">

<img src="assets/Car\_images/' . $row['image'] . '" height="300" width="420" style="display: block;">

<i class="fas fa-tachometer-alt"> <p>' .

$row['speed'] . 'km/h</p> </i>

$row['mileage'] . 'km</p> </i>

$row['battery'] . '</p> </i> '</p> </i>

$row['total\_run'] . 'km</p> </i>

. '</p> </i>

</div>

</div>

<i class="fas fa-gas-pump"> <p>' .

<i class="fas fa-car-battery"> <p>' .

<i class="fas fa-oil-can"> <p>' . $row['fuel'] .

<i class="fas fa-heartbeat"> <p>' .

<i class="fas fa-arrows-alt"> <p>' . $row['gear']

. '</h3>

method="post">

<div class="image">

<div class="pro-info">

<h2>Car </h2>

<h3>' . $row['manufacturer'] . ' ' . $row['model']

<h5>$' . $row['price'] . '</h5>

<br>

<h3>Condition:</h3>

<p>' . $row['condition'] . '</p>

<form action="product.php?id=' . $row['id'] . '"

<button class="add-to-cart" type="submit"

name="add-to-cart"> Add to Cart</button> class="fas fa-phone"></i></button>

class="fas fa-envelope"></i></button>

<button type="button" class="call"><i

<button type="button" class="mail"><i

</div>

</div>';

}

?>

</div>

</div>

</form>

</a>

</a>

</a>

</a>

<footer>

<div class="social">

<h2>FLLOW US</h2>

<a href="#"> <i class="fab fa-facebook"> <span></span> </i>

<a href="#"> <i class="fab fa-instagram"> <span></span> </i>

<a href="#"> <i class="fab fa-twitter"> <span></span> </i>

<a href="#"> <i class="fab fa-youtube"> <span></span> </i>

tarunno</h1>

</div>

<div class="credit">

<h1>TARUNNO DRIVES | Developed by mahadi hasan

</body>

</div>

</footer>

</html> Cart

<?php

session\_start();

include 'includes/cart.inc.php';

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.7.2/css/all.min.css" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/cart.css">

<title>TARUNNO DRIVES | product</title>

</head>

<body>

<div class="all">

<div class="nav">

<a href="index.php"><i class="fas fa-angle-double- left"></i></a><h1>TARUNNO DRIVES</h1>

</div>

<div class="container">

<div class="results">

<h3>My cart <span class="cost"><i class="fas fa- donate"></i> <?php echo $total; ?> </span> <a href="checkout.php?user\_id=<?php echo

$user\_id; ?>" class="checkout"> Checkout </a></h3>

<div class="product">

<?php

if($cartss->num\_rows == 0){ echo '<img class="empty-cart"

src="assets/Display\_images/empty\_cart.png">';

echo '<h1 class="empty-h1"> Empty Cart </h1>';

}

while($row = $cartss->fetch\_assoc()){

if($row['product\_id'] != -1){ echo '<div class="item">

<img src="assets/Car\_images/'.$row['product\_image'].'">

<div class="item-info">

<h2>'.$row['product\_name'].'</h2>

<p>'.$row['product\_model'].'</p>

<p>$'.$row['product\_price'].'</p>

<p style="font-size: 10px; margin-top:20px; color: gray;">survived not only five centuries, but also the leap into electronic typesetting,

remaining essentially unchanged. It was popularised in the 1960s

with the release

method="post">

btn"> Review</button>

of Letraset sheets</p>

<form class="" action="product.php?id='.$row['product\_id'].'"

<button type="submit" class="order-

</form>

<form action="cart.php?cnl\_id='.$row['id'].'" method="post">

<button class="cnl-btn" type="submit" name="cnl-

submit">Cancel</button>

</form>

</div>

</div>';

}

else{

echo '<div class="item">

<img src="assets/Product\_images/'.$row['product\_image'].'">

<div class="item-info">

<h2>'.$row['product\_name'].'</h2>

<p>'.$row['product\_model'].'</p>

<p>$'.$row['product\_price'].'</p>

<p style="font-size: 10px; margin-top:20px; color: gray;">survived not only five centuries, but also the leap into electronic typesetting,

remaining essentially unchanged. It was popularised in the 1960s

with the release

of Letraset sheets</p>

<form class=""

action="product.php?product\_id='.$row['product\_id\_2'].'" method="post">

<button type="submit" class="order-

btn"> Review</button>

</form>

<form action="cart.php?cnl\_id='.$row['id'].'" method="post">

<button class="cnl-btn" type="submit" name="cnl-

submit">Cencel</button>

</form>

</div>

</div>';

}

}

?>

</div>

</div>

</div>

</div>

<footer>

<div class="social">

<h2>FLLOW US</h2>

<a href="#"> <i class="fab fa-facebook"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-instagram"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-twitter"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-youtube"> <span></span> </i> </a>

</div>

<div class="credit">

<h1>TARUNNO DRIVES | Developed by mahadi hasan tarunno</h1>

</div>

</footer>

</body>

</html> USER

<?php session\_start();

include 'includes/order.inc.php';

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.7.2/css/all.min.css" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/checkout.css">

<title>TARUNNO DRIVES | checkout</title>

</head>

<body>

<div class="all">

<div class="nav">

<a href="cart.php"><i class="fas fa-angle-double-left"></i></a>

<h1>TILES SHOP</h1>

</div>

<div class="container">

<h2> Checkout </h2>

<div class="checkout-info">

<div id="checkout-flex" class="checkout-form">

<h3>Client information</h3>

<form action="checkout.php" method="post">

<input type="hidden" name="user\_id" value="<?php echo $\_SESSION['id']; ?>"><br>

<input type="text" name="address"

placeholder="Address..." required><br> placeholder="City..." required><br> placeholder="Phone..." required><br> placeholder="Postal code..." required><br>

<input type="text" name="city"

<input type="text" name="phone"

<input type="text" name="postal\_code"

<button type="submit" name="order-submit"

class="order-submit-btn" onclick="showAlert()"> Place order</button><br>

</form>

information</h3>

</div>

<div id="checkout-flex" class="checkout-details">

<h3 style="margin-bottom: 35px;">Checkout

<h4>Total: ₹ <?php echo $total; ?></h4>

<h4>Total items: <?php echo $count; ?></h4>

<img src="./assets/Car\_images/qrcode.png" alt="">

</div>

</div>

</div>

<script>

function showAlert() {

alert("Order successfully!!!");

}

</script>

</div>

<footer>

<div class="social">

<h2>FLLOW US</h2>

<a href="#"> <i class="fab fa-facebook"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-instagram"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-twitter"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-youtube"> <span></span> </i> </a>

</div>

<div class="credit">

<h1>TILES SHOP | Developed by US</h1>

</div>

</footer>

</body>

</html>

<?php

session\_start();

include 'includes/post.inc.php';

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.7.2/css/all.min.css" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/post.css">

<title>TARUNNO DRIVES | more</title>

</head>

<body>

<div class="nav">

<a href="javascript:history.back();"><i class="fas fa-angle-double- left"></i></a><h1>TARUNNO DRIVES</h1>

</div>

<h1 class="headling">Post</h1>

<div class="container">

<div class="results">

<?php

$row = $thoughts->fetch\_assoc();

echo '<div class="image" style="background- image:url(assets/Thought\_images/'.$row['image'].');

background-size: cover; background-position:center;">

</div>

<div class="text">

<h2>'.$row['title'].'</h2>

<i class="far fa-user"> <p style="display: inline;"> Mahdi Hasan Tarunno</p></i>

<p style="display: inline; padding:0px

5px 0px 5px;"> | </p>

<i class="far fa-calendar-alt"> <p

style="display:inline;"> '.$row['create\_time'].'</p></i>

<p style="display: inline; padding:0px

5px 0px 5px;"> | </p>

<i class="fas fa-paperclip" style="display:inline;"><p style="display:inline;"> '.$row['tag'].' </p></i>

<br><br>

<p>'.$row['body'].'</p>

<br>

</div>';

?>

</div>

</div>

<footer>

<div class="social">

<h2>FLLOW US</h2>

<a href="#"> <i class="fab fa-facebook"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-instagram"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-twitter"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-youtube"> <span></span> </i> </a>

</footer>

</body>

</html>

<?php

</div>

<div class="credit">

<h1>TARUNNO DRIVES | Developed by mahadi hasan tarunno</h1>

</div>

session\_start();

include 'includes/more.inc.php'; include 'includes/search.inc.php';

?>

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.7.2/css/all.min.css" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/more.css">

<title>TARUNNO DRIVES | more</title>

</head>

<body>

<div class="nav">

<a href="javascript:history.back();"><i class="fas fa-angle-double- left"></i></a><h1>TARUNNO DRIVES</h1>

</div>

<h1 class="headling">More</h1>

<div class="container">

<div class="results">

<?php if(isset($\_GET['type'])){

echo '<h3>'.$type.'</h3>'; while($row = $cars->fetch\_assoc()){

if($row['car\_type'] == $type){ echo '<div class="item">

<img src="assets/Car\_images/'.$row['image'].'" height="150px"

width="195px;">

<h2>'.$row['manufacturer'].'</h2>

<p>'.$row['model'].'</p>

<p>$'.$row['price'].'</p>

Cart</button>

}

<form class="" action="product.php?id='.$row['id'].'" method="post">

<button type="submit" name="add-to-cart-index"> Add to

</form>

</div>';

if($type == 'all'){

echo '<div class="item">

<img src="assets/Car\_images/'.$row['image'].'" height="150px"

width="195px;">

Cart</button>

}

<h2>'.$row['manufacturer'].'</h2>

<p>'.$row['model'].'</p>

<p>$'.$row['price'].'</p>

<form class="" action="product.php?id='.$row['id'].'" method="post">

<button type="submit" name="add-to-cart-index"> Add to

</form>

</div>';

}

} else {

foreach($thoughts as $row){ echo '

<div class="post">

<div class="post-mini1" style="background- image:url(assets/Thought\_images/'.$row['image'].');

background-size: cover; background-position:center;">

</div>

<div class="post-mini2">

Tarunno</p></i>

<h2>'.$row['title'].'</h2>

<i class="far fa-user"> <p style="display: inline;"> Mahdi Hasan

<p style="display: inline; padding:0px 5px 0px 5px;"> | </p>

<i class="far fa-calendar-alt"> <p style="display:inline;">

'.$row['create\_time'].'</p></i>

<p style="display: inline; padding:0px 5px 0px 5px;"> | </p>

<i class="fas fa-paperclip" style="display:inline;"><p style="display:inline;"> '.$row['tag'].' </p></i>

<br><br>

<p>'.substr($row['body'], 0, min(strlen($row['body']), 300)).'</p>

<br>

</div>

<div class=post-mini3>

<a href="post.php?id='.$row['id'].'"> Read more </a>

</div>

</div>';

}

}

?>

</div>

</div>

<footer>

<div class="social">

<h2>FLLOW US</h2>

<a href="#"> <i class="fab fa-facebook"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-instagram"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-twitter"> <span></span> </i> </a>

