**An-Najah National University**

**Faculty of Engineering & Information Technology**

**Computer engineering department**


**Accounting and Management System for Businesses**


**Prepared By:**

**Muhamad Maher Himuni**

**Suliaman Kurdi**

**Supervised By:**

**Dr. Emad Natsheh**


**June 2025**

## Table of Contents

# Abstract

***Fine Master*** is a comprehensive multi-platform accounting and management system designed to simplify the operations of companies of all sizes—start-ups, small, medium, and large—through a modern, user-friendly, and efficient interface. Unlike traditional accounting software that often features complicated and outdated interfaces requiring skilled accountants to operate, *Fine Master* offers an intuitive, easy-to-use design that supports both light and dark modes, enhancing usability and user comfort. The system supports three languages (Arabic, English, and Hebrew) to accommodate diverse user groups and assigns different roles and permissions tailored to each user, including administrators, warehouse managers, sales representatives, and drivers. Key features include invoice and sales management, order creation and tracking through delivery, full inventory and warehouse management, tracking of suppliers and service providers with clear linkage between each service and its respective suppliers, detailed financial and inventory reports, and an integrated internal chat system with real-time notifications powered by Firebase. Developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js) for the backend and React Native for the mobile app, *Fine Master* ensures flexible, secure access anytime and anywhere. Additionally, the system offers three subscription plans designed to meet various business needs and budgets, making it a versatile solution for companies at different stages of growth.

# Chapter 1: Introduction

## 1.1 Background

In today's rapidly evolving business environment, companies of all sizes—whether startups, small, medium, or large enterprises—face increasing challenges in managing their financial, inventory, and operational processes efficiently. Traditional accounting and management software often come with complex interfaces and require specialized accounting knowledge, making them less accessible to users without formal accounting training. Furthermore, many existing systems focus primarily on data entry, leaving the user limited to input tasks without offering a comprehensive workflow or role-based management.

With the widespread adoption of digital solutions and mobile technology, there is a growing demand for flexible, user-friendly, and integrated management systems that can be accessed anytime and anywhere. This is especially important for businesses operating in dynamic environments where various roles, such as administrators, warehouse managers, sales representatives, and delivery drivers, need to collaborate smoothly.

***Fine Master*** aims to address these challenges by providing a modern, multi-platform accounting and management system that combines simplicity with powerful features. It supports multiple languages (Arabic, English, and Hebrew) and offers role-specific access and functionalities. Moreover, *Fine Master* introduces an intuitive interface that supports both light and dark modes, making it easier for users to navigate and operate the system without requiring deep accounting expertise.

This system also integrates essential business functions such as invoicing, order tracking, inventory management, and supplier/service provider management, all within a single platform that enhances operational efficiency and business transparency.

## 1.2. Objectives

The main objectives of the *Fine Master* project are to develop a comprehensive and easy-to-use accounting and management system that meets the needs of companies of various sizes, from Start-ups to large companies. Specific objectives include:

- Design and implement a system of various platforms that integrate accounting management, inventory management, order tracking, supplier management and service providers, as well as tracking invoices, orders, payments and checks for each client in a single unified application.
- Provide a friendly interface that eliminates the need for advanced accounting knowledge, avoiding complex accounting terminology and presenting information simply and intuitively.
- To support multiple user functions, including managers, warehouse managers, sales representatives and drivers, each with custom access and features to optimize their specific tasks.
- Activate perfect communication and real -time notifications within the system using firebase, improving coordination between different users.
- Offer complete multilingual support (Arabic, English and Hebrew) to serve various user groups.
- To implement clear and dark modes, for better user experience and accessibility.
- Provide flexible subscription plans suitable for different business sizes and business needs.

## 1.3 Scope of Work

The "Fine Master" project encompasses the design, development, and deployment of a comprehensive accounting and management system tailored for companies of various sizes. The scope of work includes:

- Developing a web-based platform and a mobile application (using React Native) to provide seamless access across devices.
- Implementing core business modules including invoicing, sales management, order tracking, inventory and warehouse management.
- Integrating supplier and service provider management with detailed tracking and association of suppliers to specific services.
- Enabling detailed tracking of invoices, orders, payments, and checks for each customer to ensure transparent financial management.
- Supporting multiple user roles such as administrators, warehouse managers, sales representatives, and drivers, each with distinct permissions and dashboards.
- Designing a multilingual interface supporting Arabic, English, and Hebrew languages to accommodate a diverse user base.
- Incorporating an intuitive user interface that supports both light and dark modes to enhance usability and reduce eye strain.
- Including an internal chat system and real-time notifications powered by Firebase to improve communication and workflow coordination.
- Providing subscription-based service plans tailored to different business sizes and needs.

**This project does not cover integration with external ERP systems or advanced accounting standards compliance beyond the simplified approach intended to make accounting accessible for non-specialists.**

## 1.4 Importance of the Project

The project Fine Master is of significant importance because of the potential to transform how companies manage accounting and administrative processes. Many existing accounting systems are complex and require specialized knowledge, and limit accessibility and ease of use for many companies, especially start-ups and small companies that may lack dedicated accounting personnel. By offering a user-friendly, integrated system that simplifies accounting terminology and workflows, Fine Master provides a wider range of users-inclined administrators, warehouses, sales representatives, and drivers-to actively participate in the business management process. This democratization of accounting tasks improves operational efficiency and reduces the dependence on specialized accountants for routine activities. Furthermore, the system helps to minimize accounting errors that are often caused by personnel for data input by clearly assigning responsibility to each user for the data they provide. This accountability improves data accuracy and reliability throughout the system. Furthermore, the system's support for several languages and role-based access to different business environments addresses and promotes inclusion and flexibility. The availability of real-time communication and alerts improves collaboration and response across departments. In addition, by providing scalable subscription plans, "fine masts" ensures that companies in different growth stages can use an appropriate solution without an excessive financial burden. This flexibility supports business development and sustainability. Overall, Fine Master helps to improve business transparency, decisions, and productivity by integrating important economic and operational FU.

## 1.5 Report Organization

The report is structured as follows:

- **Chapter 2** presents the challenges faced, standards followed, and academic background.
- **Chapter 3** reviews existing systems and literature relevant to financial software.
- **Chapter 4** details the system design, tools used, implementation, and features.
- **Chapter 5** discusses the results, testing, user feedback, and comparison with alternatives.
- **Chapter 6** concludes the report and outlines future improvement recommendations.
- **Chapter 7** lists all references and sources used throughout the project.

# Chapter 2: Constraints, Standards, and Earlier Work

## 2.1 Constraints

Developing FinMaster presented several technical and operational challenges, especially in the context of real-time financial operations:

- **Security and Data Privacy:** One of the main challenges was ensuring data confidentiality and securing sensitive financial and user information. This required encryption mechanisms, secure authentication methods (like JWT), and strict access control based on user roles.
- **Cross-Platform Synchronization:** Ensuring seamless synchronization between the web and mobile platforms presented a complexity in maintaining consistent states, especially during simultaneous transactions.
- **Performance under Load:** The platform needed to handle high volumes of transactions and reports, especially for large enterprises. MongoDB was used for its scalability, but optimization and indexing were essential to maintain low latency.
- **Time and Resource Constraints:** As with most academic projects, time limitations and access to premium cloud services posed restrictions. The development team relied heavily on open-source tools and community-driven resources.

- **User Skill Levels:** The system is designed for users with minimal accounting knowledge, which limits the implementation of advanced accounting features that require professional expertise.

## 2.2 Standards

To ensure the platform meets professional expectations, development adhered to recognized standards:

- **RESTful API Architecture:** Followed for consistency, separation of concerns, and maintainability across client-server communication.
- **Authentication & Authorization:** Implemented via JWT standards, including token-based login sessions and role-based access control.
- **Database Design Principles:** NoSQL modeling strategies using MongoDB schemas supported scalable and flexible storage of financial data.
- **UI/UX Best Practices:** Both mobile and web platforms followed material design guidelines and responsive design principles.
- **Coding Standards:** Modular, reusable, and well-documented code maintained across all repositories using version control tools (Git, GitHub).

## 2.3 Related Projects and Earlier Coursework

The development of FinMaster was supported by both academic learning and analysis of existing tools:

- **Previous Coursework Influence:**
    - *Database Systems* provided the foundation for designing efficient MongoDB collections.
    - *Software Engineering* taught the principles of modularity, maintainability, and testing.
    - Database Systems: Provided knowledge on relational and non-relational databases, crucial for designing the MongoDB-based backend.
    - *Cybersecurity* informed the implementation of secure login and role management.
    - Operating Systems and Networking: Offered understanding of system resources management and communication protocols, supporting integration of real-time features with Firebase.
    - Advance Web : Included training on React.js and Node.js, enabling the development of the web interface and server-side components
- **Inspiration from Market Tools:**

  The *Fine Master* project draws inspiration from both international and local accounting software solutions. Internationally, programs such as Zoho Books, FreshBooks, Xero, Wave, and Odoo Accounting have proven successful with their user-friendly interfaces and advanced features tailored for small and medium-sized businesses. These tools highlight the importance of simplicity, flexibility, and accessibility for users without specialized accounting knowledge. Locally, Palestinian market software like Al-Arabi Accounting, Bisan, Al-Maher, Al-Quds, and Al-Shamel are well-established solutions offering various accounting and management capabilities to meet diverse business needs. This range of local tools has helped us better understand the unique requirements of the Palestinian market and guided the design of "Fine Master" to ensure it aligns closely with these needs, focusing on ease of use, multilingual support, and delivering an outstanding user experience.

# Chapter 3: Literature Review

Developing a modern, accounting system like **FinMaster** required an extensive review of current literature, existing technologies, and best practices in both accounting systems this chapter outlines the key findings and influences that shaped the system's architecture and feature set.

## 3.1 Overview of Existing Accounting Systems

Accounting software has evolved significantly over the past decades, moving from simple bookkeeping tools to integrated enterprise resource planning (ERP) systems. Traditional accounting programs primarily focus on data entry and financial reporting, often requiring specialized accounting knowledge. Popular international solutions include QuickBooks, SAP, and Oracle Financials, while many local markets have their own tailored software. Many platforms also lack strong integration with artificial intelligence, limiting their ability to provide strategic insights or real-time alerts for anomalies in financial data.

## 3.2 Limitations of Existing Systems

Despite their capabilities, many existing accounting systems face challenges such as complex user interfaces, high dependency on professional accountants, lack of role-based access, and limited support for real-time communication and multi-device use. These issues restrict the accessibility and efficiency for small and medium-sized businesses.

## 3.3 Role-Based Access and User Diversity

Recent research emphasizes the importance of role-based access control in accounting systems to distribute responsibilities and increase accountability among different users such as administrators, warehouse managers, sales representatives, and drivers. This approach can reduce errors and improve operational workflows.

## 3.4 Multilingual Support

Supporting multiple languages, especially in diverse regions, enhances usability and adoption. Localization and internationalization are essential to meet users' linguistic and cultural needs, increasing the software's market reach.

## 3.5 Technologies and Methodologies

Modern software development methodologies, including the MERN stack (MongoDB, Express.js, React, Node.js), and real-time communication platforms like Firebase, offer scalable and efficient solutions for building integrated web and mobile applications. React Native enables cross-platform mobile app development, improving accessibility and user experience.

# Chapter 4:Methodology

This chapter outlines the tools, technologies, system architecture, and implementation strategy used to develop **FinMaster**. It highlights how each component of the system was chosen and structured to support scalability, real-time responsiveness, and intelligent functionality.

### *4.1 Technical Stack*

The software system architecture is composed of several layers, each utilizing specific technologies to fulfill distinct roles:

- **Frontend (Web):**

  Built with Next.js, a React-based framework, this layer delivers a dynamic user interface. It leverages Server-Side Rendering (SSR) to enhance performance and improve search engine optimization (SEO).

- **Frontend (Mobile):**

  Developed using React Native, this layer enables the creation of cross-platform mobile applications for both iOS and Android from a single codebase, providing performance close to that of native apps.

- **Backend:**

  Powered by Express.js running on Node.js, the backend handles core business logic and serves RESTful APIs for seamless data exchange between the frontend and the database.

- **Database:**

  Utilizes MongoDB, a NoSQL database known for its flexible, schema-less structure—ideal for managing diverse and evolving data.

- **Realtime Services:**

  Integrates Firebase to support features such as user authentication, push notifications, and live data updates, enhancing user engagement and interactivity.

- **Security:**

  Implements robust measures using JWT (JSON Web Tokens) for secure, role-based authentication, along with HTTPS and input validation layers to ensure safe data transmission and protect against unauthorized access.

Together, these layers create a modern, secure, and scalable architecture designed to deliver a responsive and intelligent application experience across platforms.

## 4.2 Tools, Programming Languages, and Technologies

### 4.2.1 Mobile Application

- **Framework:** React Native
- **Languages:** JavaScript (ES6+), JSX

### 4.2.2 Website

- **Framework:** Next.js
- **Languages:** JavaScript, TypeScript (optional), CSS Modules

### 4.2.3 Backend

- **Frameworks:** Express.js, Node.js
- **Languages:** JavaScript (ES6+), JSON for data exchange
- **Authentication:** JWT
- **API Structure:** RESTful routes using controller-service architecture
- **AI Layer:** Integrated into middleware to analyze each transaction.

### 4.2.4 Real-time & Cloud Tools

- **Firebase Authentication:** Secure login, multi-device sync
- **Firebase Realtime Database:** Used for messaging and status updates
- **MongoDB Compass:** For managing schema and database structure
- **Postman:** API testing
- **VS Code + GitHub:** Source control and development

### 4.2.5 Localization

i18next and react-i18next libraries for multilingual support

## 4.3 Architecture and Implementation

### 4.3.1 System Architecture Overview

Finmaster follows a **modular microservices-inspired architecture**:

- **Frontend (Web + Mobile):** Communicates via RESTful APIs
- **Backend API Server:** Handles all core business logic
- **Database Layer:** Centralized MongoDB with Mongoose models
- **Authentication Service:** Firebase Auth + JWT token management

### 4.3.2 Integration Flow

- Users access the platform via web or mobile
- Requests are authenticated and routed through Express.js
- Results are saved in MongoDB and reflected live via Firebase

## 4.4 Feature Implementation

This section describes the implementation details of the key features integrated into the "Fine Master" system. The development focused on creating a modular, scalable, and user-friendly application that serves multiple roles with distinct functionalities.

**1. User Role Management**
The system defines several user roles including Admin, Warehouse Manager, Sales Representative, and Driver. Each role has specific permissions and access levels implemented through role-based access control (RBAC) to ensure security and data integrity.

**2. Multilingual Support**
to accommodate users in different linguistic environments, the application supports Arabic, English, and Hebrew. Language selection is available throughout the interface, and text content is dynamically loaded using localization libraries integrated in both the React web app and React Native mobile app.

**3. Super Admin Role**
A special user role called **Super Admin** is implemented to assist the application owner in managing subscription plans, monitoring partnered companies, and tracking outstanding payments from these companies. The Super Admin dashboard provides an overview of all active subscriptions, payment statuses, and company profiles

**4. Order Tracking**
Sales representatives and drivers can create and follow up on customer orders. The system tracks the status of each order from initiation through delivery, providing notifications via Firebase Cloud Messaging to keep all parties informed.

**5. Invoice and Payment Management**
the application generates invoices linked to orders and tracks payments, including cash, checks, and electronic payments. Each customer's payment history is maintained, allowing comprehensive tracking of outstanding amounts and financial records.

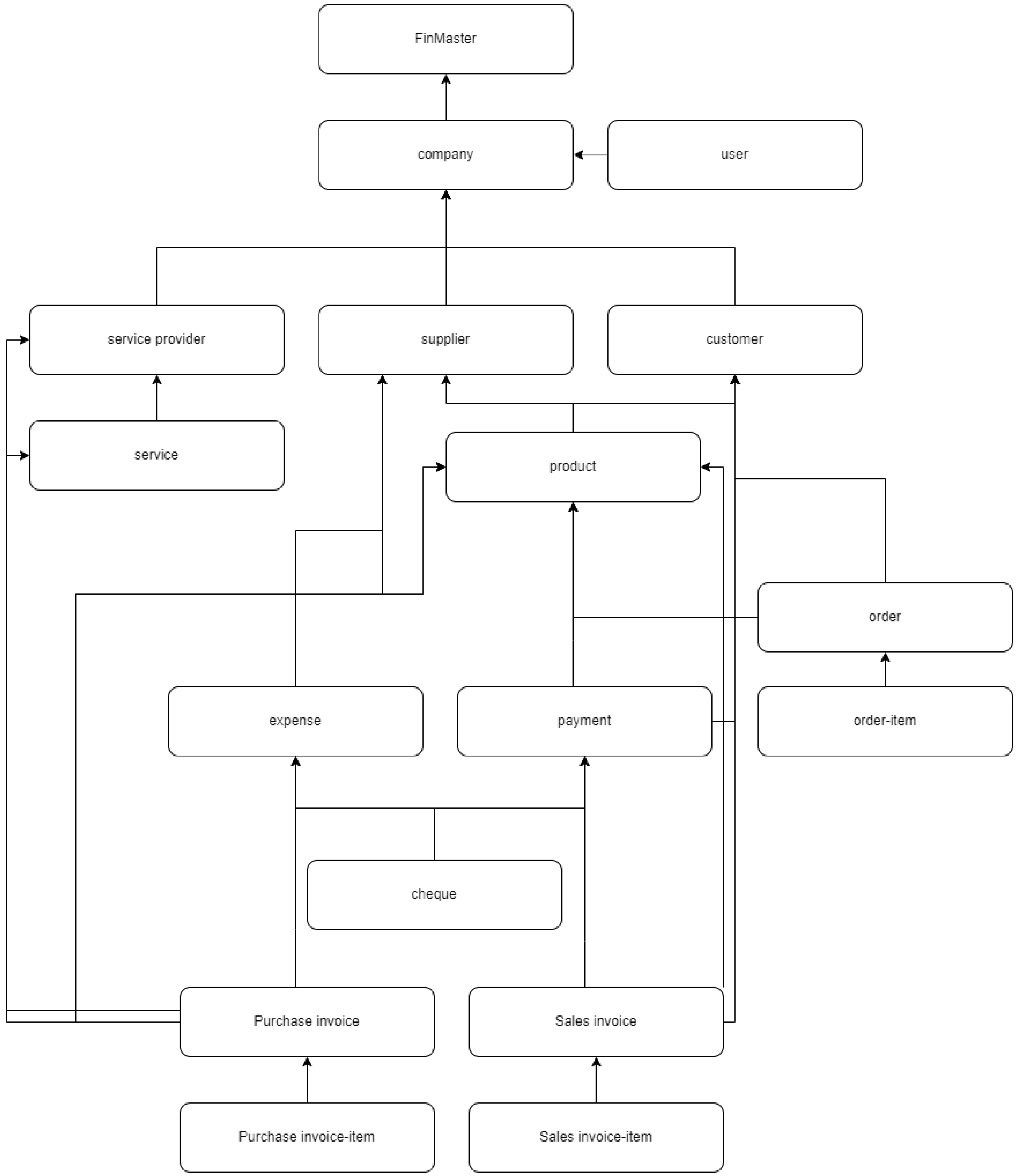
**6. Supplier and Service Provider tracking**
a dedicated module manages suppliers and service providers, linking them to specific services or products. This feature helps streamline procurement and vendor management within the system.

**7. Reporting System**
Comprehensive reports are generated to provide insights into sales, inventory status, financial transactions, and user activity. Reports are customizable and exportable for further analysis.

**8. User Interface and Experience**
The UI is designed to be intuitive, with support for both light and dark modes to enhance usability. The interface is responsive and optimized for desktop and mobile devices to ensure a consistent experience across platforms.

## 4.5 Database Design

The database of the "Fine Master" system is designed using a document-based structure (MongoDB), allowing flexibility and scalability to handle various business operations. Below is a description of the main collections (tables) used in the system:



This report details the database schema design, which is built using Mongoose for MongoDB. The design includes various interconnected collections to manage business operations such as customer and supplier interactions, invoicing, payments, expenses, product management, and company-specific settings.

### 4.5.1. Schema Overview

The database is composed of several key collections, each serving a specific business function:

- **Company**: Stores information about each registered company, including subscription details and global settings.
- **User**: Manages user accounts, roles, and their association with a company.
- **Customer**: Holds customer-specific data, including their outstanding balances and associated invoices and payments.
- **Supplier**: Stores supplier information, their type (goods or service provider), and linked invoices and expenses.
- **Product**: Contains details about products, including pricing, quantity, and associated suppliers.
- **Invoice**: Represents sales or purchase invoices, tracking items, amounts, payments, and expenses.
- **Payment**: Records payments received from customers.
- **Expense**: Tracks expenses incurred, particularly for purchases or services from suppliers.
- **Cheque**: Manages cheque details, linking them to either payments or expenses.
- **Order**: Handles sales and purchase orders, including items, delivery information, and status.
- **Service**: Defines services offered or utilized, tracking their expenses and associated service providers.
- **Subscription Request**: Stores requests from potential companies to subscribe to the system.

## 4.5.2. **Detailed Schema Breakdown**

### 2.1 Company

- **Purpose**: Central entity representing each business using the system.
- **Key Fields**:
  - name: Company's name.
  - logo: URL or path to the company logo.
  - address, location: Geographical details.
  - registrationDate, subscriptionEndDate, subscriptionType, subscriptionStatus: Crucial for subscription management.
  - contactEmail, contactPhone, website, taxNumber: Contact and legal information.
  - authorizedUsers: References to User documents, indicating users allowed to access this company's data.
  - settings: Embedded document for customizable company-specific settings (theme, currency, language, invoice prefix, fiscal year start).
  - createdBy: User who created the company.
- **Pre-save Hook**: Automatically updates subscriptionStatus to 'expired' if subscriptionEndDate is in the past.
- **Indexes**: Text indexes on name and contactEmail for efficient searching.

### 2.2 User

- **Purpose**: Manages user authentication and authorization within a specific company.
- **Key Fields**:
  - firstName, lastName, username, password, phone: User credentials and contact.
  - role: Defines user permissions (e.g., admin, salesman).
  - companyId: Links the user to a specific Company.
- **Relationships**: Many-to-one with Company.

### 2.3 Customer

- **Purpose**: Stores information about customers.
- **Key Fields**:
  - customerName, companyName: Customer identification.
  - customerType: Type of customer.
  - phone: Contact number.
  - balanceDue: Tracks outstanding balance.
  - invoiceList: References to Invoice documents (sales invoices).
  - paymentList: References to Payment documents.
  - geographicalLocation, location: Location details.
  - companyId: Links the customer to a specific Company.
  - salesmanId: User responsible for this customer.
- **Relationships**: Many-to-one with Company, many-to-one with User (salesman), one-to-many with Invoice, one-to-many with Payment.

## 2.4 Supplier

- **Purpose**: Stores information about suppliers.
- **Key Fields**:
    - supplierName, companyName: Supplier identification.
    - supplierType: Enum ('goods_supplier', 'service_provider').
    - phone: Contact number.
    - balanceDue: Tracks outstanding balance.
    - invoiceList: References to Invoice documents (purchase invoices).
    - expenseList: References to Expense documents.
    - services: References to Service documents if the supplier is a service provider.
    - geographicalLocation, location: Location details.
    - companyId: Links the supplier to a specific Company.
    - userId: User associated with this supplier.
- **Relationships**: Many-to-one with Company, many-to-one with User, one-to-many with Invoice, one-to-many with Expense, many-to-many with Service.

## 2.5 Product

- **Purpose**: Manages product inventory and details.
- **Key Fields**:
    - productName, description, category: Product details.
    - price, costPrice: Selling and cost prices.
    - quantity, unit, minQuantity: Inventory levels and units.
    - barcode: Unique identifier for the product.
    - supplierIds: References to Supplier documents that supply this product.
    - images: URLs or paths to product images.
    - userId: User who created the product.
    - companyId: Links the product to a specific Company.
- **Relationships**: Many-to-one with Company, many-to-one with User, many-to-many with Supplier.

## 2.6 Invoice

- **Purpose**: Represents a financial document for sales or purchases.
- **Key Fields**:
  - invoiceId: Unique identifier for the invoice.
  - customerId, supplierId: Links to either a customer or supplier.
  - companyId: Links to the relevant company.
  - customer / supplier: Embedded objects for denormalized customer/supplier information.
  - amount: Total invoice amount.
  - status: Enum ('pending', 'paid', 'partially_paid', 'overdue', 'unpaid', 'draft').
  - date, dueDate: Dates related to the invoice.
  - items: Array of embedded objects, each representing a product within the invoice (includes productId, productName, quantity, freeQuantity, unitPrice, totalPrice).
  - issuedBy: User who issued the invoice.
  - terms: Invoice terms and conditions.
  - type: 'sales' or 'purchase'.
  - payments: Array of embedded objects tracking associated payments.
  - expenses: Array of embedded objects tracking associated expenses.
  - paidAmount, remainingAmount: Calculated fields for payment tracking.
- **Methods**:
  - calculateTotalAmount(): Calculates the total invoice amount based on items and adds a 16% tax (txt = totalAmount * 0.16).
  - calculateRemainingAmount(): Computes the remaining amount due by subtracting total payments (for sales invoices) or expenses (for purchase invoices) from the amount. It also updates the status accordingly.
- **Pre-save Hook**: Automatically calculates paidAmount, remainingAmount, and updates status and dueDate based on payment information.
- **Relationships**: Many-to-one with Customer (for sales), many-to-one with Supplier (for purchases), many-to-one with Company, many-to-one with User (issuedBy), one-to-many with Product (through items array).

## 2.7 Payment

- **Purpose**: Records payments received from customers.
- **Key Fields**:
  - paymentId: Unique identifier.
  - customerId: Links to the paying customer.
  - companyId: Links to the relevant company.
  - invoiceId: Links to the associated invoice (optional).
  - amount, method ('cash', 'check', 'card'), date: Payment details.
  - cheques: References to Cheque documents if the method is 'check'.
  - createdBy: User who recorded the payment.
  - status: Enum ('pending', 'completed', 'failed', 'cancelled').
  - notes, reference: Additional information.
- **Relationships**: Many-to-one with Customer, many-to-one with Company, many-to-one with Invoice, one-to-many with Cheque.

## 2.8 Expense

- **Purpose**: Tracks expenses made to suppliers.
- **Key Fields**:
  - expenseId: Unique identifier.
  - supplierId: Links to the supplier.
  - invoiceId: Links to the associated purchase invoice (optional).
  - companyId: Links to the relevant company.
  - amount, method ('cash', 'check'), date: Expense details.
  - cheques: References to Cheque documents if the method is 'check'.
  - createdBy: User who recorded the expense.
  - status: Enum ('pending', 'completed', 'failed', 'cancelled').
  - notes, reference: Additional information.
- **Relationships**: Many-to-one with Supplier, many-to-one with Company, many-to-one with Invoice, one-to-many with Cheque.

## 2.9 Cheque

- **Purpose**: Manages individual cheque details.
- **Key Fields**:
  - chequeId, chequeNumber, bankName, chequeDate, amount: Cheque specifics.
  - holderName, holderPhone: Optional details about the cheque holder.
  - status: Enum ('pending', 'cleared', 'bounced').
  - type: 'received' (from customer for payment) or 'issued' (to supplier for expense).
  - customerId, supplierId: Links to either a customer or supplier.
  - paymentId, expenseId: Links to either a payment or an expense.
- **Pre-validate Hook**: Enforces critical business rules:
  - A cheque must be linked to **either** a customer or a supplier, but not both, and not neither.
  - A cheque can be linked to **either** a payment or an expense, but not both.
- **Relationships**: One-to-one with Payment or Expense, one-to-one with Customer or Supplier.

## 2.10 Order

- **Purpose**: Represents sales or purchase orders, tracking items and delivery.
- **Key Fields**:
  - orderNumber: Unique identifier.
  - customerId, supplierId: Links to either a customer or supplier.
  - companyId: Links to the relevant company.
  - customer / supplier: Embedded objects for denormalized information.
  - amount: Total order amount.
  - status: Enum ('pending', 'processing', 'completed', 'cancelled', 'draft', 'ready').
  - date, deliveryDate: Order and delivery dates.
  - items: Array of embedded objects, similar to Invoice items.
  - issuedBy: User who issued the order.
  - notes: Additional notes.
  - type: 'sales' or 'purchase'.
  - deliveryStatus: Enum ('pending', 'shipped', 'delivered', 'returned').
  - deliveryAddress, deliveryNotes: Delivery details.
  - driverId: User responsible for delivery.
- **Methods**:
  - calculateTotalAmount(): Calculates the total order amount based on items.
- **Pre-save Hook**: Generates a unique orderNumber if not provided on creation.
- **Relationships**: Many-to-one with Customer (for sales), many-to-one with Supplier (for purchases), many-to-one with Company, many-to-one with User (issuedBy, driverId), one-to-many with Product (through items array).

## 2.11 Service

- **Purpose**: Defines services and tracks their associated expenses and providers.
- **Key Fields**:
    - name, description: Service identification.
    - totalExpenses: Calculated sum of all expenses for this service.
    - serviceProviders: References to Supplier documents (specifically service_provider types).
    - invoices: References to Invoice documents related to this service.
    - expenseHistory: Array of embedded objects tracking individual expenses for the service (includes amount, date, description, expenseId).
    - isActive: Indicates if the service is active.
    - companyId: Links to the relevant company.
    - createdBy: User who created the service.
- **Pre-save Hook**: Automatically updates totalExpenses whenever expenseHistory is modified.
- **Relationships**: Many-to-one with Company, many-to-one with User, many-to-many with Supplier, one-to-many with Invoice, one-to-many with Expense (through expenseHistory).

