<u>UNIVERSITY INSTITUTE</u>
<u>OF COMPUTING</u>

MINI PROJECT
ON
Inventory Management System

Program Name: BCA
Subject Name/Code: ADVANCED WEB DEVELOPMENT (23CAH-301)

Submitted by:

Name: Utsav Raj

UID: 23BCA10816

Section: 5A

Submitted to:

Name: Abhineet Thakur

Designation: Assist. Prof.

GIT: https://github.com/Utsavrajlohani/Inventory-Managemant-System.git

Objective

To document the design, implementation, and evaluation of a minimal inventory and sales application built with Laravel. This case study explains the problem addressed, the tools and concepts used, system design, key code modules, test output, and future improvements.

Problem Statement:

The Describe the problem the project solves, the target users, and the measurable goals.

o Problem domain (inventory mismatch, duplicate sales inserts, lack of audit logs)

o Desired outcomes (accurate sales aggregation, per-user ownership, readable reports)

o Constraints (time, DB choice, hosting environment, security/privacy)

Tools and Concepts Used:

List languages, frameworks, libraries, DB and tooling. Call out important concepts.

o PHP 8.x, Laravel 10

o Blade templates, Eloquent ORM

o MySQL / MariaDB (or PostgreSQL)

o Vite, Tailwind CSS, npm

o Key concepts: transactions, migrations, Eloquent relationships, request validation, authentication

System Design:

o Operating System: Windows 10 / 11

o Local Server: XAMPP / WAMP

o Browser: Chrome, Firefox (latest versions)

o Languages Used: web server (PHP/Laravel), front-end (Vite/Tailwind), background jobs (optional), storage

o Database: MySQL (phpMyAdmin)

o Text Editor: VS Code / Sublime Text

o Data model: key tables and relationships (users, products, suppliers, sales)

Code Snippet or Module Explanation

1. Authentication Module (Register & Login)
Purpose:
This module manages secure user registration and login using Laravel's built-in authentication system.

Controller Example:
```php
public function register(Request $request) {
    $request->validate([
        'name' => 'required|string|max:255',
        'email' => 'required|email|unique:users',
        'password' => 'required|confirmed|min:6'
    ]);
    User::create([
        'name' => $request->name,
        'email' => $request->email,
        'password' => bcrypt($request->password),
    ]);
    return redirect('/login');
}
```

2. Dashboard Module
Purpose:
Displays system summary including product count, supplier count, and sales count.

Controller Example:
```php
public function index() {
    return view('dashboard', [
        'productsCount' => Product::count(),
        'suppliersCount' => Supplier::count(),
        'salesCount' => Sale::count(),
    ]);
}
```

3. Product Management Module
Purpose:
Allows the admin to manage product details including name, price, quantity, and supplier.

Controller Example:

```
public function store(Request $request) {
    $request->validate([
        'name' => 'required',
        'supplier_id' => 'required',
        'unit_price' => 'required|numeric',
        'quantity_in_stock' => 'required|integer'
    ]);
    Product::create($request->all());
    return redirect()->back()->with('success', 'Product added successfully!');
}
```

4. Supplier Module

Purpose:

Manages supplier information and maintains links between suppliers and products.

Controller Example:

```
public function store(Request $request) {
    $request->validate(['name' => 'required', 'phone' => 'required']);
    Supplier::create($request->all());
    return redirect()->back()->with('success', 'Supplier added successfully!');
}
```

5. Sales Module

Purpose:

Handles all sales transactions and automatically reduces stock quantity after each sale.

Controller Example:

```
public function store(Request $request) {
    $sale = Sale::create([
        'customer_id' => $request->customer_id,
        'total_amount' => $request->total_amount,
        'sale_date' => now(),
    ]);
    foreach ($request->products as $item) {
        SaleDetail::create([
```

```
            'sale_id' => $sale->id,
            'product_id' => $item['product_id'],
            'quantity' => $item['quantity'],
            'unit_price' => $item['unit_price'],
            'subtotal' => $item['quantity'] * $item['unit_price'],
        ]);
        Product::where('id', $item['product_id'])->decrement('quantity_in_stock',
$item['quantity']);
    }
}
```

6. Sales Report Module

Purpose:

Generates daily, weekly, or monthly sales reports based on stored transaction data.