<a href="#"> <i class="fab fa-youtube"> <span></span> </i> </a>

</footer>

</body>

</html>

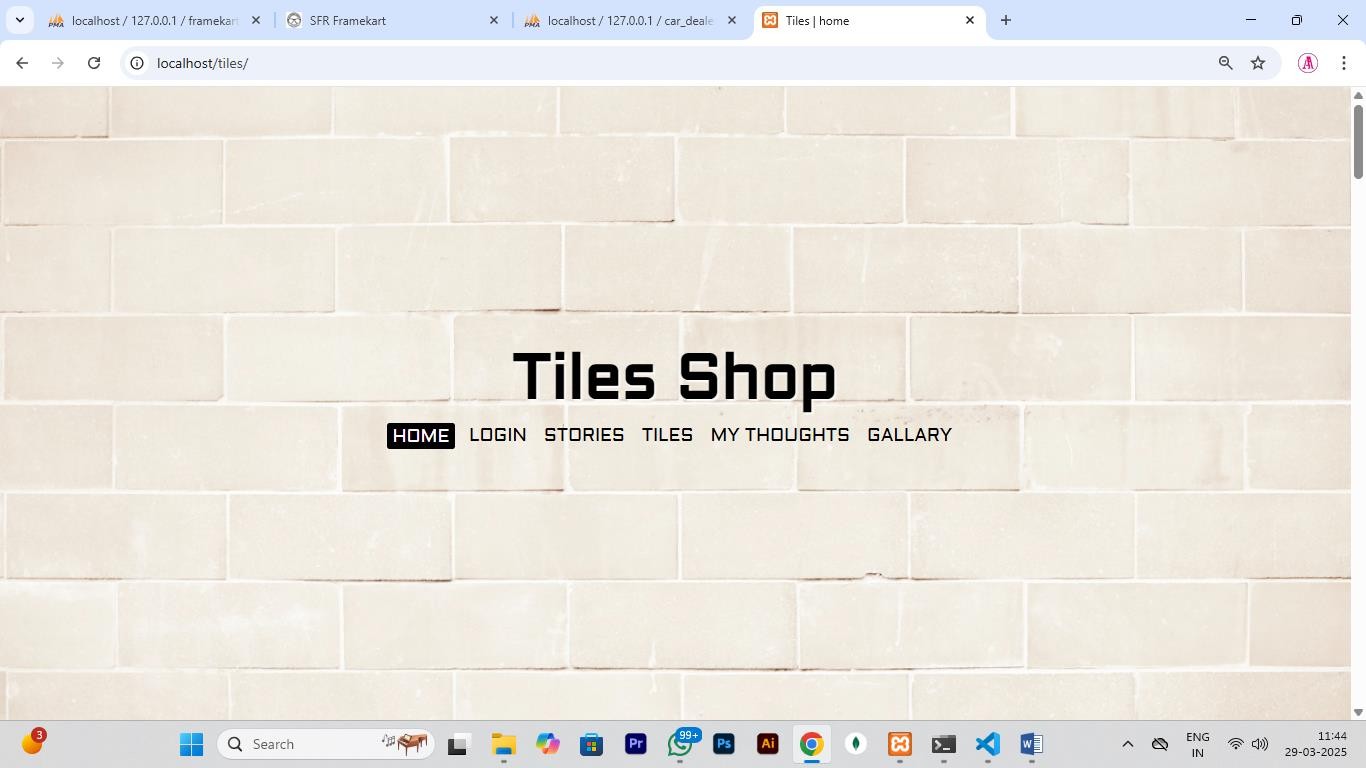
</div>

<div class="credit">

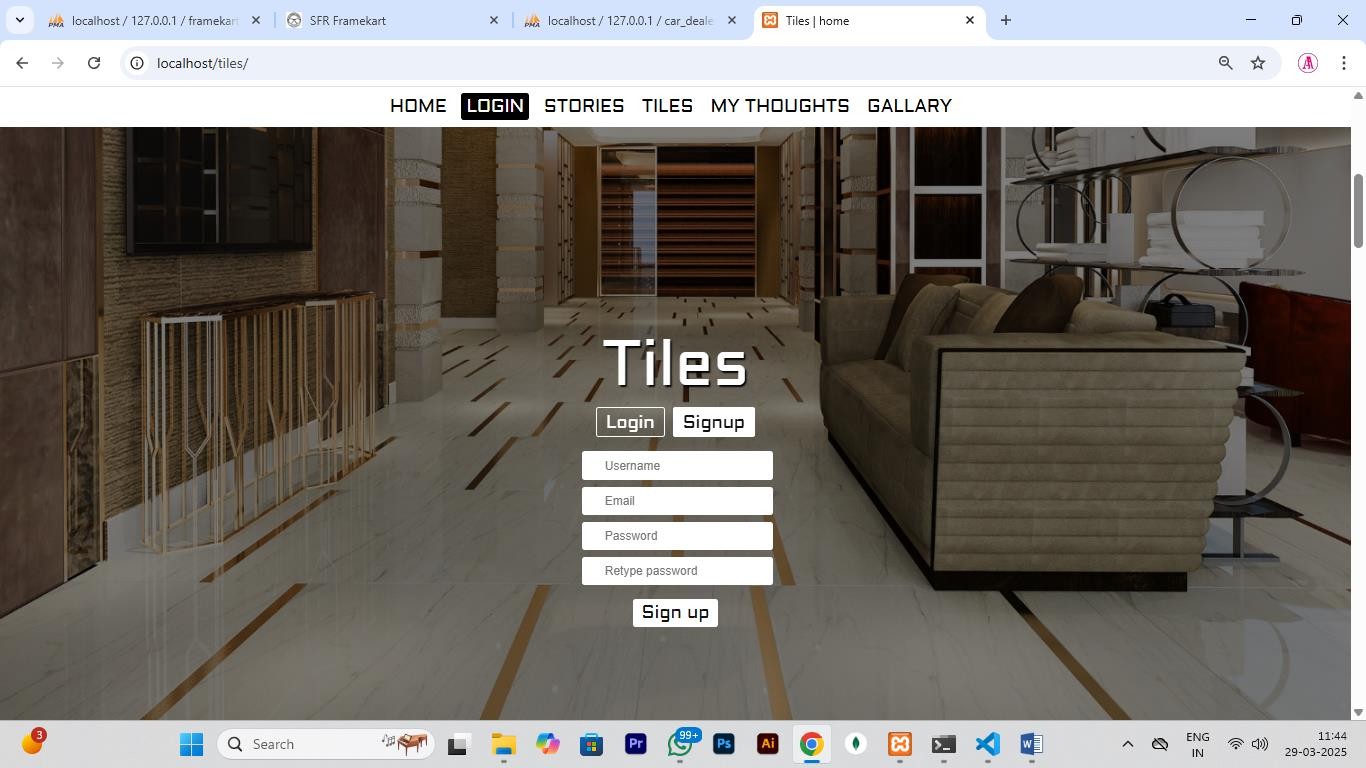
<h1>TARUNNO DRIVES | Developed by mahadi hasan tarunno</h1>