## 2.12 SubscriptionRequest

- **Purpose**: Handles new company subscription requests.
- **Key Fields**:
    - company: Embedded object for company name and avatar.
    - contactName, email, phone: Contact information for the request.
    - country, companySize, industry, plan: Details about the requesting company and their desired plan.
    - additionalInfo: Any extra notes.
    - status: Enum ('pending', 'approved', 'rejected').
    - processedBy: User who processed the request (optional).
    - processedAt: Date when the request was processed (optional).
- **Relationships**: Many-to-one with User (processedBy).

### 4.5.3Relationships and Data Flow

The database design implements a relational structure using MongoDB's document-oriented capabilities. Key relationships are established through mongoose.Types.ObjectId references, allowing for efficient querying and data integrity.

- **Core Company Linkage**: Almost all documents (User, Customer, Supplier, Product, Invoice, Payment, Expense, Order, Service) are linked to a Company via companyId. This ensures data multi-tenancy, where each company's data is isolated.
- **User Activity Tracking**: createdBy and issuedBy fields (referencing User documents) are used across several schemas to track who performed certain actions.
- **Financial Flow**:
  - Invoice documents serve as the central hub for sales and purchase transactions.
  - Payment documents record incoming funds, typically linked to Invoices and Customers.
  - Expense documents record outgoing funds, typically linked to Invoices and Suppliers.
  - Cheque documents are specifically used when payments or expenses involve cheques, enforcing rules to ensure they are correctly associated with either a Payment or an Expense.
- **Product and Order Management**:
  - Product details are embedded within Invoice and Order items, providing a snapshot of the product at the time of the transaction.
  - Orders manage the pre-invoice process, focusing on goods movement and delivery.
- **Service and Supplier Interconnection**: Suppliers can be of service_provider type and are linked to Service documents. Service documents, in turn, track their expenses and link to Expense documents.

### 4.5.4. Validation and Business Logic

Several schemas incorporate Mongoose's validation features and pre-save/pre-validate hooks to enforce business rules and maintain data consistency:

- **Cheque Schema**: Strong validation ensures a cheque is linked to a customer/supplier and a payment/expense correctly, preventing ambiguous data states.
- **Company Schema**: Automatically updates subscription status based on the end date.
- **Invoice Schema**: Contains methods to calculate total and remaining amounts, and a pre-save hook to update these values and the invoice status (paid, partially_paid). It also includes a 16% tax calculation on the total amount.
- **Service Schema**: Automatically sums totalExpenses from its expenseHistory array.

# Chapter 5: Results, Discussion, and Analysis

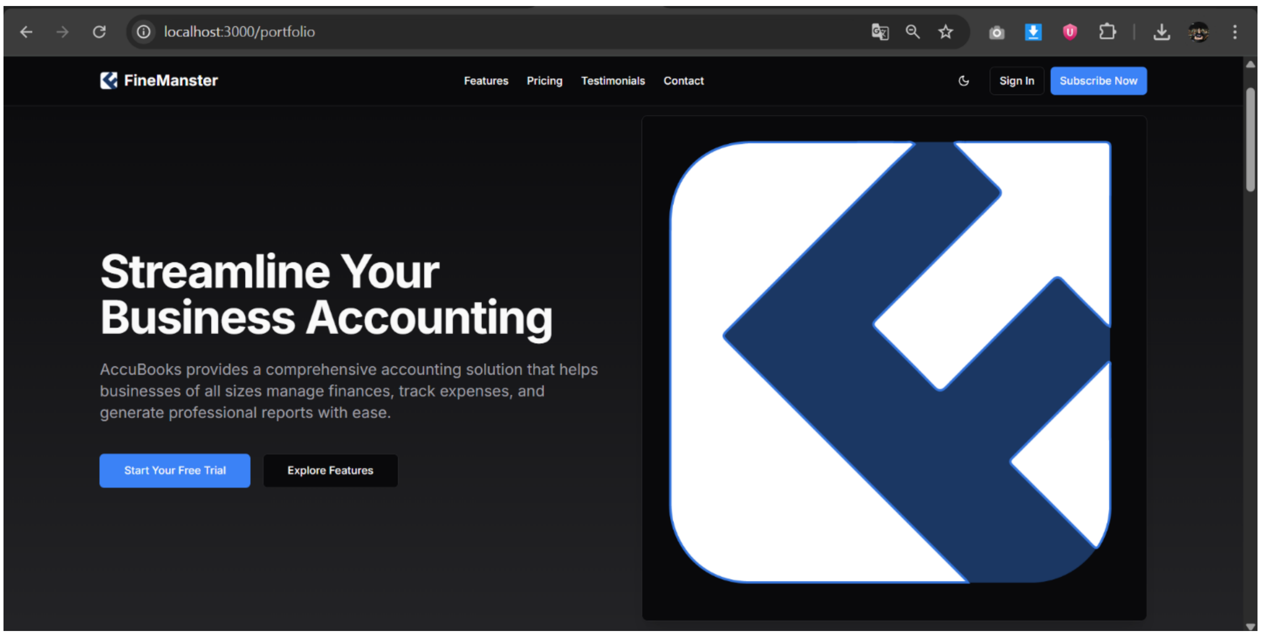
## 5.1 Implementation Result

This section presents the implementation results of the "Fine Master" system in a detailed, step-by-step manner, illustrating the full user journey from the initial interaction to complete usage of the system.

## 5.1 Landing Page & System Introduction

When the user first opens the website, they are presented with a portfolio page showcasing the features and advantages of Fine Master.

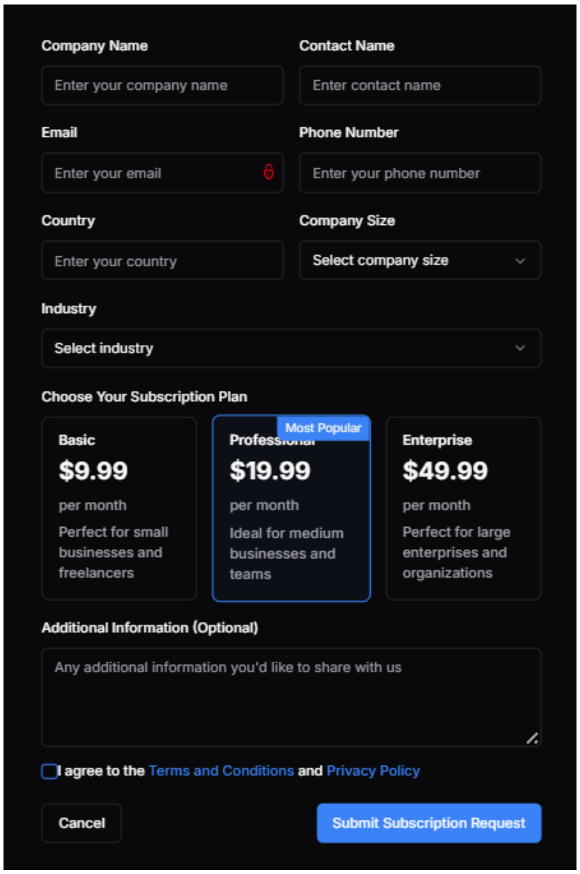
- Implemented using React with a responsive and multilingual design.
- Highlights benefits, subscription plans, screenshots, and testimonials.
- Provides options to request a subscription or start a free trial.



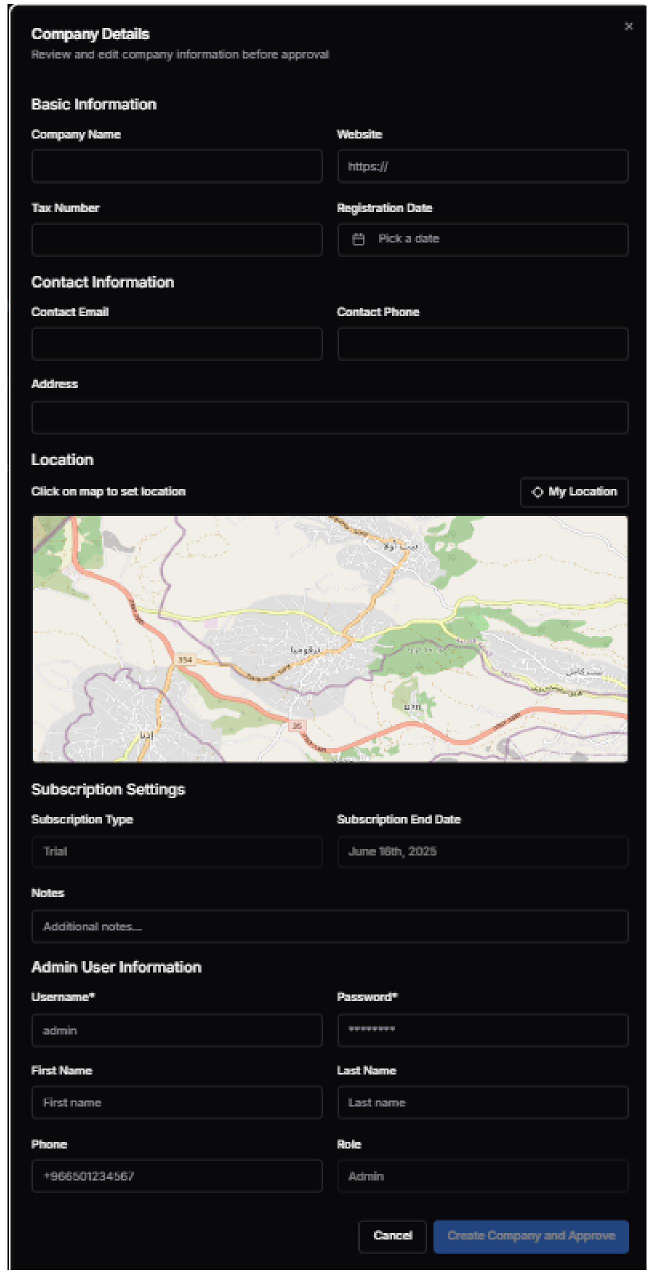

*Portfolio Page*

From the landing page, the user has two options:

- **Request a Company Subscription** – filling in company info and contact details.
- **Start a Free Trial** – limited-time access to explore the platform.
  *Form data is stored and managed via MongoDB.*
  *Request is logged under* SubscriptionRequests *collection for Super Admin review.*



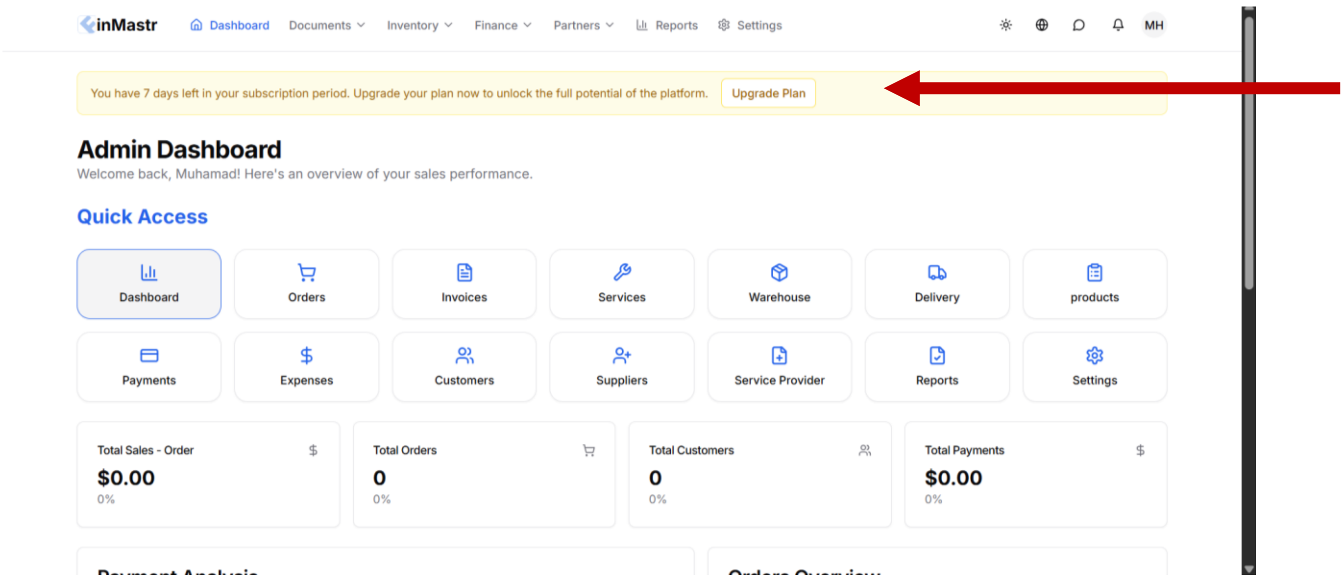

*Request a Company Subscription*




*Start a Free Tria*

If the user chooses to start the **free trial**, they are automatically redirected to the **main application dashboard**, where they receive:

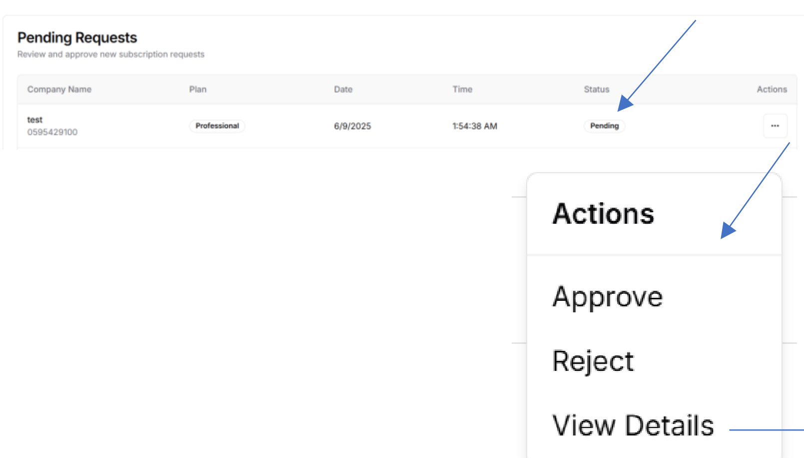
- **7 days of full access** to explore the system.
  This allows users to test all modules without entering payment information.

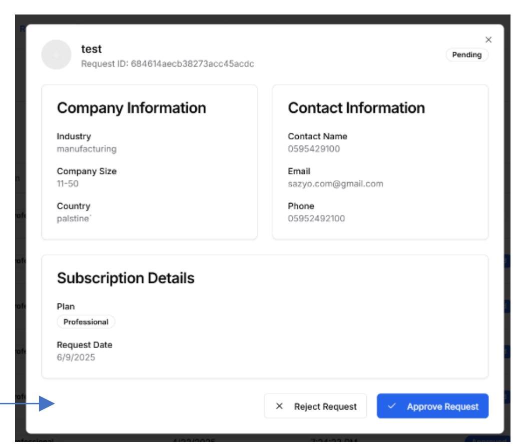



If the user instead chooses to **submit a subscription request**:

- A form is filled with company and contact information.
- The request is sent to the **Super Admin** for review.

The request is stored in the SubscriptionRequests collection.
Super Admin receives a real-time notification (via Firebase).



If the user initially registered for the **free trial**, the system keeps track of the 7-day duration using timestamp data linked to the company account.

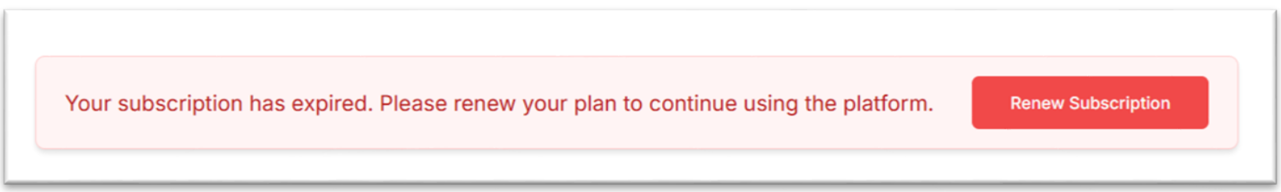
- Once the 7-day trial ends, the **system is automatically locked** for that company.
- On every login attempt by any user from that company, a clear message is displayed:

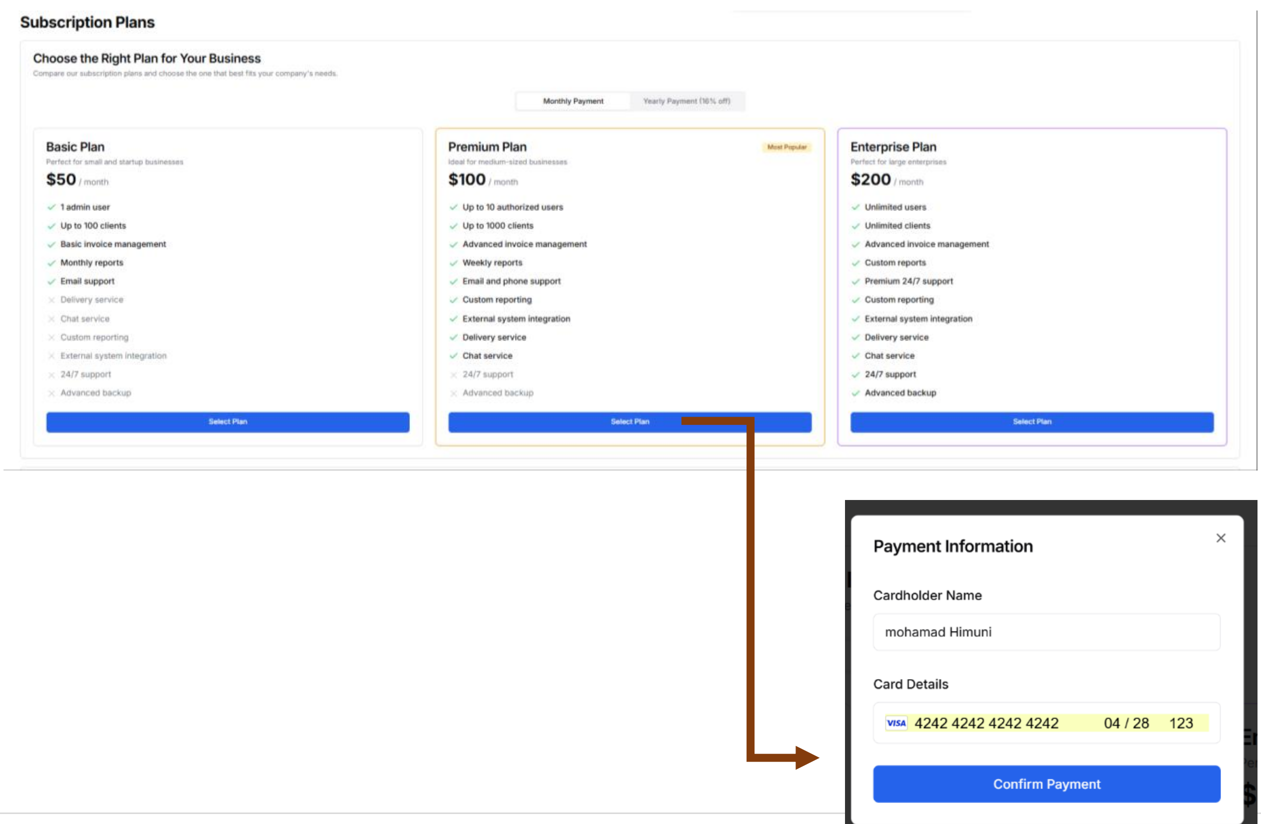
  "Your free trial has ended. Please choose a subscription plan and complete the payment to continue using Fine Master."

All user access is disabled until the payment process is completed.
The system redirects the user to the **subscription upgrade page**, where they must:

- Select one of the three available plans.
- Choose monthly or yearly billing.
- Complete the payment using a credit/debit card.

Once the payment is successful, the system reactivates access for all company users, and normal usage resumes.

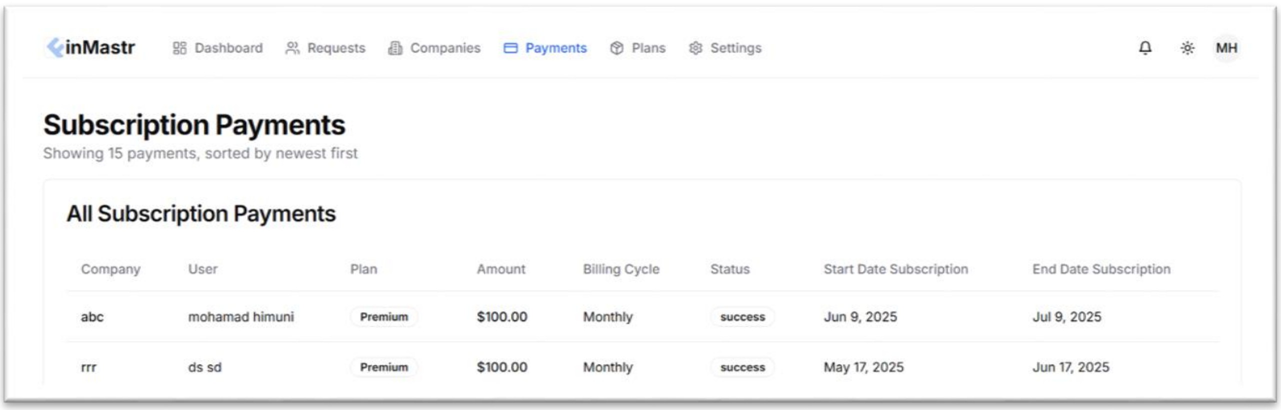


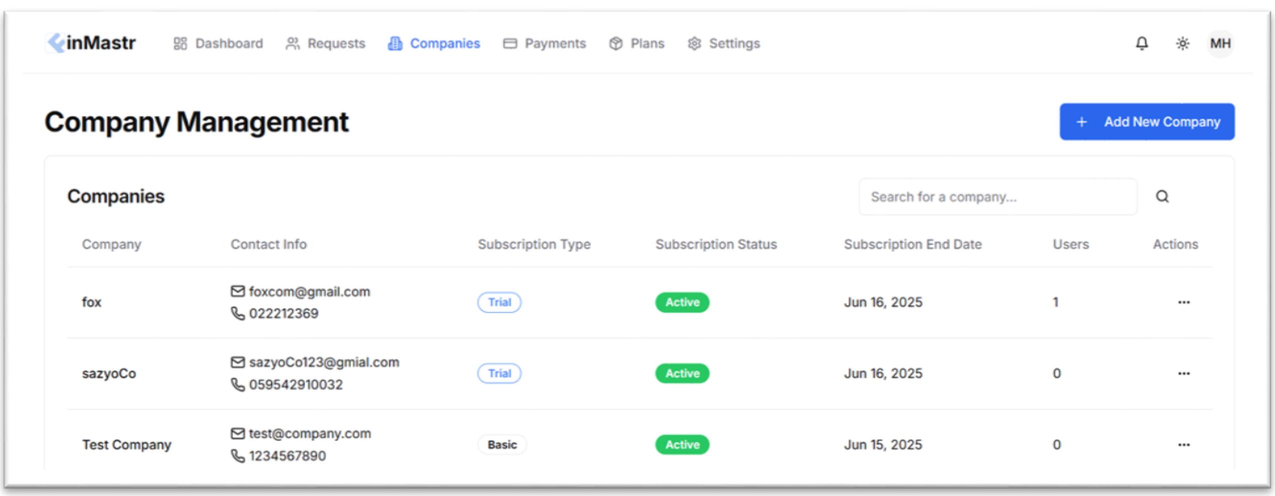



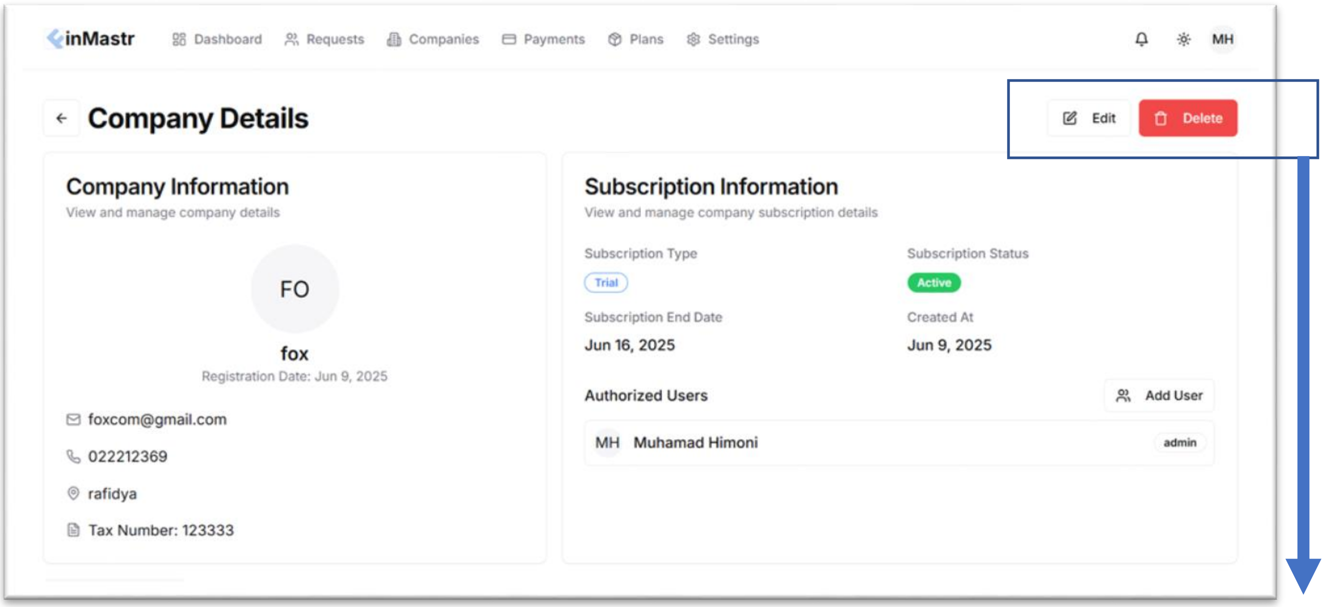
Once the user completes the payment successfully through the subscription page:

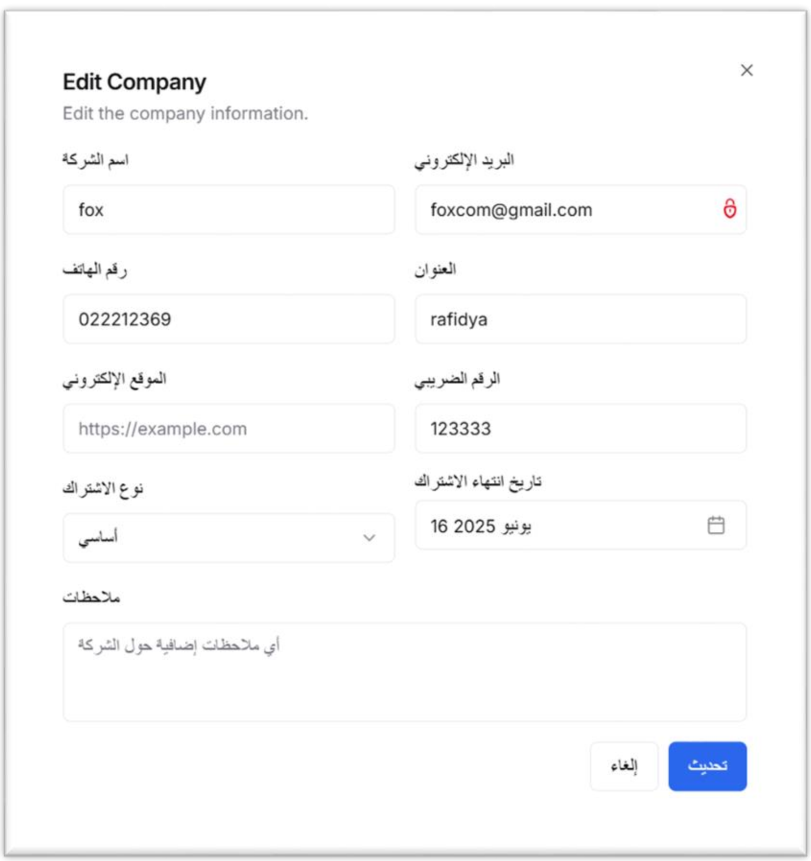
- The company's subscription is immediately extended based on the selected plan (monthly or yearly).
- All users within the company regain access to the system with no restrictions.
- The system automatically logs the payment transaction, including:

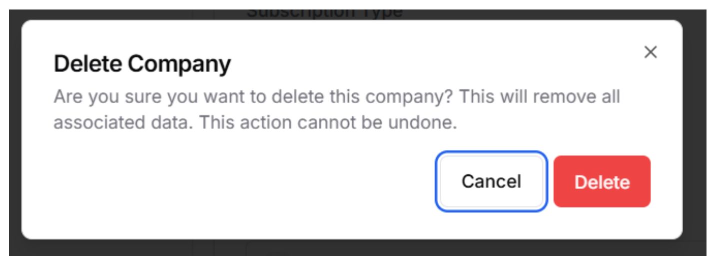
- Company name, Amount paid, Payment date and Plan type
- This payment information is stored securely in the payment table.
- A copy of the transaction is also visible to the Super Admin through the "Payments" section in their dashboard.
- This allows the Super Admin to monitor:
- All payments from all subscribed companies
- Subscription start and end dates




inMastr
Dashboard
Requests
Companies
Payments
Plans
Settings
MH
Company Details
Edit
Delete
Company Information
View and manage company details
FO
fox
Registration Date: Jun 9, 2025
foxcom@gmail.com
022212369
rafidya
Tax Number: 123333
Subscription Information
View and manage company subscription details
Subscription Type
Trial
Subscription Status
Active
Subscription End Date
Jun 16, 2025
Created At
Jun 9, 2025
Authorized Users
Add User
MH Muhamad Himoni
admin


Edit Company
Edit the company information.
اسم الشركة
fox
البريد الإلكتروني
foxcom@gmail.com
رقم الهاتف
022212369
العنوان
rafidya
الموقع الإلكتروني
https://example.com
الرقم الضريبي
123333
نوع الاشتراك
أساسي
تاريخ انتهاء الاشتراك
16 يونيو 2025
ملاحظات
أي ملاحظات إضافية حول الشركة
إلغاء
تحديث


Delete Company
Are you sure you want to delete this company? This will remove all associated data. This action cannot be undone.
Cancel
Delete

**Once the subscription is successfully activated, each registered company is granted access for four user roles, each with defined permissions and capabilities.**

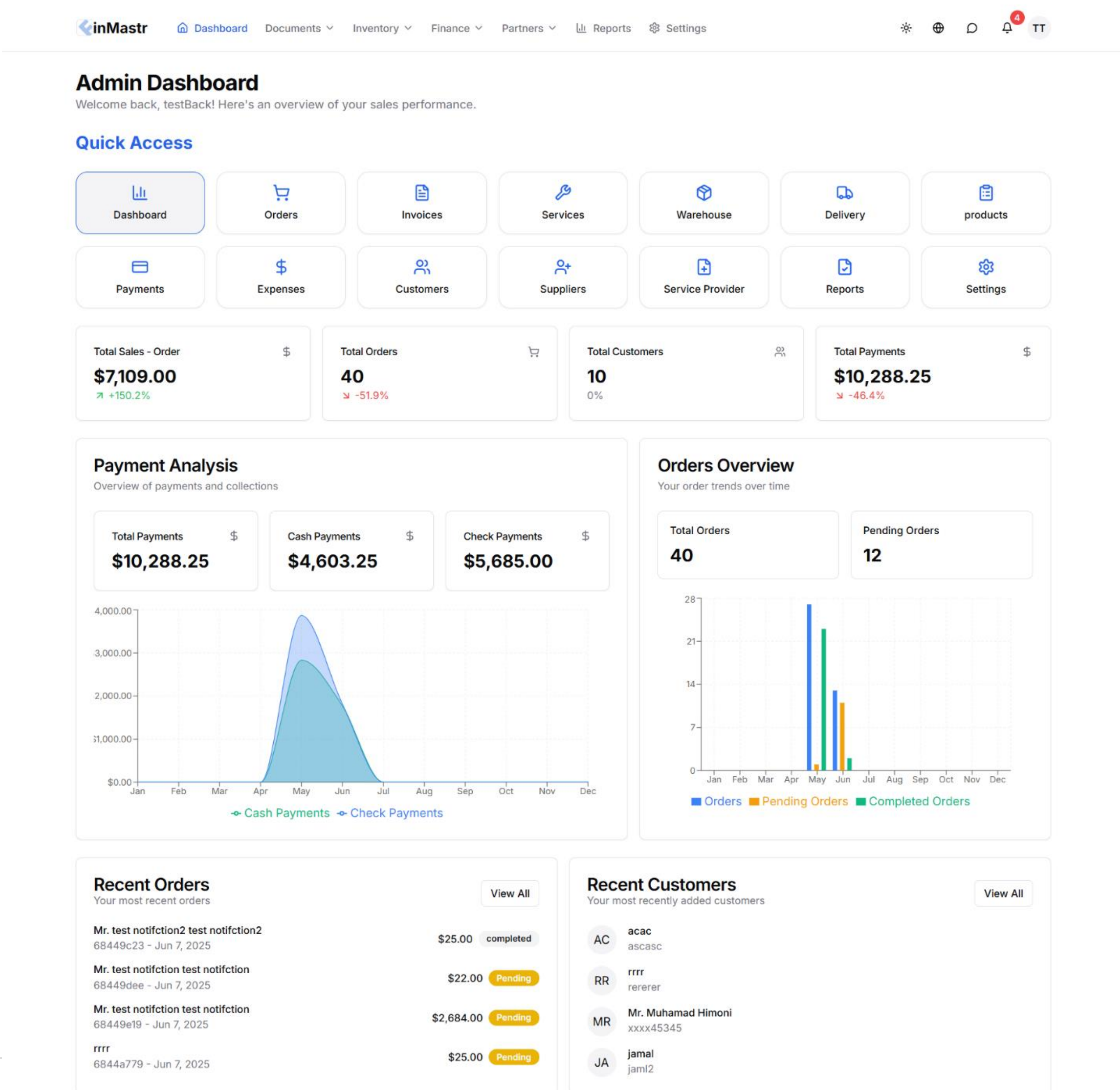
## Admin user:

The Admin has full visibility and control over all operations and user activities within the company system.

Upon logging in, the Admin is directed to a main dashboard that displays real-time statistics and a quick overview of recent actions, such as orders, payments, and deliveries.

A quick access button is available, allowing the Admin to seamlessly navigate through all company-related pages, including:

1. **Dashboard: View overall business insights and KPIs.**



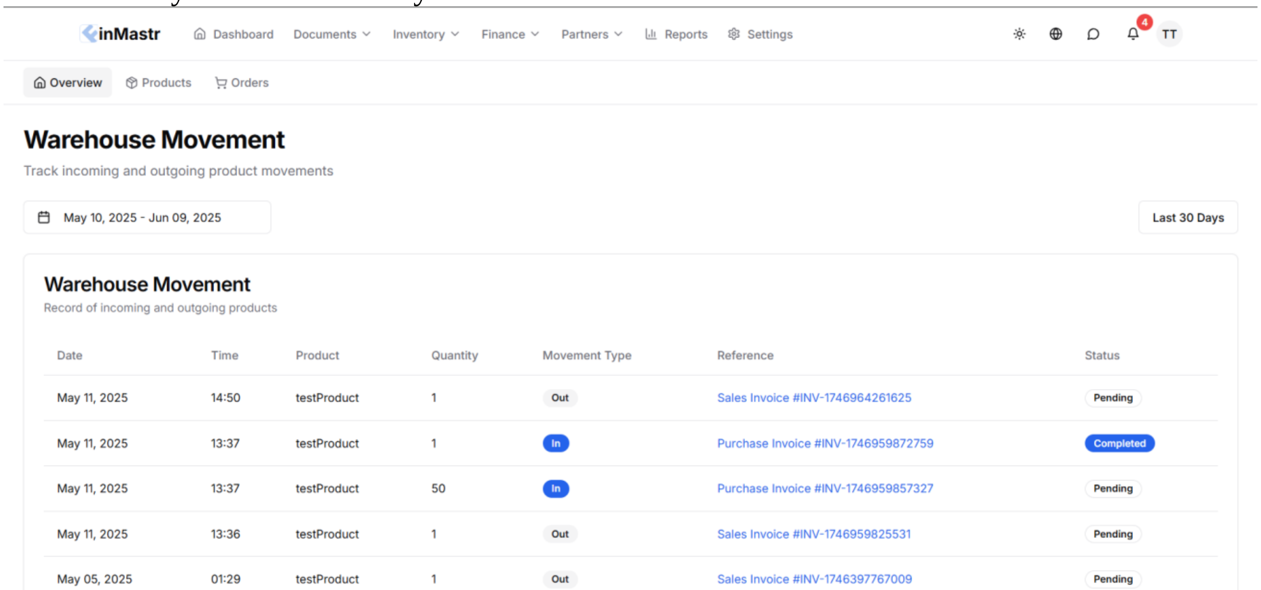

*Admin Dashnoard*

## 2. Warehouse Movement Module

**The Warehouse Movement feature is designed to offer complete control and visibility over stock operations. It is structured into three dedicated pages, each serving a specific purpose:**

### 1. Warehouse Movement Module

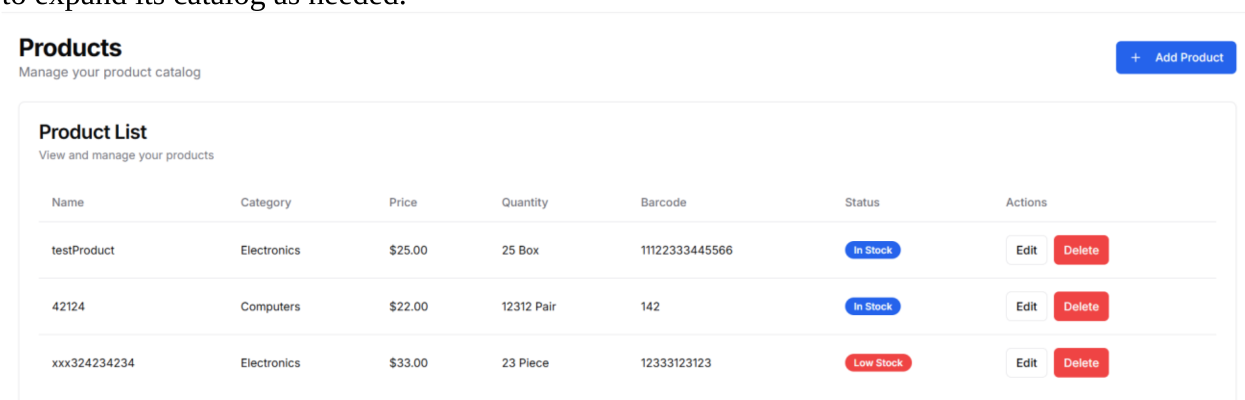
Displays a dashboard-style summary of current stock levels, recent movements
For inventory turnover and daily movements.



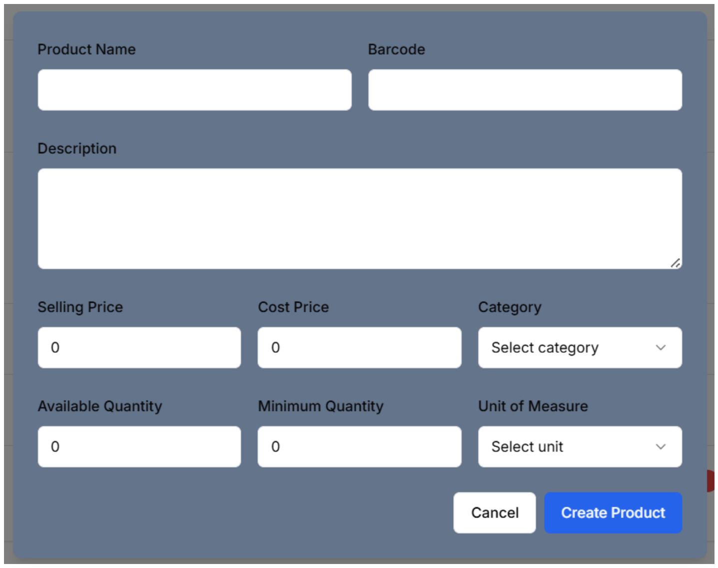

*Warehouse Movement - Overviews Page*

### 2. Product Page

Displays a complete list of all products registered within the company's inventory.
Allows the warehouse manager or admin to add new products to the system, enabling the company to expand its catalog as needed.



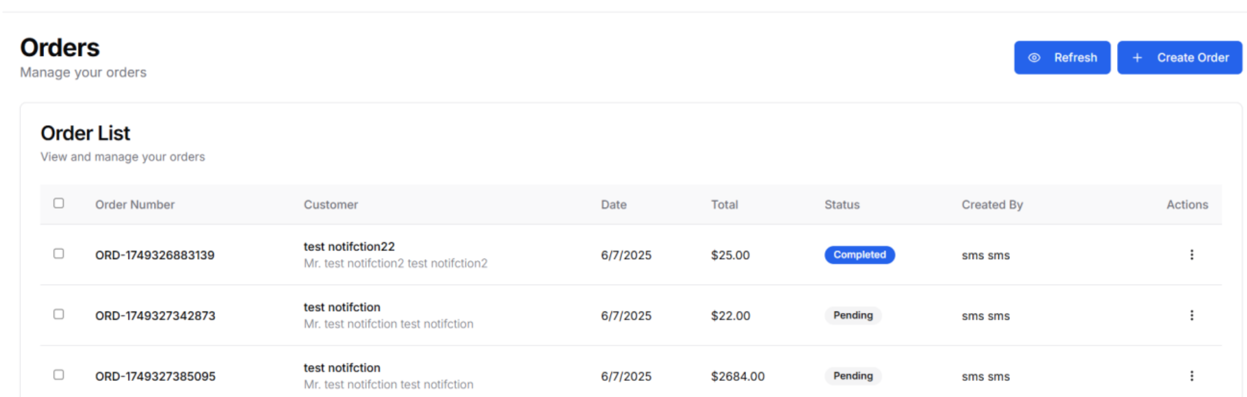

*Warehouse Movement -Product Page*



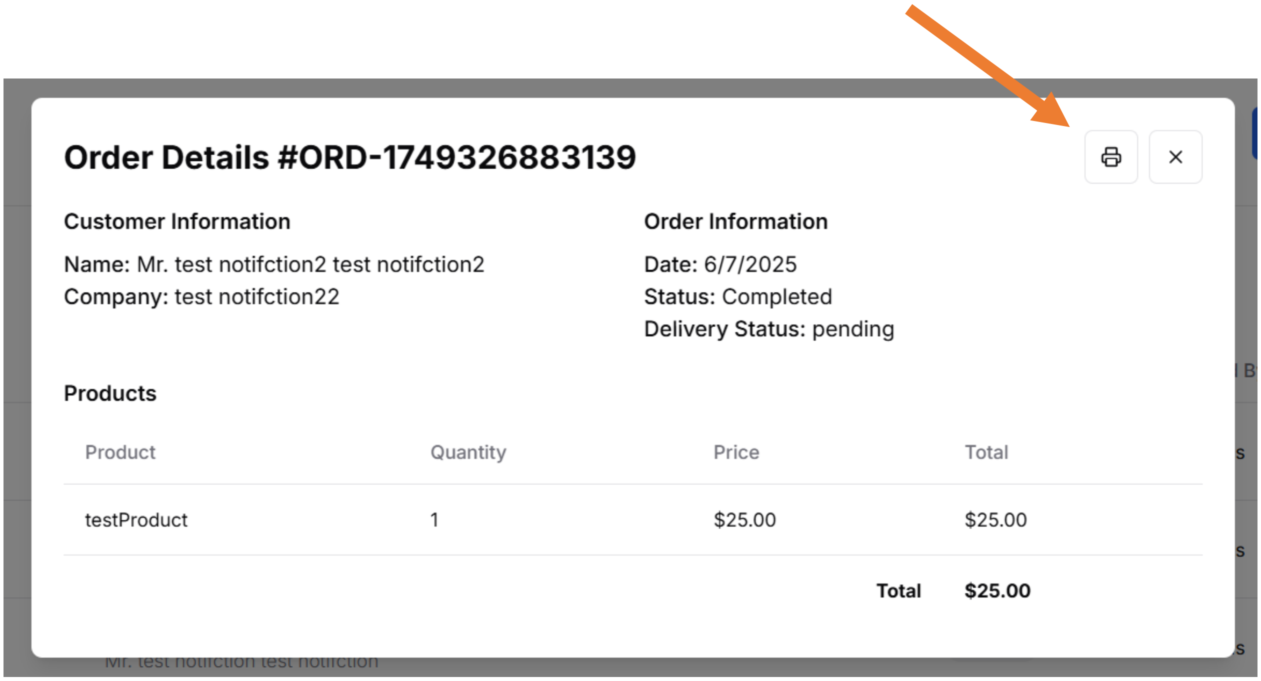

*Add New Product*

### 3. Order Page

Lists all warehouse-related orders, including both incoming shipments and outgoing deliveries. Tracks order statuses and fulfillment progress in real time.
Provides the functionality to create new customer orders, streamlining the order-processing workflow directly from the warehouse interface.




*Warehouse Movement - Order Page*

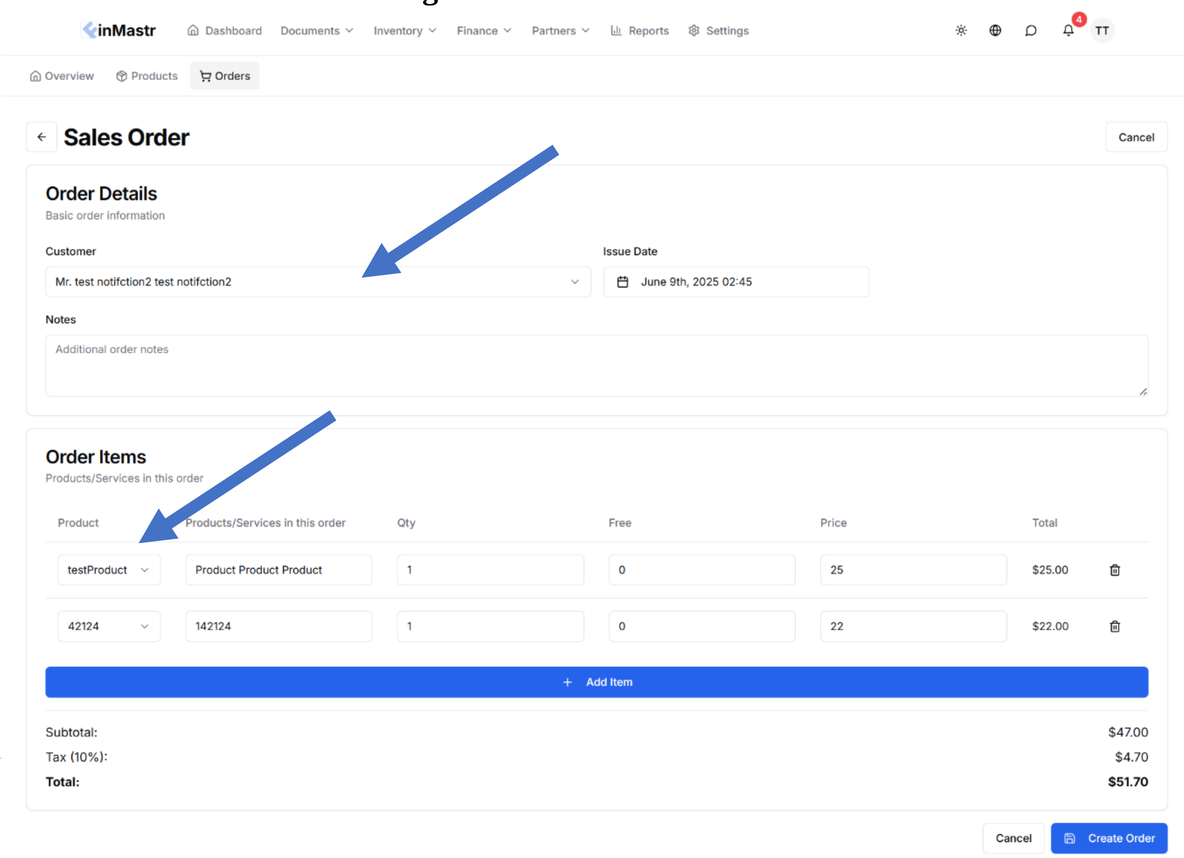



*Order Details*

Upon accessing the **order creation interface**, the user is presented with:

- A **dynamic customer list** showing all registered customers associated with the company.
- A **product list** displaying only the products that belong to the company's catalog and are available in stock.

The user can select the desired customer, choose multiple products, define quantities, and confirm the order details. Once submitted, the order is automatically **recorded in the system** and marked with a status of **"Pending"**

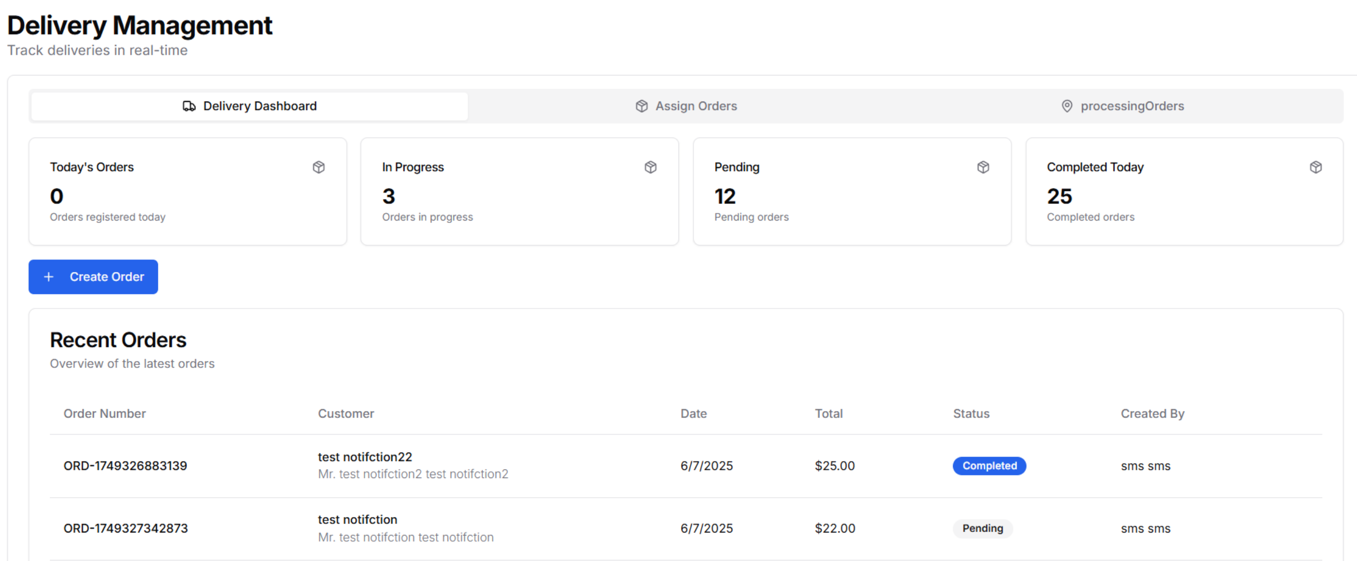


## 3. Delivery Management Module

The Delivery Management system in the project consists of several key pages that help organize and monitor delivery operations efficiently and accurately. These pages include:

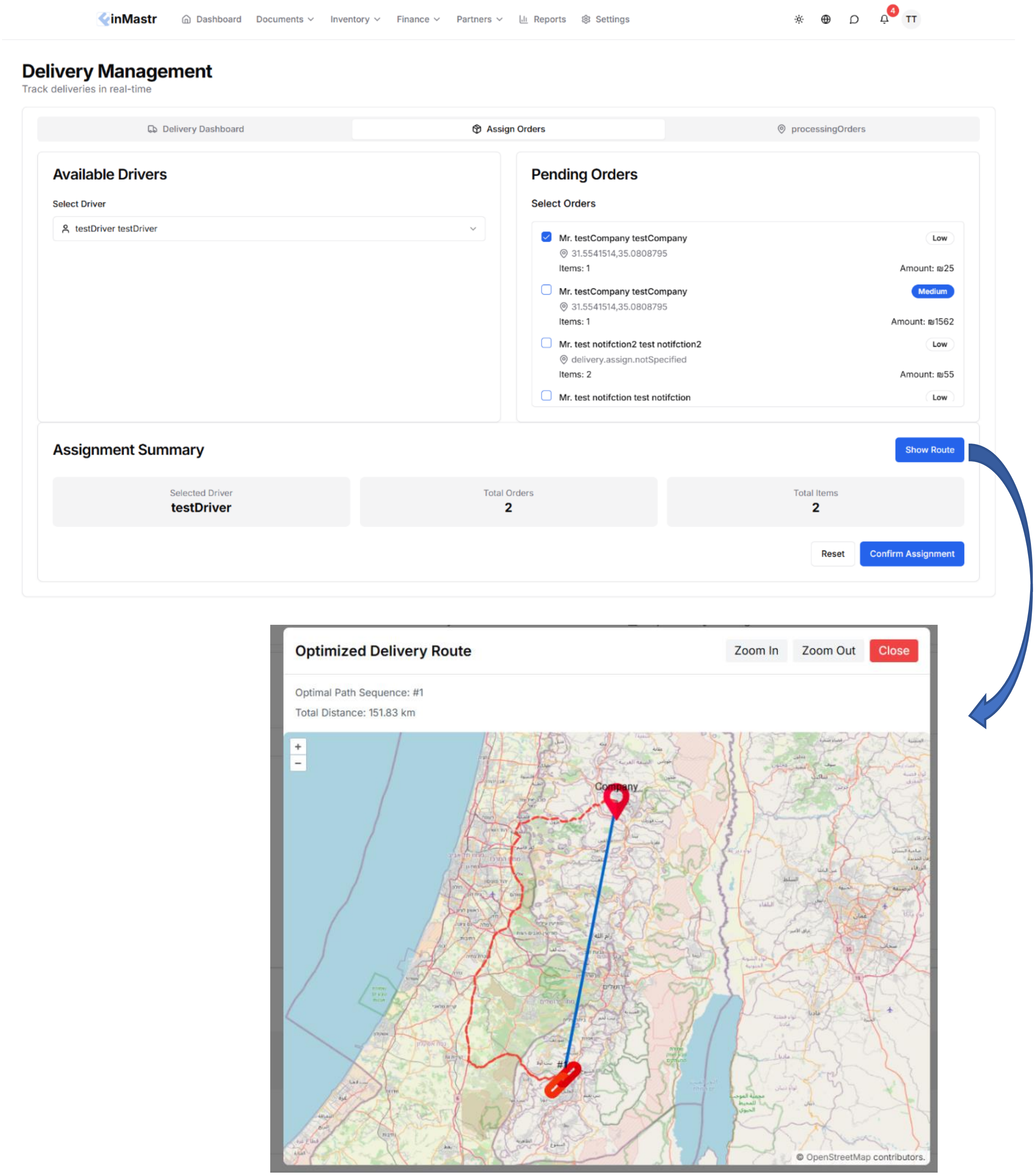
1. Delivery Dashboard

- Provides an overall summary of the current delivery operations.
- Displays detailed statistics such as the number of pending orders, orders in process, and completed deliveries.
- Shows driver performance metrics, including the number of completed deliveries and delivery times.
- Helps management track workflow and make quick decisions to improve efficiency.



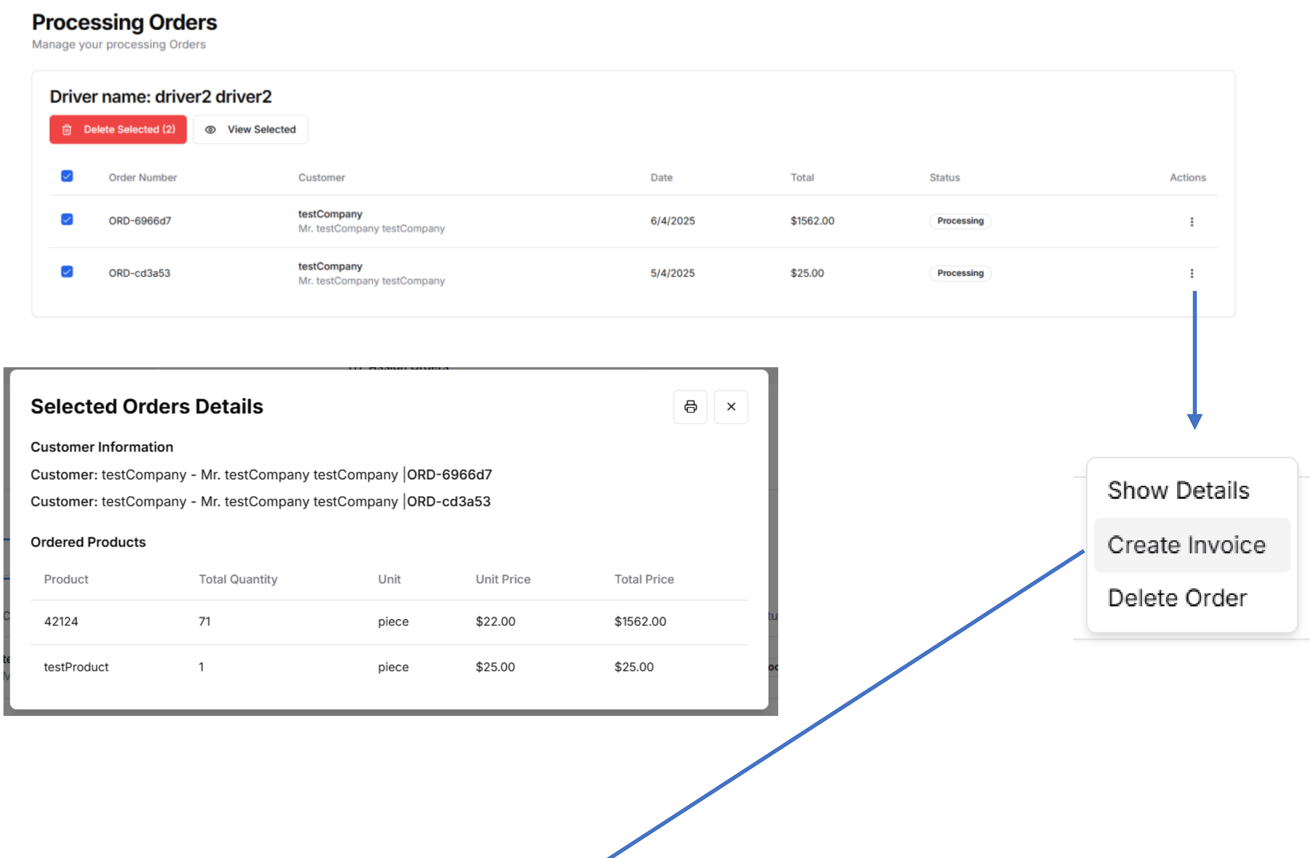
2. Assign Orders

- Allows the admin to assign pending orders to drivers.
- Upon assignment, the order status changes from **"Pending"** to **"Processing"**.
- Real-time notifications are sent to the assigned drivers with details of their orders.
- Features an advanced route visualization using an **AI algorithm (Simulated Annealing)** that optimizes the delivery path based on customer locations and the company warehouse location.
- This helps management plan and distribute orders more intelligently and efficiently.



