**Department of Collegiate & Technical Education**

**Capstone Project**

**Continuous Internal Evaluation – CIE – III**

**Capstone Project Name:** INVENTORY MANAGEMENT SYSTEM

**Capstone Project Members:** SHESHAGIRI S (149CS21046)

DIVYA J (149CS21021)

DILEEP G P (149CS21020)

DHANAYSHREE A P (149CS21061)

**Testing and validation: Details of laboratory experiments/ programming/ modelling/ simulations/ analysis/ fabrication/ construction etc.,**

**Testing:** In inventory management, testing involves ensuring that the inventory tracking system accurately records incoming and outgoing stock, updates inventory levels in real-time, and handles various scenarios such as stockouts, overstock situations, and backorders. Testing might include barcode scanning, RFID verification, and stress testing the system with large volumes of transactions.

**Validation:** Validation confirms that the inventory management system meets the specified requirements and provides accurate data for decision-making. This involves verifying that the system effectively manages stock levels, minimizes inventory costs, and meets regulatory compliance requirements.

**Laboratory Experiments :** For specialized IMS features or integrations with hardware (e.g., barcode scanners, robots), controlled lab environments might be used to test specific functionalities.

**Simulation and Modelling :**

* Focus: Simulate real-world scenarios using models to predict inventory behavior and system performance.
* Method: Develop simulation models to represent inventory fluctuations, order patterns, and analyze potential impact on stock levels and resource allocation.
* Example: Simulate seasonal demand changes, supplier delays, or unexpected sales spikes to assess the system's ability to handle dynamic inventory conditions.

**Data Analysis:**

* Analyze data collected during testing phases to identify trends, errors, and areas for improvement.
* Metrics like test case pass/fail rates, user feedback, performance data, and security vulnerabilities are analyzed.
* Provides insights into overall system stability, user experience, and identifies areas requiring further refinement.

**Fabrication & Construction**

* **Testing:**
  + **Prototyping:** Create physical or digital prototypes of new inventory storage solutions (e.g., shelving units) to test functionality, capacity, and ease of use.
  + **Stress Testing:** Test the capacity and durability of storage solutions under various loads and conditions to ensure they can handle anticipated inventory weight and volume.

**TESTING**

**Overview**

Testing is a procedure, which uncovers blunder in the program. It is the significant quality measure utilized during programming advancement. During testing, the program is executed with a lot of experiments and the yield of the program for the experiments is assessed to decide whether the program is proceeding as it is relied upon to perform.

The testing has been uses to find errors in the following categories:

• Incorrect or missing functions

• Interface errors

• Error in data structure or external database access

• Performance errors

**1. Unit Testing (Programming):**

* Focus: Each individual software module or function is tested in isolation to ensure it operates as intended.
* Methods: Unit testing frameworks like PHP Unit or JUnit can be used to write automated test cases that verify specific functionalities.
* Benefits: Catches bugs early in the development process, improving code quality and reducing overall testing time.

**2. Integration Testing (Programming):**

* Focus: Checks how different software modules interact and exchange data with each other.
* Methods: Simulate real-world data flow between components and verify data integrity throughout the system.
* Benefits: Ensures smooth communication and data exchange within the IMS, preventing integration issues later.

**3. System Testing (Functionality & Usability):**

* Focus: Evaluates the overall functionality and user experience of the IMS from the end-user's perspective.
* Methods: Functional testing verifies if the system performs core functionalities as designed. Usability testing assesses if the interface is user-friendly, intuitive, and meets user needs.
* Benefits: Identifies usability issues and ensures the system meets user expectations for both functionality and ease of use.

**4. Acceptance Testing (User Validation):**

* Focus: End-users (employees, managers) evaluate the IMS in a simulated or real-world environment to confirm it meets their needs and business requirements.
* Methods: User Acceptance Testing (UAT) involves user walkthroughs, task completion, and feedback collection.
* Benefits: Provides valuable user feedback for potential improvements and ensures the IMS is accepted and adopted by the target users.

**5. Performance Testing (Load & Stress):**

* Focus: Evaluates the system's performance under various load conditions, simulating real-world usage patterns.
* Methods: Load testing assesses system behavior under increasing user load. Stress testing pushes the system beyond expected usage to identify potential bottlenecks and breakage points.
* Benefits: Ensures the IMS can handle anticipated usage volumes, preventing performance issues during deployment.

**6. Security Testing (Penetration Testing):**

* Focus: Identifies potential security vulnerabilities that could lead to unauthorized access, data breaches, or system malfunctions.
* Methods: Penetration testing simulates attacks by hackers to expose potential security weaknesses.
* Benefits: Improves system security posture and protects sensitive inventory data from unauthorized access.

**Testing Suits To Inventory Management System Project**

**1. Unit Testing Suits**

* Focuses on individual modules of the inventory system code.
* Tests include:
  + Adding, editing, and deleting inventory items.
  + Stock level calculations.
  + Data validation for item information (e.g., price, quantity).
  + Error handling for invalid inputs.

**2. Integration Testing Suits**

* Evaluates how different modules interact within the system.
* Tests include:
  + Functionality of adding/removing items from inventory affecting stock levels.
  + Communication between inventory system and other connected systems (e.g., purchasing, accounting).
  + Data flow between modules (e.g., order processing impacting stock levels).

**3. System Testing Suits**

* Tests the entire inventory system as a whole from a user's perspective.
* Tests include:
  + User login, access control, and permission checks.
  + Searching and filtering of inventory items.
  + Stock level reports and inventory history.
  + Generating reports (e.g., low stock alerts, sales reports).
  + Functionality across different user interfaces (web, mobile app).

**4. Regression Testing Suits**

* Ensures existing functionalities remain intact after modifications or new features are introduced.
* Re-runs previously successful test cases after code changes.
* Helps identify regressions (bugs) introduced in updates.

**5. Usability Testing Suits**

* Evaluates the user-friendliness and ease of use of the inventory system.
* Tests include:
  + Interface clarity and intuitiveness.
  + Workflow efficiency for adding, managing, and reporting inventory.
  + User experience across different devices and screen sizes.

**How you can test your inventory management project effectively:**

**1. Define your Testing Strategy:**

* **Project Goals:** Clearly define what success looks like for your inventory system. Are you aiming to reduce stockouts, improve order fulfillment speed, or optimize storage space?
* **Testing Levels:** Identify the testing suits you'll implement (Unit, Integration, System, etc.) based ! on your project goals (refer to previous explanation of testing suits).
* **Test Cases:** Create detailed test cases for each testing suit. These should outline specific scenarios and expected outcomes for each test.

**2. Leverage Testing Tools:**

* **Automated Testing Tools:** Utilize automation tools to streamline repetitive test cases, saving time and effort during regression testing.
* **Inventory Management Software:** If using dedicated inventory management software, explore its built-in testing functionalities.

**3. Execute the Testing Suits:**

* **Unit Testing:** Developers typically perform unit testing during the development phase.
* **Integration Testing:** Test integration between different modules of the system after development is complete.
* **System Testing:** Simulate real-world user scenarios to comprehensively test the entire system.
* **Regression Testing:** Re-run successful test cases after code changes or new feature implementations to identify regressions (introduced bugs).
* **Usability Testing:** Involve potential users in testing the system's usability and gather feedback on its user-friendliness.

**4. Analyze and Report Results:**

* **Document Test Results:** Record the outcome of each test case (passed/failed) and identify any bugs encountered.
* **Defect Tracking:** Utilize a defect tracking system to manage and prioritize bug fixes.
* **Reporting:** Prepare reports summarizing the testing process and highlighting identified issues.

**5. Refine and Iterate:**

* Bug Fixes: Address identified bugs and retest the impacted functionalities.
* Test Automation Refinement: Continuously improve and refine your automated test cases for efficiency.
* Lessons Learned: Document lessons learned from the testing process to improve future testing endeavors.

**To test a Inventory management system project** **what resources we need to test**

The resources you'll need to test your inventory management project can be categorized into three main areas:

**1. People:**

* **Testers:** You'll need testers with a good understanding of inventory management principles and the functionalities of your system. This could involve your development team, dedicated QA testers, or even potential end-users for usability testing.
* **Subject Matter Experts (SMEs):** Involve individuals familiar with your specific inventory management workflows and data to create realistic test cases and evaluate the system's effectiveness from a user's perspective.

**2. Tools & Technologies:**

* **Testing Framework:** Consider using a testing framework like JUnit (for Java) or NUnit (for .NET) to structure and automate test cases, especially for unit and regression testing.
* **Inventory Management Software (if applicable):** If you're using pre-built inventory management software, explore any built-in testing functionalities it might offer. Some software may have test data generators or reporting tools to aid testing.
* **Defect Tracking System:** A system like Jira or Bugzilla helps you efficiently track and manage bugs identified during testing.
* **Test Data Management Tool (optional):** For complex systems or large datasets, consider using a test data management tool to create and manage realistic test data sets.