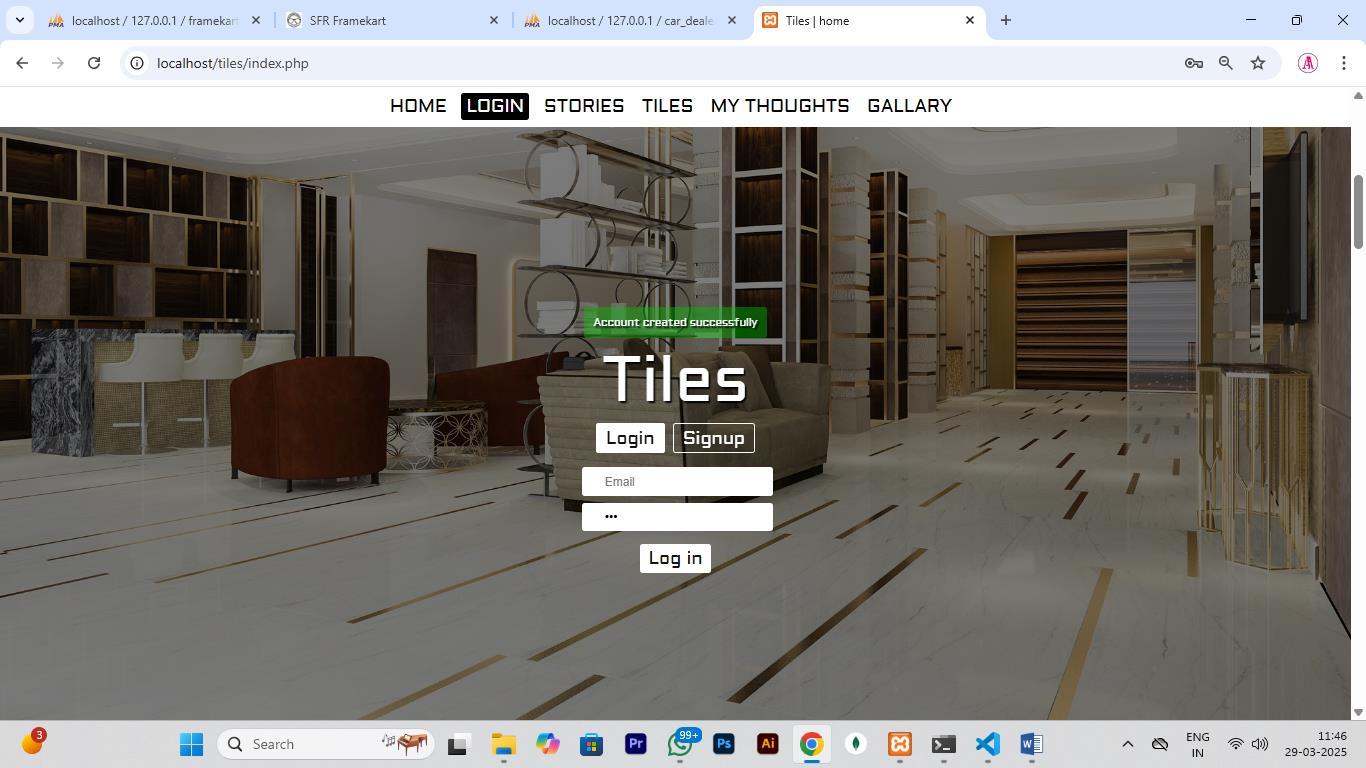
</div>

* 1. **SCREEN LAYOUTS HOME PAGE**

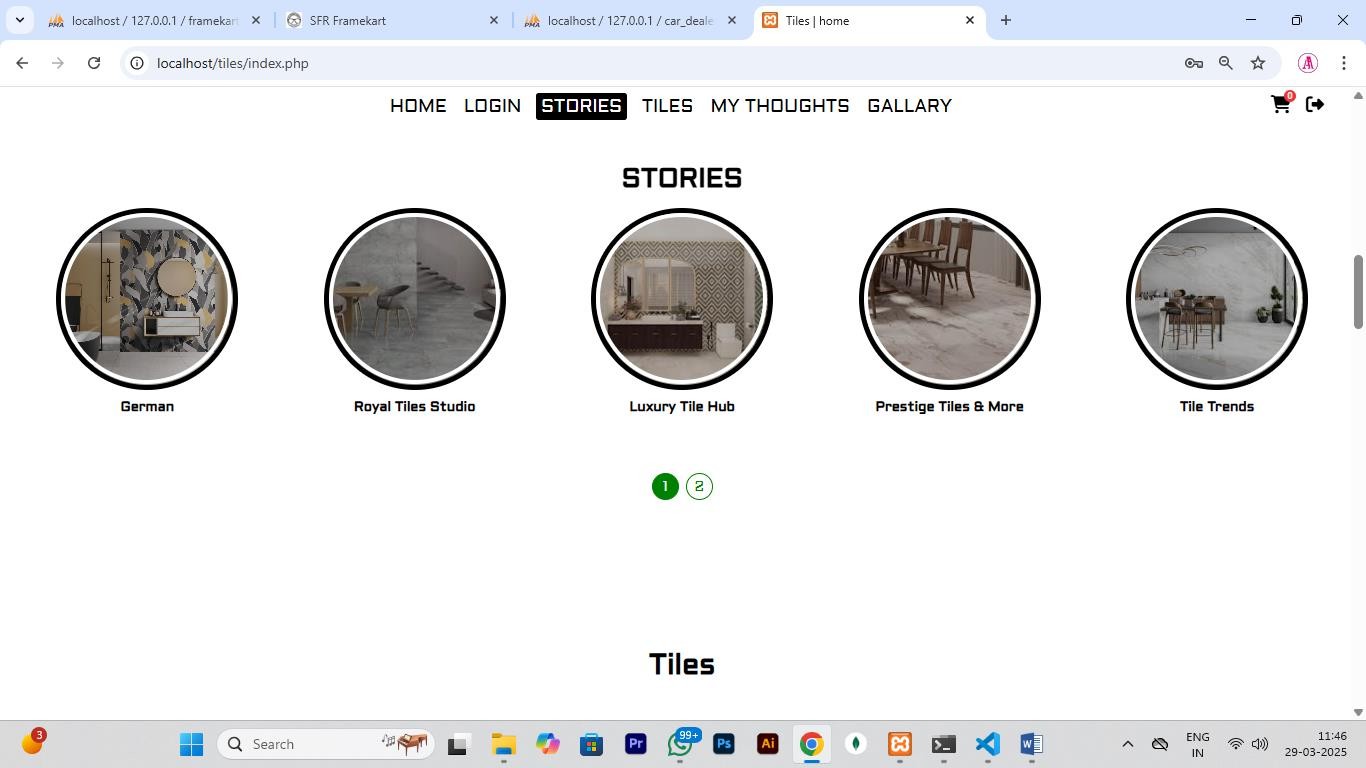


**LOGIN PAGE**

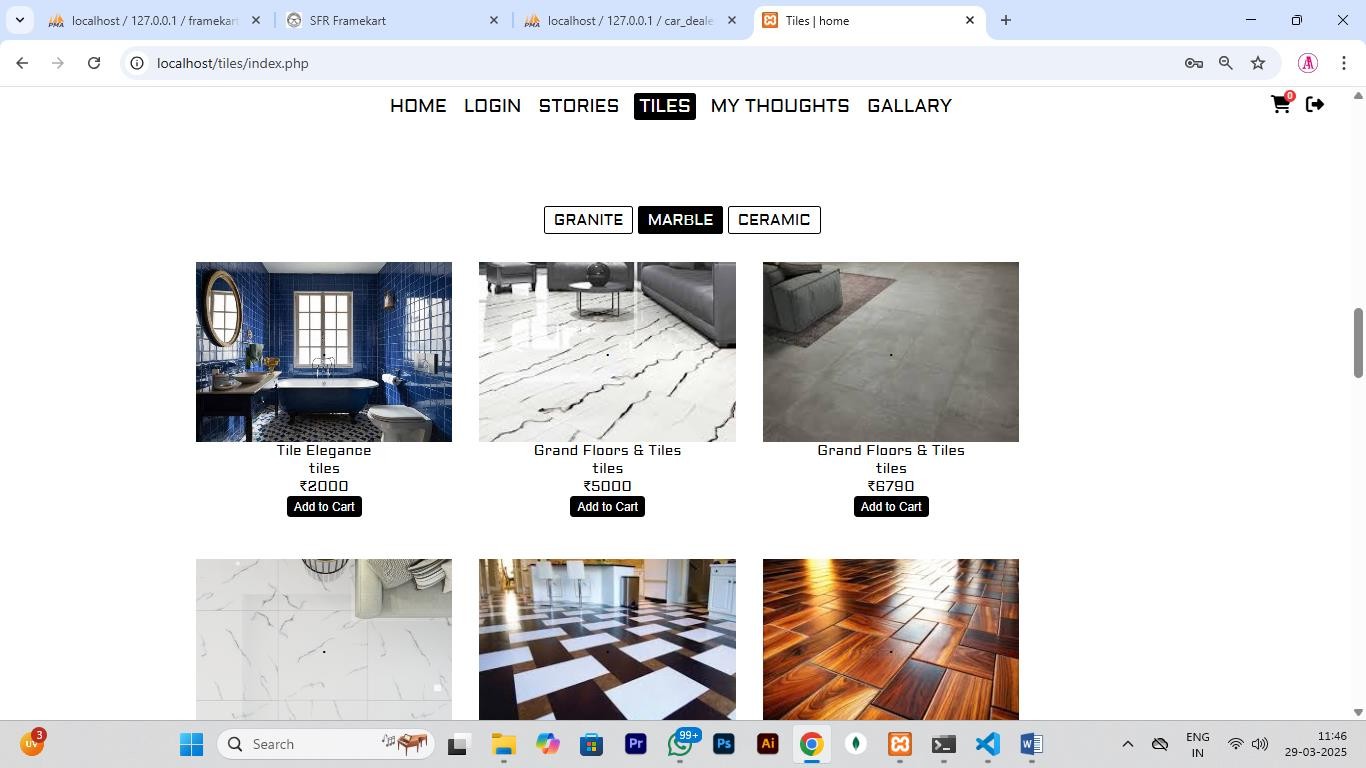
****

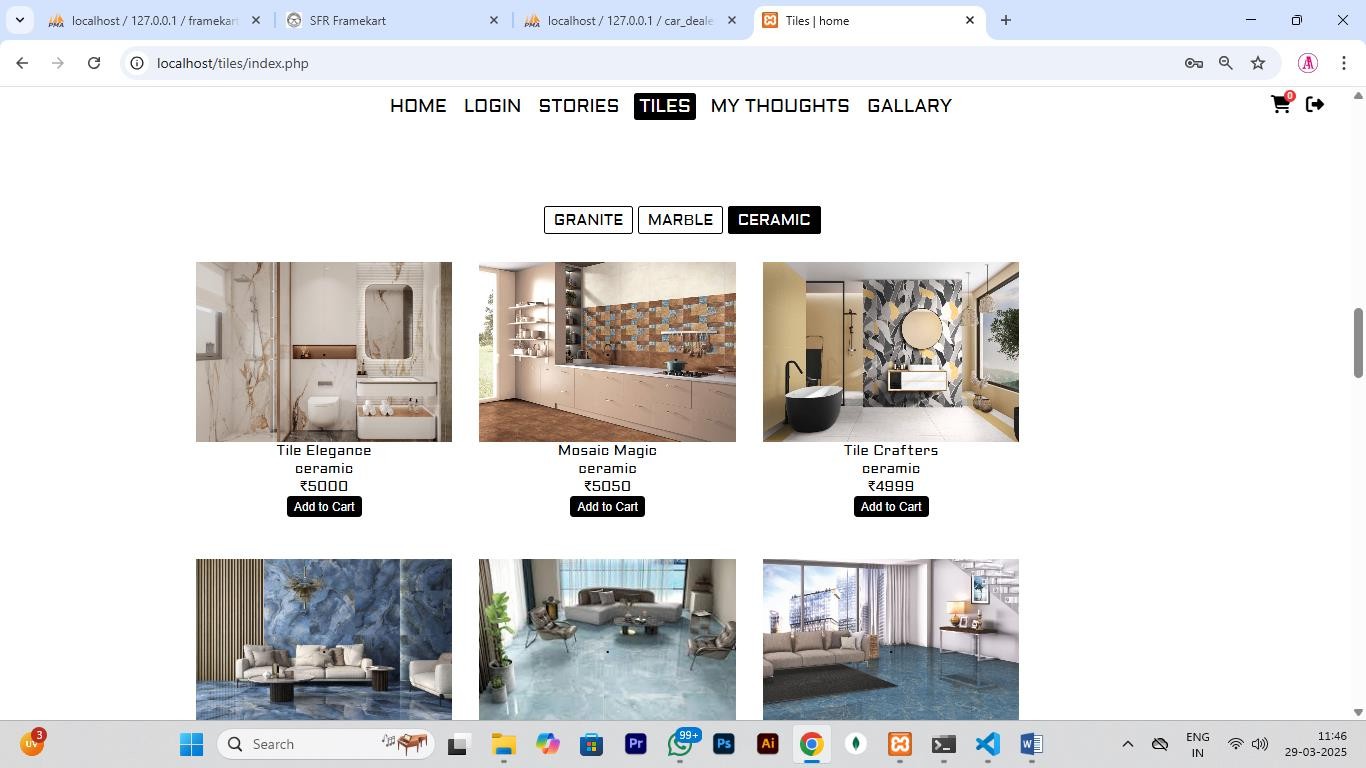


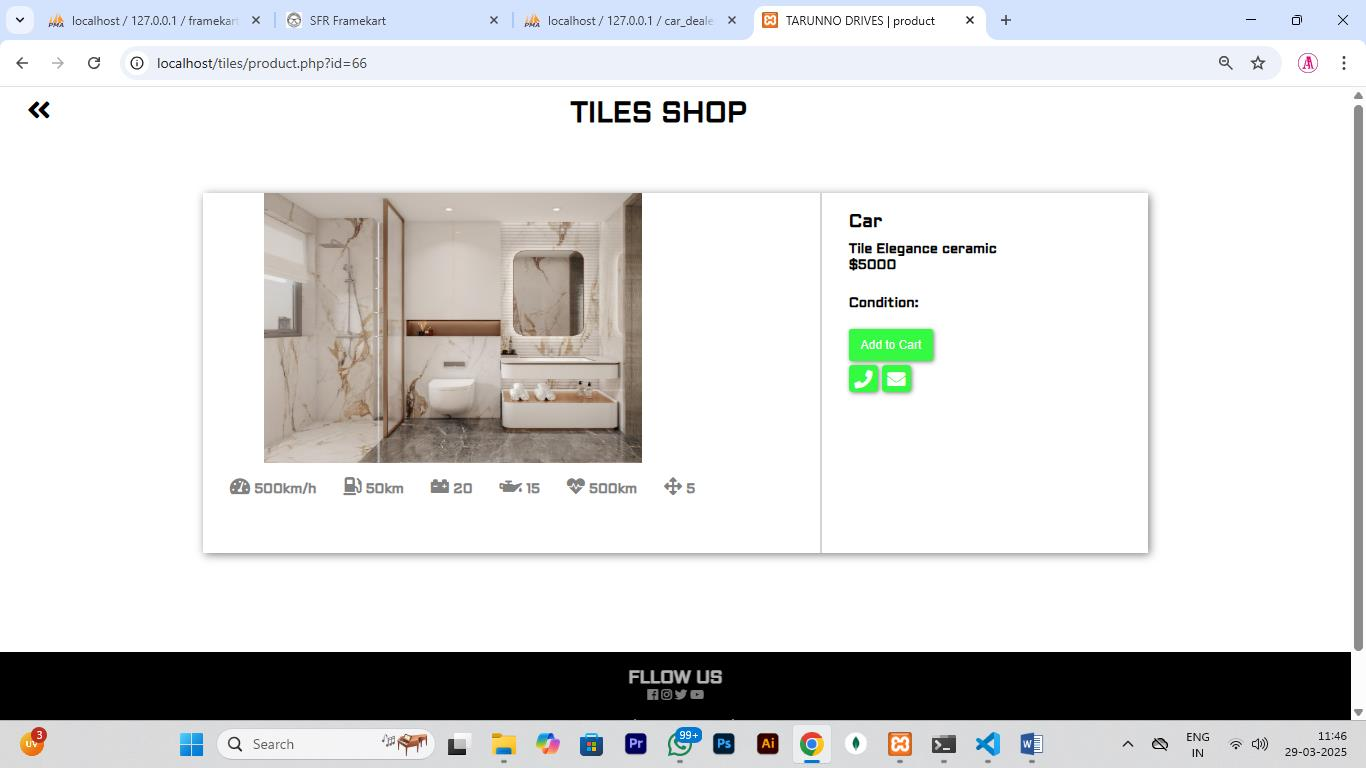