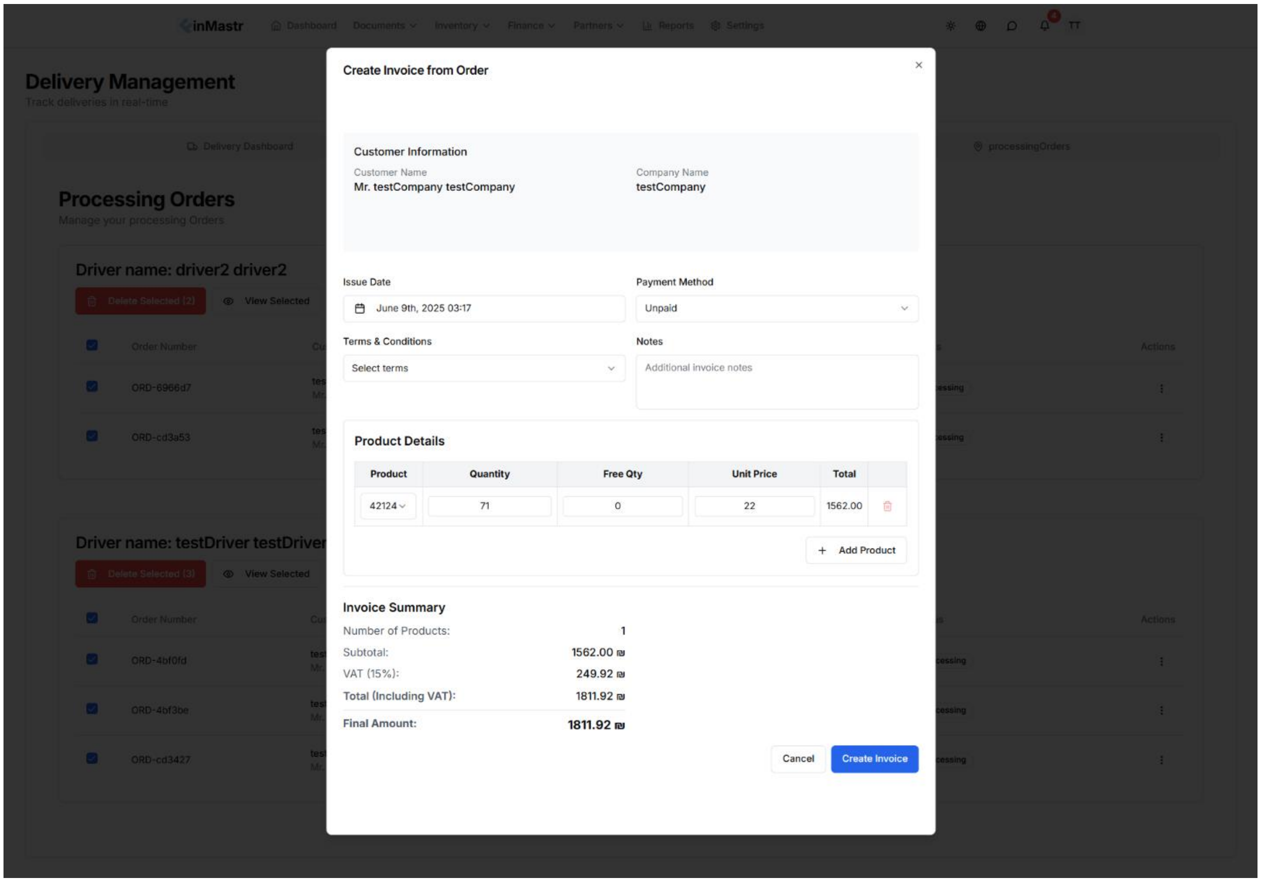
### 3. Processing Orders

Displays a list of active drivers along with the orders each driver is currently handling.
Enables real-time tracking of the progress of each order, from assignment to delivery.
Allows administrators to monitor which orders are in progress, who is responsible for them, and what stage they are in.
The page also includes the ability to print and generate a summary report of all orders assigned to each driver.
This summary can be used by drivers during deliveries or by warehouse staff for preparation.
The report includes key order details such as customer names, order items, quantities, and delivery addresses.




Invoice Generation and Order Finalization

- After order preparation, the admin can generate an invoice for each order.
- The invoice allows editing of products or quantities before final confirmation.
- The customer’s balance is automatically updated based on the finalized invoice details.
- This ensures invoice accuracy and enhances financial tracking.




## 4. Partners Management

The **Partners** section of the system is divided into three key modules:

- **Customers**
- **Suppliers**
- **Service Providers**

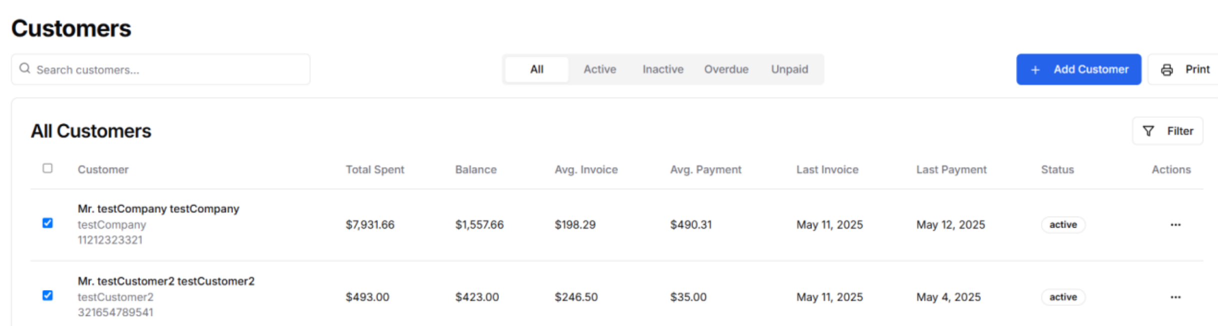
Each of these modules provides a structured and comprehensive view of the company's relationships with its external stakeholders.

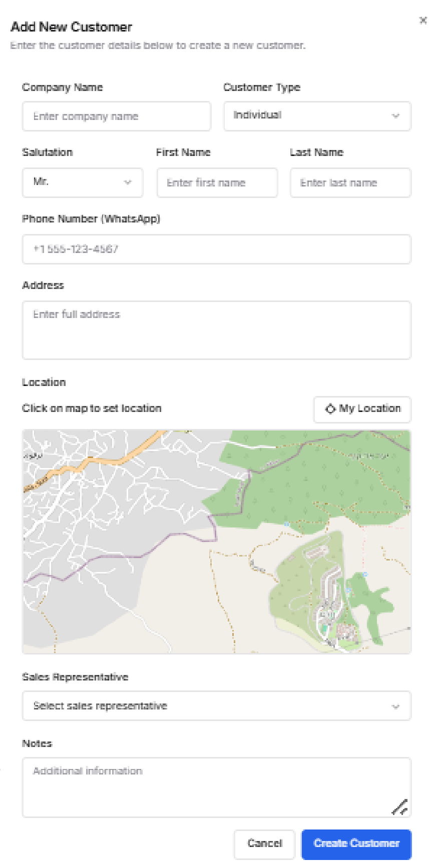
# 1. Customers Page

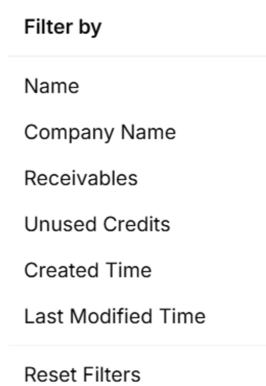
**Description:**
This page manages all clients and buyers that interact with the company. It provides a complete overview of each customer's financial and transactional activity.

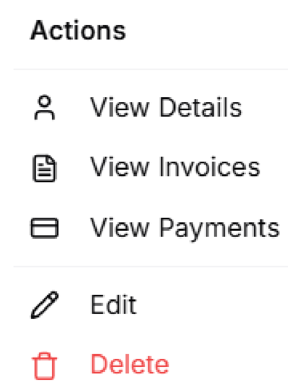
**Key Features:**

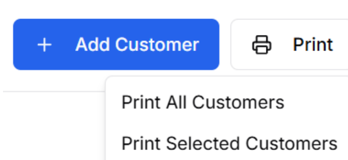
- Displays customer details: total spent, current balance, average invoice/payment, last invoice/payment, and status.
- **View Details** shows the full customer profile, including a map location and transaction history.
- Linked to **Invoices**, **Payments**, and **Cheques** modules.
- Allows filtering customers by name, company name, receivables, unused credits, creation time, and modification time.
- Option to **add a new customer** with address location on the map.

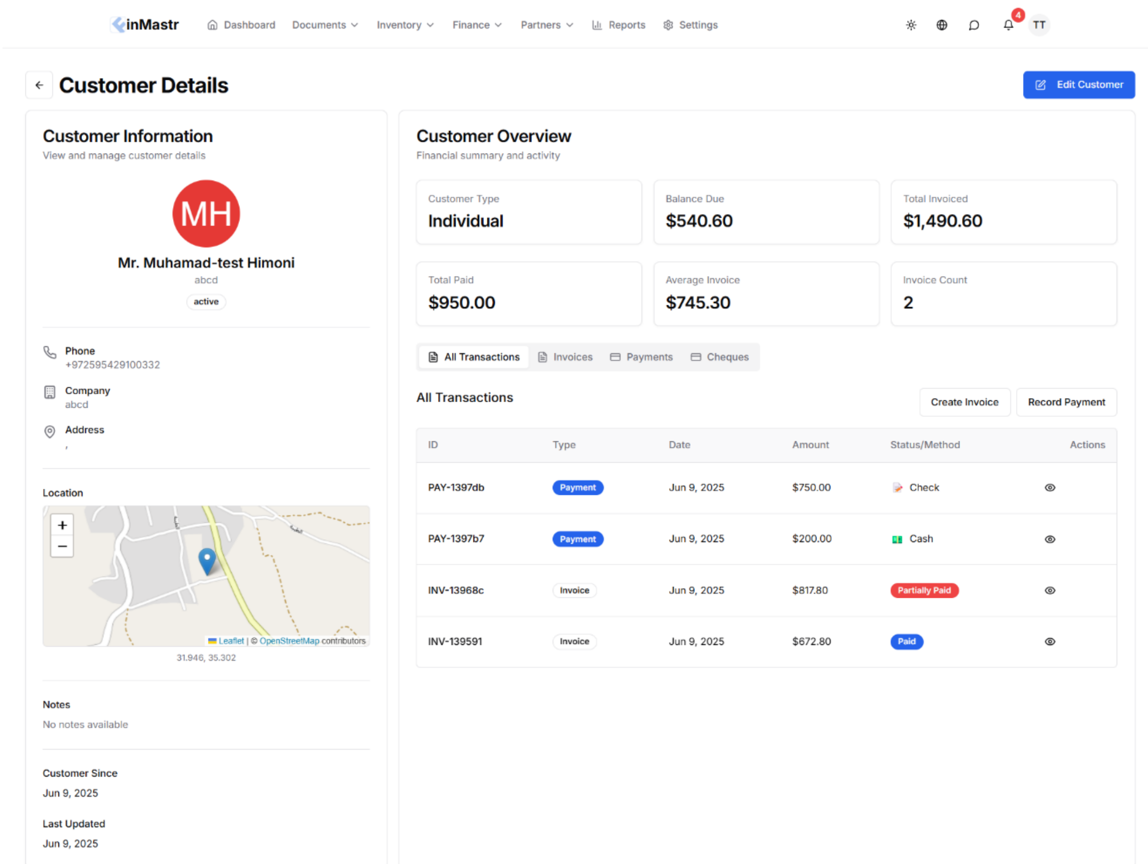





## 2. Suppliers Page

**Description:**
This page handles all product suppliers and vendors that provide goods to the company. It focuses on tracking procurement-related transactions and financial data.

**Key Features:**

- Shows supplier information: total expenses, outstanding balances, and payment history.
- Linked directly with the **Expenses** and **Products** modules.
- Enables detailed supplier profile viewing, editing, and deletion.
- Map integration for location-based logistics planning.
- Ability to add new suppliers and print supplier lists.

## 3. Service Providers Page

**Description:**
This page is designed for managing external contractors or service partners who offer services to the company (e.g., maintenance, installation, support).

**Key Features:**

- Tracks service-specific expenses and engagements.
- Linked with the **Services** and **Expenses** modules.
- Shows provider transaction history, current balance, and operational region.
- Includes options to add new providers, view service logs, and manage service-related financials.
- Integrated map for visualizing provider coverage areas.
- Printable views for documentation.

## Shared Functionality Across All Partner Pages

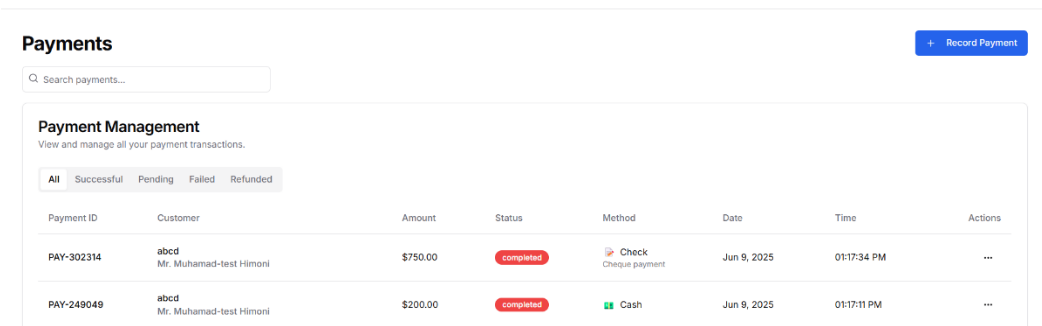
- **Data Table Columns:**
  - Name / Company
  - Total Spent
  - Balance
  - Avg. Invoice
  - Avg. Payment
  - Last Invoice
  - Last Payment
  - Status
  - Actions (View, Edit, Delete)
- **Actions Menu Includes:**
  - **View Details**
  - **View Invoices**
  - **View Payments**
  - **Edit / Delete**
- **Advanced Filters:**
  - Filter by name, company, financial metrics, and timestamps.
  - Reset and customize filters based on business needs.
- **Map Integration:**
  - For visual management of partner locations and route planning.
- **Bulk Actions:**
  - Select multiple entries for printing or data export

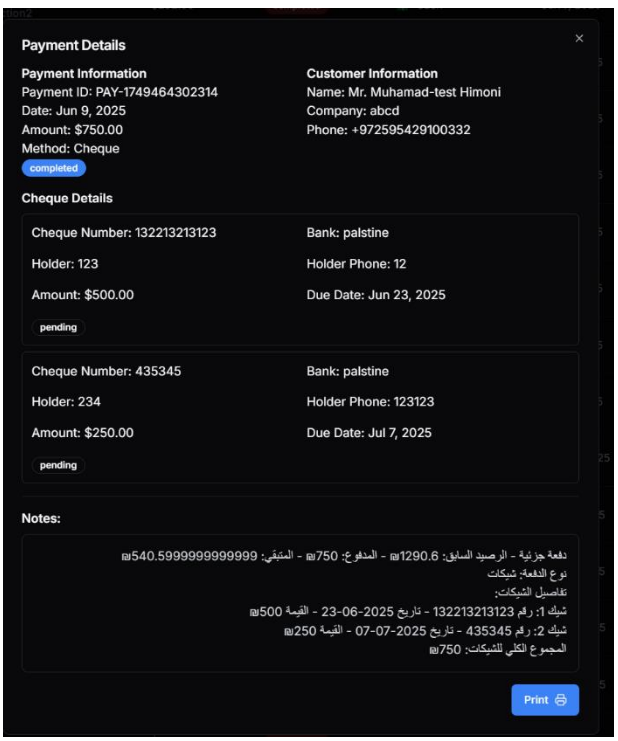
## 5. Finance Section

# 1. Payments Page

The **Payments** page also includes the ability to **register a new payment** manually. This can be done through either:

- **Cash Payment**
- **Cheque Payment**



**Create Payment:**

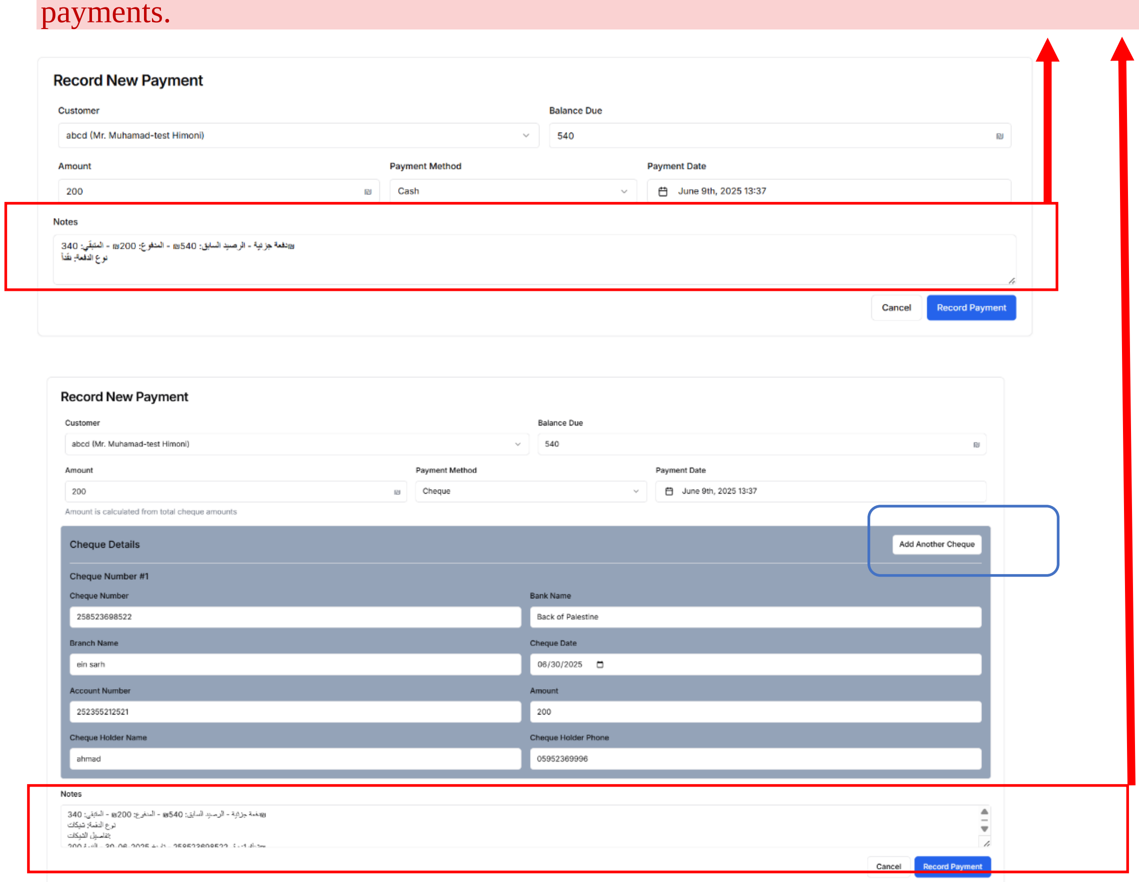
- Displays a list of all **customers** with their respective **current balances**.
- Allows users to **select a customer** and record a payment toward their outstanding invoices.
- For cheque payments:
    - Multiple cheques can be recorded for a single payment.
    - Each cheque entry includes: Cheque number ,Bank name, Amount and Due date

In the **New Payment Creation** page, the system automatically **generates a detailed payment summary** based on the input provided by the user.

*Auto-Generated Payment Summary Includes:*

- Total payment amount
- Payment method (Cash or Cheque)
- Associated customer and their current balance
- Invoice(s) covered by the payment
- Cheque details (if applicable)
- Remaining balance (if any)

This smart summary generation feature is not available in most traditional accounting applications, which often require manual entry or offer limited insight before submitting payments.



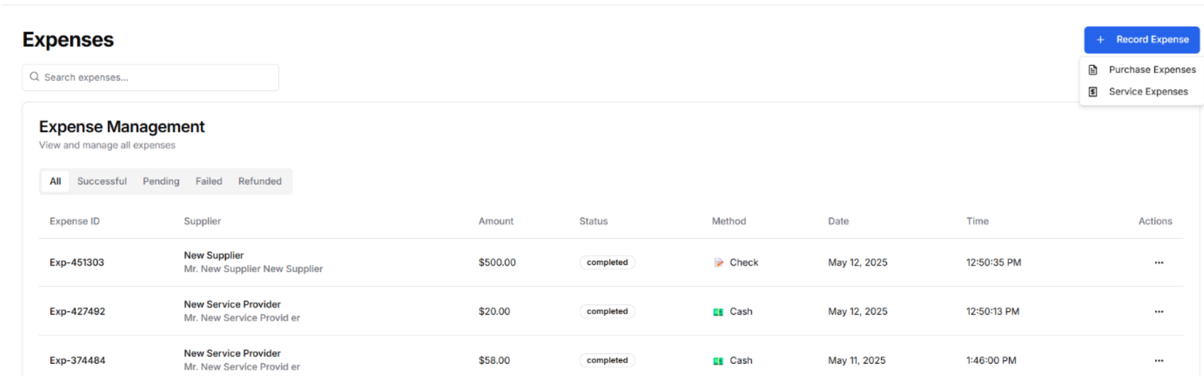
## 2. Expenses Page

The **Expenses** module is divided into three integrated pages to handle various types of business expenditures efficiently and with clarity. These are:

1. **Expenses Overview Page**

This main page displays a unified table view of all recorded expenses, including both purchase and service expenses.

- Displayed Columns: Expense ID, Type (Purchase/Service), Provider (Supplier or Service Provider), Amount, Date & Time, Actions (View, Edit, Delete, Download)
- Advanced filtering by date, provider, category, and amount.
- Option to export or print full reports.

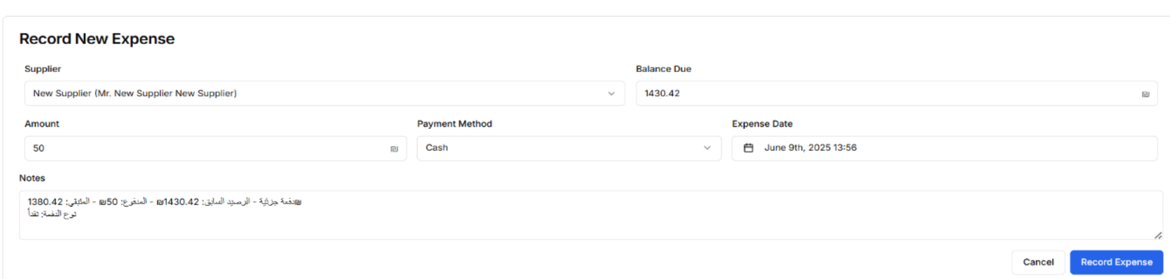



### 2. Purchase Expenses Page

This page is designed to register expenses related to **product purchases** and **inventory suppliers**. It shares a similar layout and logic with the payment registration page, but it is specifically linked to **purchase invoices** and supplier accounts.

### Key Features:

- Select supplier and purchase invoice.
- Input payment method (cash, cheque, etc.).
- Automatically calculates the remaining balance for the supplier.
- Links each expense with the relevant warehouse and product records.
- Generates a clear summary for each transaction.



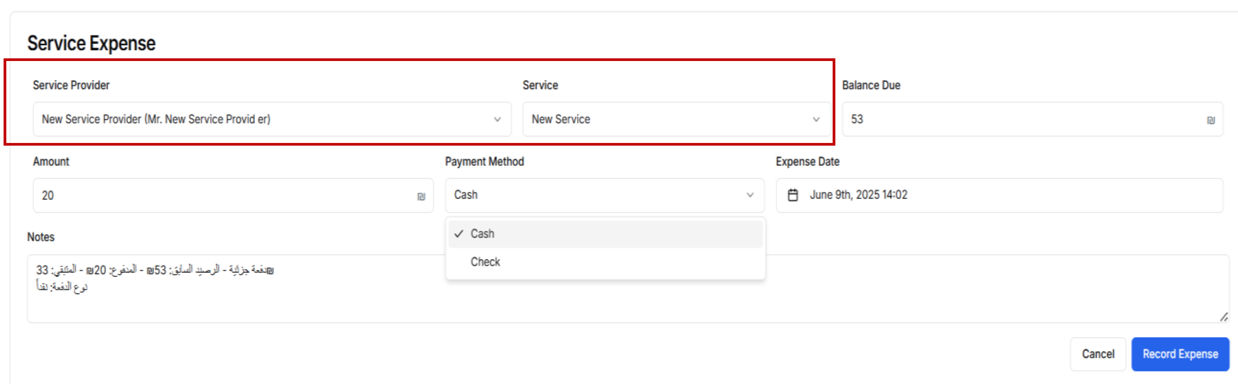
### 3. Service Expenses Page

This page is tailored for **service-related expenses** and follows a **double-entry concept** to ensure accurate internal tracking — without requiring users to know accounting terminology.

**Key Features:**

- Select a **service provider** and the **specific service** offered.
- Enter expense amount and description.
- The system automatically assigns the cost to both:
    - The **service provider's account**, and
    - The **service** it belongs to.
- A summary is generated showing how the expense impacts both internal and external financial records.
- Sure! Here's the paragraph in English, rewritten in a professional and concise form:

The **Service Expenses** page is designed based on the **double-entry accounting principle**, ensuring that each recorded expense is accurately reflected. When creating a service expense, the system automatically registers the amount under both the selected **service provider** and the **service category**. This is particularly useful in cases where multiple providers can offer the same service, and a single provider can offer multiple services. This structure allows for precise tracking of financial activity per provider and detailed analysis of service-specific spending, all without requiring advanced accounting knowledge.



## 6. Invoice Management Page

The Invoice Management page serves as the central hub for managing all invoice transactions within the system, including sales invoices, purchase invoices, and service invoices. It provides a comprehensive and user-friendly interface for tracking, filtering, sorting, and managing invoices with advanced functionalities.

**Page Components**

### A. Invoice Table

The main table displays a list of all invoices with the following columns: Invoice: Invoice number or unique identifier. Name: Customer name (for sales) or supplier name (for purchases/services).Date: Issue date of the invoice. Due Date: Payment due date. Status: Current status of the invoice (Draft, Pending, Paid, Overdue, Unpaid, And Partially Paid).Amount: Total invoice amount. Actions: Available operations for each invoice.

### B. Filters & Sorting

Users can filter and sort invoices based on multiple criteria:

#### 1. Invoice Type Filter

- Sales Invoices: Issued to customers (affects customer balance).
- Purchase Invoices: Received from suppliers (for goods).
- Service Invoices: A subset of purchase invoices for services (e.g., internet, transportation, utilities).

#### 2. Status Filter

- Draft: Unfinalized invoices (not yet sent).
- Pending: Awaiting payment.
- Paid: Fully paid.
- Overdue: Past due date with unpaid/partial payment.
- Unpaid: No payments recorded.
- Partially Paid: Partial payments received.

#### 3. Advanced Sorting

Invoices can be sorted in Ascending (A-Z, Old-New, Low-High) or Descending (Z-A, New-Old, High-Low) order by:

- Issue Date
- Due Date
- Amount
- Customer/Supplier Name

### C. Invoice Actions

Each invoice in the table has actionable options:

- View Invoice: Displays full invoice details (customer/supplier info, items, taxes, discounts, totals, notes).
- Edit Invoice: Modify invoice details (if in draft status or permitted).
- Download PDF: Export the invoice as a PDF file.
- Send Invoice: Email the invoice directly to the customer/supplier.
- Record Payment: Log payments to update the invoice status (paid/partially paid).