**3. Hardware & Software Environment:**

* **Testing Environment:** Set up a dedicated testing environment separate from your production system to avoid impacting live data. This allows you to freely test functionalities and introduce bugs without affecting real operations.
* **Hardware Resources:** Ensure you have adequate computing power and storage capacity to run your testing tools and manage test data, especially if dealing with large datasets.

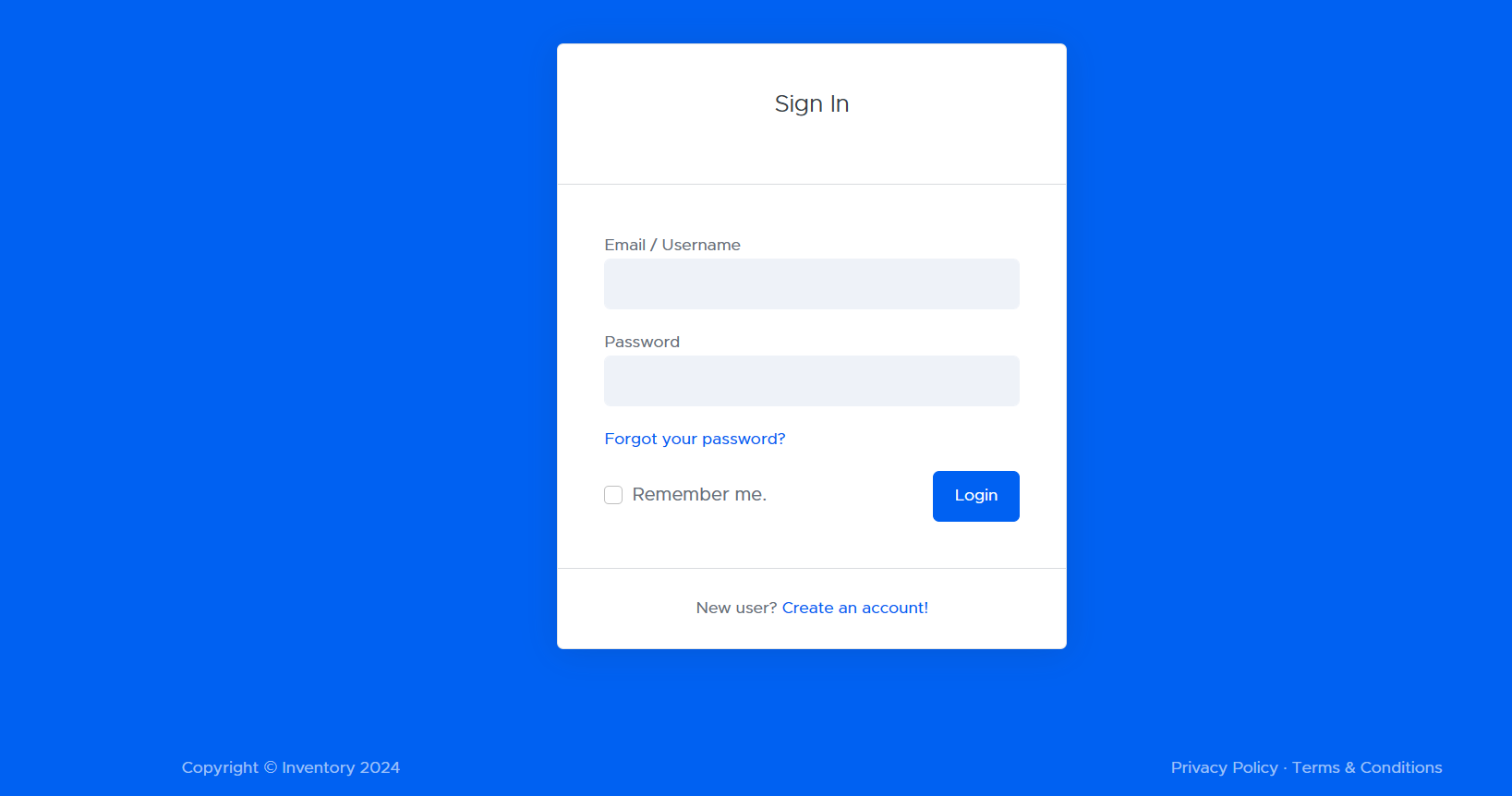
**Test Cases: Admin Functionality for Inventory management system**

This table outlines test cases specifically for the Admin functionalities of a Inventory management system project.

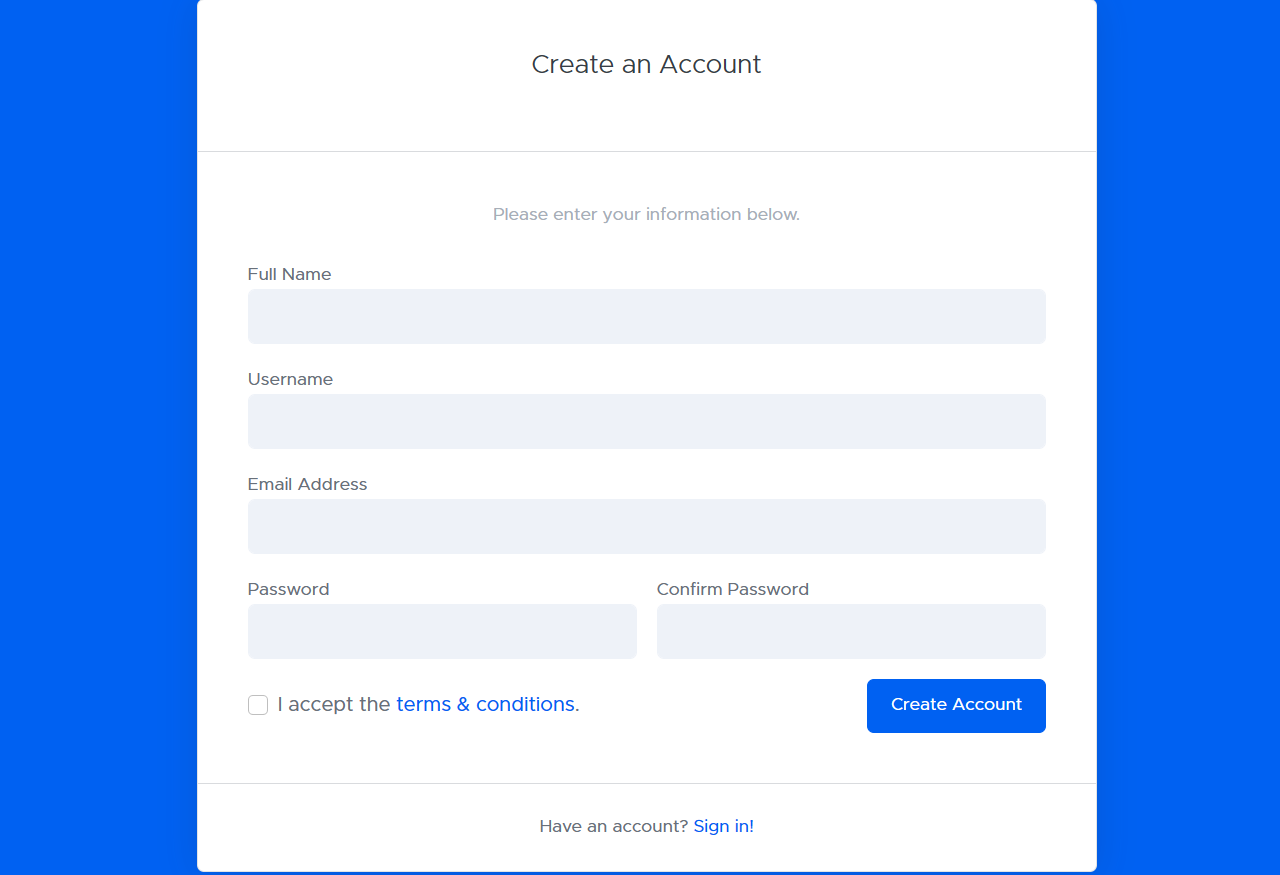
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Description | Expected Result | Pass/Fail | Notes |
| TC-01 | Admin attempts to add a new product to the inventory. | - New product information is saved successfully. - System displays confirmation message. | Pass | - Test with valid product information (name, description, quantity, etc.). |
| TC-02 | Admin attempts to add a new product with an existing product name. | - System displays an error message indicating duplicate product name. - New product is not added. | Pass | - Validate system enforces unique product names. |
| TC-03 | Admin attempts to edit an existing product's information. | - Updated product information is saved successfully. - System displays confirmation message. | Pass | - Test with valid changes to product details. |
| TC-04 | Admin attempts to edit a non-existent product. | - System displays an error message indicating the product cannot be found. | Pass | - Validate system handles attempts to edit invalid product IDs. |
| TC-05 | Admin attempts to delete an existing product. | - Product is removed from the inventory. - System displays confirmation message. | Pass | - User is prompted to confirm deletion before proceeding. |
| TC-06 | Admin attempts to delete a non-existent product. | - System displays an error message indicating the product cannot be found. | Pass | - Validate system handles attempts to delete invalid product IDs |
| TC-07 | Admin views a list of all products in the inventory. | - A complete and accurate list of all products is displayed. | Pass | - List includes product details (name, description, quantity, etc.). |
| TC-08 | Admin filters the product list by a specific category. | - Only products belonging to the chosen category are displayed. | Pass | - System allows filtering by pre-defined categories (if applicable). |
| TC-09 | Admin searches for a product by name or ID. | - The specific product matching the search criteria is displayed. | Pass | - Search functionality works accurately. |