**STORIES PAGE**

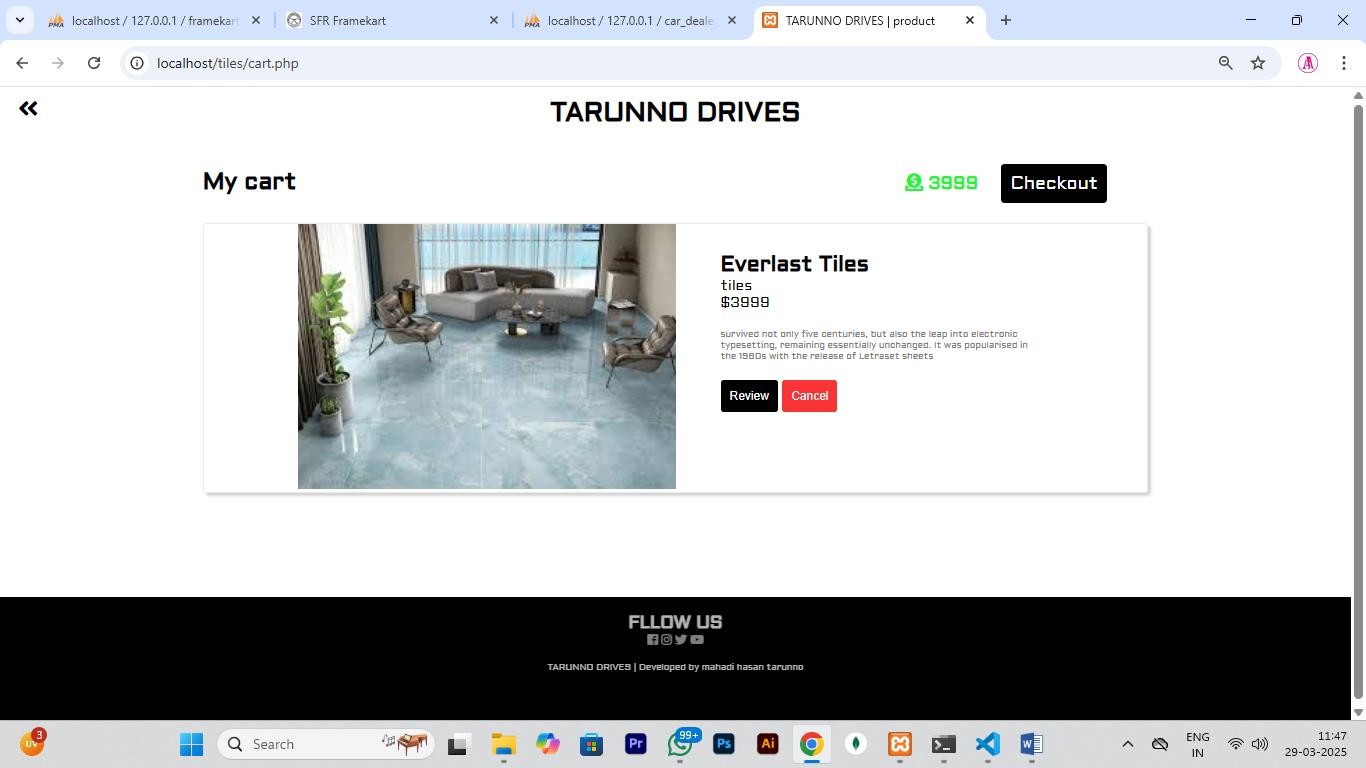
****

**TILES PAGE**

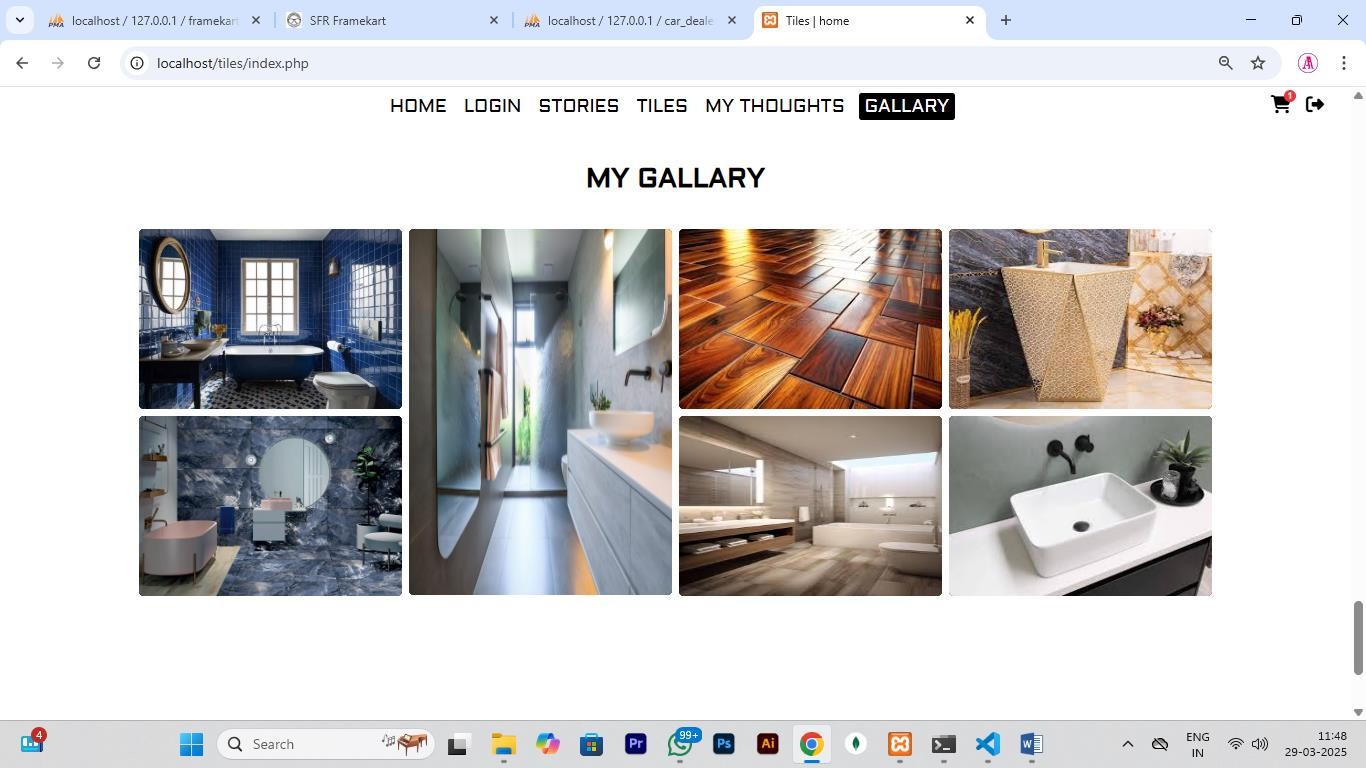
****

****

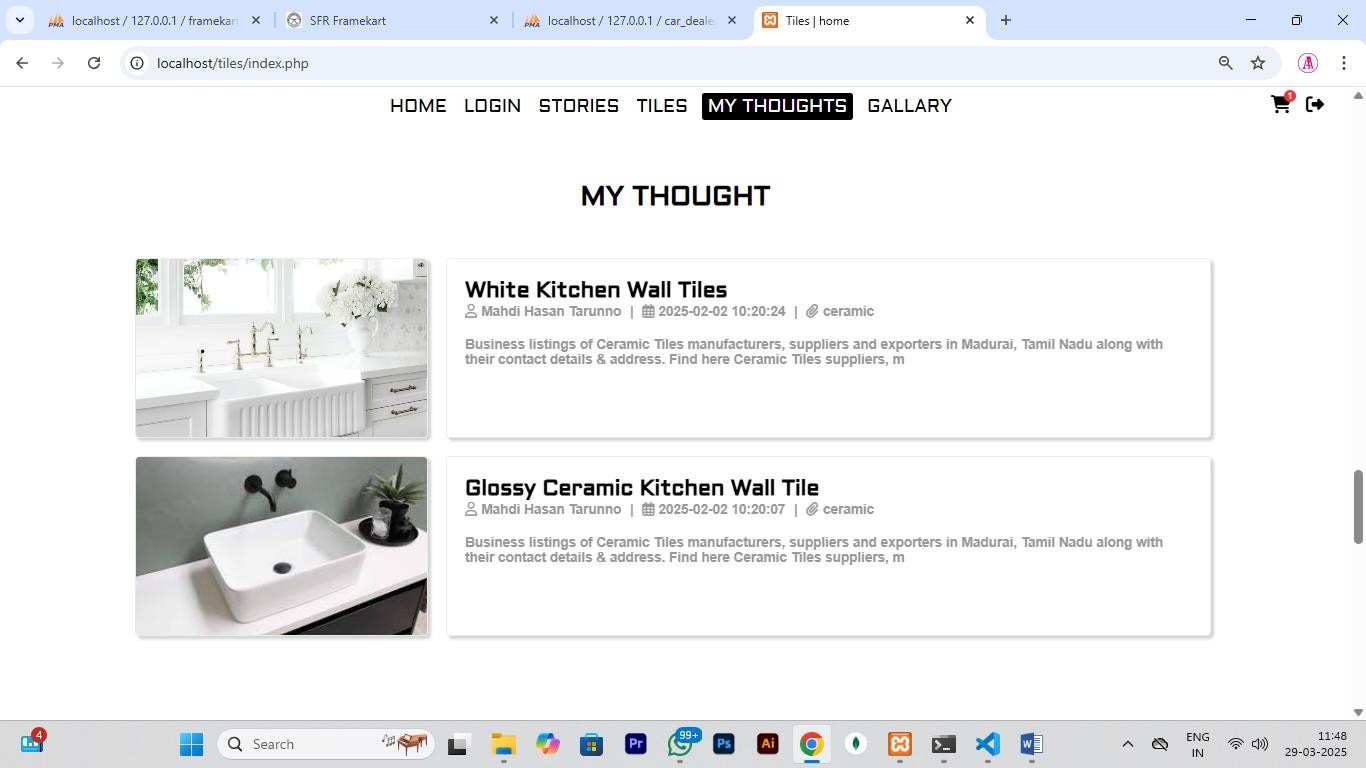


****

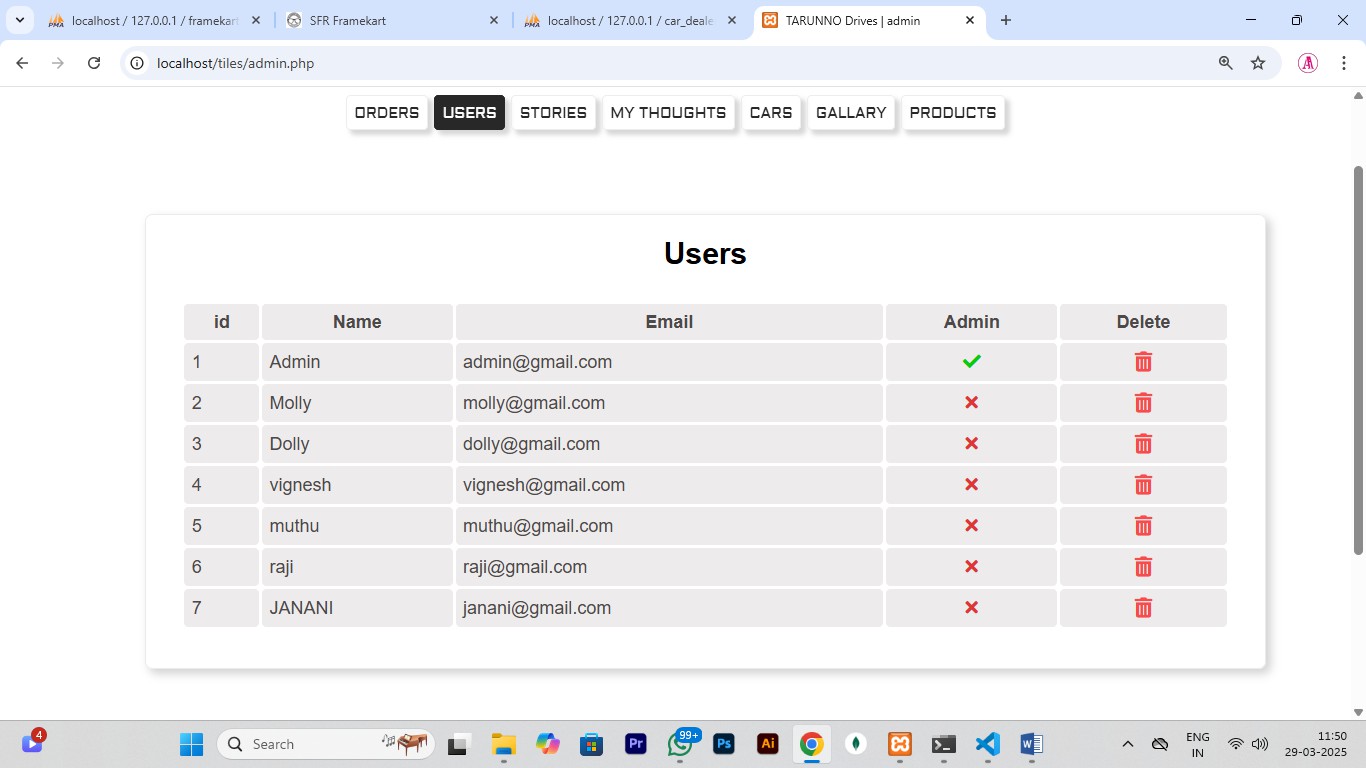
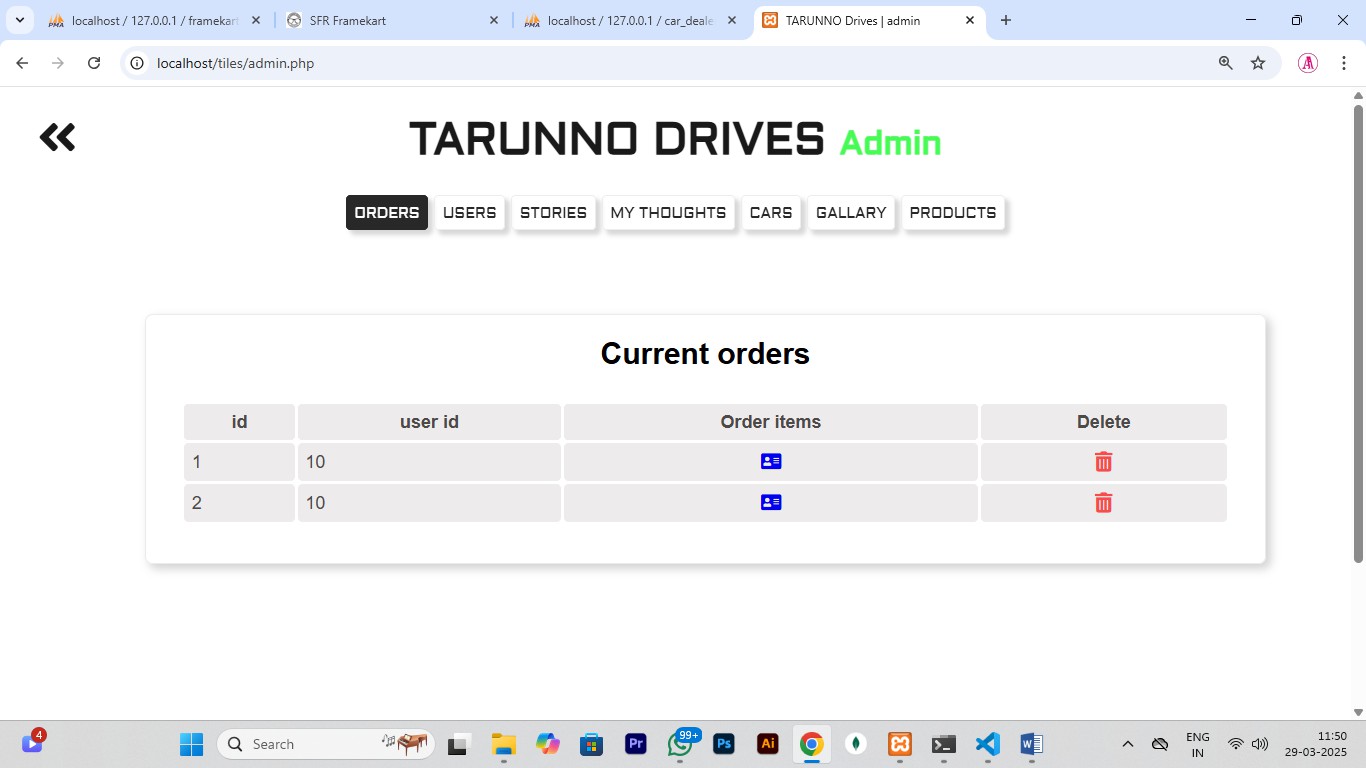
**GALLARY PAGE**