### D. Invoice Details & Additional Features

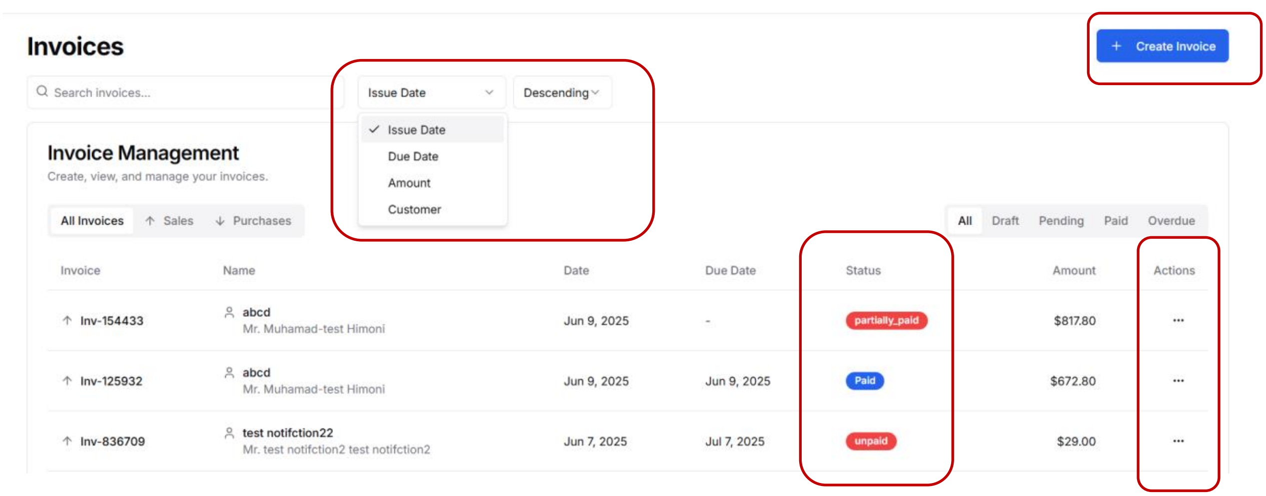
When viewing an invoice (View Invoice), users can:

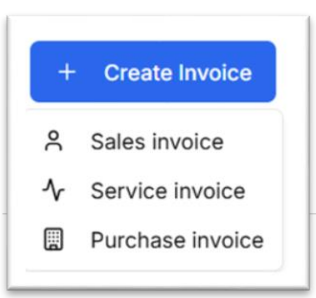
- Download as PDF for offline use.
- Share via WhatsApp (PDF or link).
- Print the invoice directly.

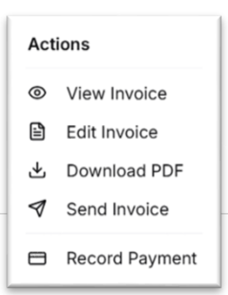
### E. Invoice Status Mechanism

The system **automatically updates** the invoice status based on recorded payments:

- Unpaid: No payments received.
- Partially Paid: Partial payment received (less than total amount).
- Paid: Full payment received (equal to or exceeding the total).
- Overdue: Due date passed with unpaid/partial payment.



**Sales invoice INV-1749464154433**

View invoice details and download as PDF.

×

**BillFlow Inc.**
123 Business Street
San Francisco, CA 94107
billing@billflow.com

**Invoice**
Invoice Number: INV-1749464154433
Issue Date: Jun 9, 2025
Due Date: -

Bill To:

Mr. Muhamad-test Himoni
N/A
Customer Address Line 1
Customer City, State, ZIP

| Product Name | Quantity | Free Qty | Unit Price | Total |
|---|---|---|---|---|
| testProduct | 15 | 0 | $25.00 | $375.00 |
| 42124 | 15 | 0 | $22.00 | $330.00 |

| | |
|---|---|
| Subtotal: | $705.00 |
| Tax (10%): | $70.50 |
| **Total:** | **$817.80** |
| Status: | Partially_paid |

Payment Terms: Net 7 days
Please make payment to: BillFlow Inc.
Bank: National Bank
Account: 1234567890
Routing: 987654321

Print | Download PDF | Share on WhatsApp | Send Invoice

**Invoice Details**
Invoice Number: INV-1749464154433
Customer: Mr. Muhamad-test Himoni
Date: Jun 9, 2025
Due Date: -
Status: partially_paid
Total Amount: $817.80

**Items:**

- testProduct
  Quantity: 15
  Price: $25.00
  Total: $375.00

- 42124
  Quantity: 15
  Price: $22.00
  Total: $330.00

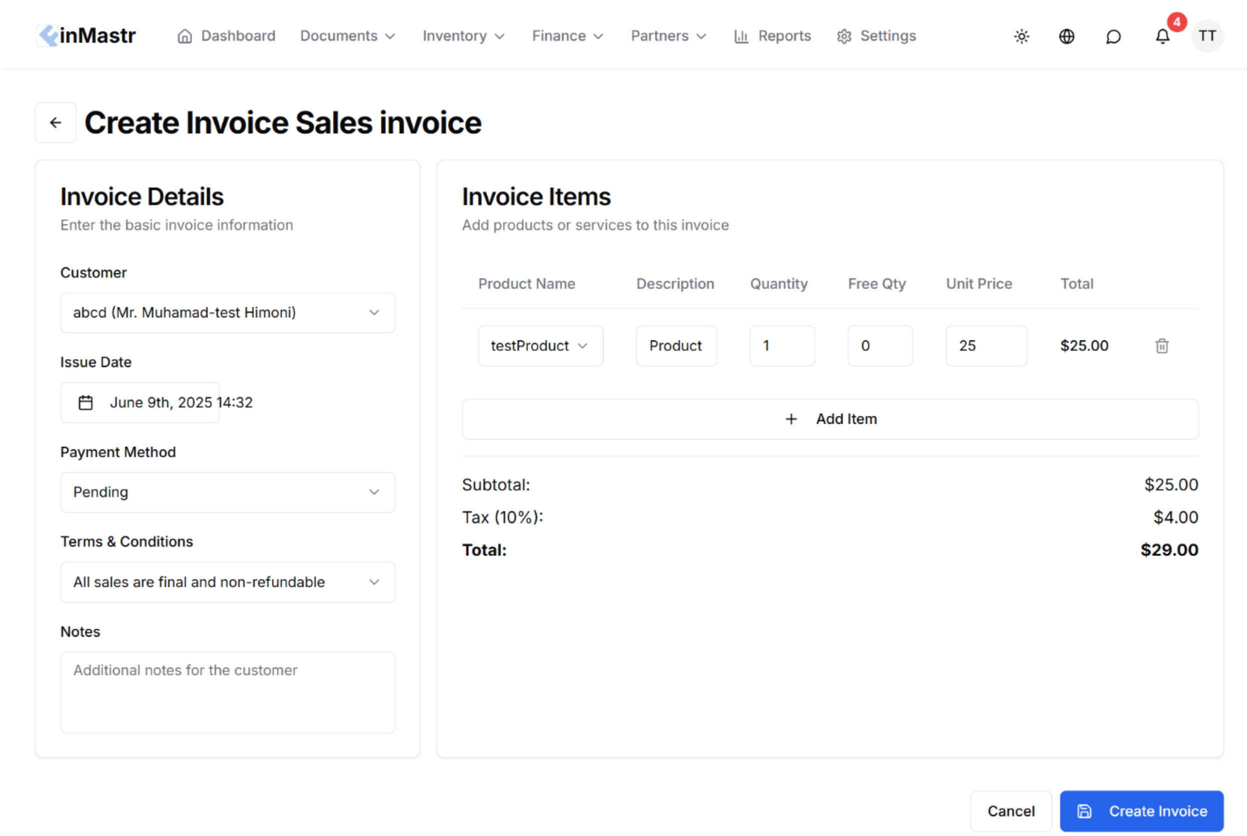
**Payment Information**
Please make payment to: BillFlow Inc.
Bank: National Bank
Account: 1234567890
Routing: 987654321
2:28 PM

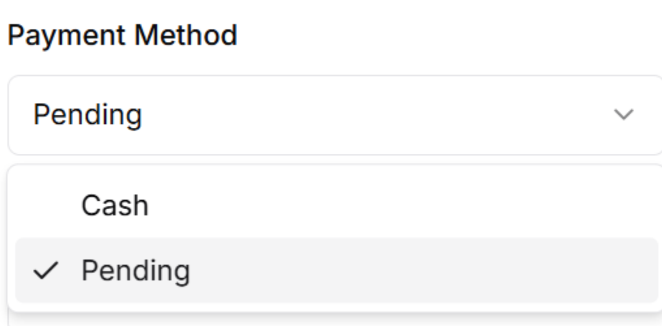
## F. Types of Invoices

The system supports three invoice types:

### A. Sales Invoices

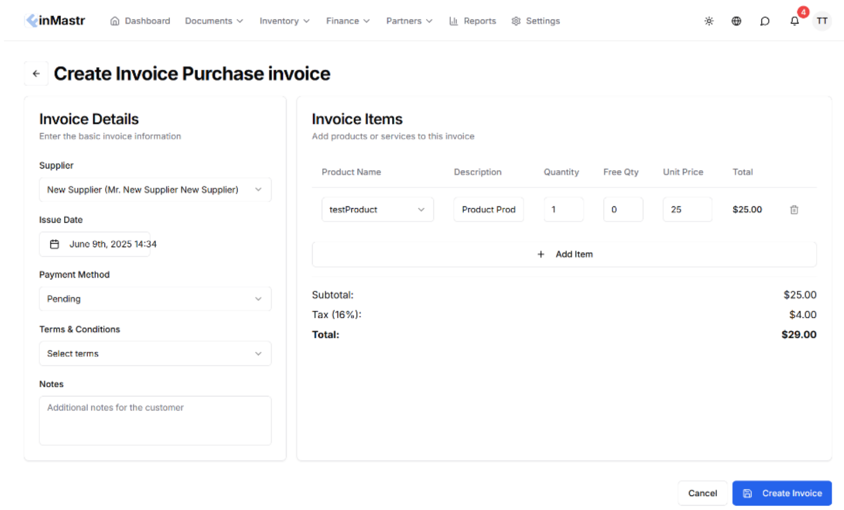
- Issued to customers.
- Increases customer balance (accounts receivable).
- Affects sales revenue.



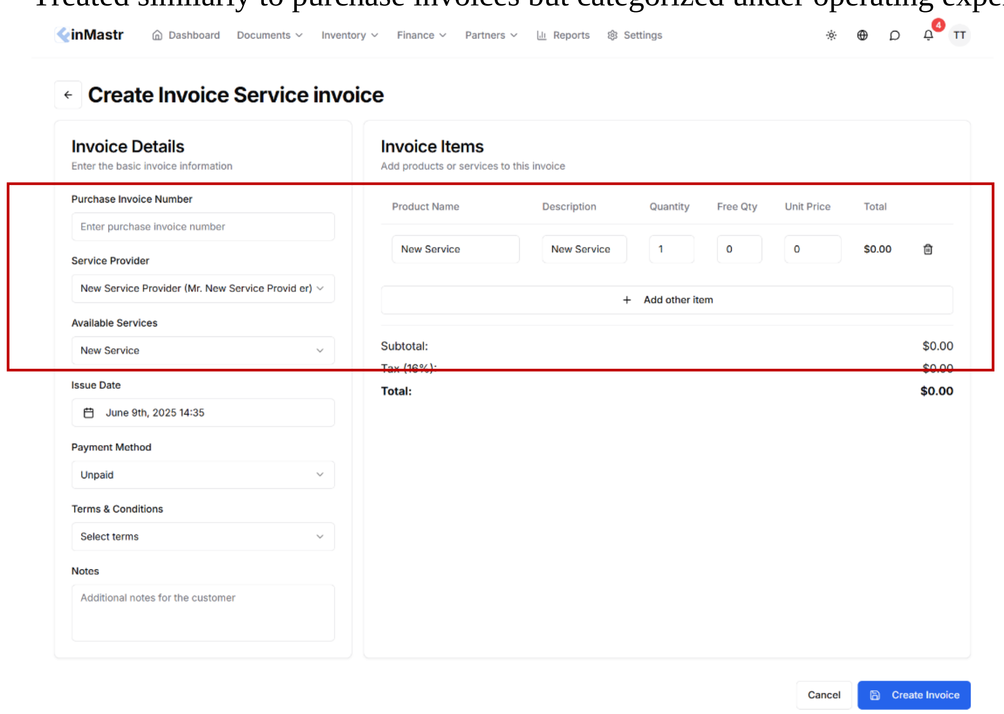
## B. Purchase Invoices

- Received from suppliers (for goods).
- Increases supplier balance (accounts payable).




## C. Service Invoices

- A subset of purchase invoices for services (e.g., internet, utilities, transportation).
- Treated similarly to purchase invoices but categorized under operating expenses.



## 7. Service Management Page

The **Service Management** page allows companies to view and manage all their available services in one centralized location. It provides a clear interface to monitor service status, associated providers, and expenses.

### Page Layout and Features

#### A. Services Table

The main table displays all company services with the following columns:

- **Name:** Service name
- **Description:** Brief details about the service
- **Service Providers:** List or count of providers linked to this service
- **Total Expenses:** Total amount spent on this service
- **Status:** Current status (Active / Inactive)
- **Created At:** Date the service was added
- **Actions:** Available options for each service

#### B. Add New Service

The page includes a form or modal to add a new service with the following fields:

- **Service Name:** Text input to enter the service name
- **Description:** Text input for detailed description
- **Service Providers:** Select one or multiple service providers from the existing list

#### C. Service Providers Management

- Each service can be linked to **multiple service providers**.
- Providers represent companies or individuals who deliver the service.

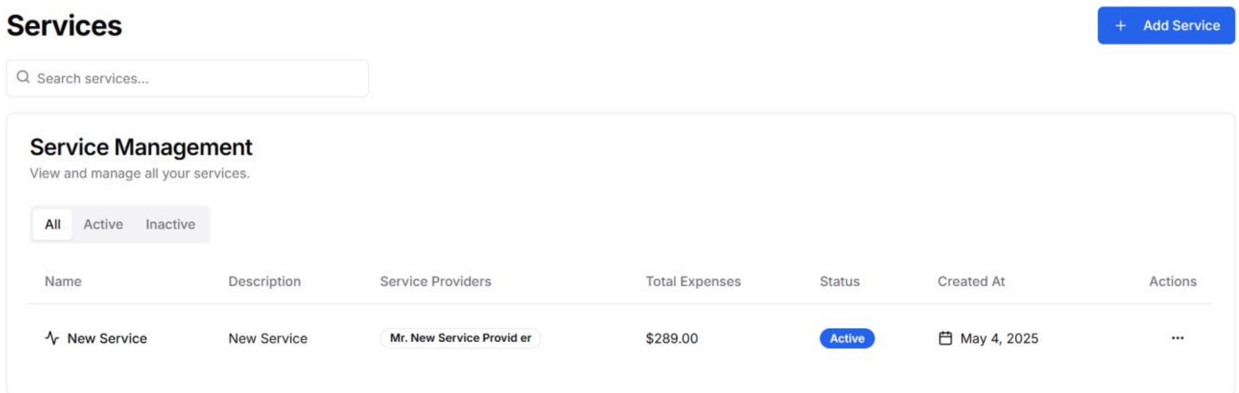
#### D. View Details

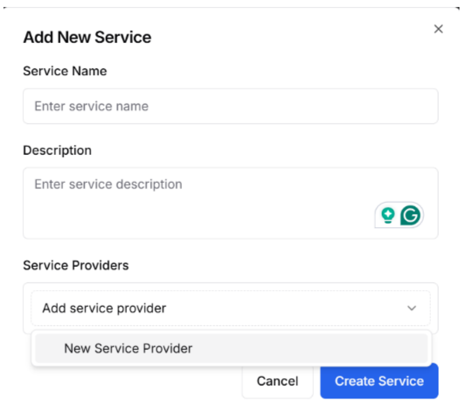
Selecting **View Details** for a service shows:

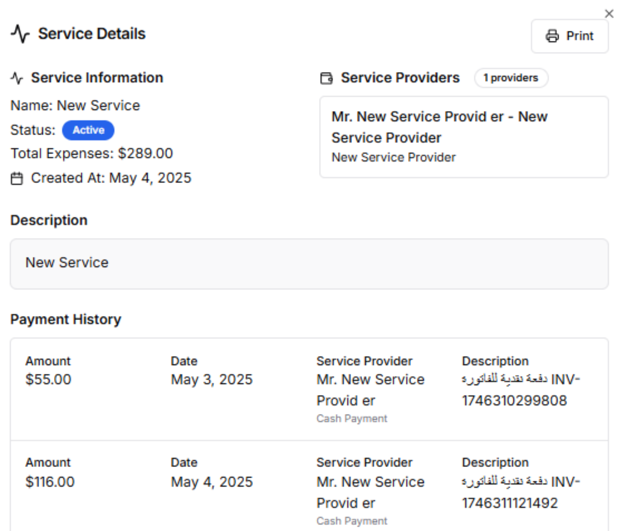
- Full service description
- List of all expenses related to that service, aggregated from all linked providers
- Option to **print** the detailed service expense report

#### E. Edit Service

Allows modification of service details, including changing the name, description, and associated providers.


Services
Add Service
Search services...
Service Management
View and manage all your services.
All
Active
Inactive
Name
Description
Service Providers
Total Expenses
Status
Created At
Actions
New Service
New Service
Mr. New Service Provid er
$289.00
Active
May 4, 2025


Add New Service
Service Name
Enter service name
Description
Enter service description
Service Providers
Add service provider
New Service Provider
Cancel
Create Service


Service Details
Print
Service Information
Name: New Service
Status: Active
Total Expenses: $289.00
Created At: May 4, 2025
Service Providers
1 providers
Mr. New Service Provid er - New Service Provider
New Service Provider
Description
New Service
Payment History
Amount
$55.00
Date
May 3, 2025
Service Provider
Mr. New Service Provid er
Cash Payment
Description
دفعة نقدية للفاتوره INV-1746310299808
Amount
$116.00
Date
May 4, 2025
Service Provider
Mr. New Service Provid er
Cash Payment
Description
دفعة نقدية للفاتوره INV-1746311121492

## 8. **Reports Page**

The **Reports** page provides a comprehensive analytical dashboard that enables businesses to visualize and understand their financial and operational performance. It is divided into multiple dynamic and interactive sections, designed to offer valuable insights and support data-driven decision-making. The key strength of this page lies in its **modular structure**, **real-time data integration**, **advanced filters**, and **export capabilities**.

### Main Features and Tools

- **Date Range Selector:** Users can filter reports by custom date ranges.
- **Export Options:** Export reports as **PDF** or **Excel**, supporting document sharing and record keeping.
- **Refresh Data Button:** Ensures up-to-date reports synced with recent transactions.
- **Advanced Filters:** Filters based on time intervals, categories, methods, etc.

### Reports Sections

### A. Overview Report

This section summarizes the overall financial health and activity of the company. It includes:

- **Revenue, Expenses, Net Profit, and Growth Widgets:**
  Visually present current performance and compare it to previous periods.
- **Business KPIs:**
  Quick insights into number of customers, products, services, orders, and suppliers.

**Key Feature:**
Color-coded metrics and arrows indicate performance trends (growth or decline), providing immediate at-a-glance insights.

### B. Sales Report

This section focuses on sales performance and order trends.

- **Chart & Table Views:**
  Switch between visual graphs and data tables to view revenue, order volume, and customer behavior over time.
- **Sales Analysis:**
  Shows average order value, new vs repeat customers, and order distribution by time.
- **Top Customers Widget:**
  Highlights customers with the highest spending.

**Key Feature:**
Supports daily, weekly, and monthly breakdowns for flexible sales tracking.

### C. Expenses Report

Designed to monitor and analyse the company's spending patterns.

- **Monthly Breakdown Charts:**
  Display trends in expenses by time and method (e.g., cash, cheque).
- **Top Suppliers & Methods:**
  Identify which vendors or payment types account for most of the spending.
- **Purchase Invoice Tracking:**
  Link expenses directly to related purchase invoices for auditing and transparency.

**Key Feature:**
Combines graphical data with tabular insights for full control over cost management.

### D. Customers Report

Provides an in-depth view of the customer base and their financial interaction with the business.

- **Customer Summary:**
  Number of total, active, and inactive customers with average balances and transactions.
- **Balance Distribution Graphs:**
  Visualize how customer balances are spread across defined ranges.
- **Demographics:**
  Segment customers by type or location for targeted marketing or support.

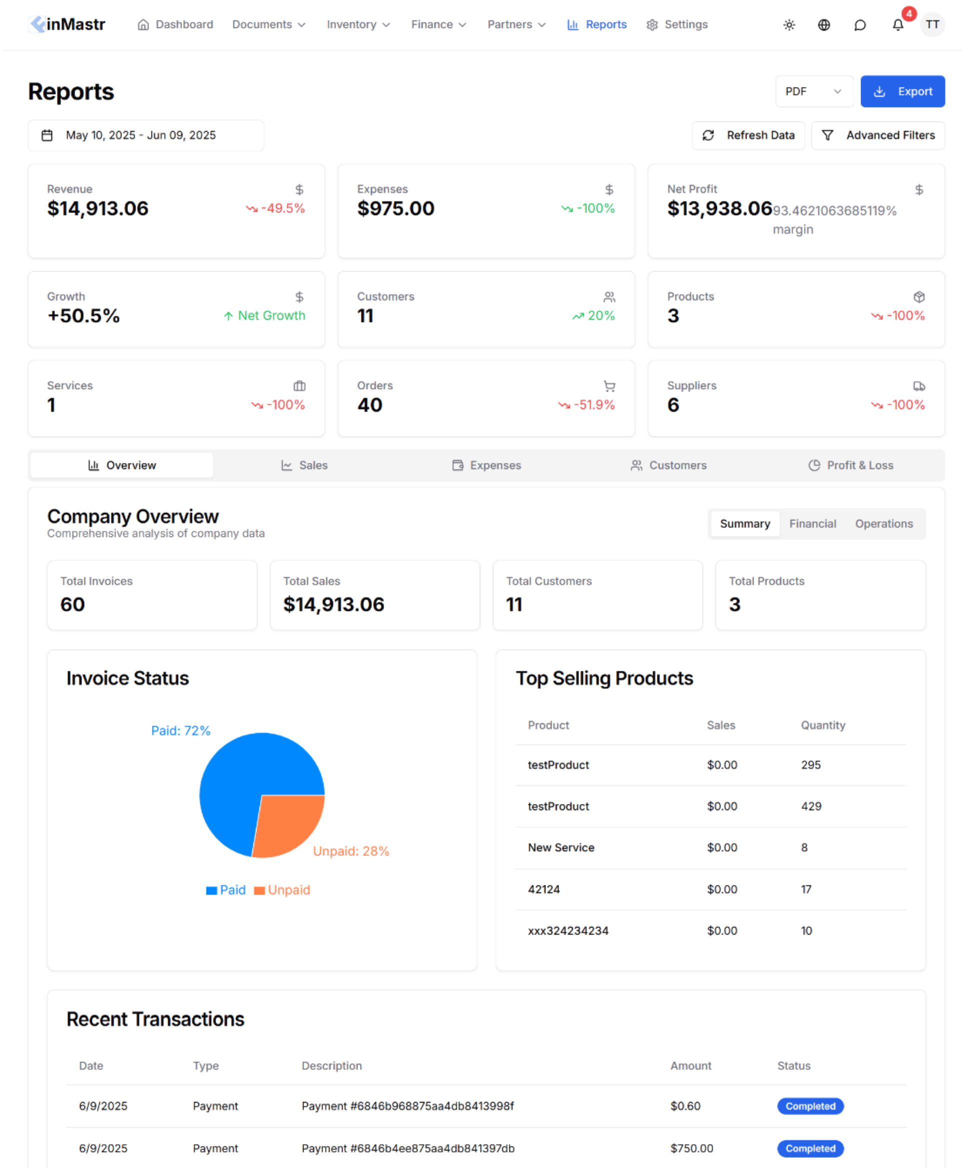
**Key Feature:**
Helps businesses identify their most valuable customers and detect at-risk accounts.

### E. Profit & Loss Report

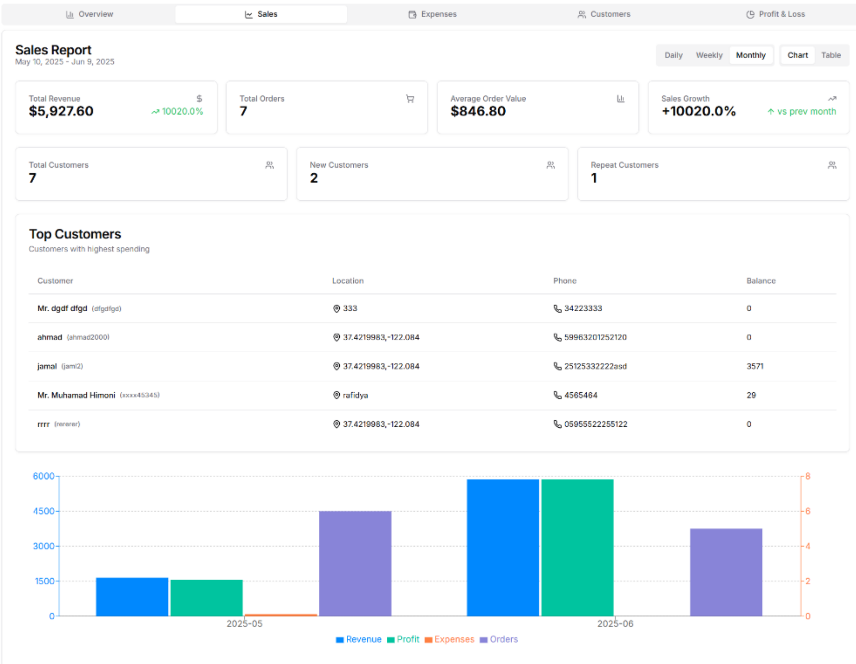
This section offers a clear financial breakdown of the business's earnings versus expenses.

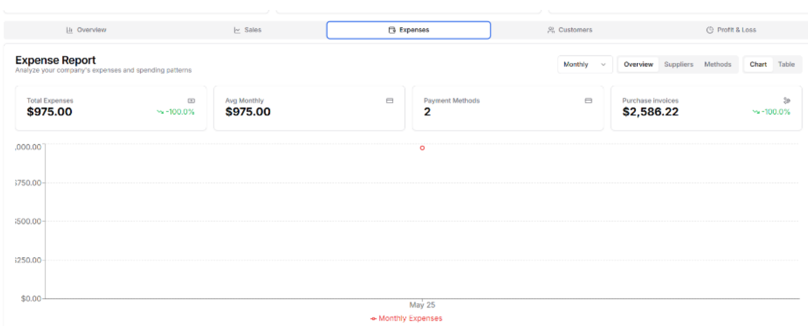
- **Bar & Line Charts:**
  Show monthly revenue, expenses, and net profit over time.
- **Profit Margin Calculation:**
  Automatically computes profitability ratios to support long-term strategy.
- **Date Range Navigation:**
  Enables comparative analysis between different periods.