**Result and Inferences :**

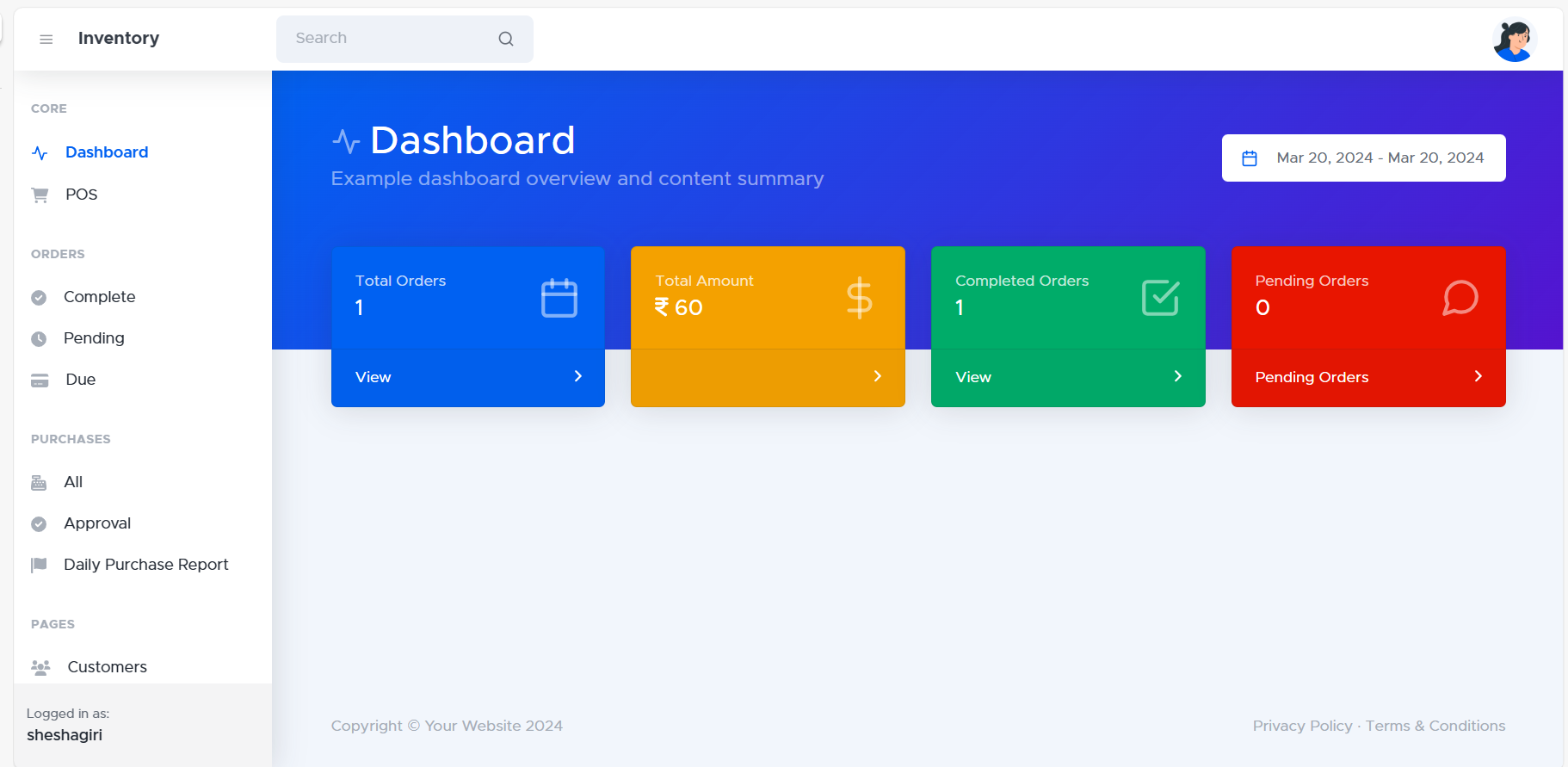
**Log in:** The login of an inventory management system (IMS) will indicate whether the user's credentials (username and password) were successfully validated or not.

****

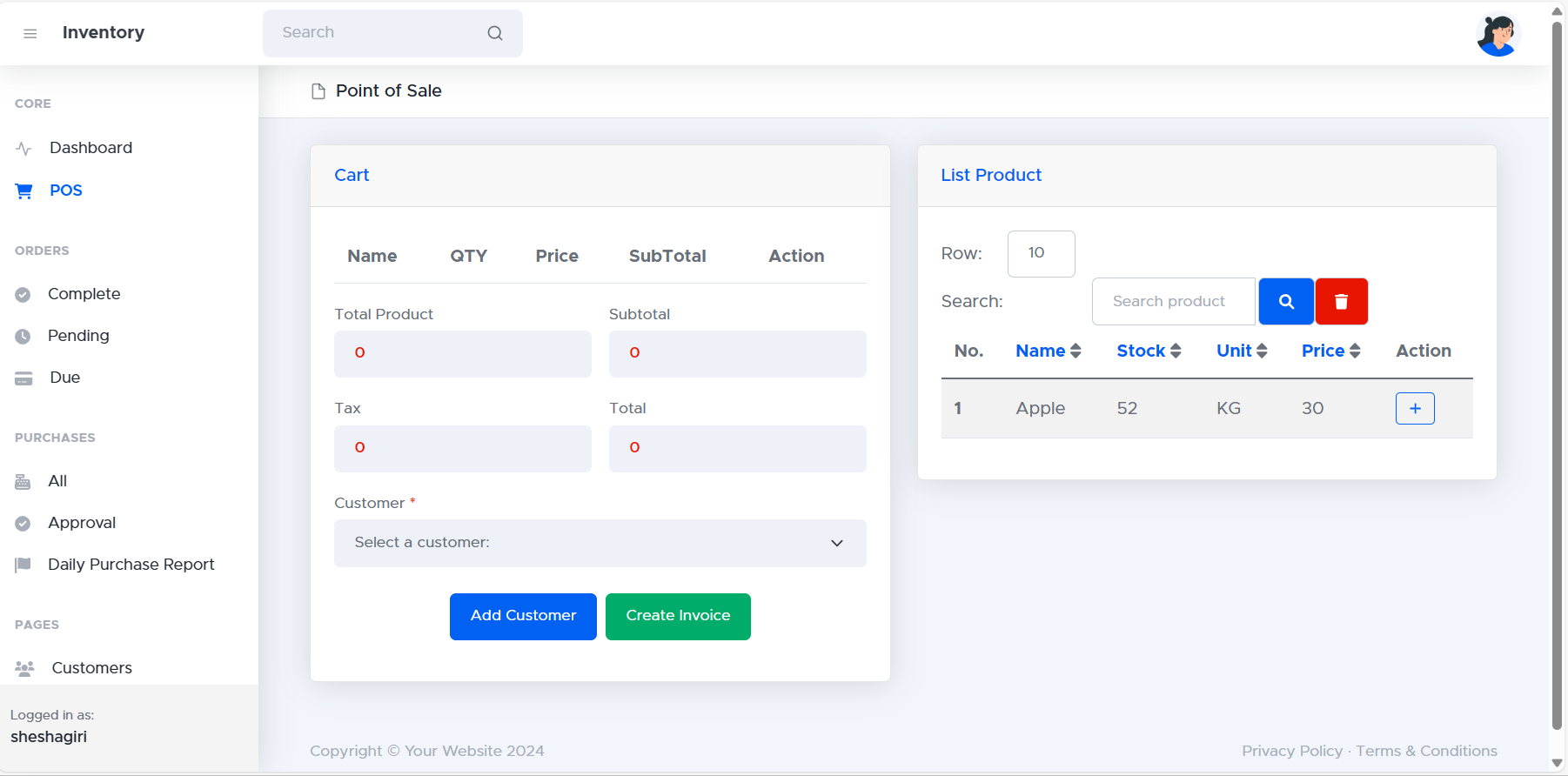
**Create an account :** The account creation process for the inventory management system (IMS) resulted in:



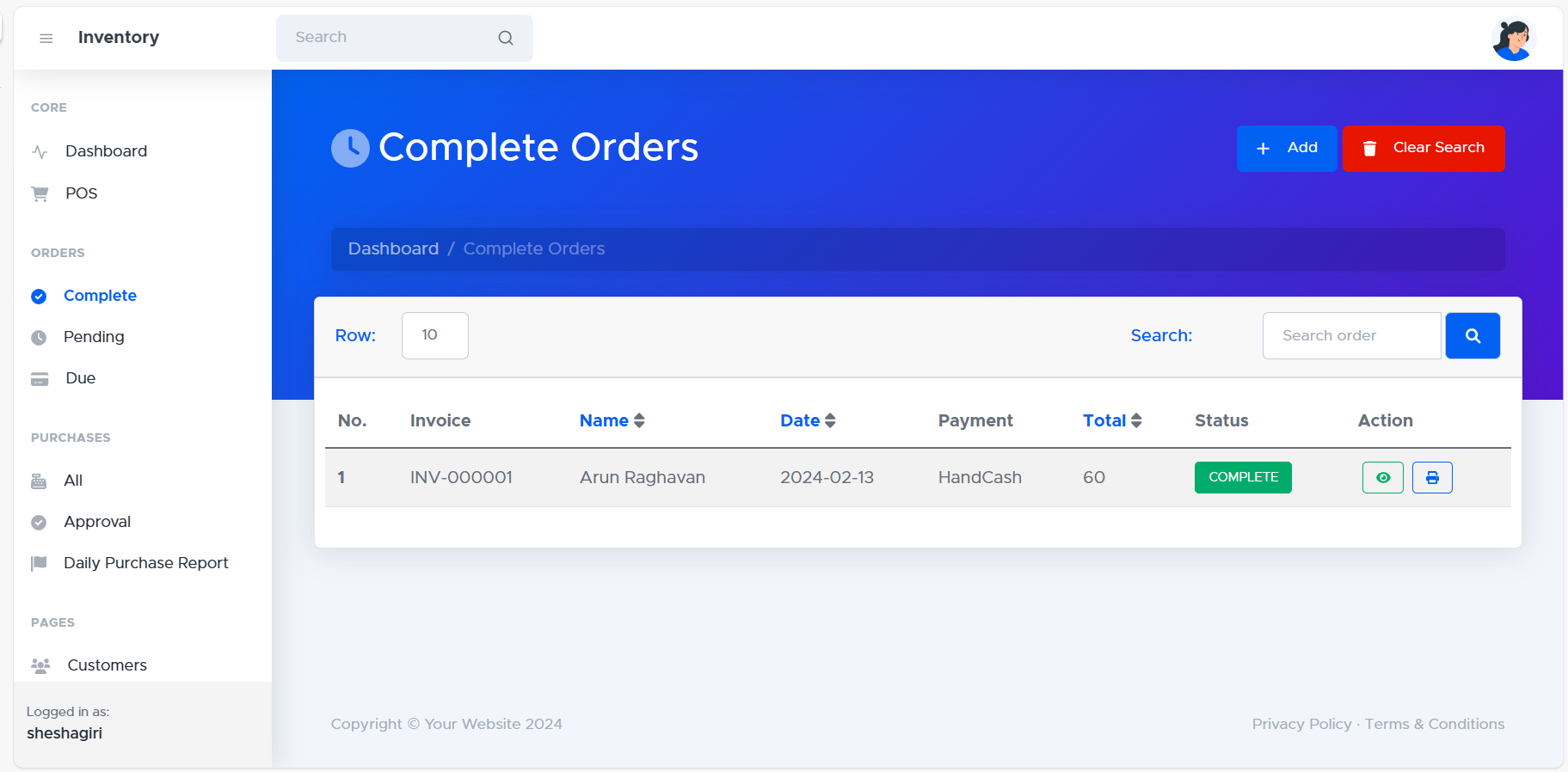
**Dashboard :** The inventory management system (IMS) dashboard provides a centralized location to visualize and analyze your inventory data at a glance.



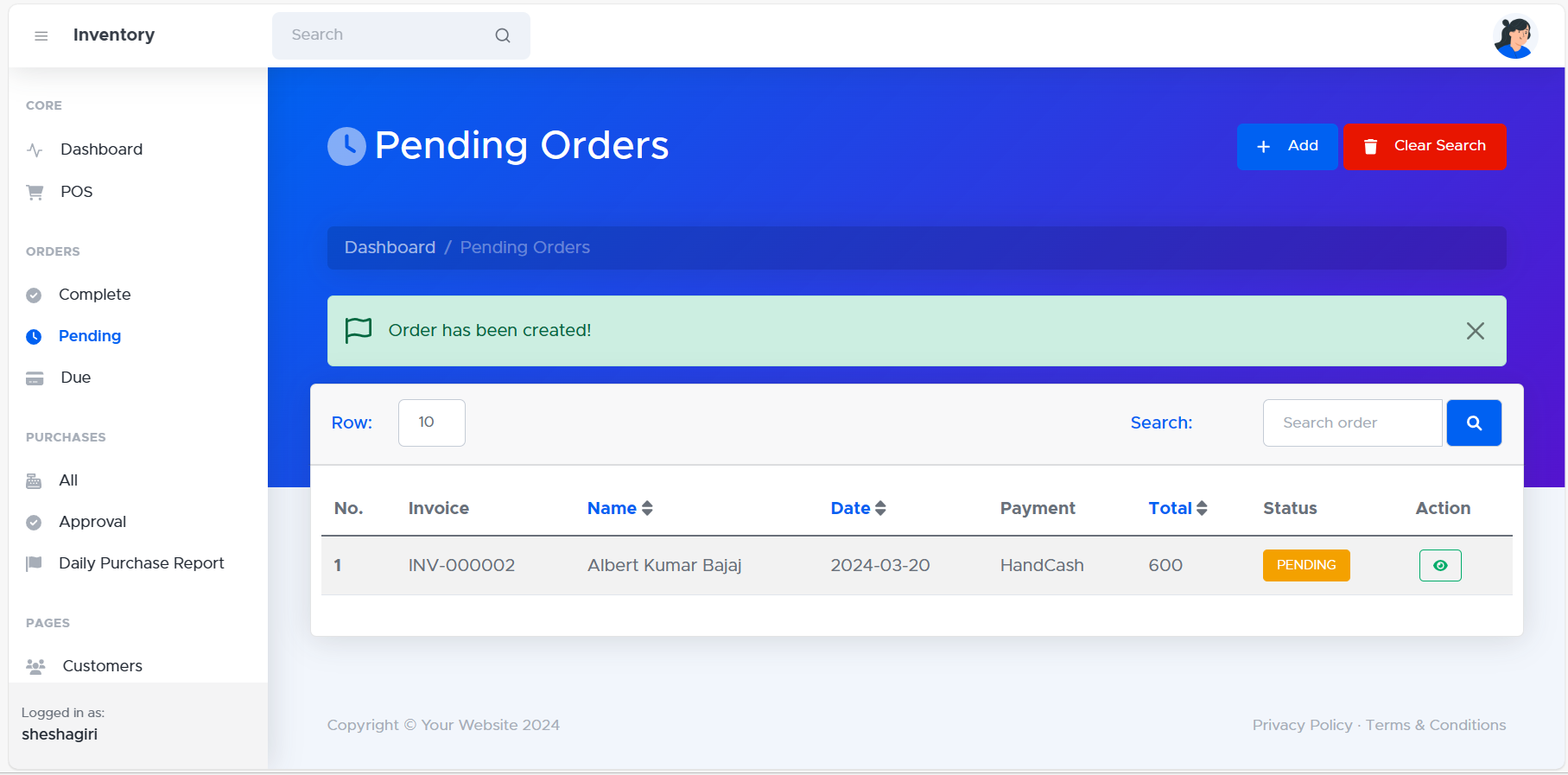
**Point of sale :** Integrating your POS with your IMS can lead to a more efficient and profitable business. You'll benefit from improved inventory accuracy, enhanced operational efficiency, and data-driven insights that empower you to make informed decisions for your business.