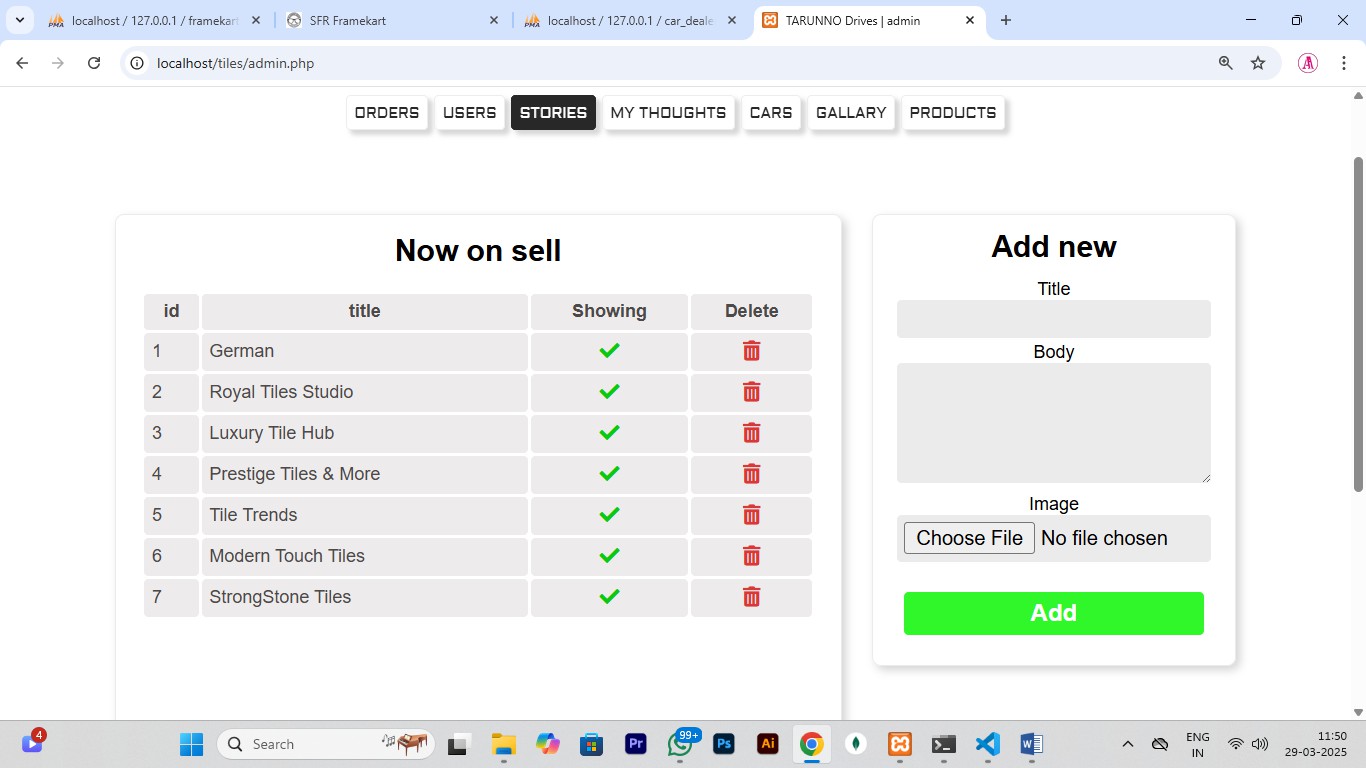
****

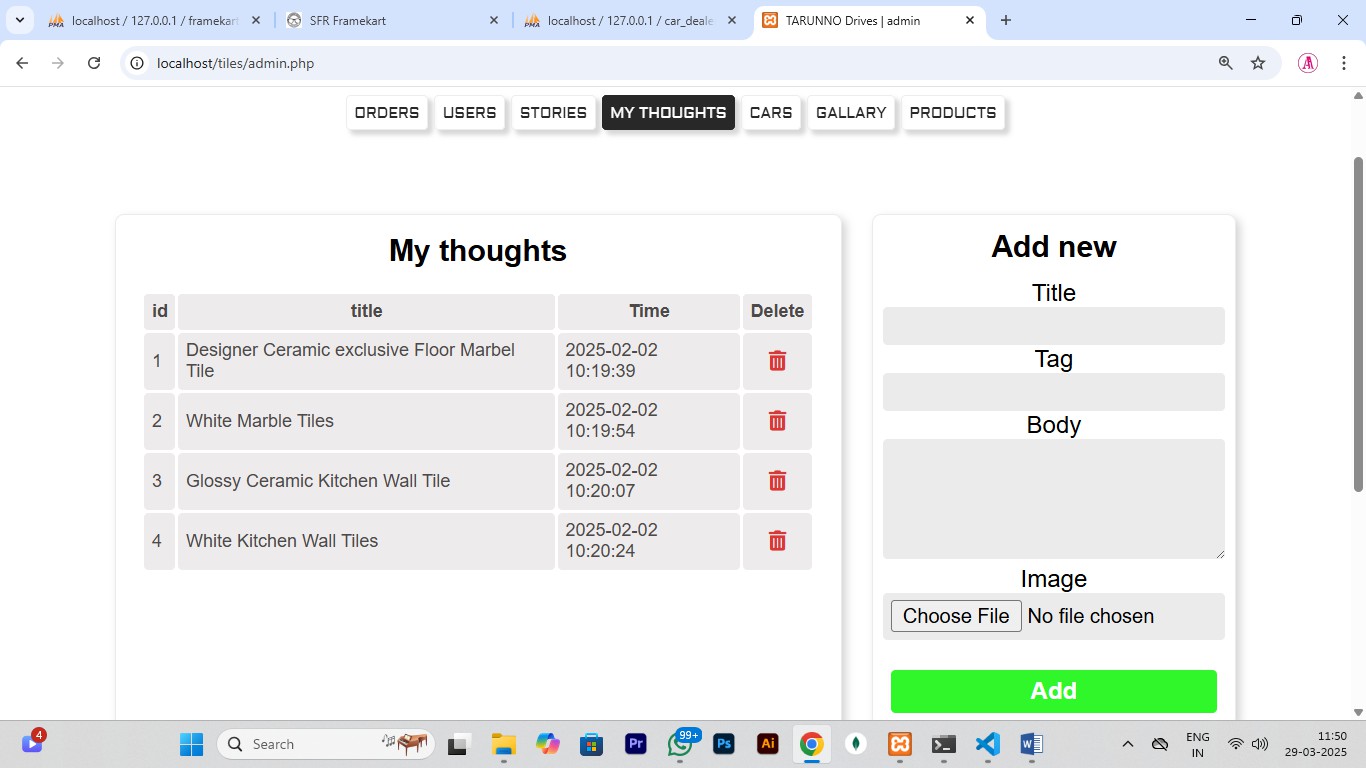
**MY THOUGHTS PAGE**

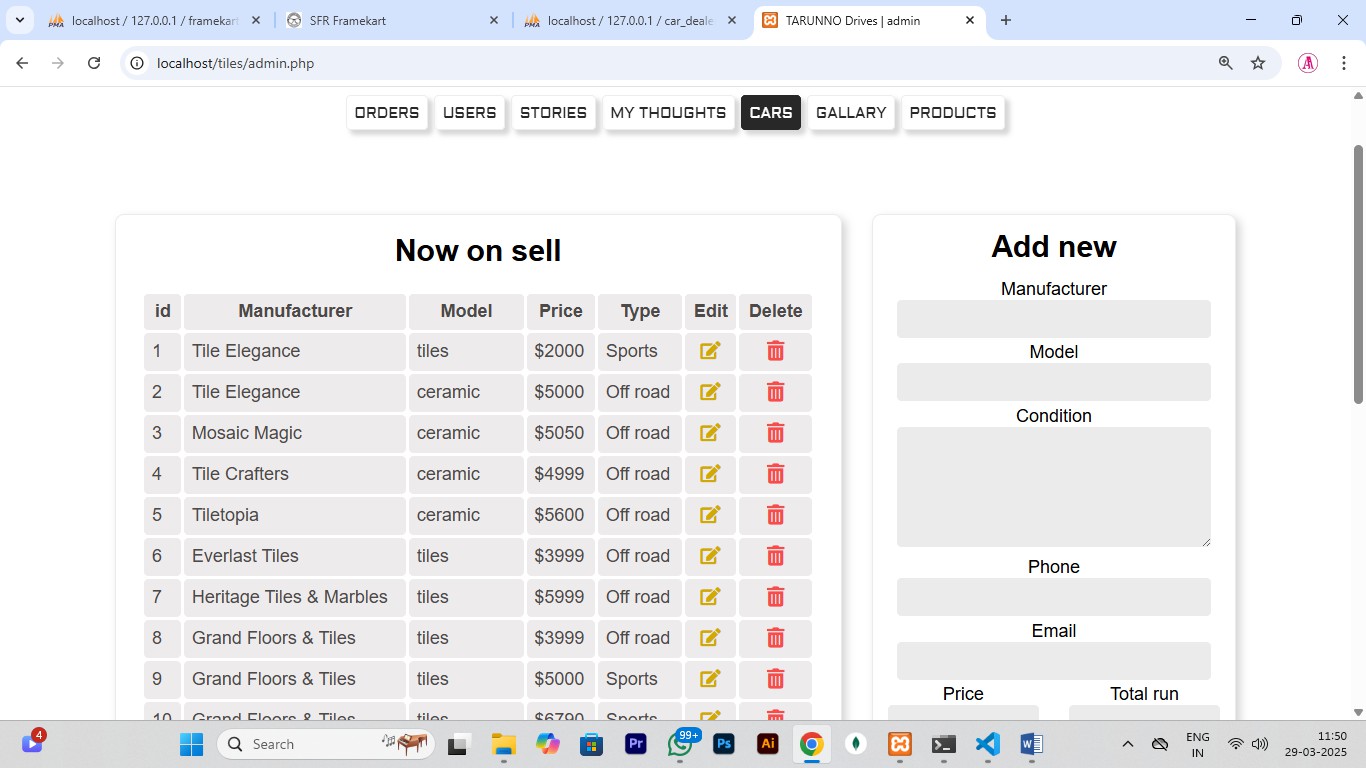
****

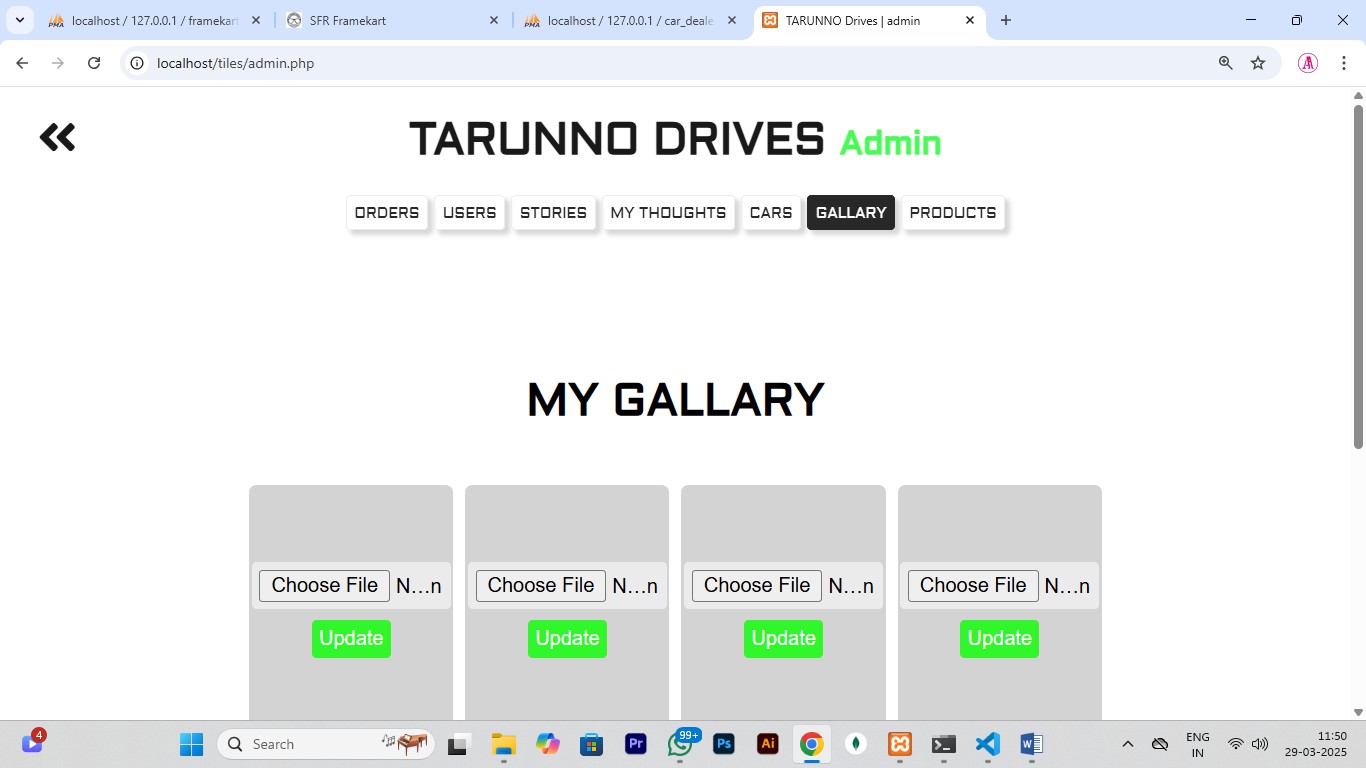
**ADMIN PAGE**

****

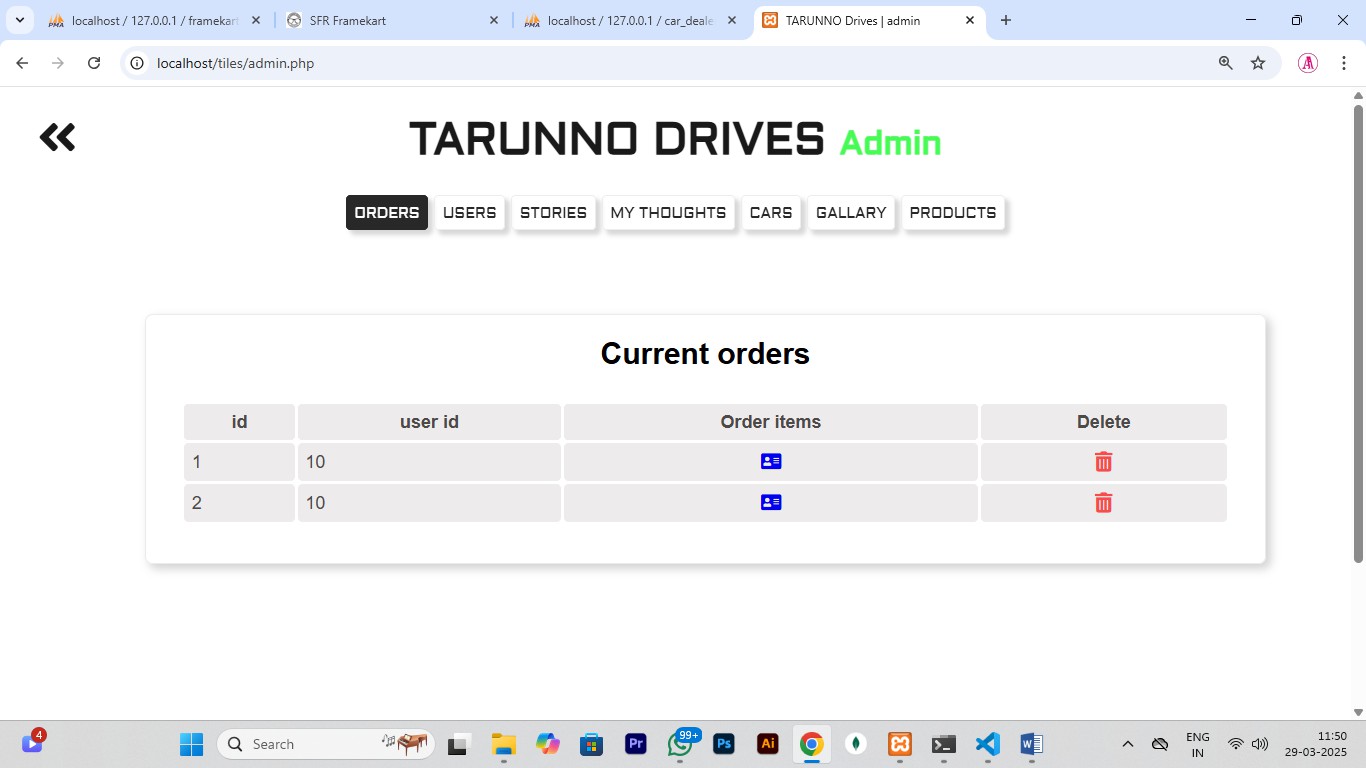


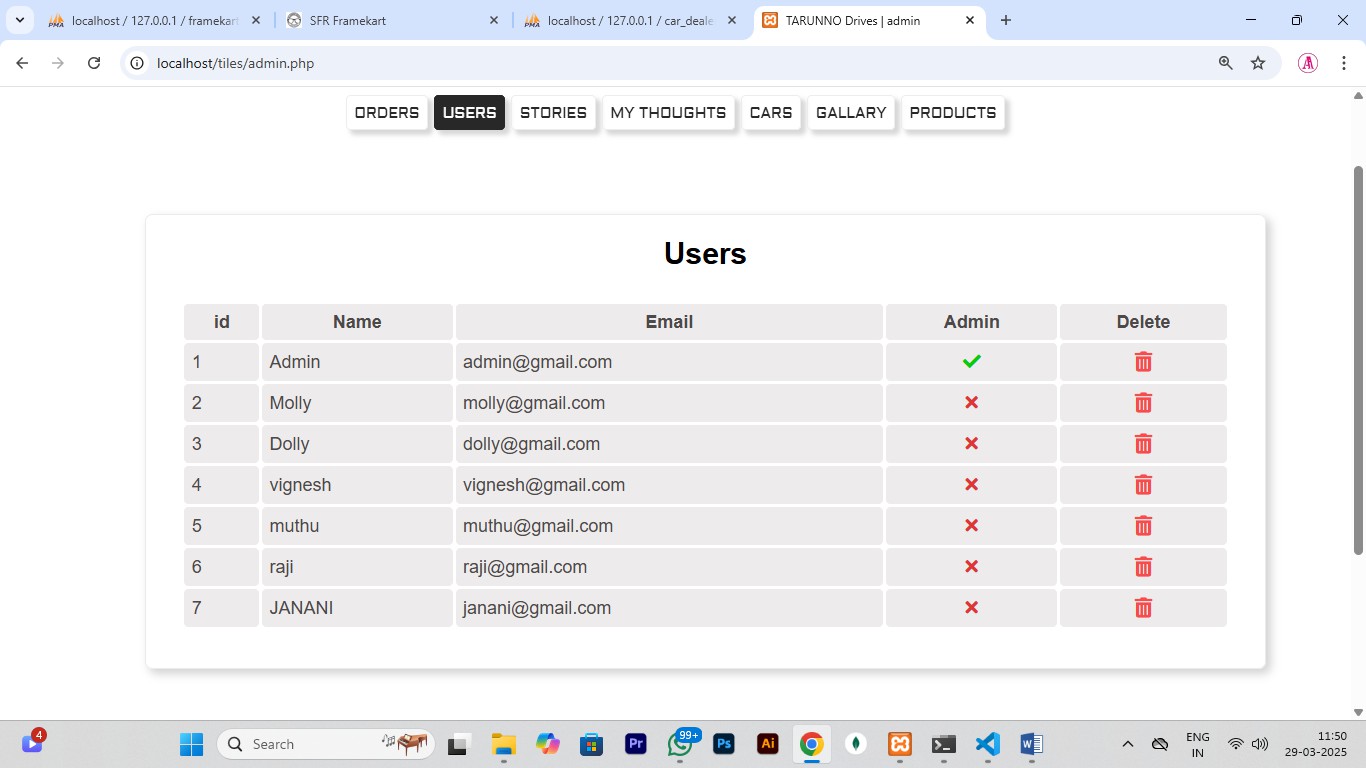
****

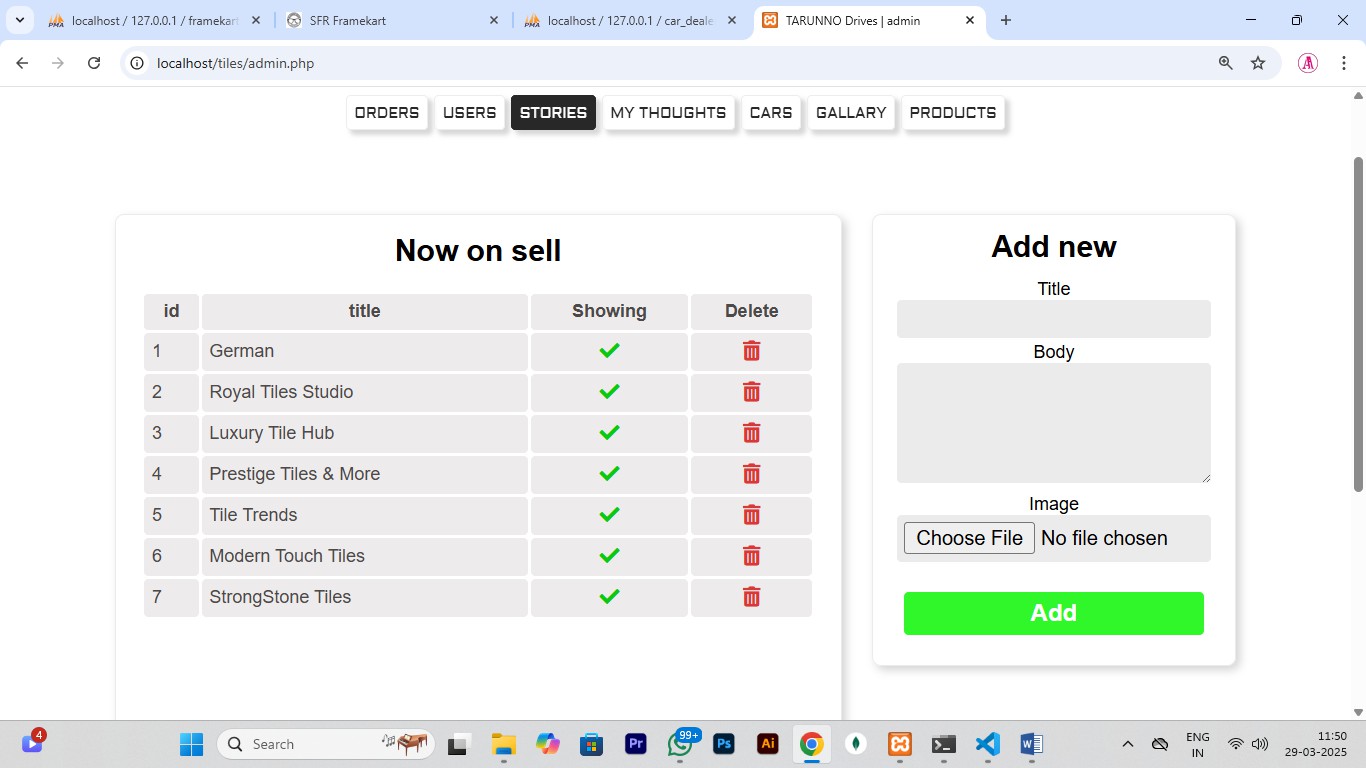


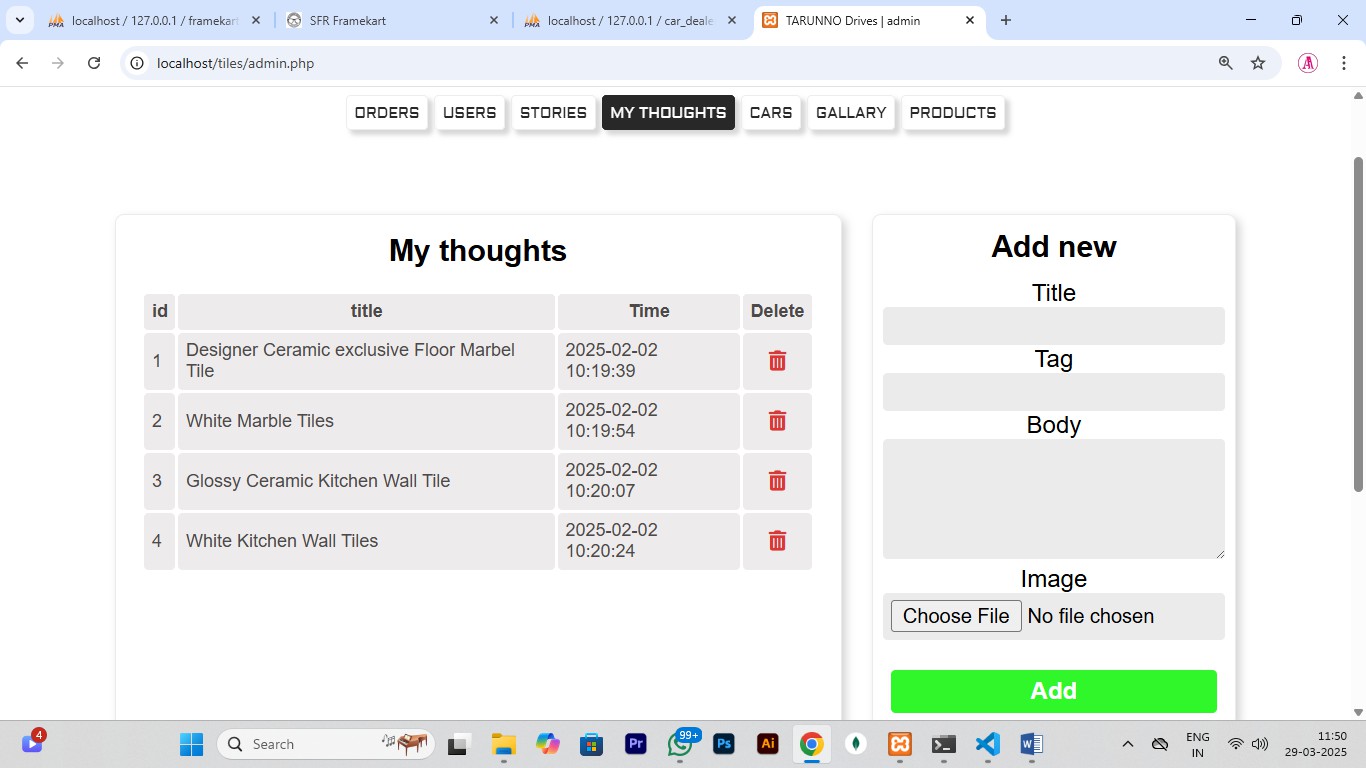
****

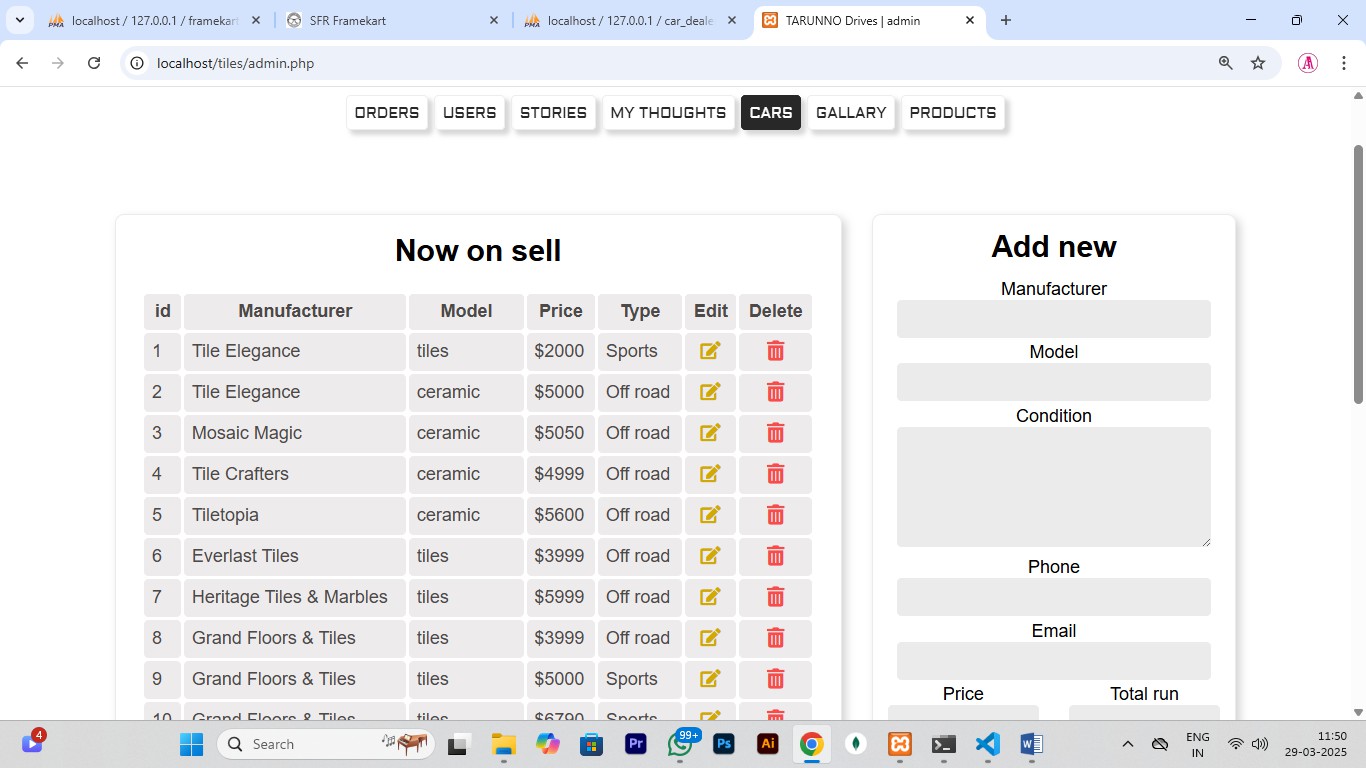
* 1. **Report Admin page:**