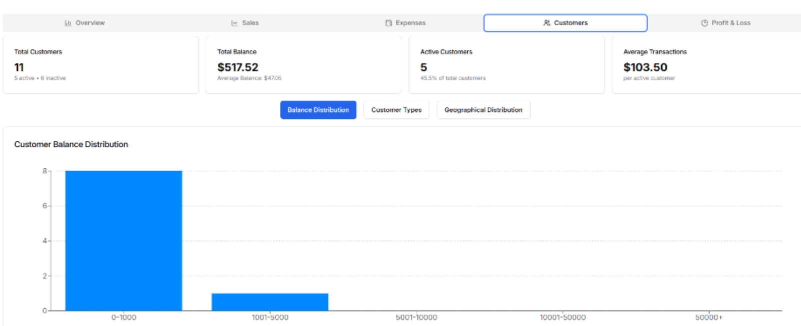
**Key Feature:**
Simplifies accounting by aggregating income and cost data into a visual, actionable format.


inMastr
Dashboard
Documents
Inventory
Finance
Partners
Reports
Settings
4
TT
Reports
PDF
Export
May 10, 2025 - Jun 09, 2025
Refresh Data
Advanced Filters
Revenue
$14,913.06
-49.5%
Expenses
$975.00
-100%
Net Profit
$13,938.06 93.4621063685119% margin
Growth
+50.5%
Net Growth
Customers
11
20%
Products
3
-100%
Services
1
-100%
Orders
40
-51.9%
Suppliers
6
-100%
Overview
Sales
Expenses
Customers
Profit & Loss
Company Overview
Comprehensive analysis of company data
Summary
Financial
Operations
Total Invoices
60
Total Sales
$14,913.06
Total Customers
11
Total Products
3
Invoice Status
Paid: 72%
Unpaid: 28%
Paid
Unpaid
Top Selling Products
Product Sales Quantity
testProduct $0.00 295
testProduct $0.00 429
New Service $0.00 8
42124 $0.00 17
xxx324234234 $0.00 10
Recent Transactions
Date Type Description Amount Status
6/9/2025 Payment Payment #6846b968875aa4db8413998f $0.60 Completed
6/9/2025 Payment Payment #6846b4ee875aa4db841397db $750.00 Completed


Profit & Loss Report
Revenue, expenses, and profit analysis
Bar
Line
Monthly
Total Revenue
$7,522.02
Total Expenses
$607.00
Net Profit
$6,915.02
Profit Margin
91.9%
$4.5k
$3k
$1.5k
$0k
$-1.5k
2025-05-11
2025-05-12
2025-06-04
2025-06-05
2025-06-07
2025-06-09
Revenue
Expenses
Profit


Overview
Sales
Expenses
Customers
Profit & Loss
Sales Report
May 10, 2025 - Jun 9, 2025
Daily
Weekly
Monthly
Chart
Table
Total Revenue
$5,927.60
10020.0%
Total Orders
7
Average Order Value
$846.80
Sales Growth
+10020.0%
vs prev month
Total Customers
7
New Customers
2
Repeat Customers
1
Top Customers
Customers with highest spending
Customer
Location
Phone
Balance
Mr. dgdf dfgd (dfgdfgd)
333
34223333
0
ahmad (ahmad2000l
37.4219983,-122.084
59963201252120
0
jamal (jaml2)
37.4219983,-122.084
25125332222asd
3571
Mr. Muhamad Himoni (xxxx45345)
rafidya
4565464
29
rrrr (rwrerer)
37.4219983,-122.084
05955522255122
0
6000
4500
3000
1500
0
2025-05
2025-06
Revenue
Profit
Expenses
Orders

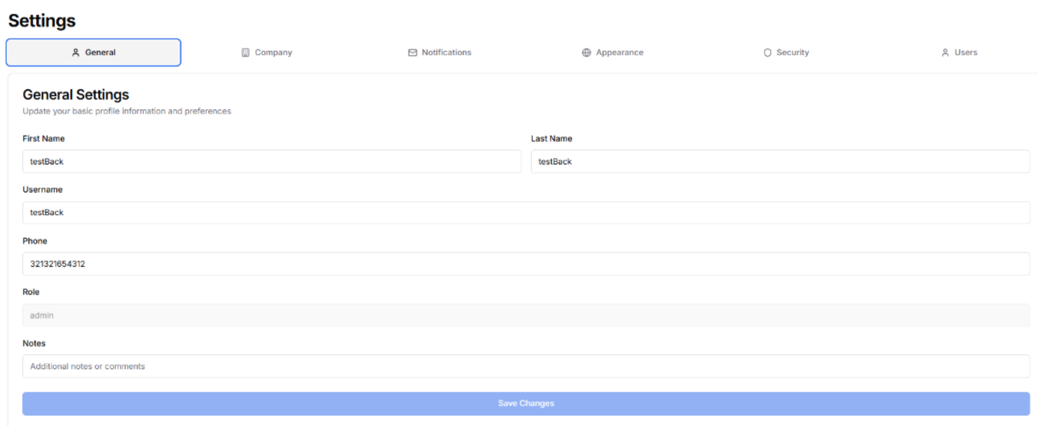

Overview
Sales
Expenses
Customers
Profit & Loss
Expense Report
Analyze your company's expenses and spending patterns
Monthly
Overview
Suppliers
Methods
Chart
Table
Total Expenses
$975.00
-100.0%
Avg Monthly
$975.00
Payment Methods
2
Purchase Invoices
$2,586.22
-100.0%
$0.00
May 25
Monthly Expenses


Overview
Sales
Expenses
Customers
Profit & Loss
Total Customers
11
Total Balance
$517.52
Active Customers
5
Average Transactions
$103.50
per active customer
Balance Distribution
Customer Types
Geographical Distribution
Customer Balance Distribution
8
6
4
2
0
0-1000
1001-5000
5001-10000
10001-50000
50000+
Number of Customers

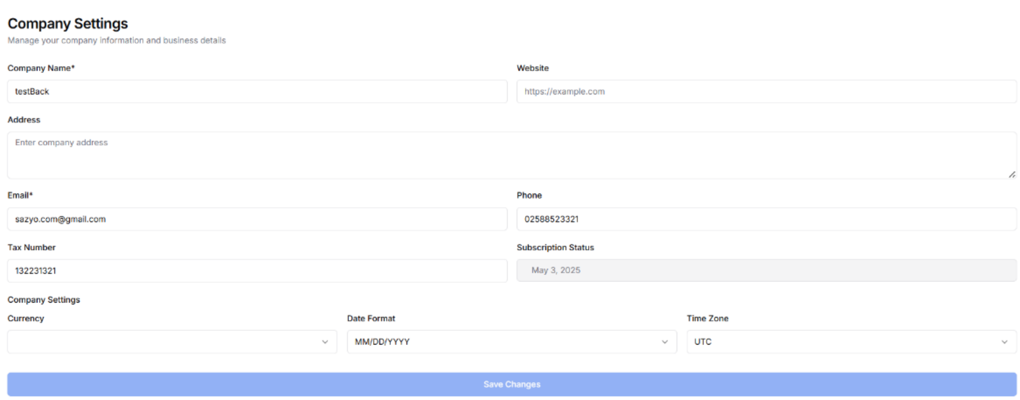
## 9. Settings Page

The **Settings** page serves as the central configuration hub for the system. It enables users and administrators to customize their profile, company information, system appearance, notification preferences, and security settings. The page is divided into **five main sections**, each with specific functionalities and user permissions.

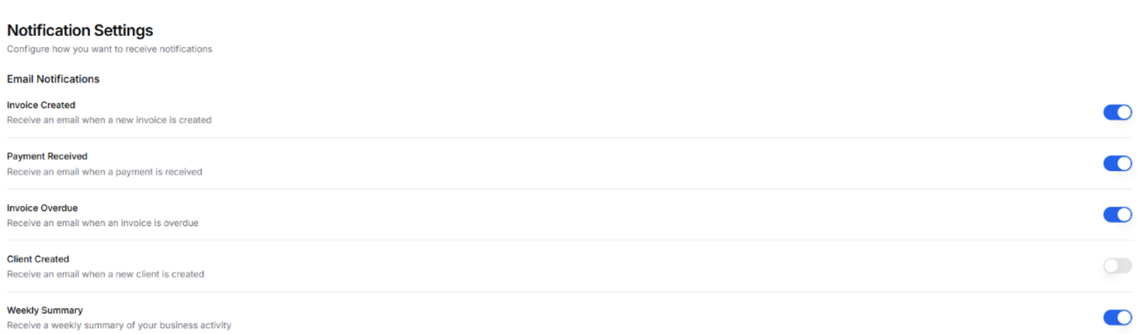
**A. General Settings:** Allows users to update their personal profile and account preferences.



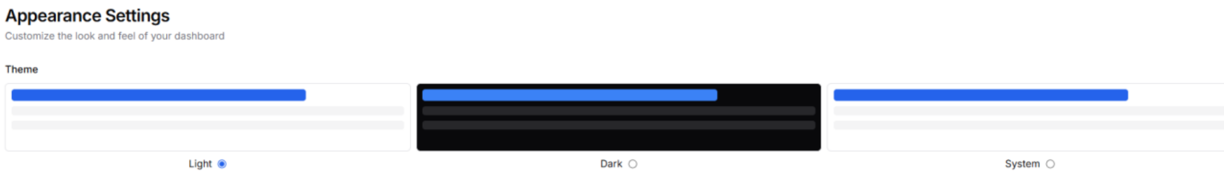

**B. Company Settings :Used** to manage business identity and financial configuration.




**C.Notification Settings:** Enables users to configure how they receive system notifications.

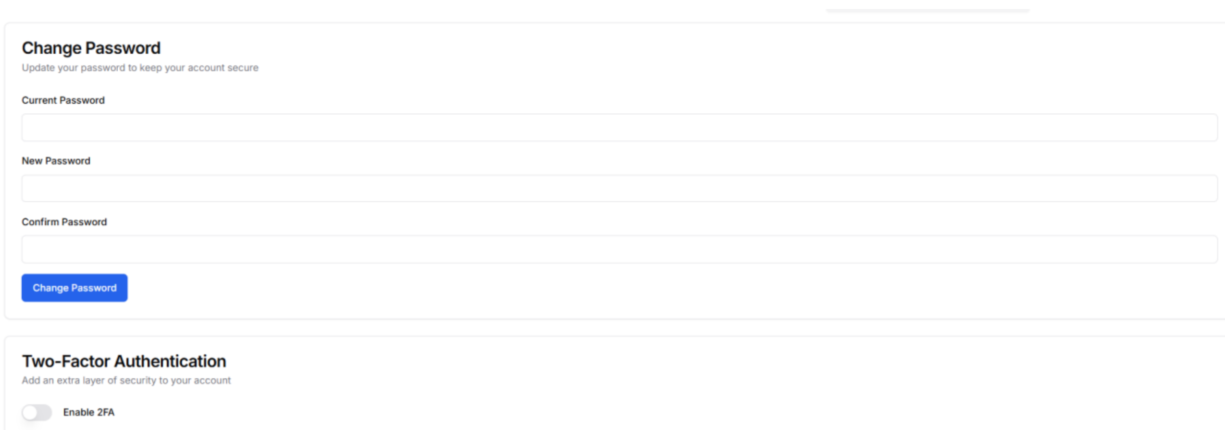


**D. Appearance Settings**: Allows users to personalize the look and feel of their dashboard.



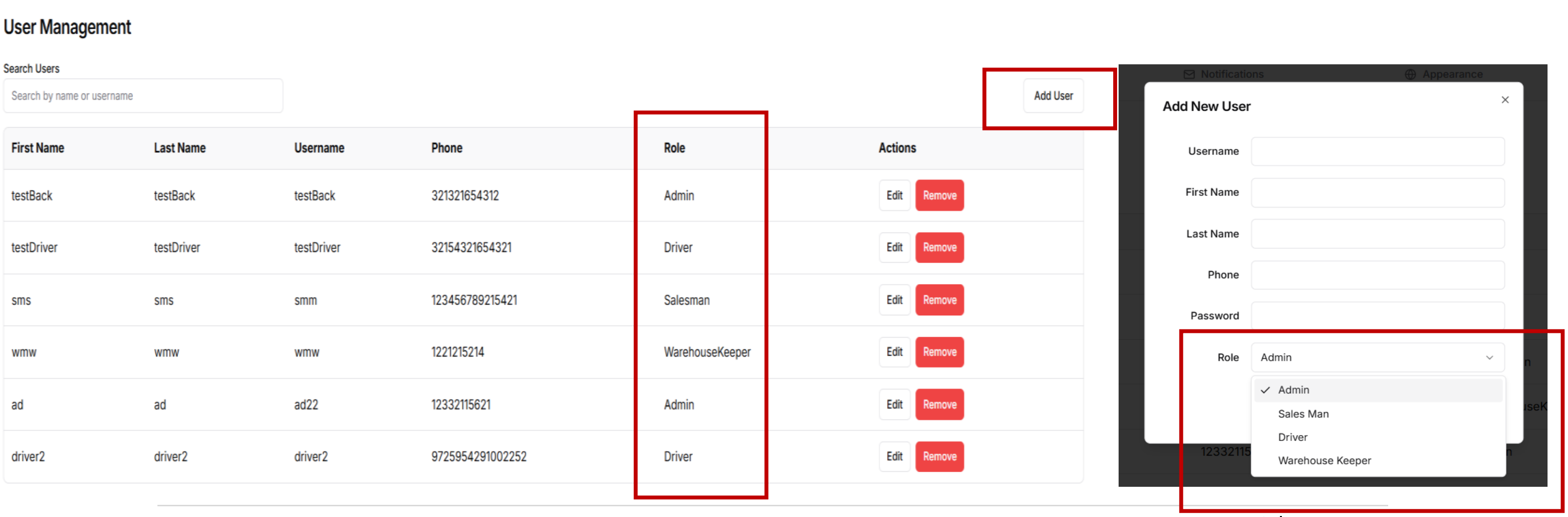

E. Security Settings: Focuses on account security and access control.

- **Change Password**:
  - Current password verification
  - New password entry and confirmation
- **Two-Factor Authentication (2FA)**:
  - Adds an extra security layer via OTP or authenticator apps




**F. User Management (Admin Only):** Provides administrators with tools to manage system users.
**Capabilities:**

- Search users by name or username
- View a list of registered users
- Add new users and assign roles



## 10. Notification System

The application includes a **real-time, role-based notification system** built using **Firebase Firestore**. This system ensures seamless communication and workflow coordination between users with different roles (Admin, Sales Rep, Warehouse Manager, Driver, etc.). It plays a crucial role in improving system responsiveness and user awareness.

### Technical Implementation

The notification system is implemented in the `notificationService` module using Firebase Firestore. Key operations include:

#### 1. Add Notification

- `addNotification`: Adds a new notification with automatic timestamp generation.
- Used when a new action requires informing a specific user.

#### 2. Real-Time Listener

- `listenToUnreadNotifications`: Establishes a real-time listener on unread notifications for a specific user.
- Automatically updates the UI when new notifications arrive using Firebase's `onSnapshot`.

#### 3. Mark as Read

- `markAsRead`: Marks a specific notification as read.
- `markAllAsRead`: Marks **all unread notifications** for a user as read in bulk using Firebase's `writeBatch`.

#### 4. Fetch Notifications

- `getAllNotifications`: Retrieves all notifications for a specific user, ordered by timestamp.
- `getUnreadCount`: Returns the count of unread notifications for badge display.

#### 5. Delete Notifications

- `deleteNotification`: Deletes a specific notification.
- `deleteReadNotifications`: Deletes **all read notifications** for a user using batch deletion.

**Role-Based Notification Scenarios**

The notification system facilitates communication and event tracking across the entire business workflow. Below are real-world examples of how notifications are triggered between roles:

**Admin → Sales Representative**

- **Action:** Admin assigns a new client to a representative.
- **Notification:** "A new client has been assigned to you."

**Sales Representative → Warehouse Manager**

- **Action:** Sales rep places a new product order.
- **Notification:** "A new order has been created and is awaiting fulfillment."

**Sales Representative → Admin**

- **Action:** Rep registers a new client or records a payment.
- **Notification:** "New client registered" or "A new payment has been recorded."

**Warehouse Manager → Driver**

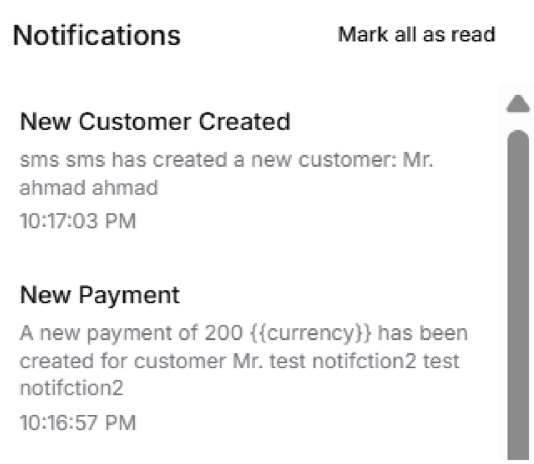
- **Action:** A delivery is assigned to a driver.
- **Notification:** "A new delivery task has been assigned to you."

**Driver → Sales Man & Warehouse Manager**

- **Action:** Driver completes a delivery.
- **Notification:** "The delivery has been successfully completed."

**Any User → Any User**

- **Action:** A user sends a direct message to another user.
- **Notification:** "You received a new message from [sender]."



## 11. Chat System

The Company Chat System is a real-time communication platform designed specifically for internal use within organizations. Built with React and Firebase, it allows company employees to exchange messages instantly, ensuring smooth and secure communication. The system supports essential features such as user authentication, unread message tracking, push notifications, and a user-friendly interface. This project enhances workplace collaboration by providing a fast and efficient way for team members to stay connected.

1. ***User Interface (UI)***

- Displays a list of authorized users within the same company (excluding the current user).
- Chat window showing real-time messages between users.
- Messages are visually distinguished based on sender or receiver.
- Unread messages are highlighted with visual indicators.
- Unread message count is shown next to each user in the user list.

2. ***Key Features***

***Sending & Receiving Messages***

- Messages are grouped using a `conversationId` generated by combining sender and receiver IDs.
- Messages are stored in the `messages` collection.
- A `conversations` document is created or updated whenever a message is sent.

***Unread Message Tracking***

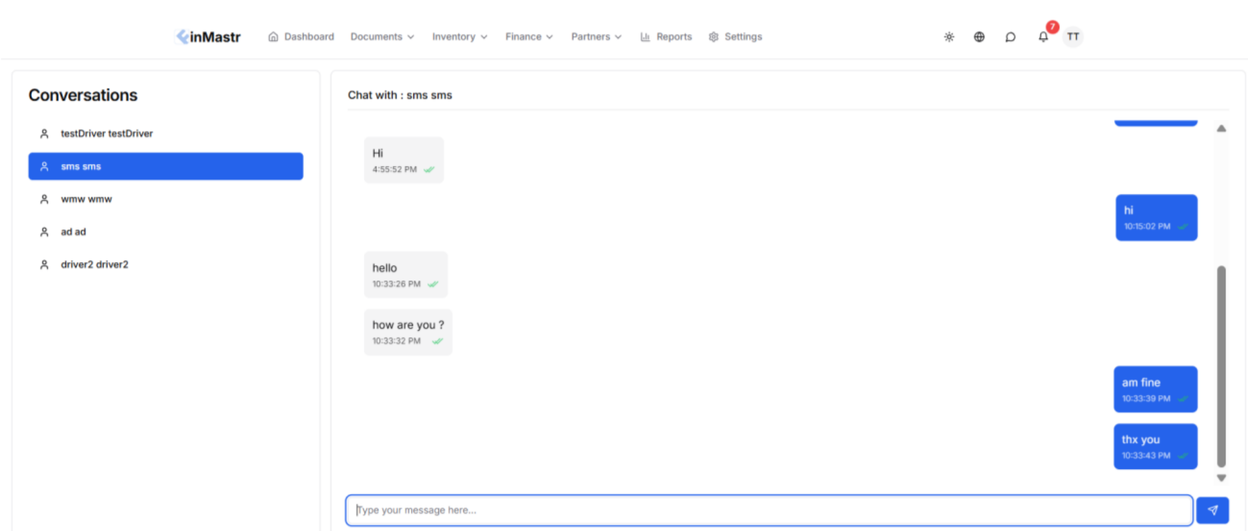
- The UI shows the number of unread messages next to each user.
- When a chat is opened, all messages from that user are marked as read.

***Real-Time Updates***

- `onSnapshot` listeners ensure real-time updates for messages and conversations.

***Notifications***

- A custom `notificationService` is triggered when a message is sent, alerting the receiver.



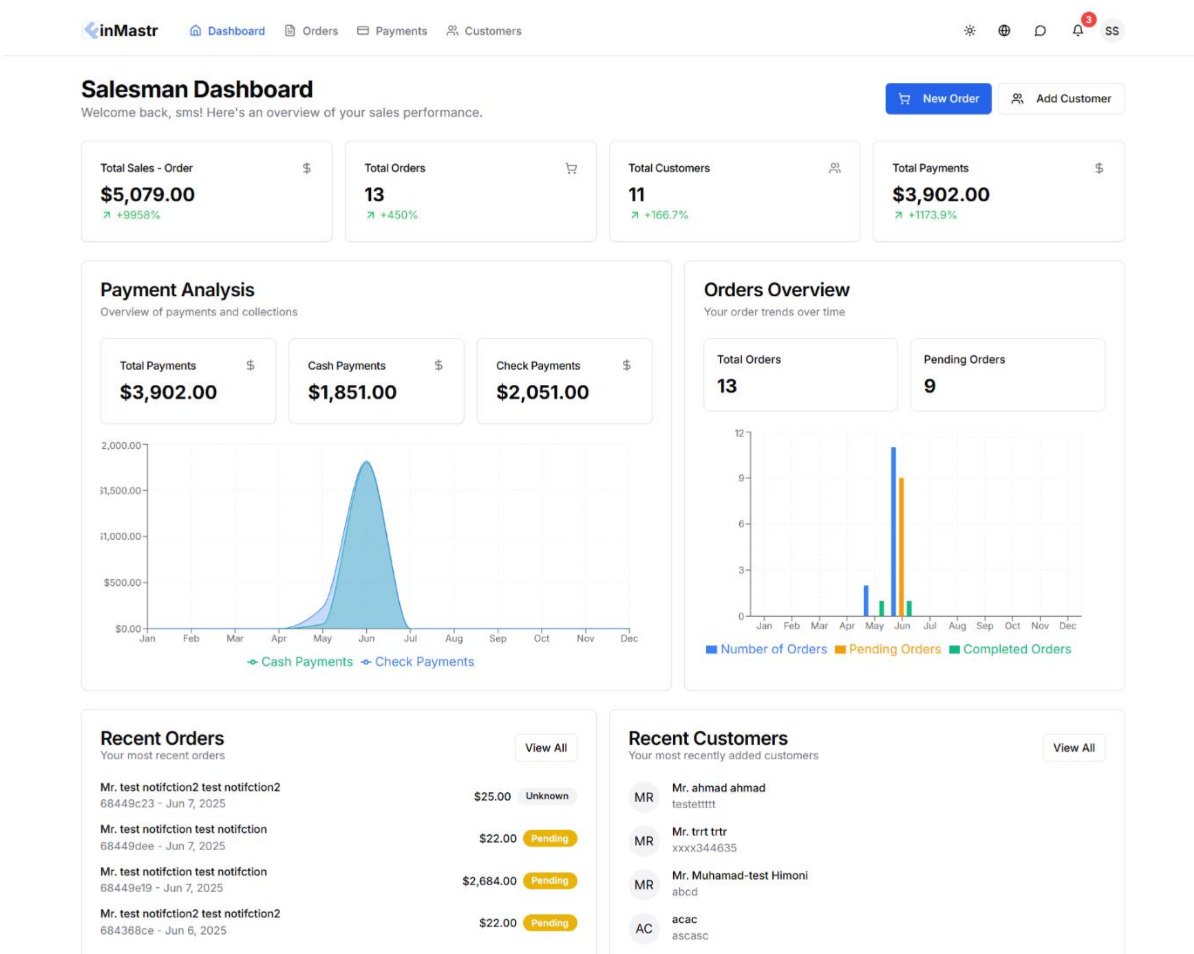
## Seles Man user:

The **Salesman User** is provided with both a **web interface** and a **mobile application**, each tailored to support the daily operations of field sales representatives. These interfaces allow the salesman to manage orders, payments, and customer data efficiently, whether in the office or on the go.

## Web Interface

The **web application** was developed using the **MERN stack** (MongoDB, Express.js, React.js, Node.js). This stack enables real-time data handling, responsive UI, and fast interactions with backend services. The web dashboard serves as the central hub for the salesman, offering access to various modules such as orders, payments, and customers. The interface is designed to be clean, intuitive, and focused on sales productivity.

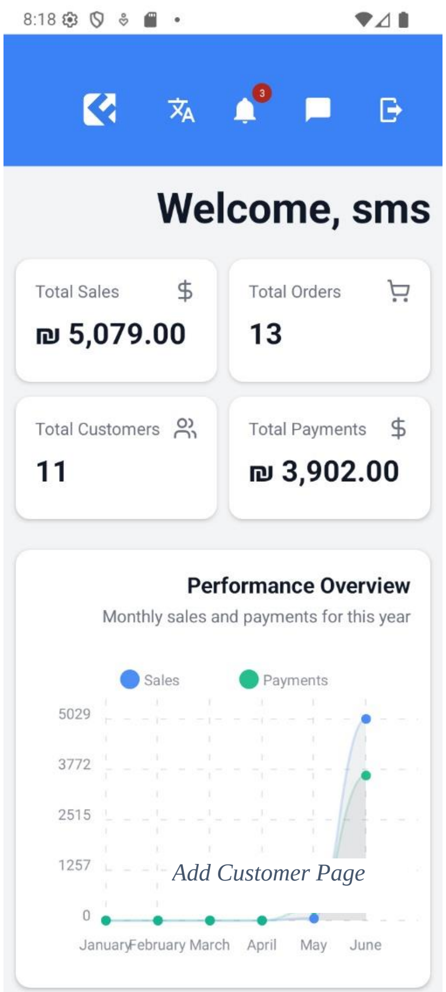
The dashboard provides a comprehensive overview of the salesman's performance and recent activity. It includes visual indicators for order statuses, payment summaries, and customer engagement. From this central page, the user can quickly navigate to other functional areas to perform tasks such as creating new orders, registering customers, or processing payments.



## Mobile Interface

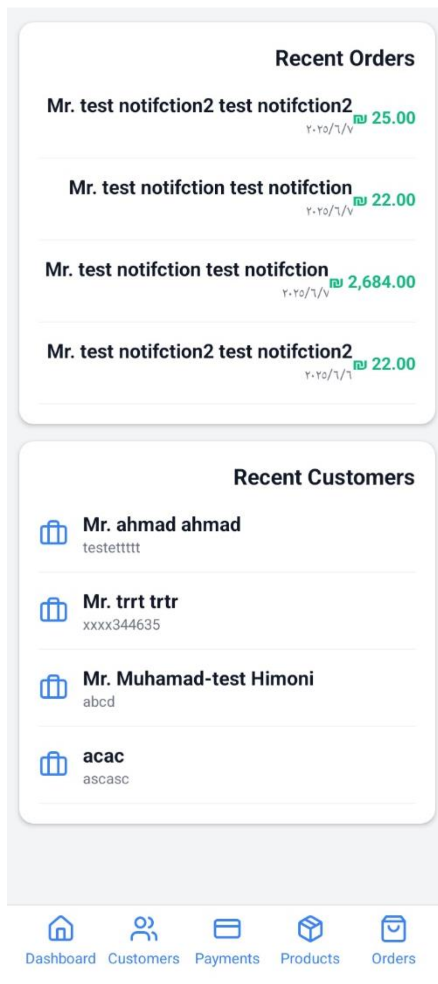
To complement the web experience, a **mobile application** was developed using **React Native**. This allows for cross-platform support on both iOS and Android with a single codebase, ensuring consistent behavior across devices. The mobile app mirrors the core functionality of the web version, with optimizations for smaller screens and touch interactions.

Salesmen can access their dashboards, view order and payment statuses, and interact with customer records directly from their mobile devices. This flexibility enhances field efficiency and ensures that all sales activities are synced with the central system in real time.