Controller Example:

```
$sales = Sale::whereBetween('sale_date', [$startDate, $endDate])->get();
$totalRevenue = $sales->sum('total_amount');
```

- **SUMMARY:**

This project offers a complete solution for event registration with:
- Clean and modern UI with responsiveness and animation
- Dark mode functionality using JavaScript and local storage
- Form validation and modal-style feedback popup
- Backend integration using PHP to insert user input into MySQL
- Secure handling of form data using prepared statements
- Easy configuration using XAMPP/WAMP for local deployment

- **CONCLUSION:**

Summarize what was achieved vs the objective. Call out any metrics (bugs fixed, performance improvements, tests added).
- Fixed duplicate-sale insertion bug by adding transaction guard and server-side duplicate checks
- Dashboard now shows aggregated SUM(quantity) instead of row counts
- Improved timestamp handling with proper casting and formatting
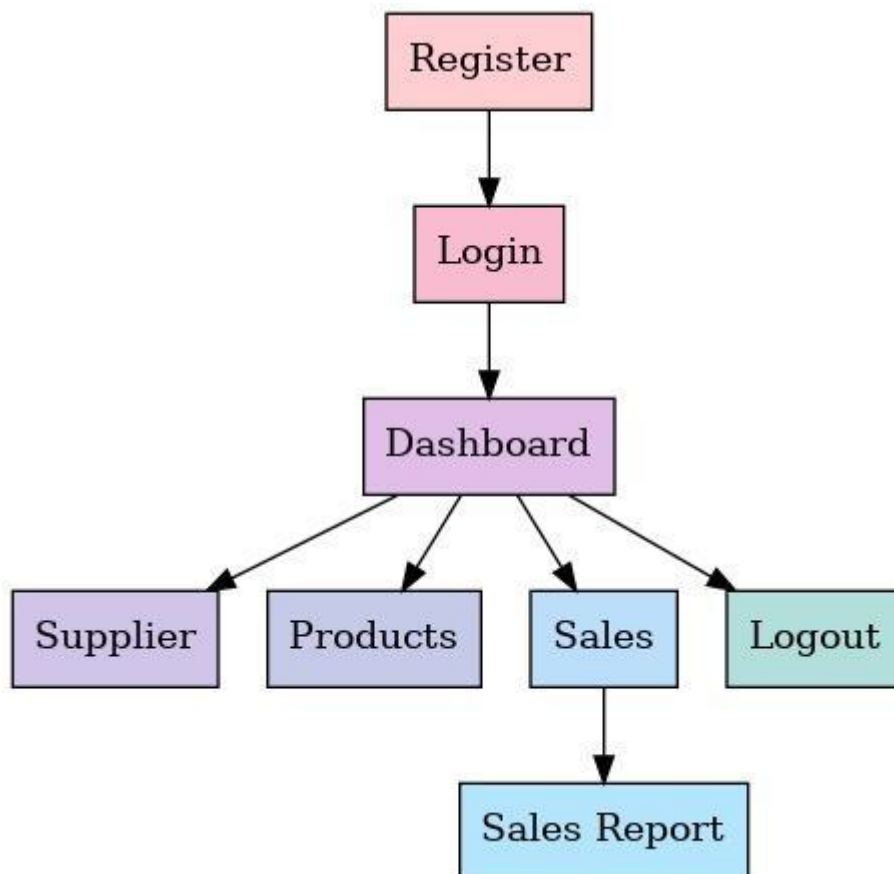
- **Limitations and Future Scope:**
  Note known limitations, assumptions and concrete next steps.
  o No audit trail for edits to sales (add activity logs)
  o Would benefit from worker queue for heavy report generation (Laravel queues)
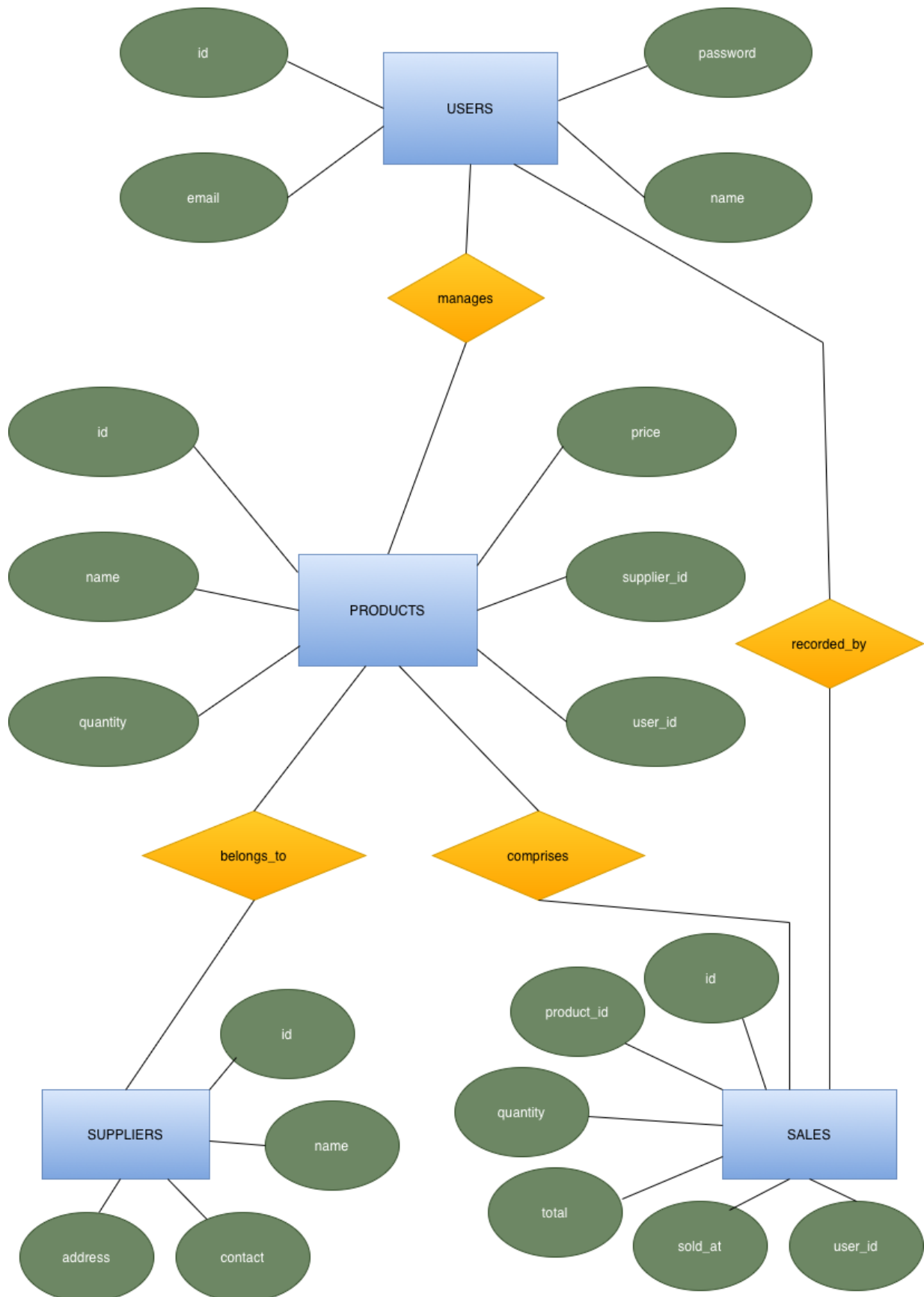 o Add stricter client-side debounce or server-side idempotency tokens to avoid double submits