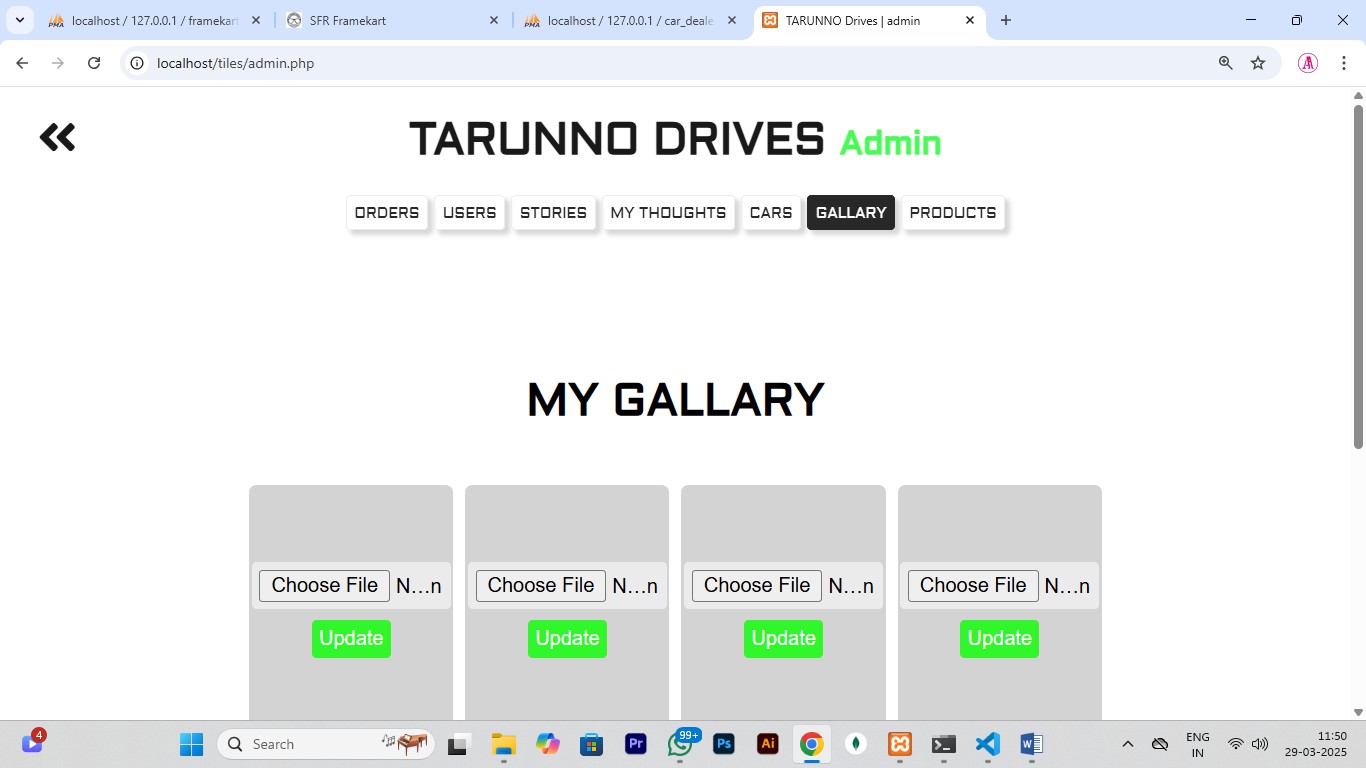


****



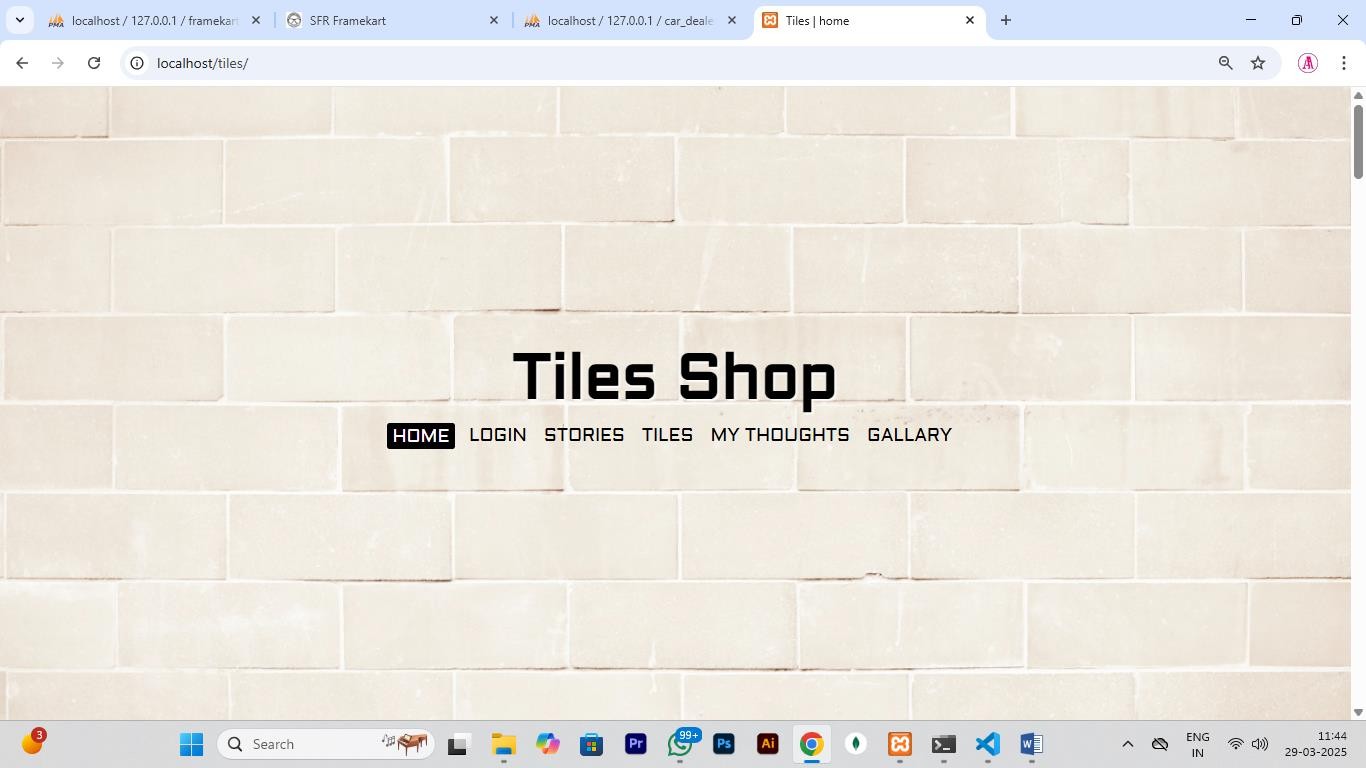
****



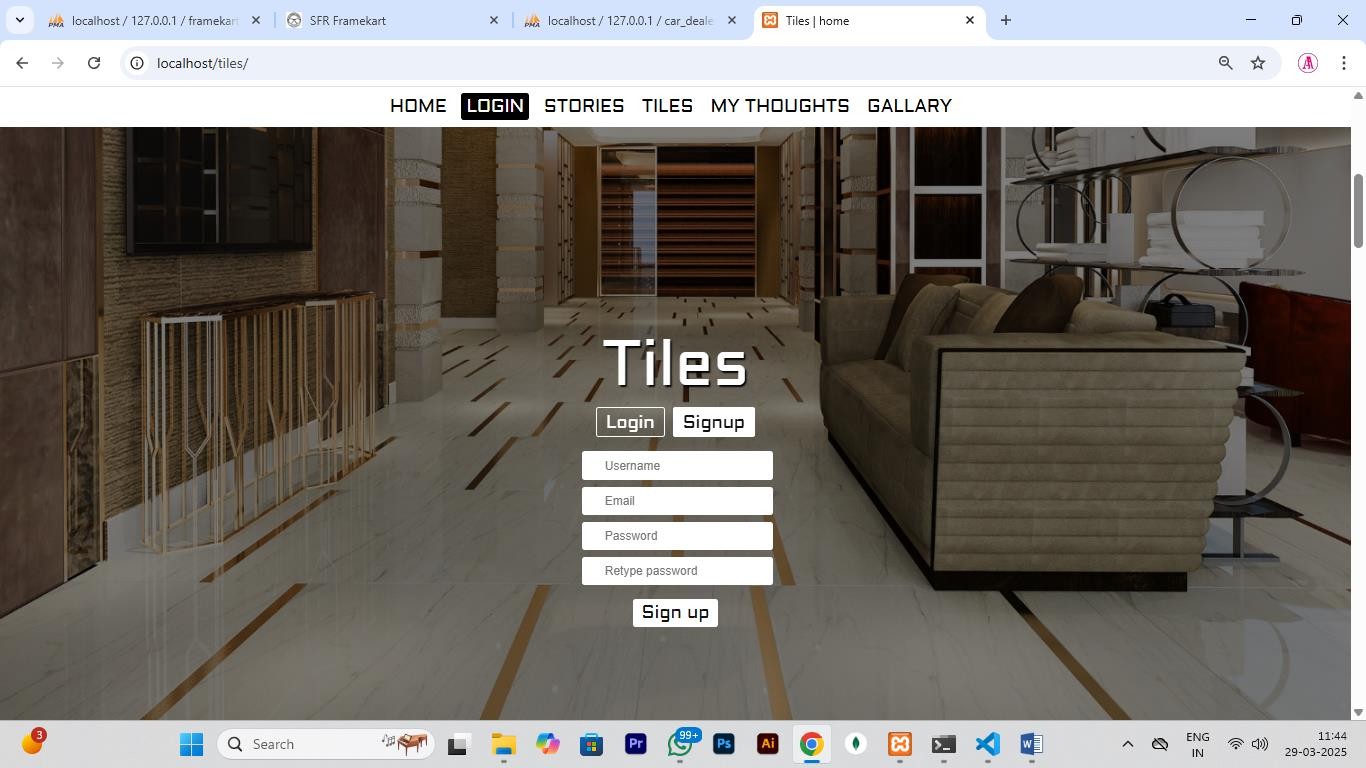
****

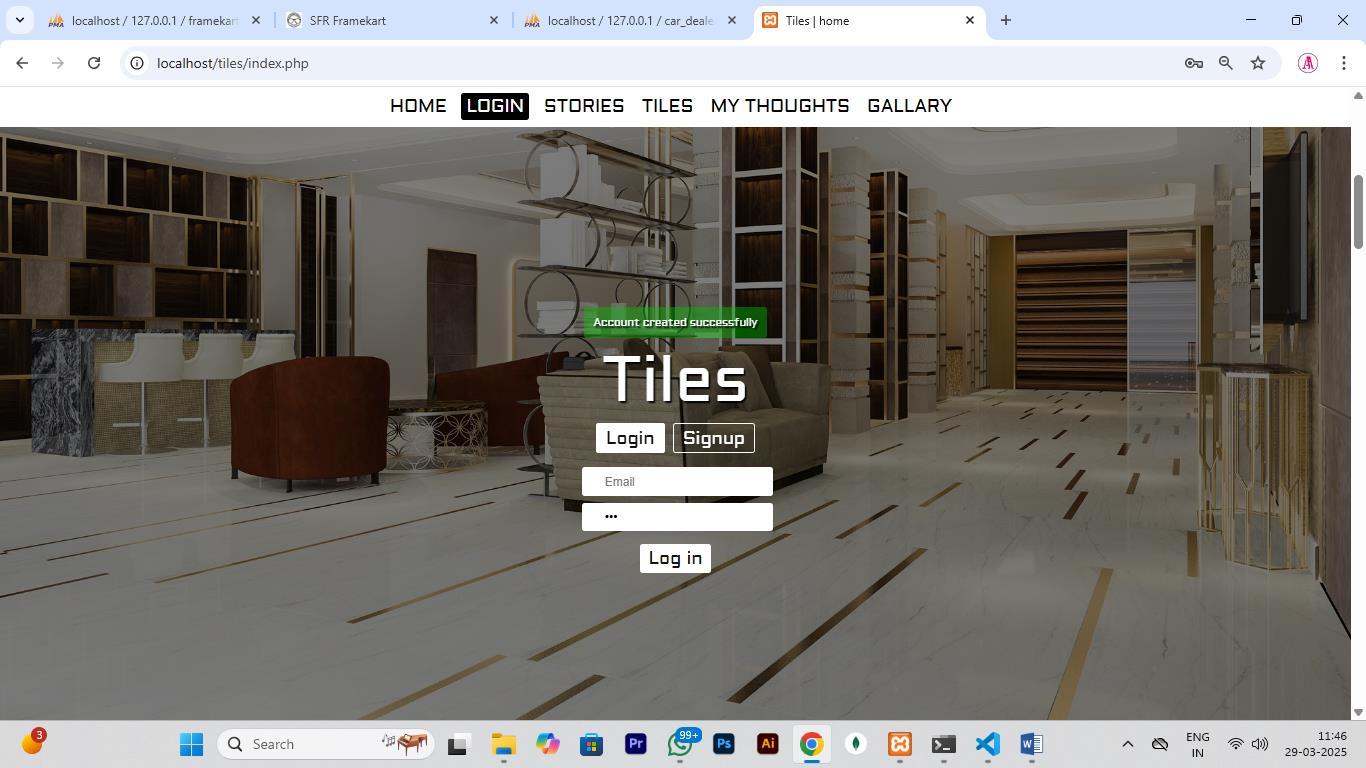
**Users Page:**

**HOME PAGE**

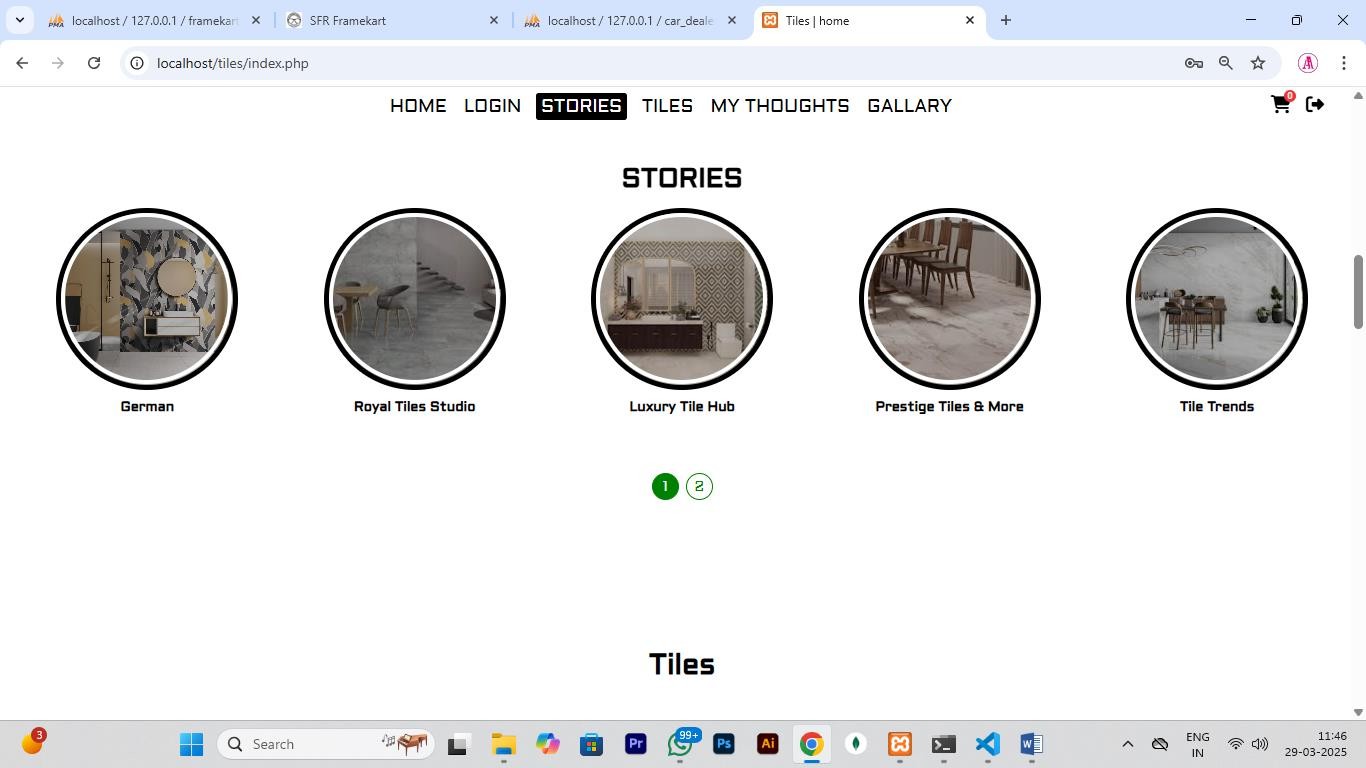
****

**LOGIN PAGE**

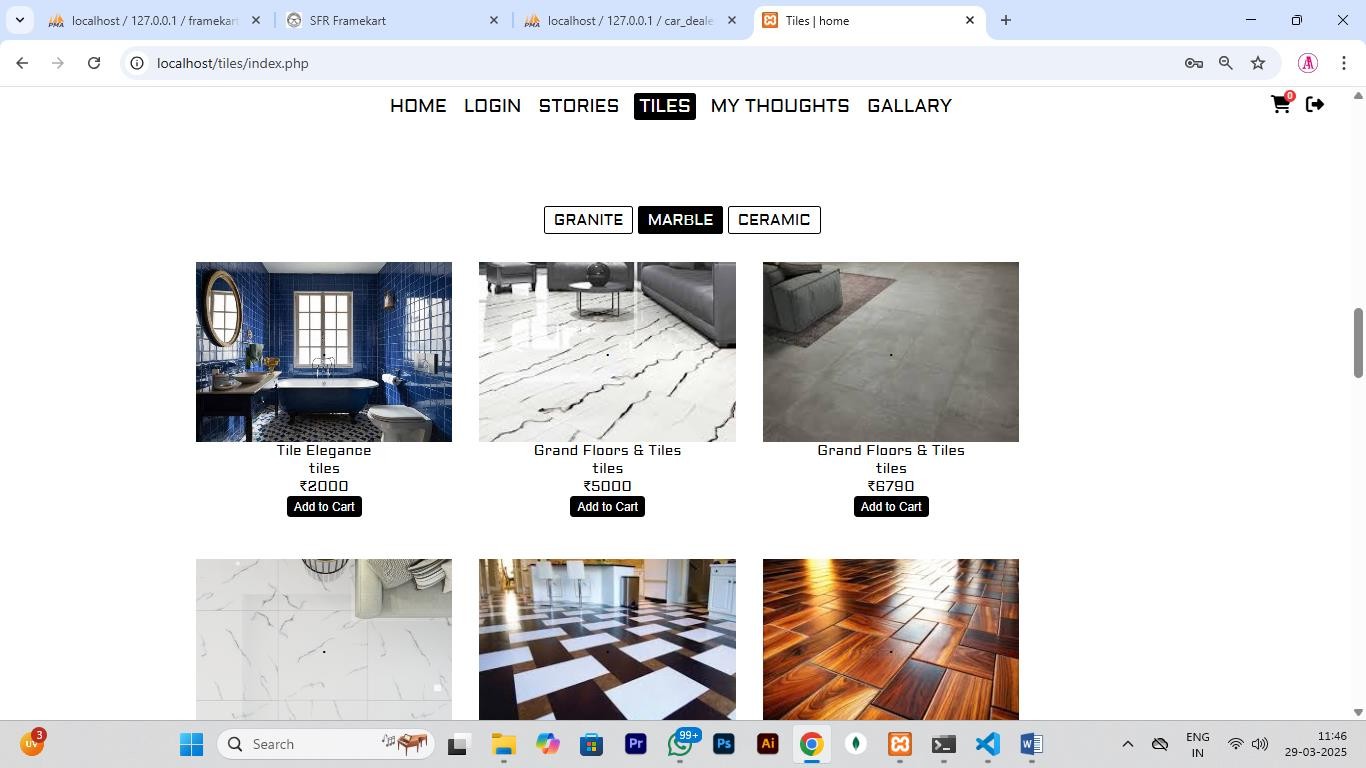
****

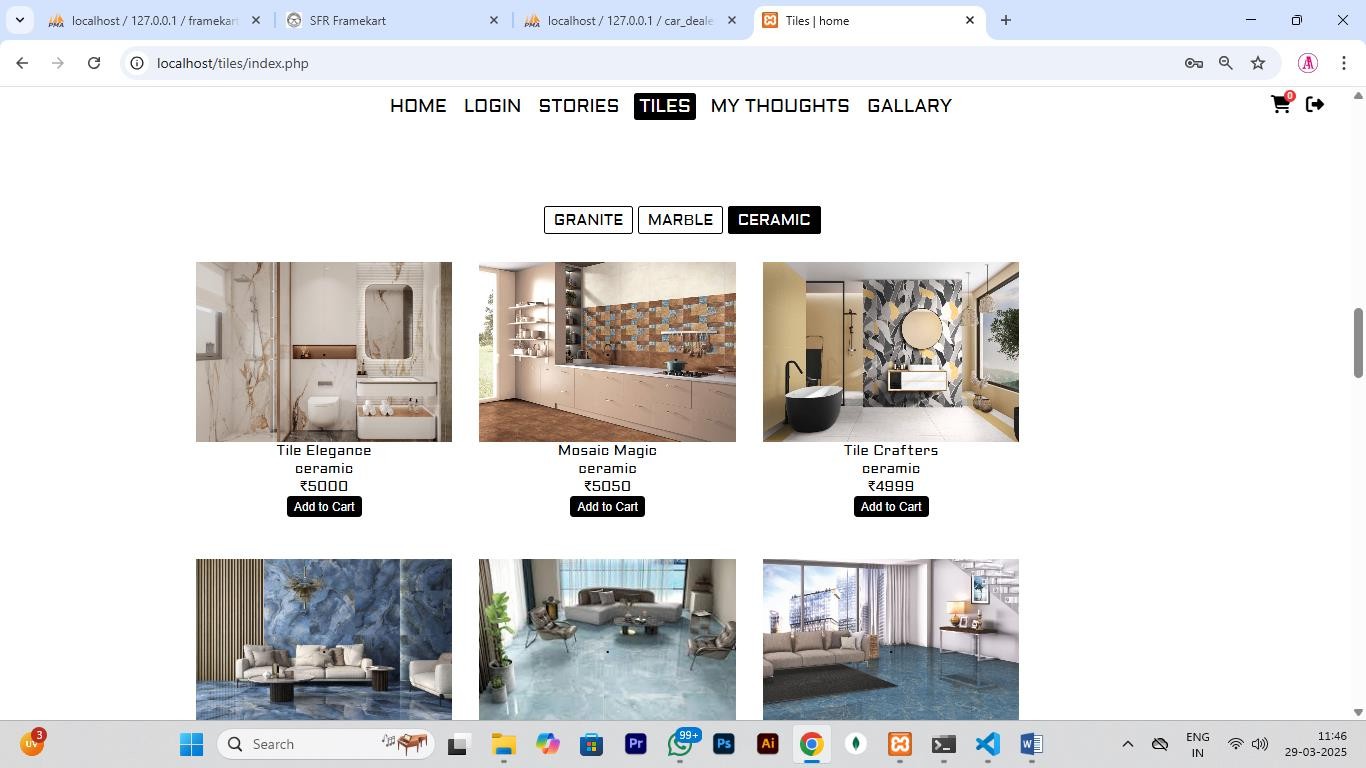


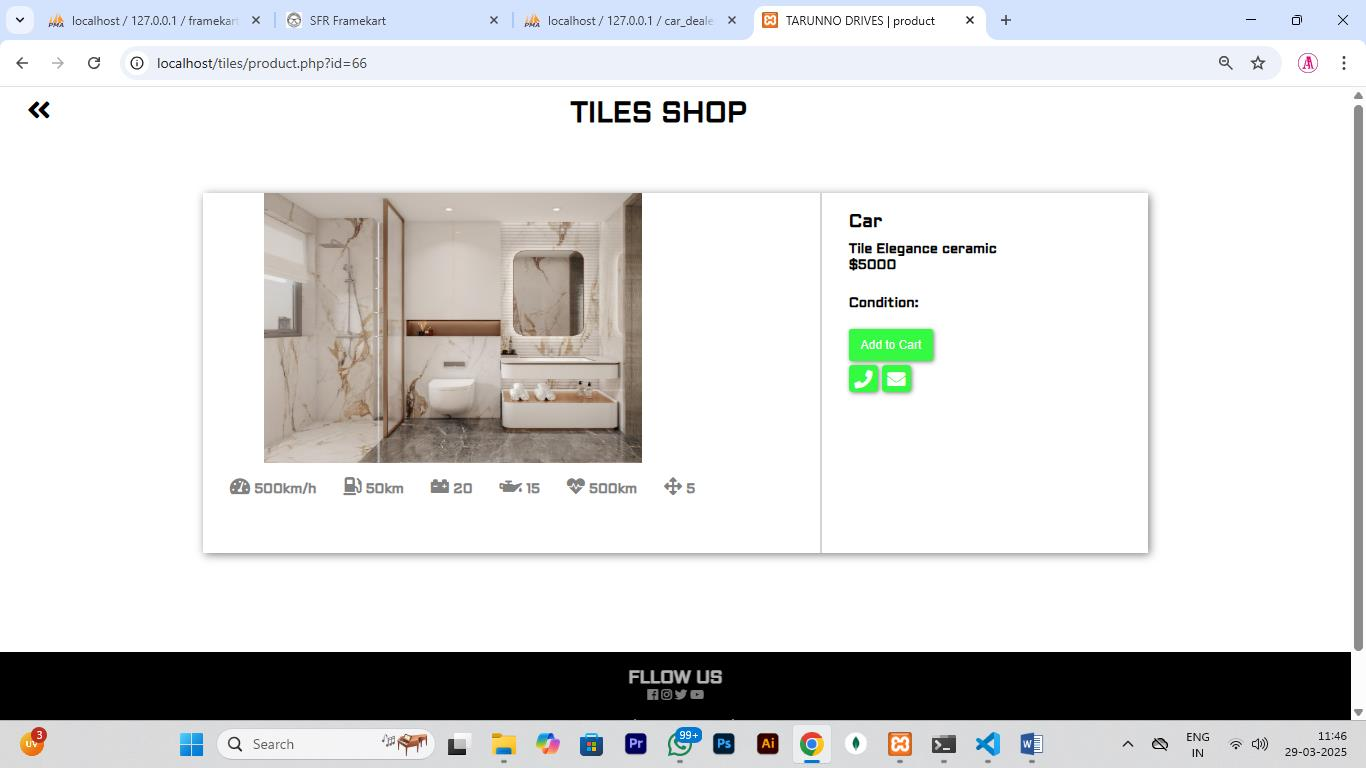
**STORIES PAGE**

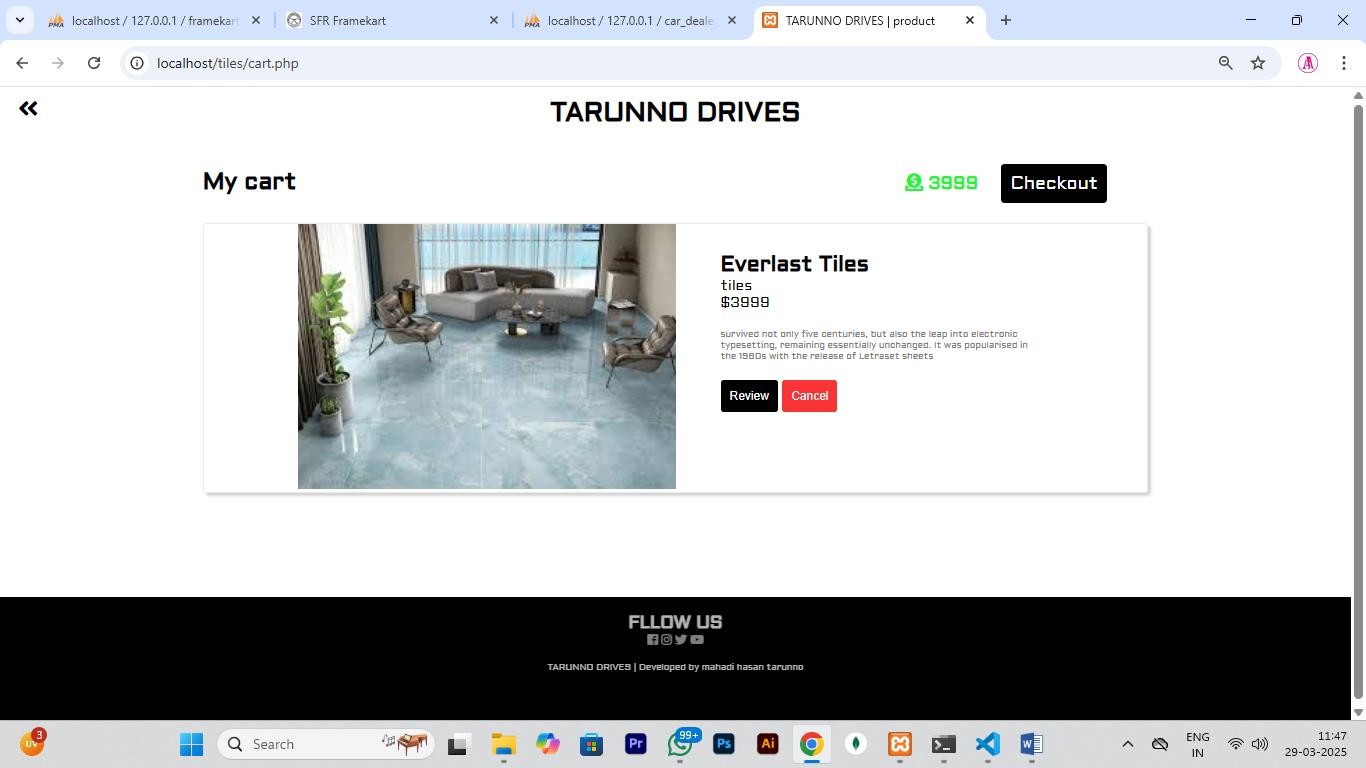