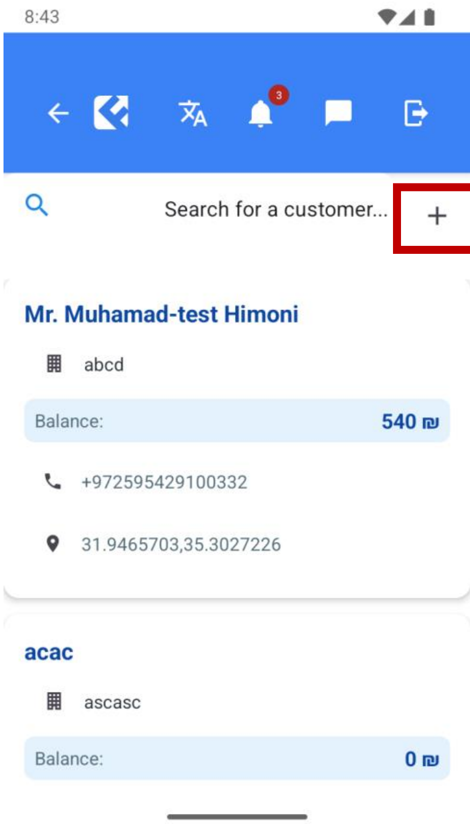


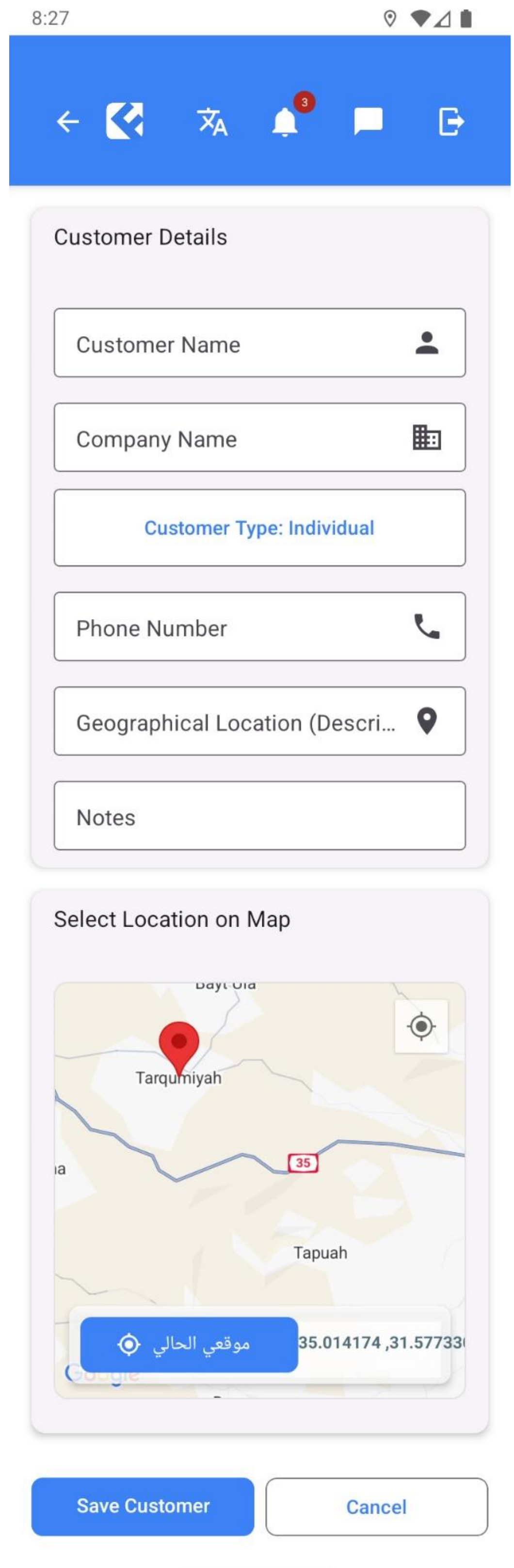

*Add Customer Page*

*Customer Page*

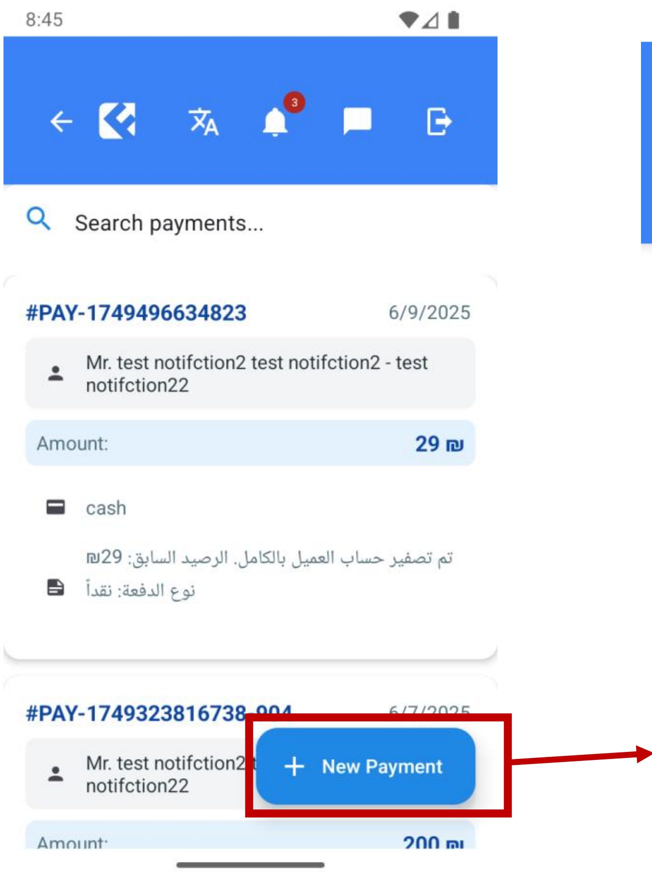


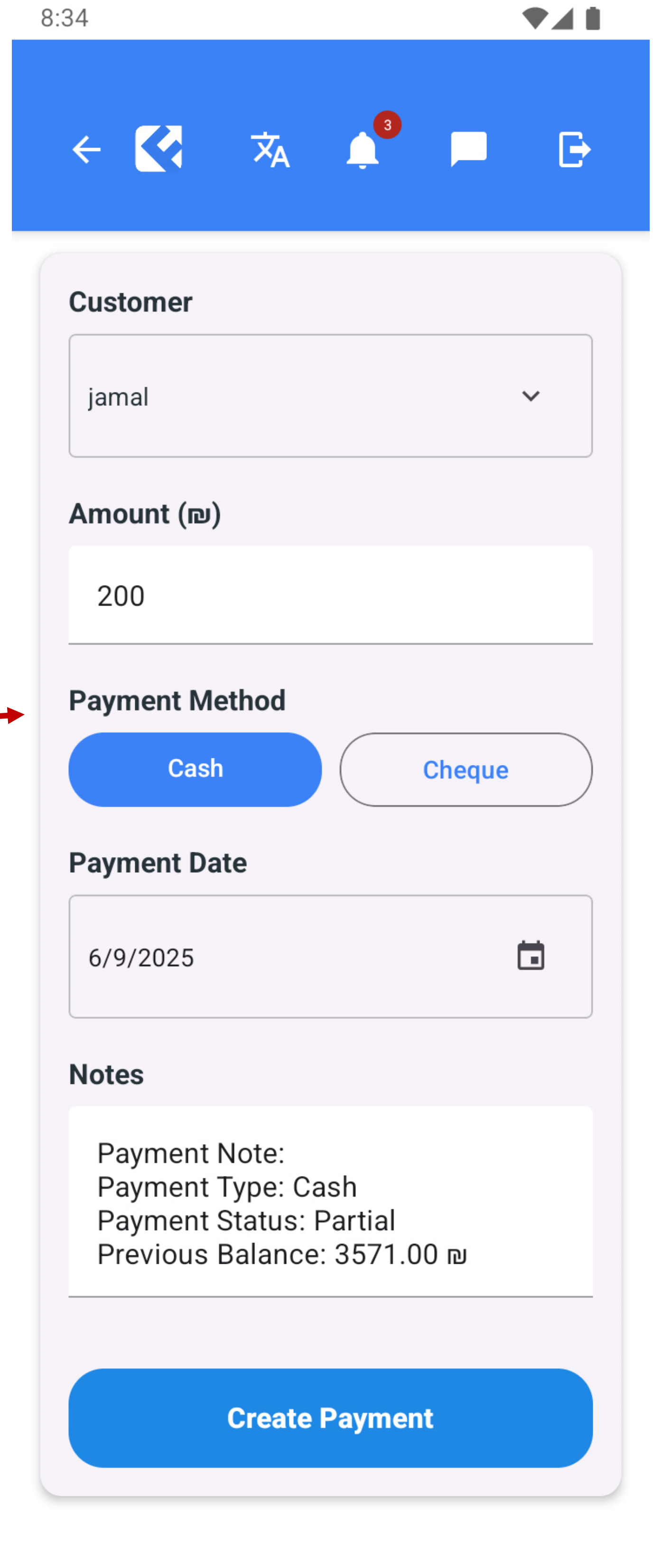
**Customer Page:**



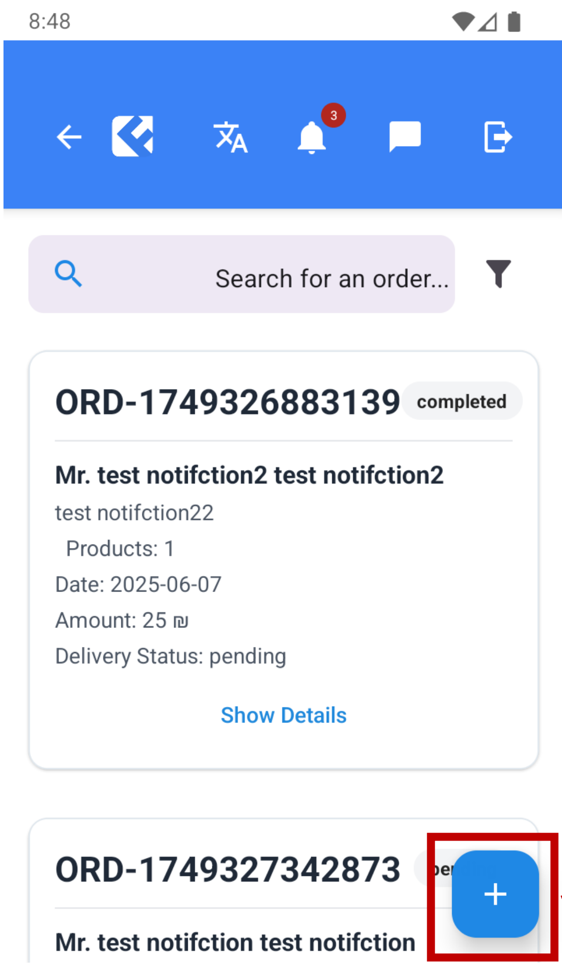



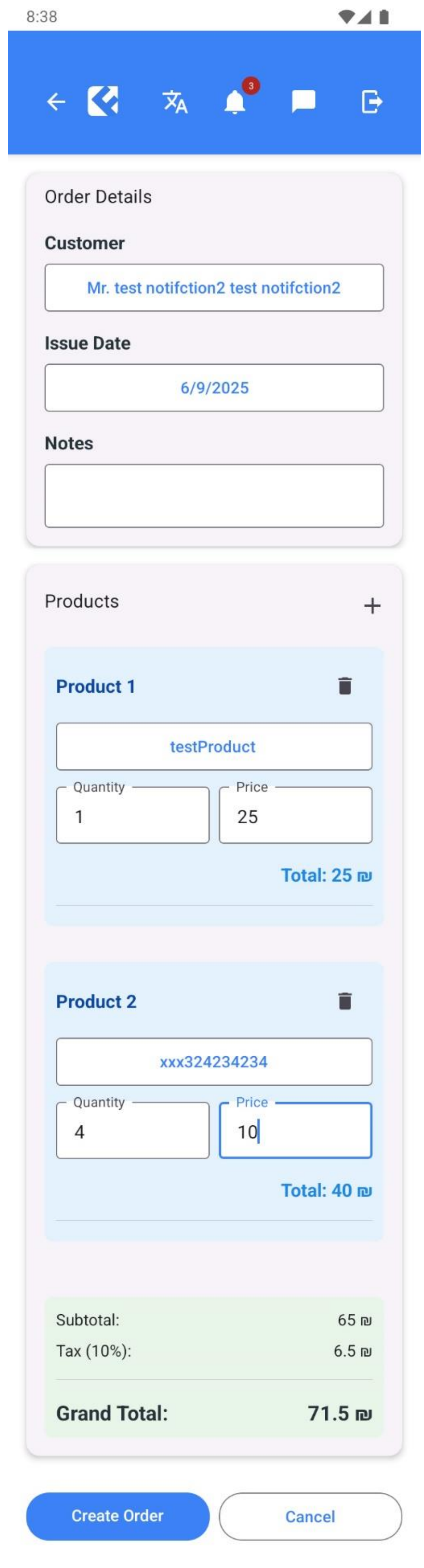
**Payment Page:**



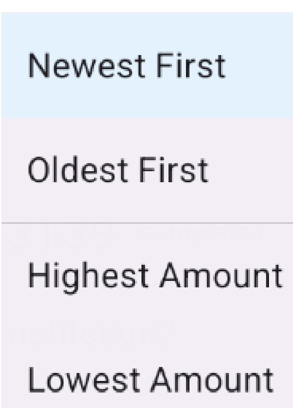



## Order Page:




## Filters:



## Navigation & Functional Modules

Both interfaces include access to essential modules that cover the core business operations of the salesman:

- **Dashboard**: A quick overview of performance indicators, recent orders, and customer activity.
- **Orders**: Interface for viewing and managing current and past orders, with support for creating new orders.
- **Payments**: Allows users to review payment history, categorize payments by type, and track outstanding collections.
- **Customers**: Enables adding, editing, and viewing customer profiles and associated sales records.
- **Product**


*Navigation Mobile*

**Note:**

- **On the customer page, only the customers the Salesman is responsible appear.**
- **On the orders page and the payments page, only the transactions that the salesman records appear.**

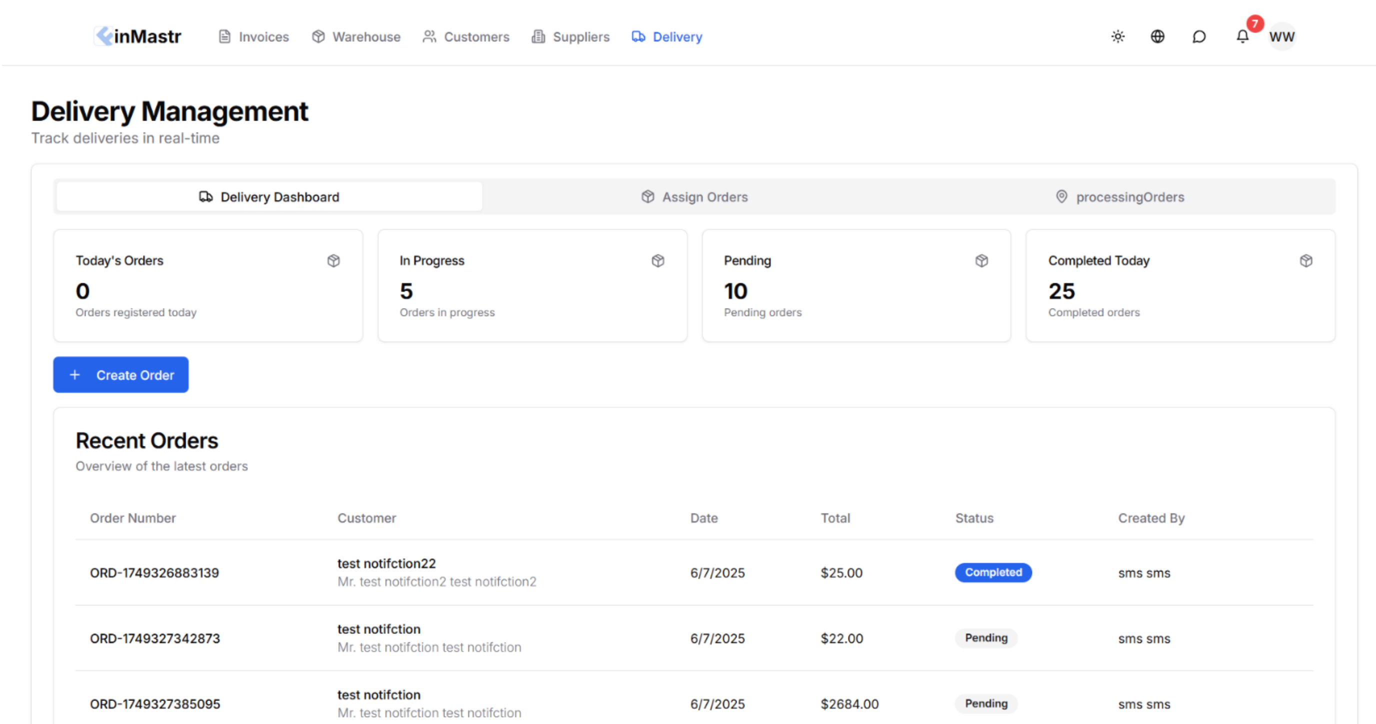
## Warehouse Keeper user:

The **Warehouse Keeper** user plays a central role in managing warehouse operations and coordinating delivery logistics. This user is responsible for overseeing the flow of goods in and out of the warehouse, assigning delivery tasks to drivers, and maintaining accurate records of inventory and transactions.

The Warehouse Keeper has access to the following modules:

- **Invoices**: Create and manage sales invoices, ensuring all outgoing goods are properly documented and billed.
- **Warehouse**: Monitor stock levels, receive incoming goods, and register items into the system. This includes the ability to track inventory changes and validate product entries.
    - **Warehouse Overviews**
    - **Product**
    - **Order + Create Order**
- **Customers**: View and manage customer data to ensure accurate delivery and invoicing information.
- **Suppliers**: Manage supplier records and link them to purchase transactions.
- **Delivery**: Assign orders to drivers, track delivery progress, and confirm order status updates in real time.

This interface is available both on the web platforms, ensuring the Warehouse Keeper can operate efficiently whether on-site or remotely. The system is designed to streamline warehouse tasks, reduce manual errors, and maintain an up-to-date inventory and delivery status.



## Driver user:

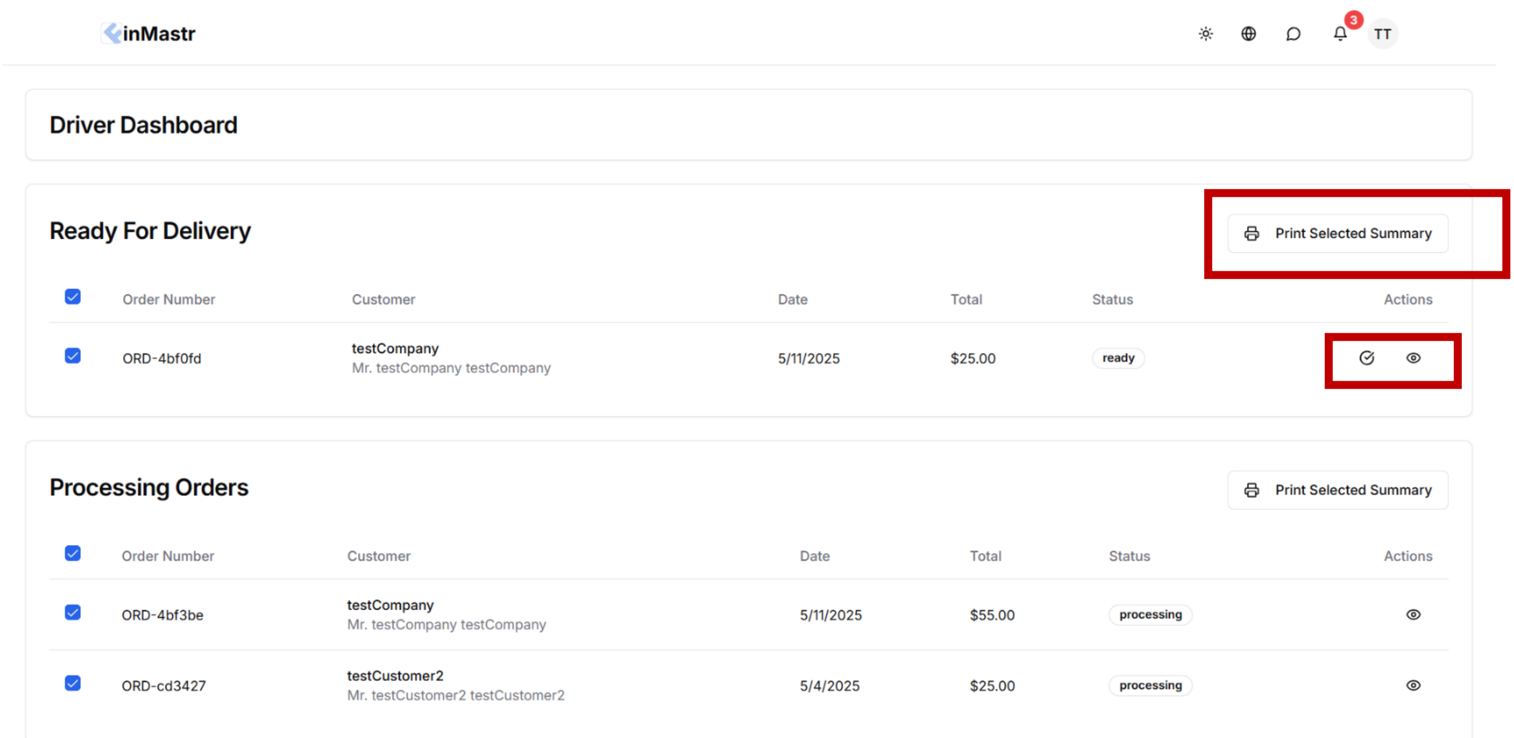
The **Driver Dashboard** is designed to assist delivery personnel in managing and completing their assigned tasks efficiently. This interface is available on both **web and mobile platforms**, allowing drivers to access their deliveries whether they are in the office or on the move.

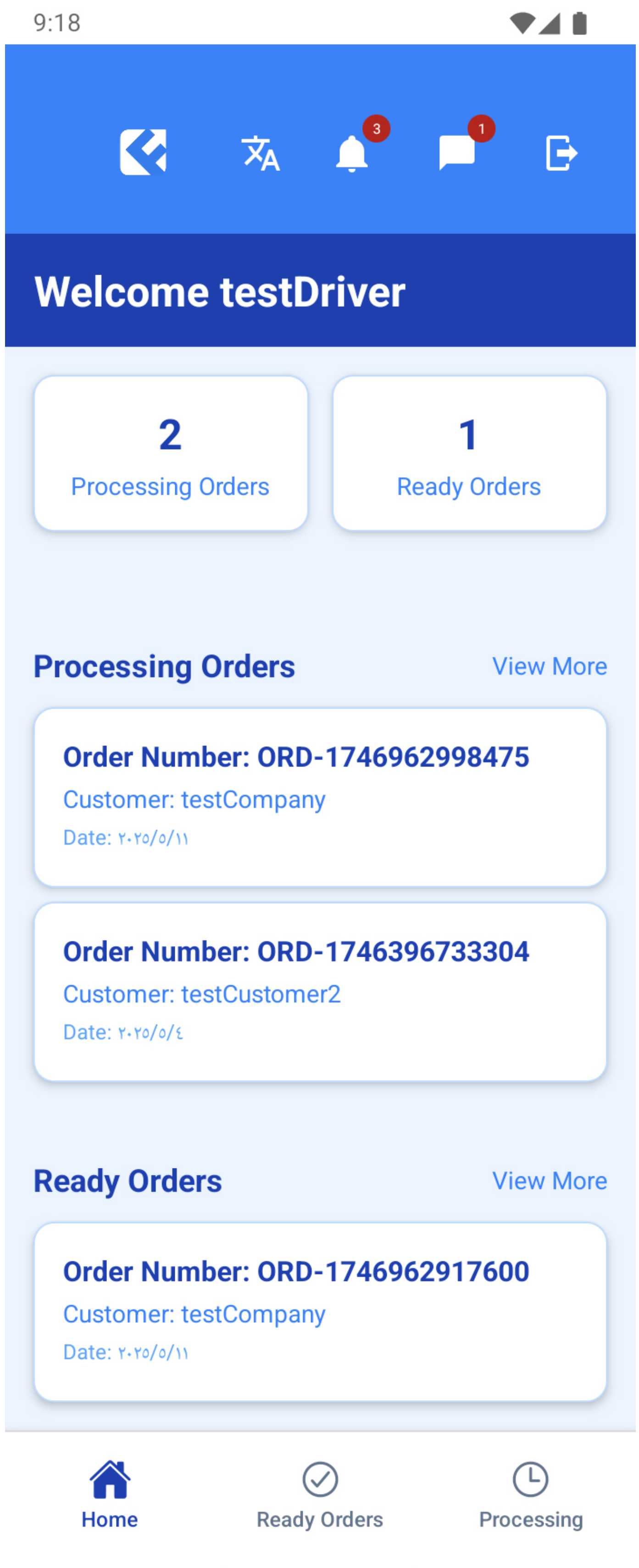
The dashboard provides two key sections:

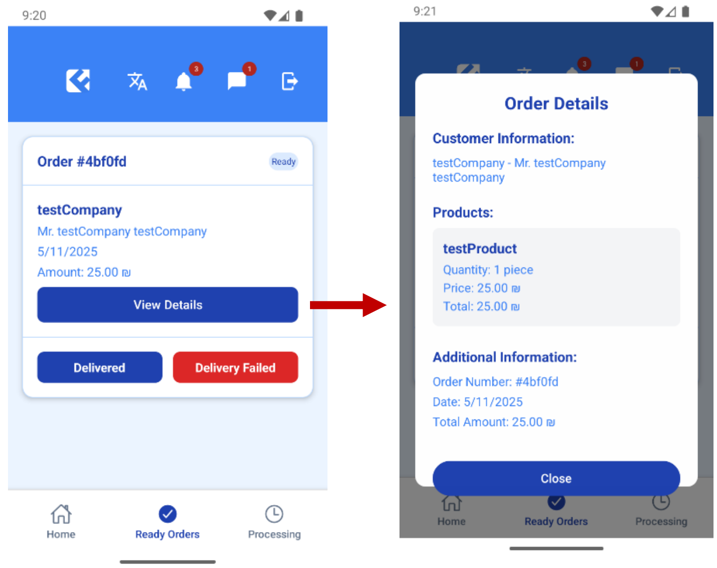
- **Ready for Delivery**: Lists all orders that are prepared and awaiting delivery. Each order is displayed in a table format containing:
  **Order Number**, **Customer**, **Date**, **Total**, **Status**, and **Actions**.
  Drivers can:
  - View full order details and item breakdowns.
  - Print the order for delivery reference.
  - Confirm that the customer has received the order, which updates the status from "Ready" to "Completed".
- **Processing Orders**: Displays ongoing deliveries that are currently in progress.

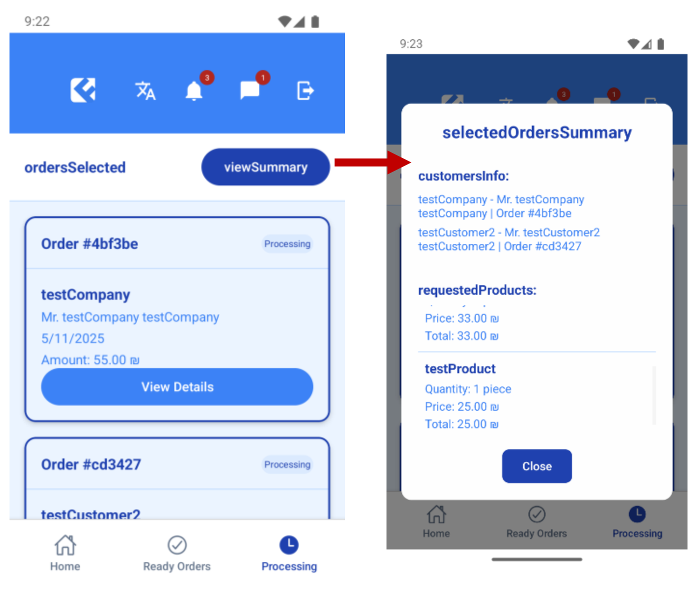
***Key Features:***

- **Multi-Order Selection & Summary Print**: Drivers can select multiple orders simultaneously and generate a consolidated delivery summary that includes customer names and total requested items. This summary can be printed for logistical use.
- **Mobile & Web Compatibility**: The dashboard is responsive and accessible through both web browsers and mobile apps (built with React Native), ensuring usability in different environments.




9:18
Welcome testDriver
2
Processing Orders
1
Ready Orders
Processing Orders
View More
Order Number: ORD-1746962998475
Customer: testCompany
Date: ٢٠٢٥/٥/١١
Order Number: ORD-1746396733304
Customer: testCustomer2
Date: ٢٠٢٥/٥/٤
Ready Orders
View More
Order Number: ORD-1746962917600
Customer: testCompany
Date: ٢٠٢٥/٥/١١
Home
Ready Orders
Processing


9:20
Order #4bf0fd
Ready
testCompany
Mr. testCompany testCompany
5/11/2025
Amount: 25.00 ₪
View Details
Delivered
Delivery Failed
Home
Ready Orders
Processing
9:21
Order Details
Customer Information:
testCompany - Mr. testCompany testCompany
Products:
testProduct
Quantity: 1 piece
Price: 25.00 ₪
Total: 25.00 ₪
Additional Information:
Order Number: #4bf0fd
Date: 5/11/2025
Total Amount: 25.00 ₪
Close


9:22
ordersSelected
viewSummary
Order #4bf3be
Processing
testCompany
Mr. testCompany testCompany
5/11/2025
Amount: 55.00 ₪
View Details
Order #cd3427
Processing
testCustomer2
Home
Ready Orders
Processing
9:23
selectedOrdersSummary
customersInfo:
testCompany - Mr. testCompany testCompany | Order #4bf3be
testCustomer2 - Mr. testCustomer2 testCustomer2 | Order #cd3427
requestedProducts:
Price: 33.00 ₪
Total: 33.00 ₪
testProduct
Quantity: 1 piece
Price: 25.00 ₪
Total: 25.00 ₪
Close

## Super Admin user:

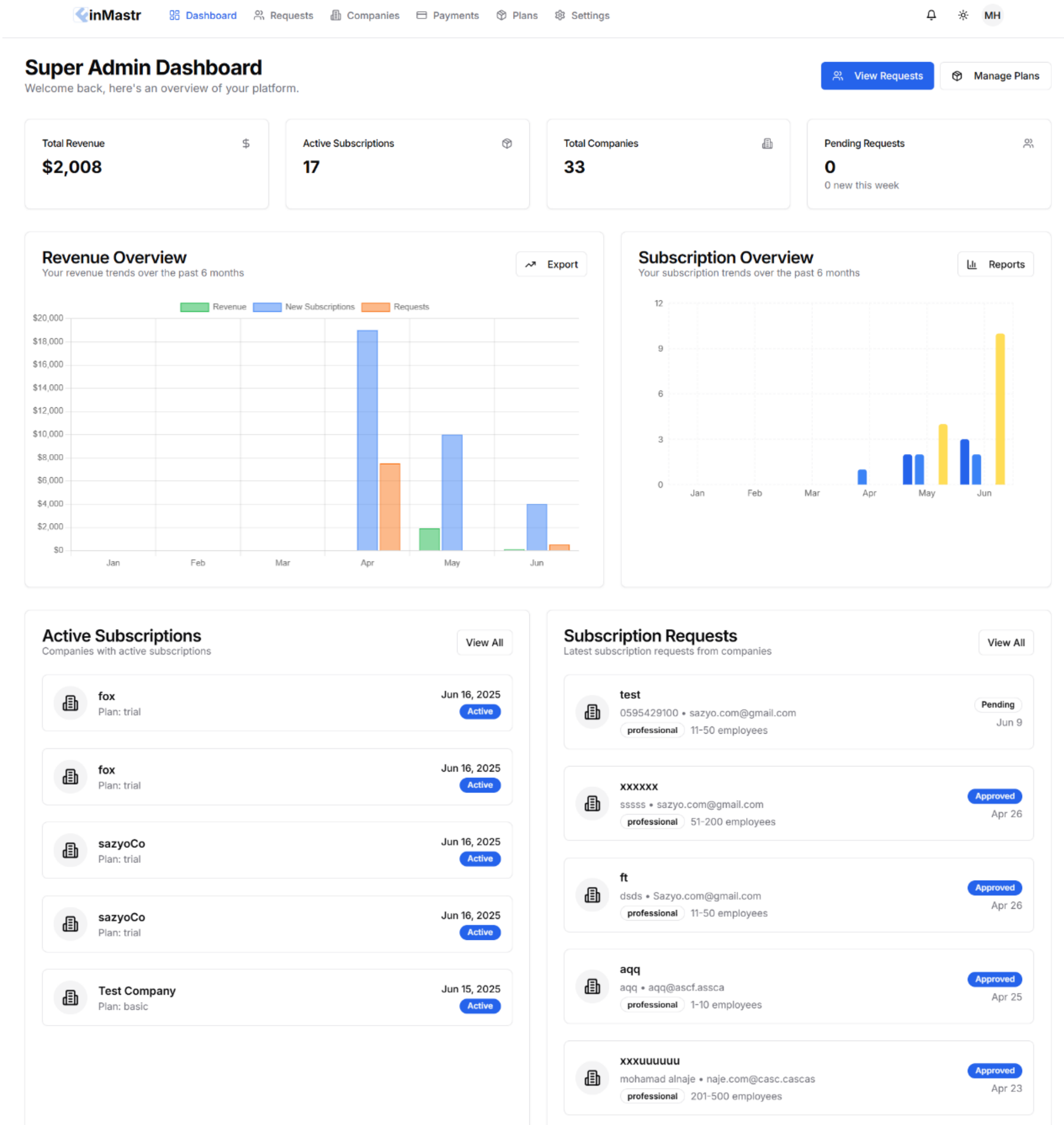
The **Super Admin** is the highest-level user in the system, responsible for overseeing the entire platform's operations. This user has access to a dedicated interface that provides full administrative control over registered companies, their subscription plans, users, and payments.