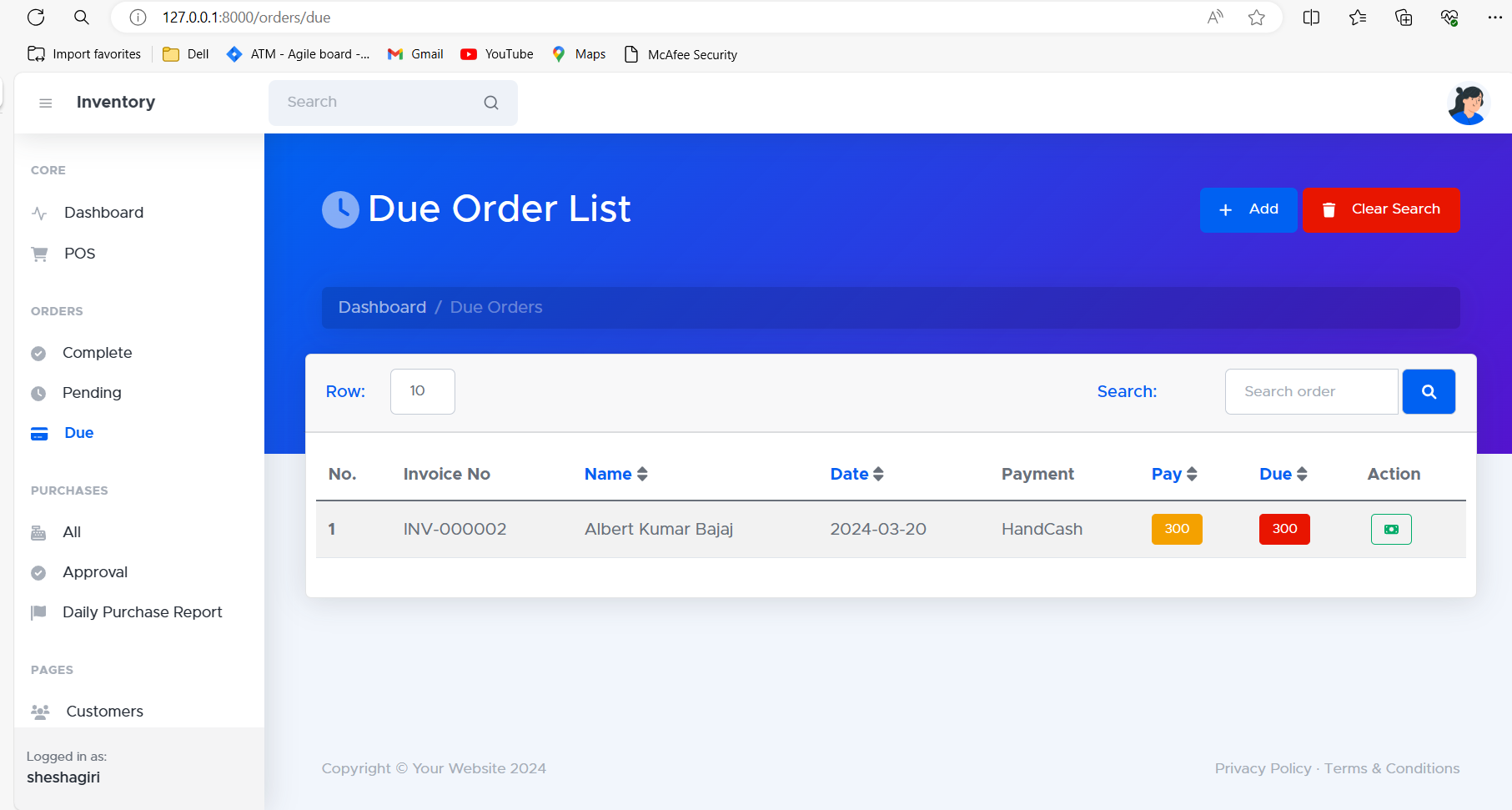
**Completed order :** A completed order in an IMS serves as a record of a successful customer transaction. The associated information provides valuable insights for sales analysis, inventory management, and customer service.



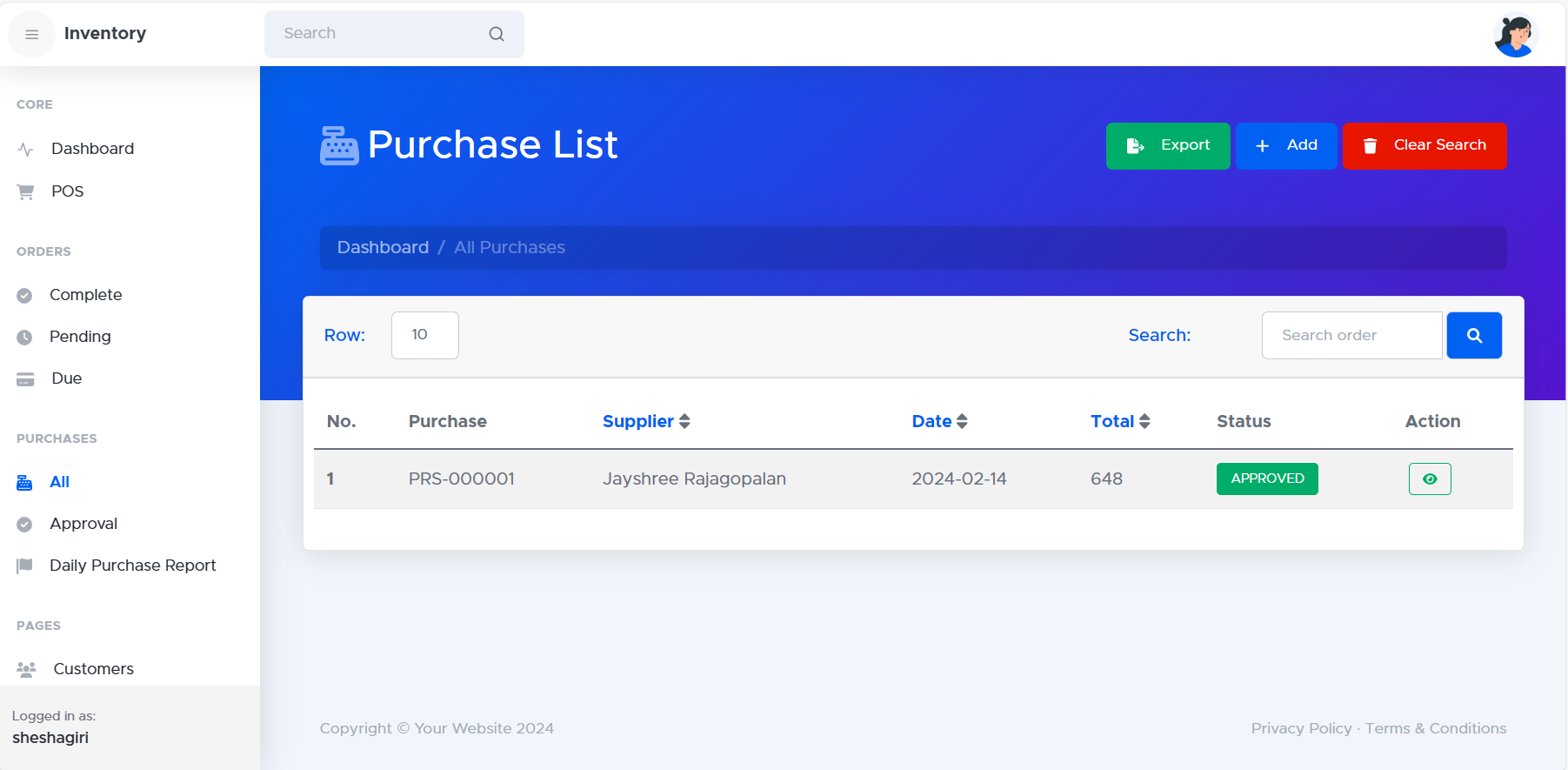
**Pending orders:** The "Pending Orders" section within your inventory management system (IMS) displays a list of orders that have been placed but haven't yet been fulfilled for various reasons.