****

**TILES PAGE**

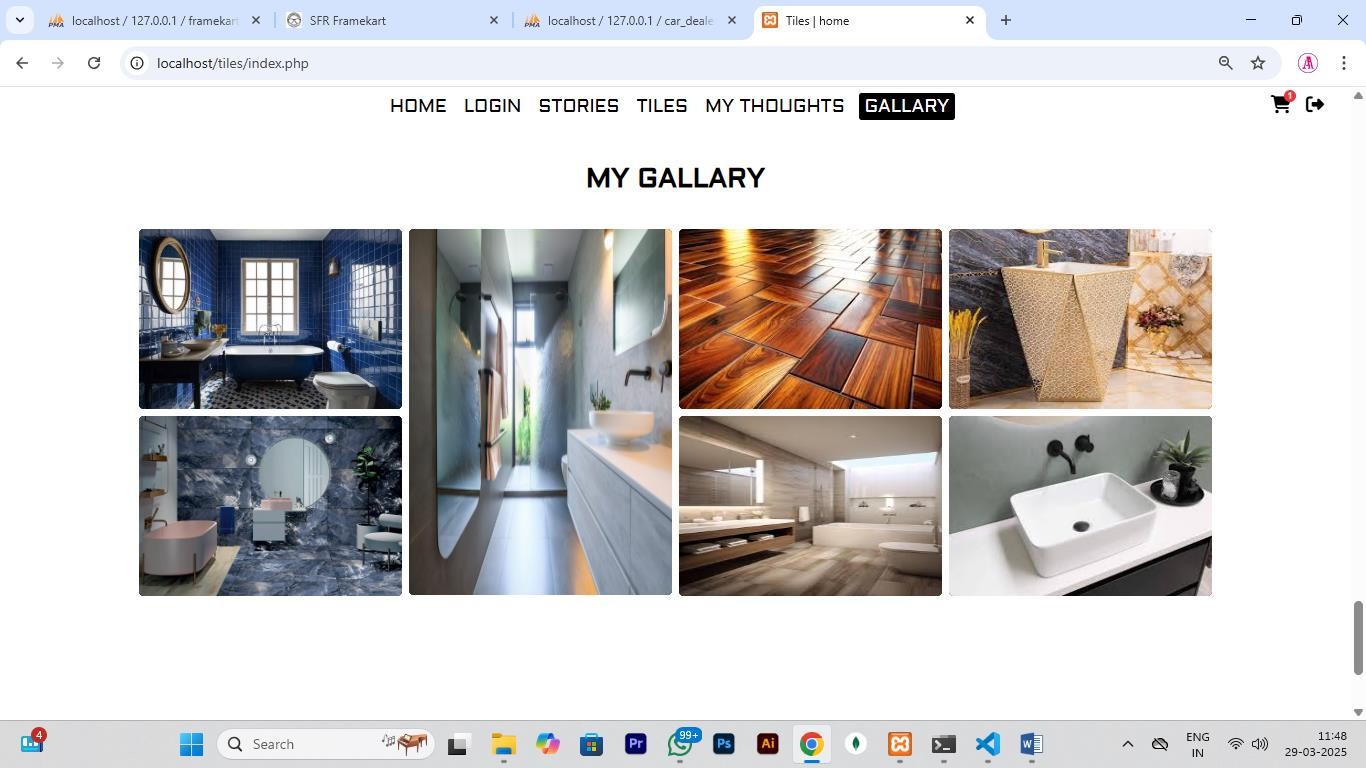
****

****

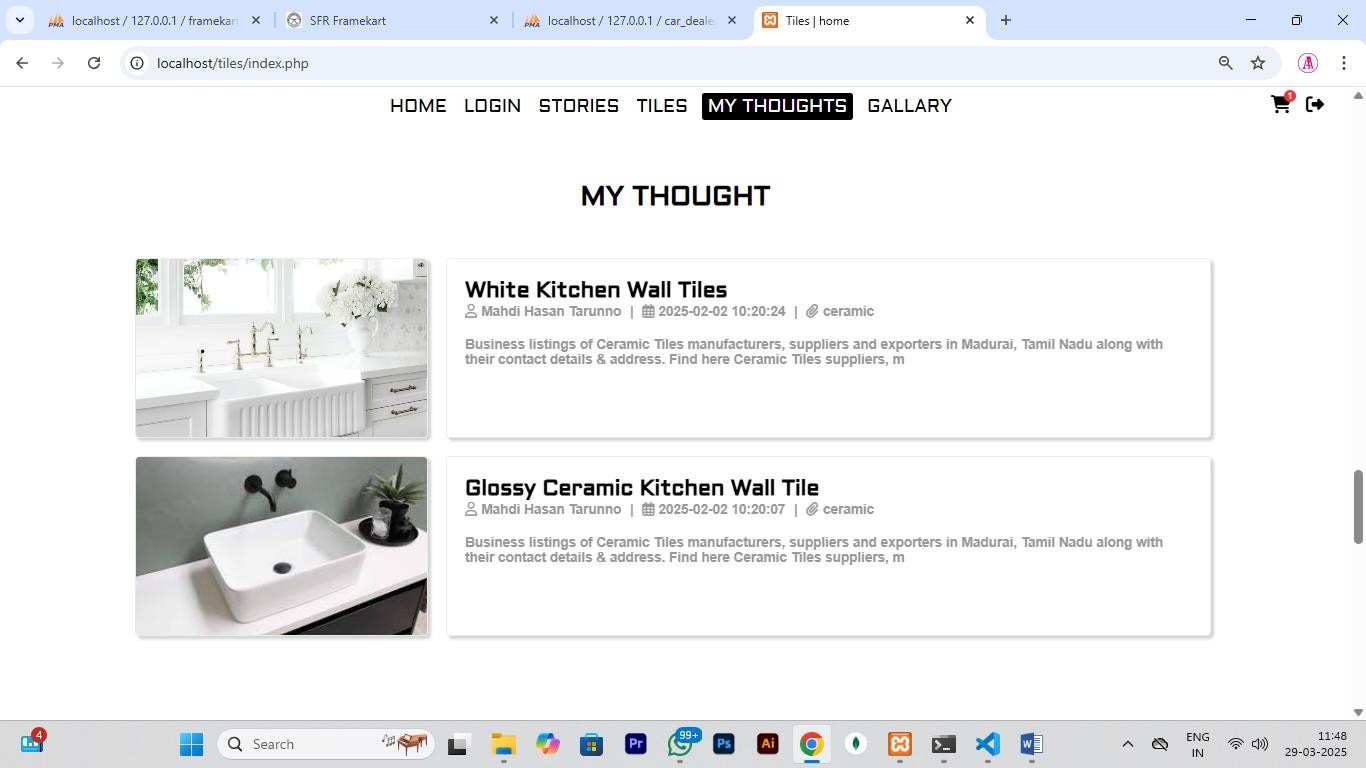


****

**GALLARY PAGE**

****

**MY THOUGHTS PAGE**

****

1. **Bibliography**

# Referred Websites

* **[www.kashipara.com/project/php/98/ceramic-company-tiles-product-project-in-](http://www.kashipara.com/project/php/98/ceramic-company-tiles-product-project-in-php) [php](http://www.kashipara.com/project/php/98/ceramic-company-tiles-product-project-in-php)**
* **<https://github.com/openlayers/website/blob/master/tile.php>**
* **<https://www.scribd.com/document/410248945/Marble-Tiles-Report-doc>**
* **<https://www.studocu.com/in/document/mahatma-gandhi-university/bachelor>**

# Referred Books .

* Tom Butler & Kevin Yank, ”PHP &MySQL”,Ninja”,6th Edition ,site point,2020 .
* Vikram Vaswani,”PHP:A Beginers Guide”,5TH Edition ,MeGraw-Hill Education,2019
* Lynn Beighley & Michael Morrison, ”Head first PHP & MYSQL “,First Edition,O’Reilly , 2018.