***Role and Responsibilities***

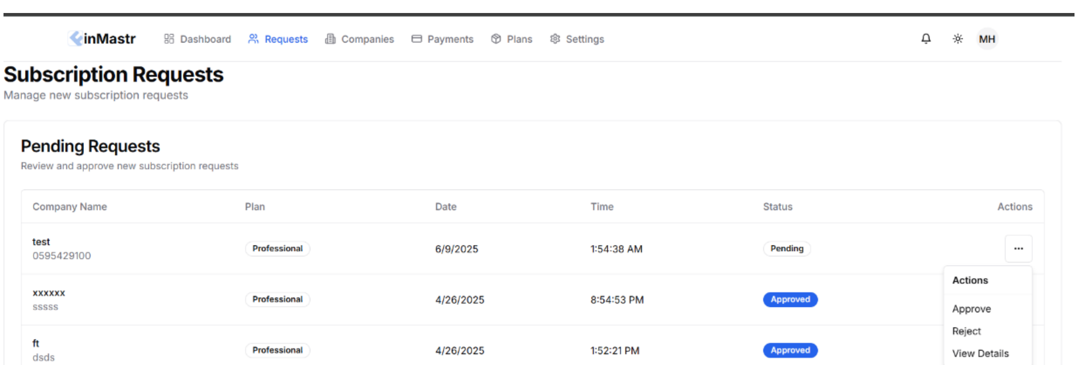
- **Platform Oversight**: Monitor platform performance through a dedicated dashboard displaying key metrics such as total revenue, number of companies, active subscriptions, and pending requests.
- **Company Management**: View, add, edit, and delete registered companies. Access detailed information about each company's contact info, users, and subscription status.
- **User Management**: Manage all users across the platform, including the ability to create and assign new Super Admins, giving them equal access and permissions.
- **Request Handling**: Approve or reject new subscription requests submitted by companies, with options to view request details before taking action.
- **Payment Tracking**: Review and manage all subscription-related payments, including billing cycles, plans, payment status, and subscription durations.
- **Plan Management**: Define and update available subscription plans that companies can choose from.
- **System Settings**: Configure general settings, appearance themes, notification rules, and security policies from a centralized settings page.

***Key Pages Accessible by Super Admin***

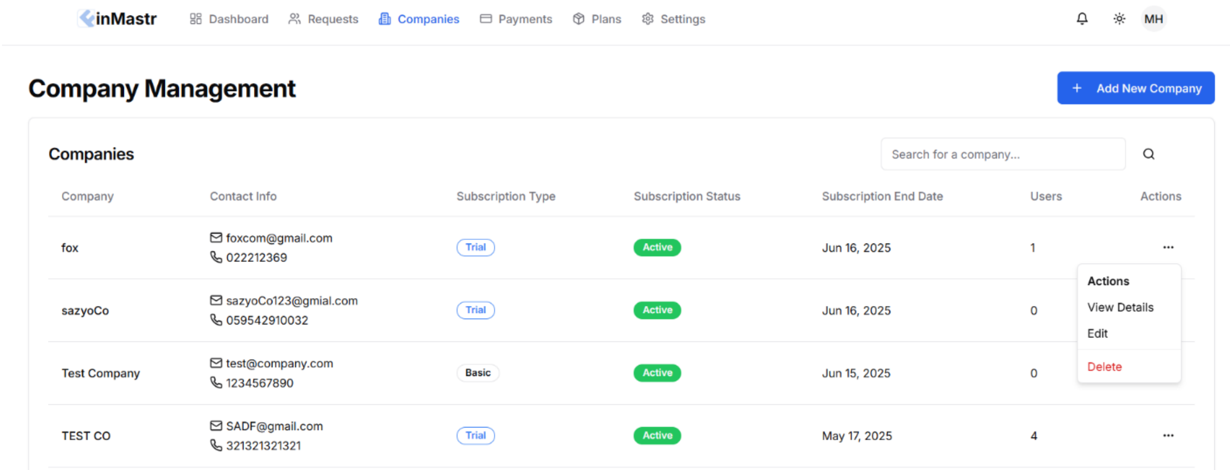
- **Dashboard**: Overview of revenue, subscription trends, and system statistics.
- **Requests**: View and manage pending subscription requests.
- **Companies**: Full access to company profiles, users, and subscriptions.
- **Payments**: Track all subscription payments made by companies.
- **Plans**: Create, update, or deactivate subscription plans.
- **Settings**: Configure platform-wide settings including users, appearance, notifications, and security.



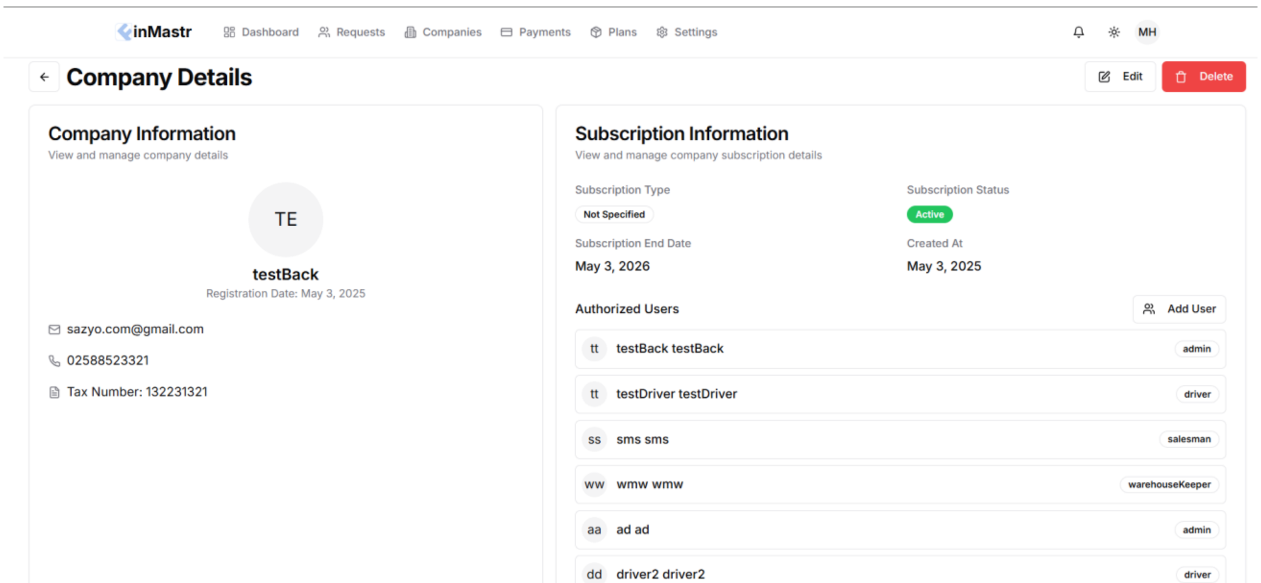

*Super Admin Dashboard*



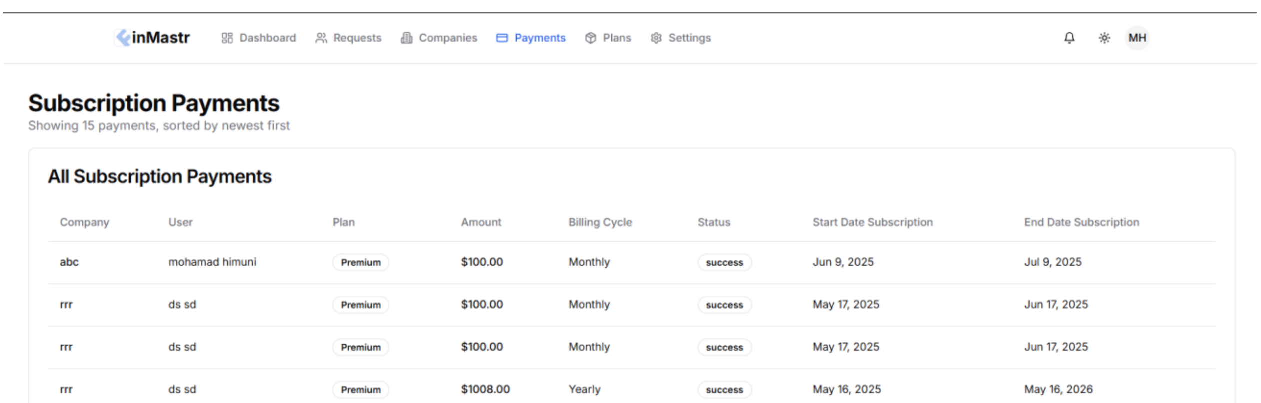

*Subscription Requests*




*Company Management*




*Company Details*

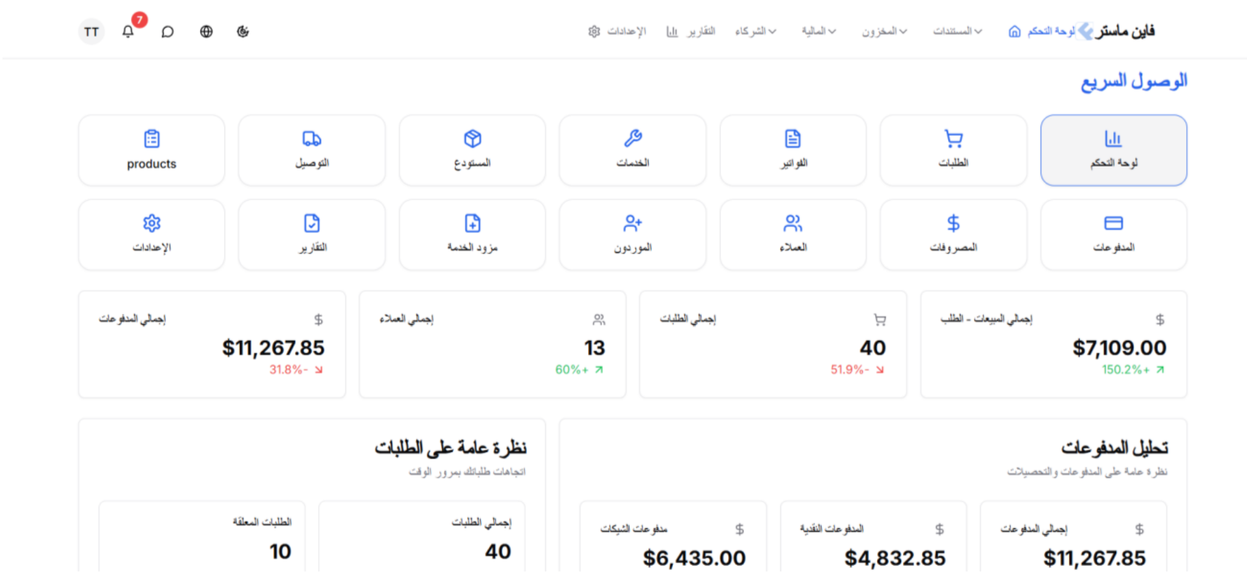



*Subscription Payments*

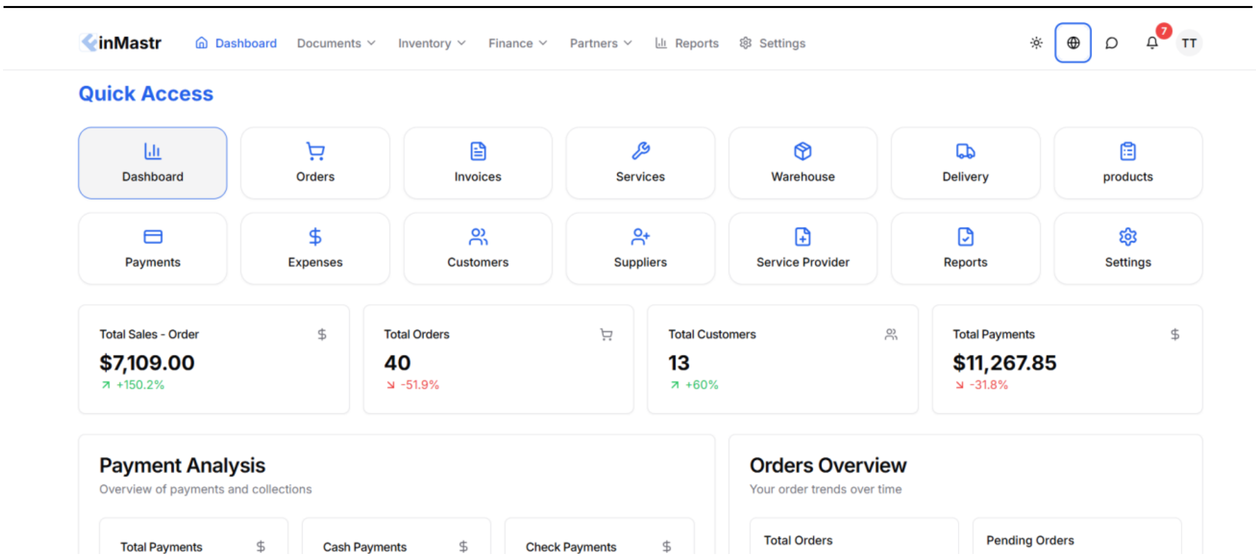
## Multi-language Support

Both the **web** and **mobile** versions of the application are fully multilingual, supporting **Arabic**, **English**, and **Hebrew**. This feature ensures accessibility and usability for users from different linguistic backgrounds across the region.

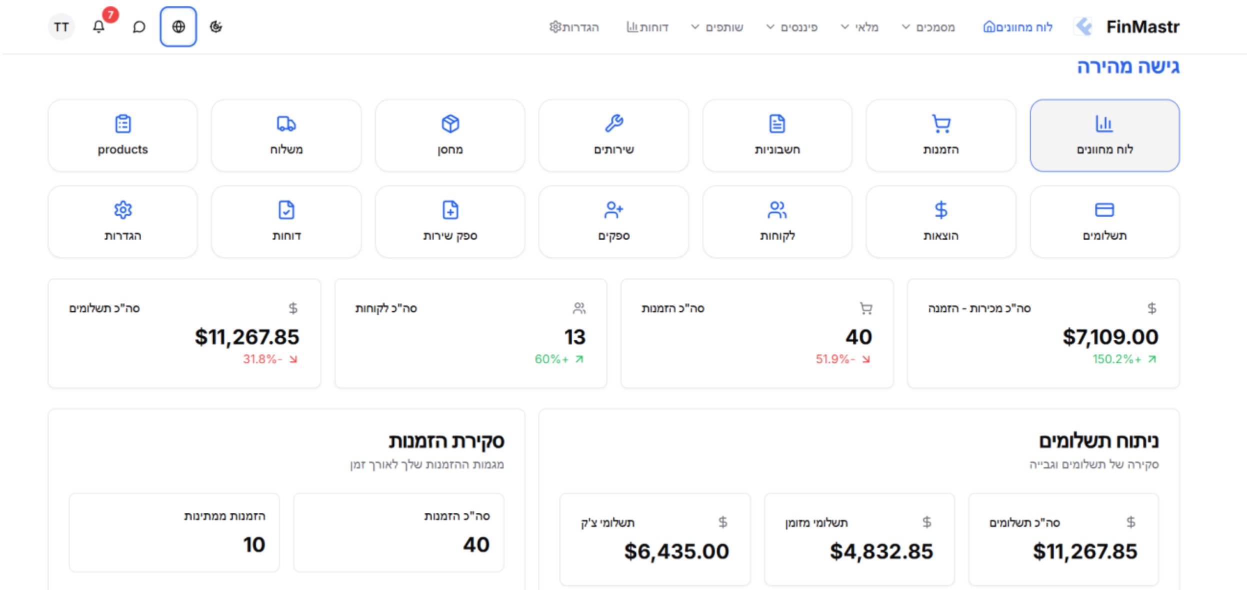
The multilingual support was implemented using a **dictionary-based approach**, where a separate dictionary (language file) is maintained for each supported language. Each dictionary contains approximately **3,000 key-value pairs**, covering all interface elements such as labels, buttons, messages, errors, and system notifications.



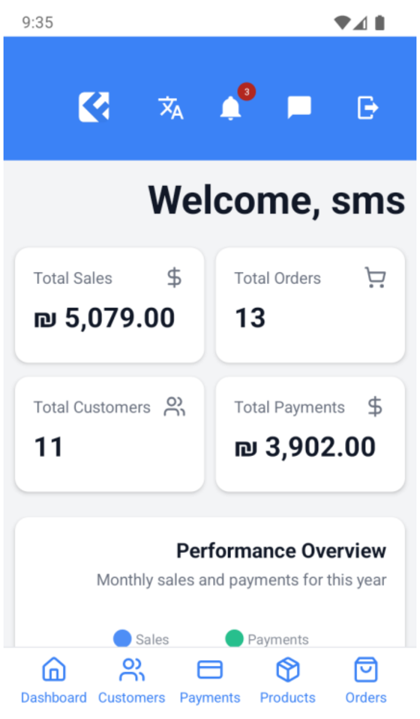

*Arabic*



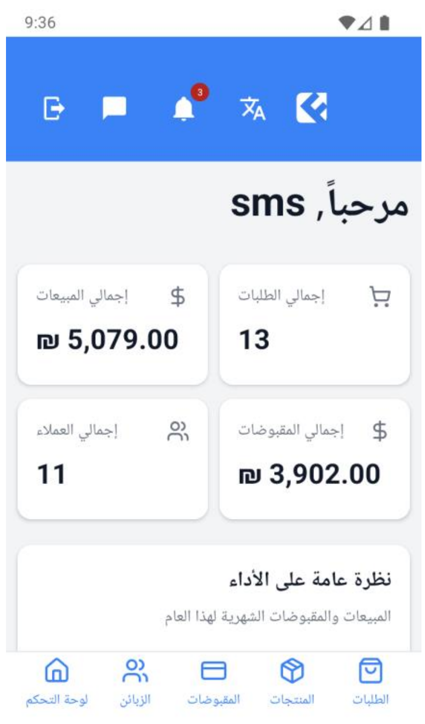

*English*



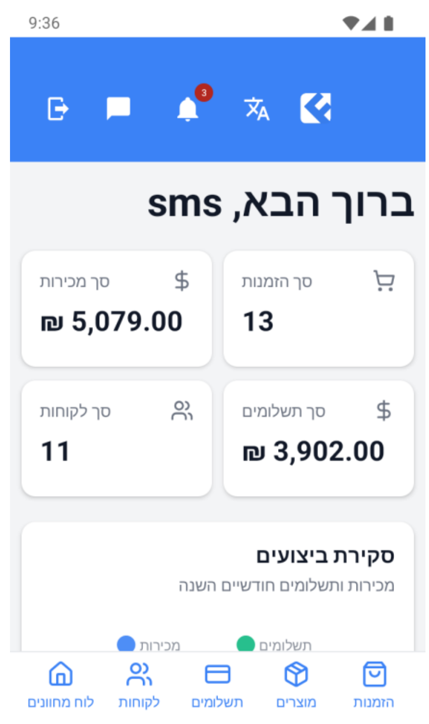

*Hebrew*




*English*



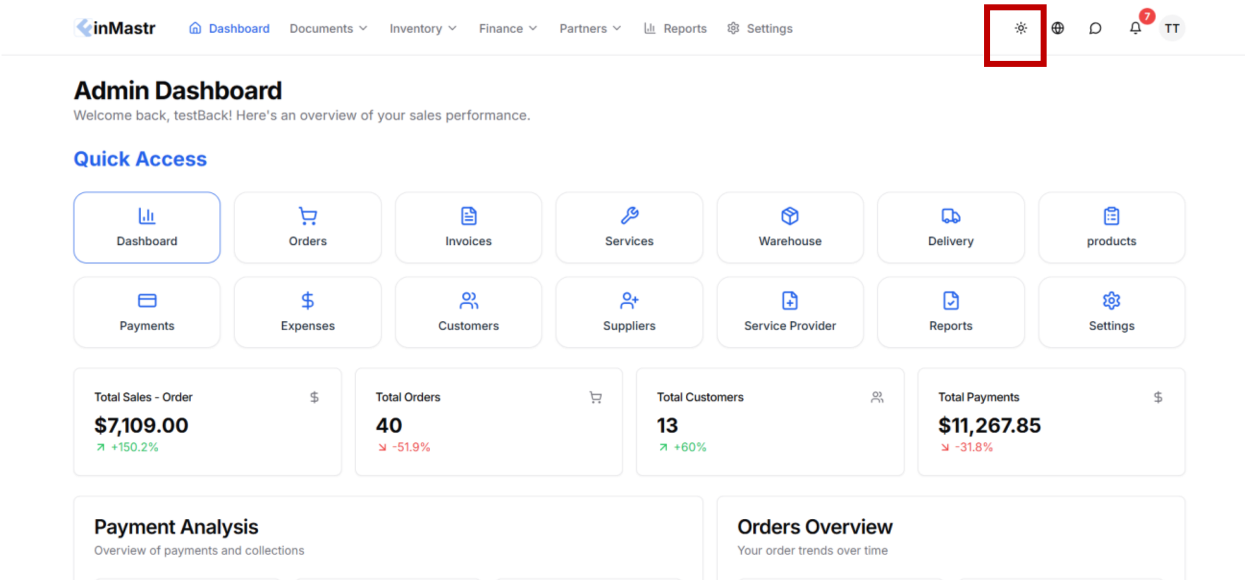

*Arabic*



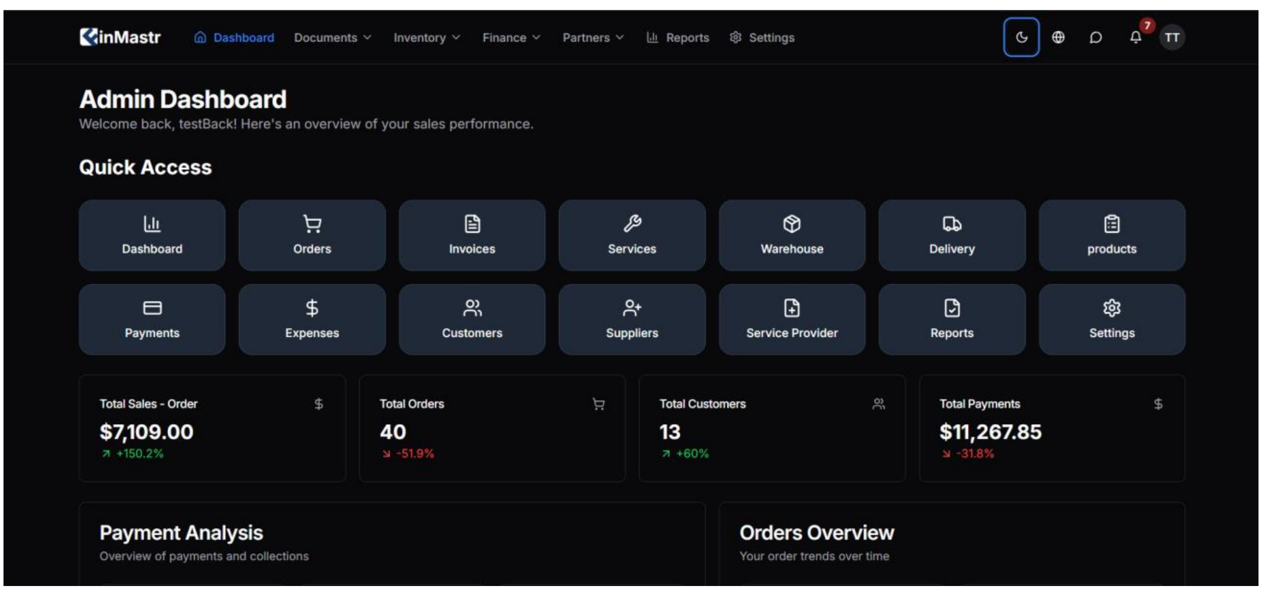

*Hebrew*

## Theme Support: Light & Dark Modes

To enhance user experience and accessibility, the application supports both **light mode** and **dark mode** across the **web**. Users can switch between themes based on their preferences or device settings.




*Light mode*




*Dark mode*

## 5.1 Challenges

Throughout the development and deployment of Finmaster, several key challenges emerged:

- **Integration Complexity:** Connecting Firebase real-time updates with RESTful backend services required thoughtful architecture to avoid data duplication or inconsistencies.
- **Cross-Platform Consistency:** Ensuring seamless synchronization and identical experiences between web and mobile platforms was difficult, especially with complex financial workflows and multi-user role scenarios.
- **Security and Compliance:** Implementing industry-standard encryption and secure role-based access control to handle financial data added complexity to the backend logic and user management layers.
- **Testing and Debugging:** Verifying complex financial flows, especially those involving multiple currencies and automated calculations, was time-intensive and required thorough unit and integration testing.

## 5.2User Feedback

To evaluate the system's usability and performance, a group of users from varying business sectors was invited to test the application. Their feedback was essential in shaping the final structure.

**Positive Feedback:**

- Users appreciated the **intuitive UI** and ease of creating invoices and managing reports.
- Business owners found the **multi-role access** and **real-time dashboard** highly effective for team collaboration.

**Constructive Feedback:**

- Some users requested an **offline mode** for the mobile app in areas with poor connectivity.
- Auditors suggested adding **audit trails** and **approval logs** for high-value transactions.
- A few users requested deeper integration with **external banking APIs** for direct import of transactions.

# Chapter 6: Conclusion and Recommendations

## 6.1 Conclusion

In conclusion, the developed system offers a comprehensive and user-friendly platform for managing sales, warehouse operations, drivers' tasks, and overall business workflows. With dedicated dashboards tailored for different roles such as Salesmen, Warehouse Keepers, and Drivers, the system ensures each user accesses the exact tools and data they need to perform efficiently. The use of modern technologies—**MERN stack** for the web and **React Native** for the mobile app—provides a scalable and responsive user experience across platforms.

Multilingual support (Arabic, English, and Hebrew) and theme switching (light/dark modes) further enhance accessibility and usability for a diverse user base. The integration of features like bulk order processing, invoice management, real-time status updates, and customizable summaries empowers users to streamline operations and reduce manual overhead.

## 6.2 Recommendations and Future Enhancements

While Fine master has achieved its goals, several potential areas for future enhancement have been identified to increase its functionality and competitiveness:

- **Bank API Integration:** Enable secure syncing with banks for importing transactions directly from bank feeds.
- **AI-Driven Budget Suggestions:** Implement a budgeting assistant that provides financial planning advice based on income, spending trends, and business goals.
- **Offline Mode (Mobile):** Add caching mechanisms for mobile users in low-connectivity environments to manage data offline and sync automatically when reconnected.
- **Audit Trail System:** Include a full audit log for tracking changes in sensitive financial data, improving transparency for accountants and auditors.
- **Third-Party Integrations:** Offer integrations with external platforms such as payroll systems, inventory tools, or tax reporting services.
- **Machine Learning Personalization:** Enhance user experience by tailoring dashboards and suggestions using ML models based on behavior.

# Chapter 7: References


## Technologies & Frameworks

1. Next.js Documentation
2. Express.js Official Docs
3. MongoDB Developer Docs
4. React Native Documentation
5. Firebase Documentation (Auth, Realtime Database)
6. JWT Authentication Guide
7. Mongoose ODM Docs

## UI/UX & Design Principles

11. Material UI Guidelines
12. Responsive Web Design – Mozilla Developer Network

13. Tellwind css

## General Tools

13. Postman API Testing Tool
14. MongoDB Compass
15. Visual Studio Code
16. Git & GitHub Documentation
17. **Docker Documentation**: Resources for containerizing applications to ensure consistency across environments.
18. **Jest Documentation**: Guide for setting up and running tests with the Jest testing framework.
19. **Insomnia API Client**: Alternative to Postman for testing and managing APIs.

## Learning Resources

17. Academind React Native Course (YouTube)
18. Traversy Media - MERN Stack Tutorials
19. Fireship - Firebase in 100 Seconds

FreeCodeCamp.org - REST API Concepts and Examples