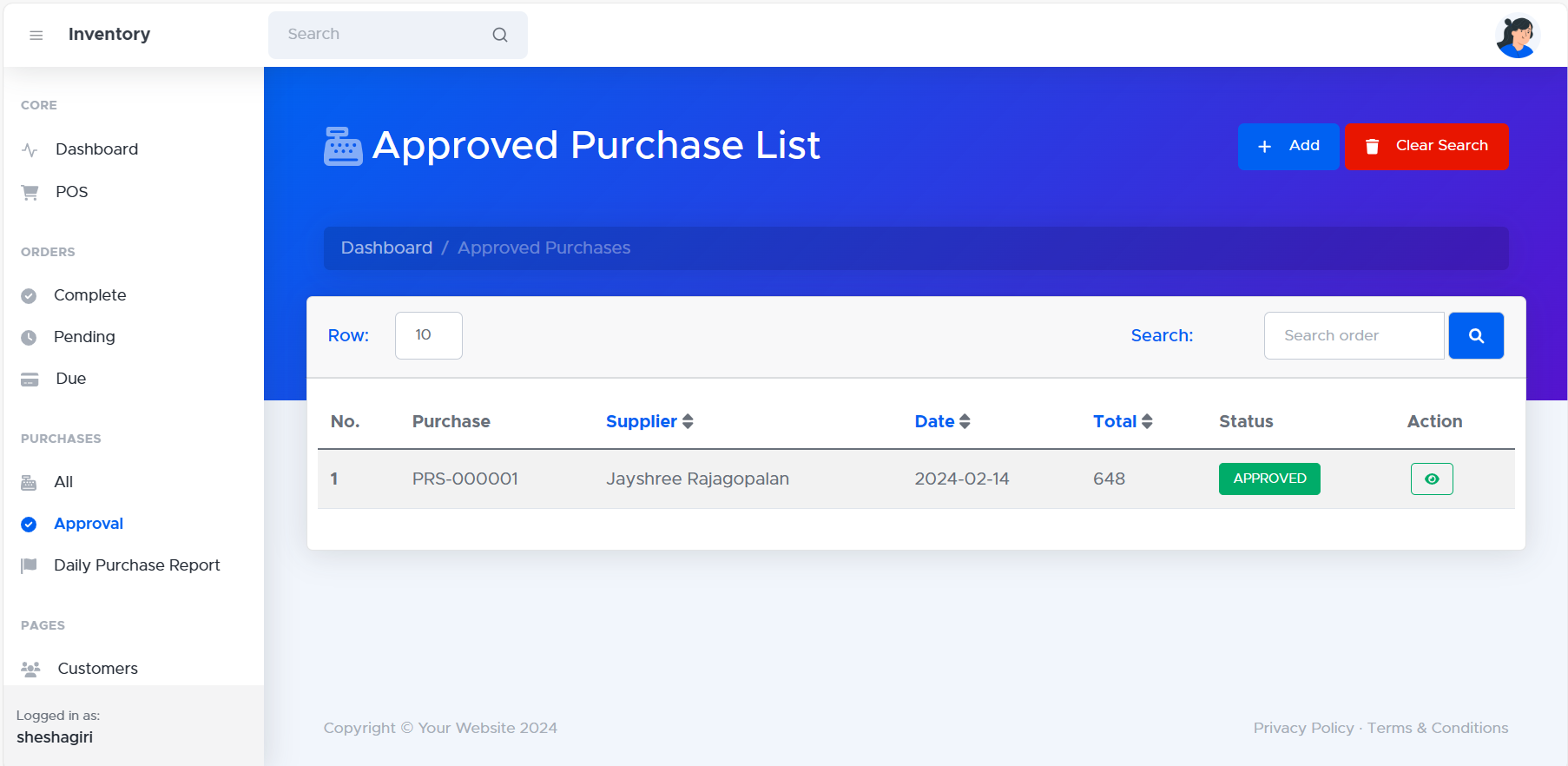
**Due order list :** the Due Order List is a valuable tool for inventory management. It provides crucial insights into replenishment needs and helps maintain optimal stock levels, ultimately improving business efficiency and profitability.



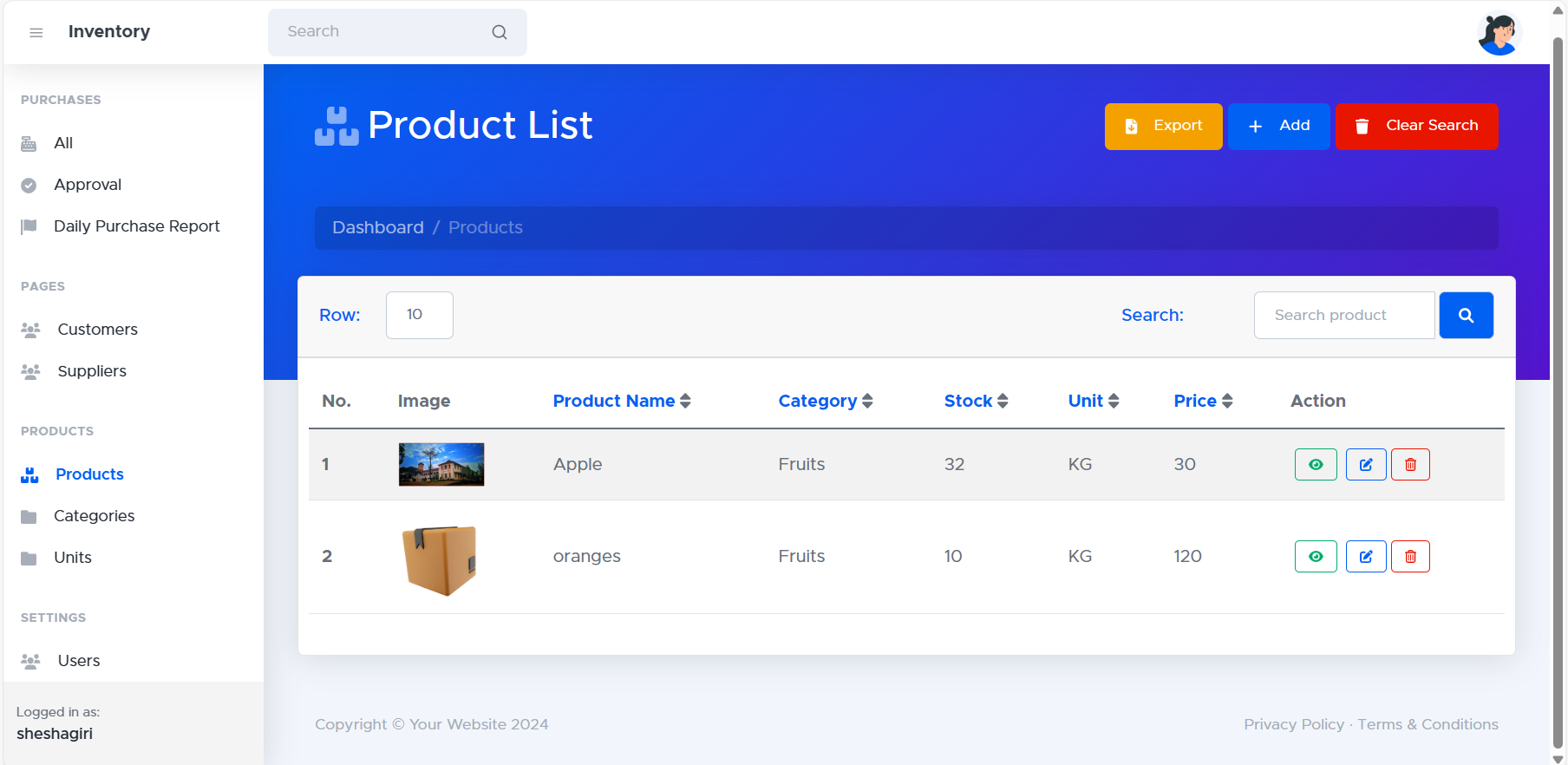
**Purchase List:** The Purchase List within your inventory management system (IMS) provides a detailed record of all purchases made for your inventory.



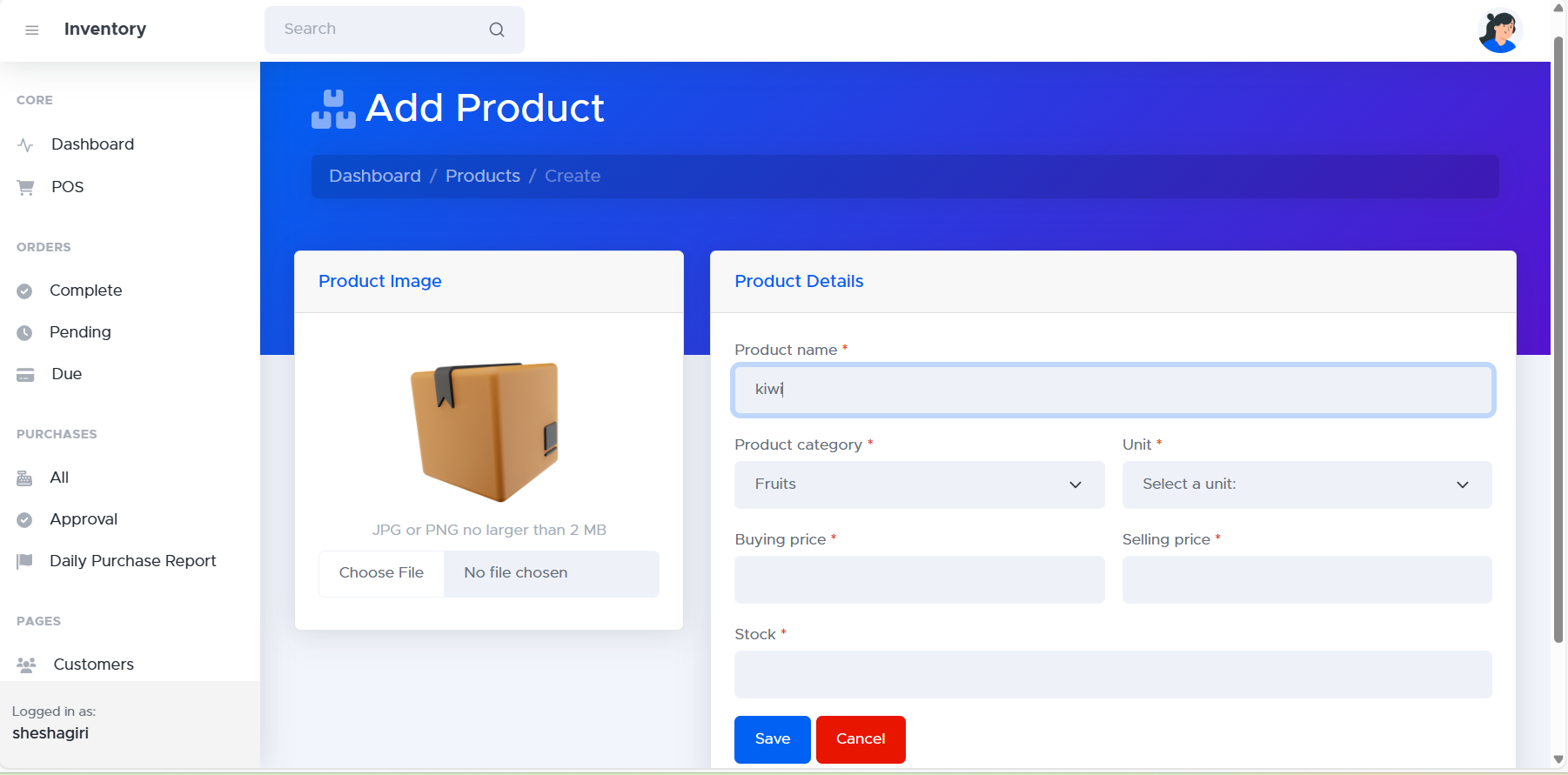
**Approved Purchase List :** The Approved Purchase List (APL) is a generated report or list within your inventory management system (IMS) that identifies items authorized for purchase.