- **Commands:**
  o cmd
  o composer install
  o php artisan migrate
  o php artisan serve

Flowchart or algorithm

o **DATA MODEL (ER Diagram):**

Outputs:

Suppliers

Add Supplier

ID	Name	Contact	Email	Address	Products	Actions
1	Anand	8541030170	kish868787@gmail.com	namaste	1	
2	Rajgir Traders	785545452	anandkumar780434@gmail.com	asdfg	1	

Products

Add Product

#	Name	Supplier	Qty	Price	Image	Actions
1	sprite	Rajgir Traders	35	₹20.00		
2	coco cola	Anand	15	₹40.00		

Sales Summary

Record Sale

Product	Number of Sales	Total Quantity	Total Amount	Last Sale
sprite	1	15	₹300.00	Nov 05, 2025 03:34:57 AM
coco cola	1	10	₹400.00	Nov 05, 2025 03:35:22 AM

Recent Transactions

Product	Quantity	Amount	Sold At
coco cola	10	₹400.00	Nov 05, 2025 03:35:22 AM
sprite	15	₹300.00	Nov 05, 2025 03:34:57 AM

Sales Report

Daily Weekly Monthly

November 05, 2025

Product	Number of Sales	Total Quantity	Total Amount
coco cola	1	10	₹400.00
sprite	1	15	₹300.00
Total	**2**	**25**	**₹700.00**