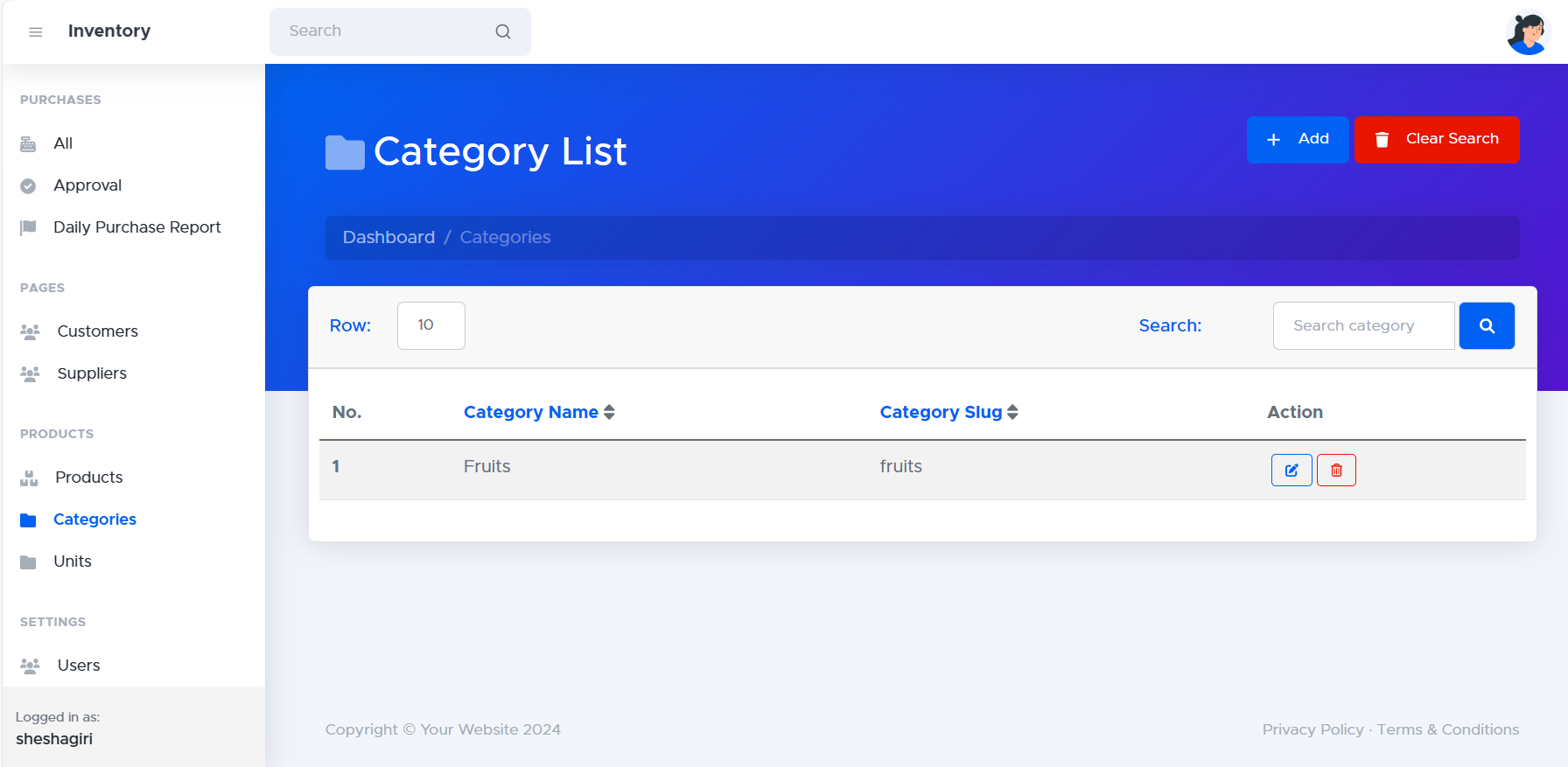
**Product List :** The "Product List" within your inventory management system (IMS) displays a comprehensive overview of all the items currently stored in your inventory.



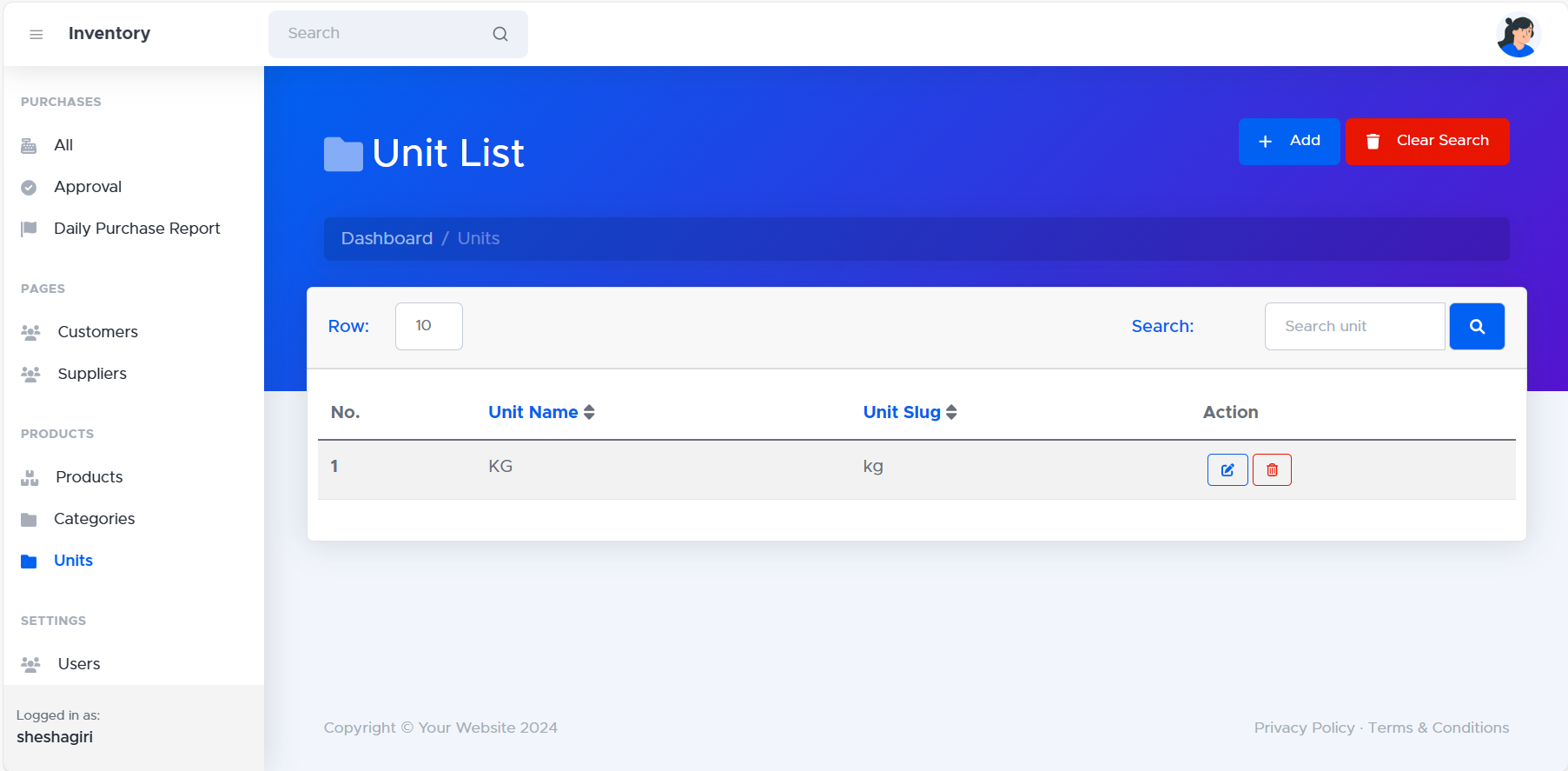
**Add Product :** The "Product List" within your inventory management system (IMS) displays a comprehensive overview of all the items currently stored in your inventory.



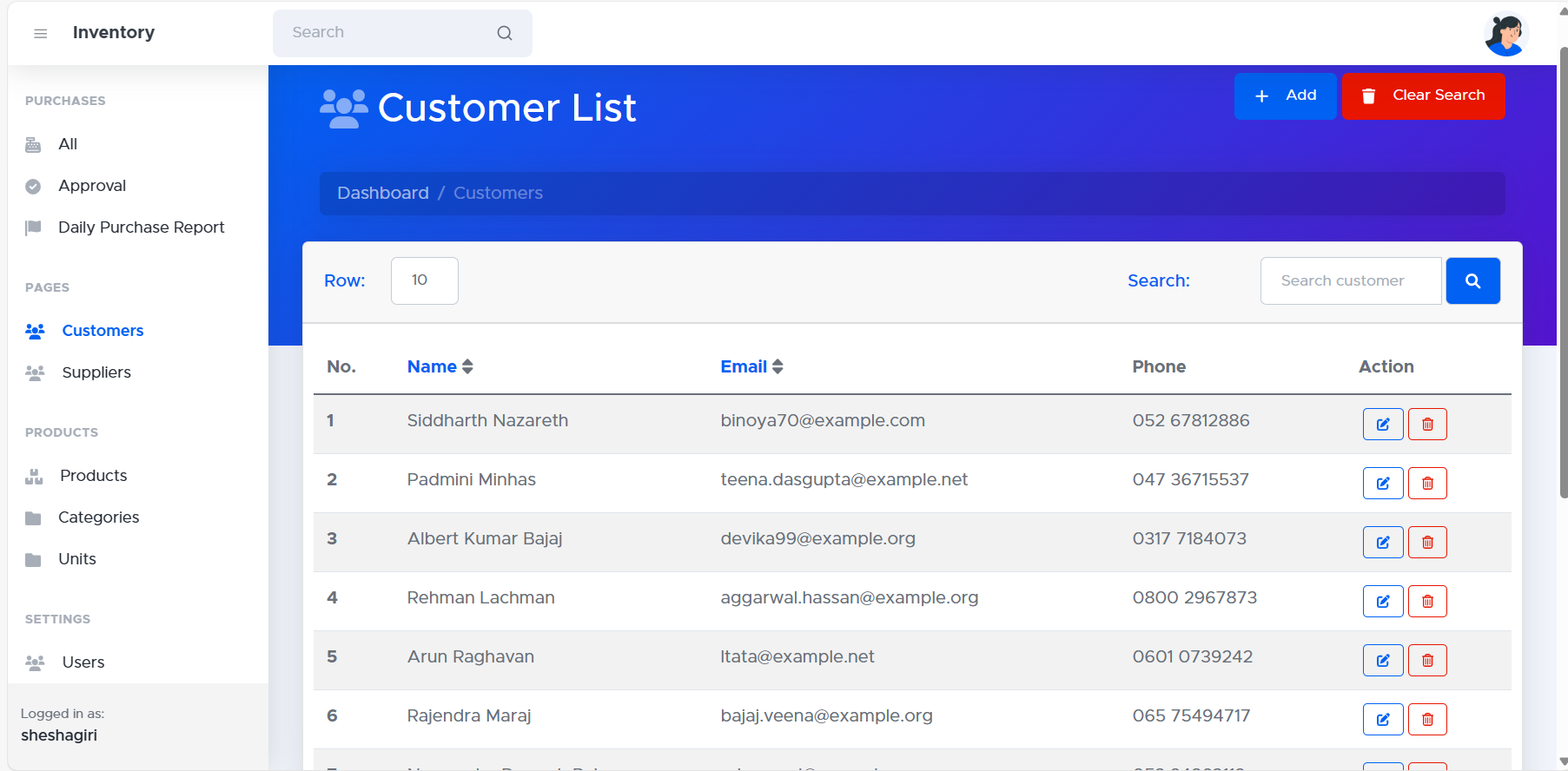
**Category List :** The "Category List" in your inventory management system (IMS) displays a structured breakdown of all product categories used to organize your inventory.



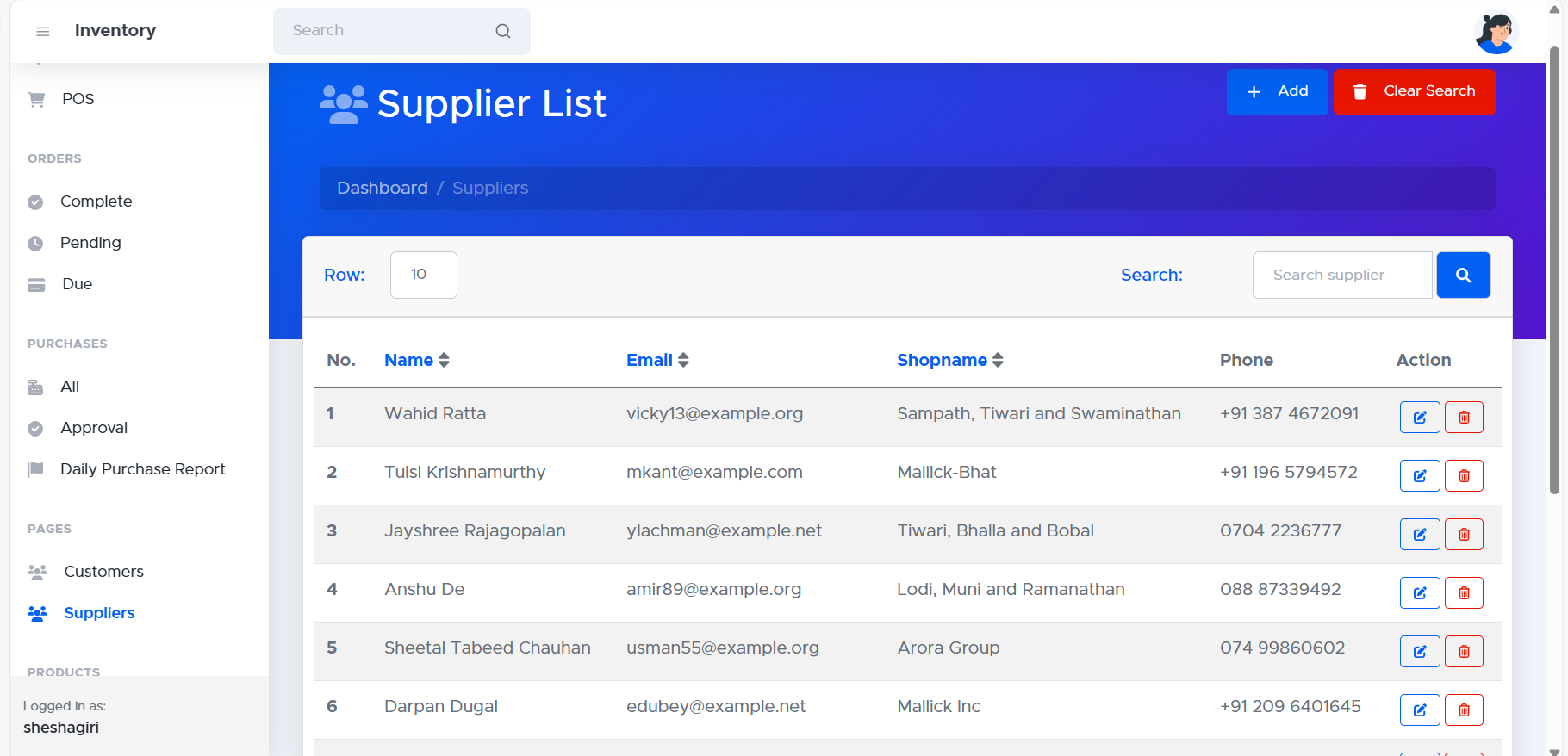
**Unit List :** The Unit List provides an overview of all the units of measurement used within your IMS for quantifying inventory items.



**Customer List :** The "Customer List" within your inventory management system (IMS) provides a central repository for managing information about your customer base.



**Supplier List :** The "Supplier List" within your inventory management system (IMS) provides a central repository of information about all the vendors or suppliers you source your products from.



**User List:** The "User List" in your inventory management system (IMS) displays a record of all individuals authorized to access and utilize the system.

