**Inventory Management System**

**1. Introduction**

The Inventory Management System is a web application designed to streamline the management of inventory for a business. It features two user types: Managers (Superusers) and Suppliers. This document provides an overview of the system, its architecture, and key functionalities.

**2. System Architecture**

The system is built using the Django framework. This Django project manages a system with two types of users: admins and suppliers. It employs a relational database to store user information, product details, and inventory data.  
  
**2.1 Tables  
1. User(from django.contrib.auth.models):** Represents the core user model already provided by Django for authentication purposes. Likely extended by the admin through the Django admin interface.

**2. Supplier :**

* One-to-one relationship with User: Ensures a unique supplier account is associated with each user.
* Additional fields: phone\_number (unique) and address

**3 .Product :**

* Foreign key relationship with **Supplier**: Links products to their corresponding supplier.
* Fields:

**name** (required, not null): Product name.

**description** (optional): Product description.

**unit\_price** (DecimalField, minimum value 0.01): Unit price of the product.

**stock** (PositiveIntegerField): Current product stock.

**active\_status** (BooleanField, default=True): Indicates if the product is active

**4. Inventory :**

* One-to-one relationship with **Product**: Each product has a unique inventory entry.
* Fields:

**selling\_unit\_price** (DecimalField, minimum value 0.01): Selling price of the product in the inventory.

**stock** (PositiveIntegerField): Current product stock in the inventory

**2.2 Relationships**

* A User can have a one-to-one relationship with a Supplier.
* A Supplier can have many Products through the foreign key relationship.
* A Product can have a one-to-one relationship with an Inventory entry.

**3.User Management (Admin vs. Supplier):**

The User Management in the Inventory Management System begins with the creation of an admin user. This is achieved by using the Django command:

**python manage.py createsuperuser**

This command prompts the administrator to input a username, email, and password for the superuser. Once created, this superuser gains access to the Django Admin View.

Through the Django Admin View, the admin can manage suppliers. This includes creating new supplier instances. The Supplier model, connected to the Django User model through a one-to-one relationship, allows the admin to associate each supplier with a user account.

**3.1 AdminLogin**

* The admin, after creating the superuser account, logs in using the provided credentials.
* The system uses a **AdminLoginMixin** to restrict access to views meant only for administrators.

**3.2 SupplierLogin**

* Suppliers, created by the admin, can log in with their credentials.
* The system uses a SupplierLoginMixin to limit access to views specifically designed for suppliers.

**4.Permissions**

**4.1 Supplier Permissions :**

Upon logging in, suppliers have permissions restricted to Product CRUD operations:

* **Create Product**: Suppliers can add new products to the system.
* **View Own Products**: Suppliers have the ability to view only the products they added.
* **Edit Own Products**: Suppliers can edit the details of the products they added.
* **Delete Own Products**: Suppliers can delete the products they added.

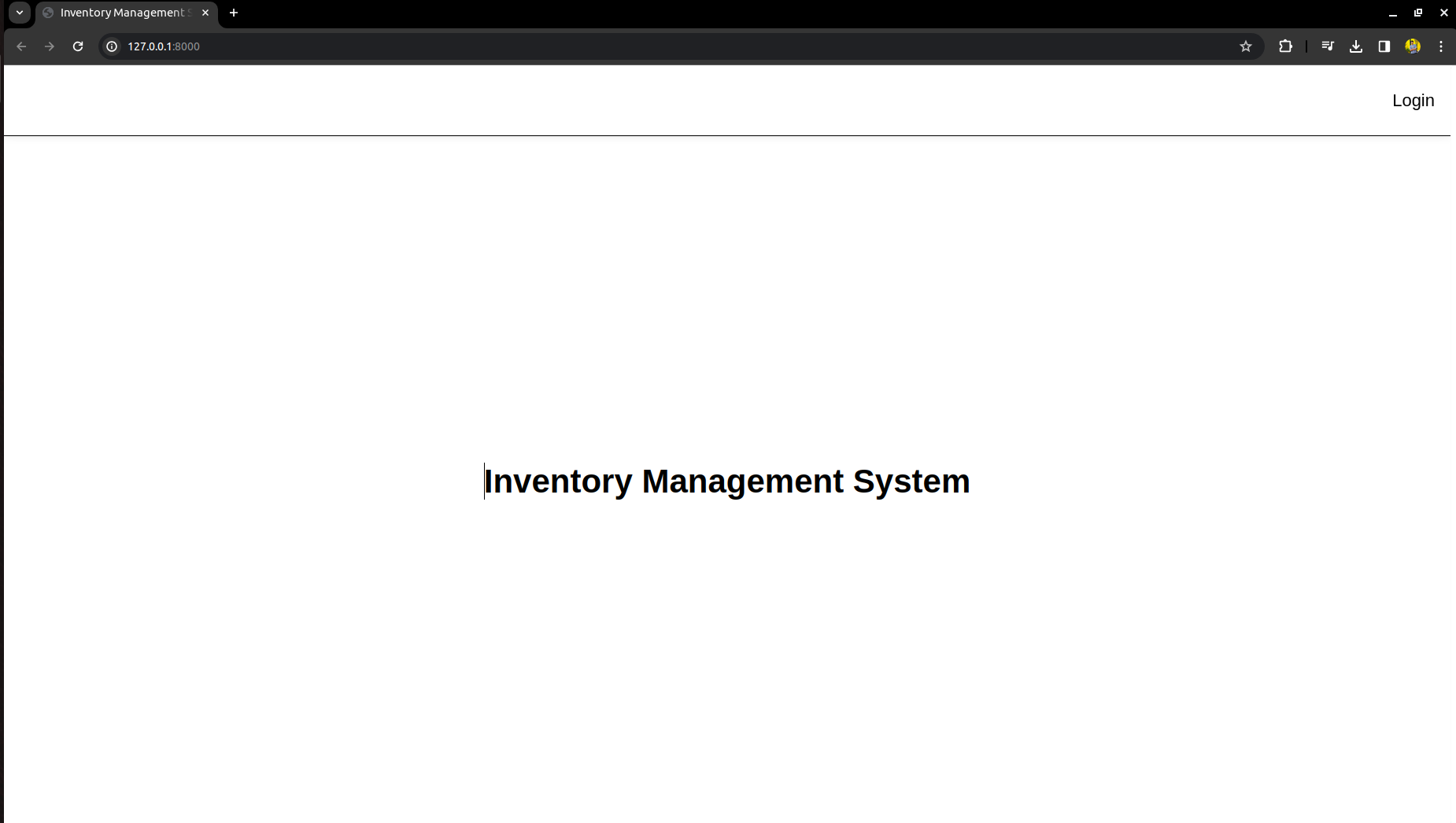
**4.2 Admin Permissions :**

The Admin Dashboard is a dedicated view for the administrator. Here, the admin can perform various actions related to inventory management and supplier details.

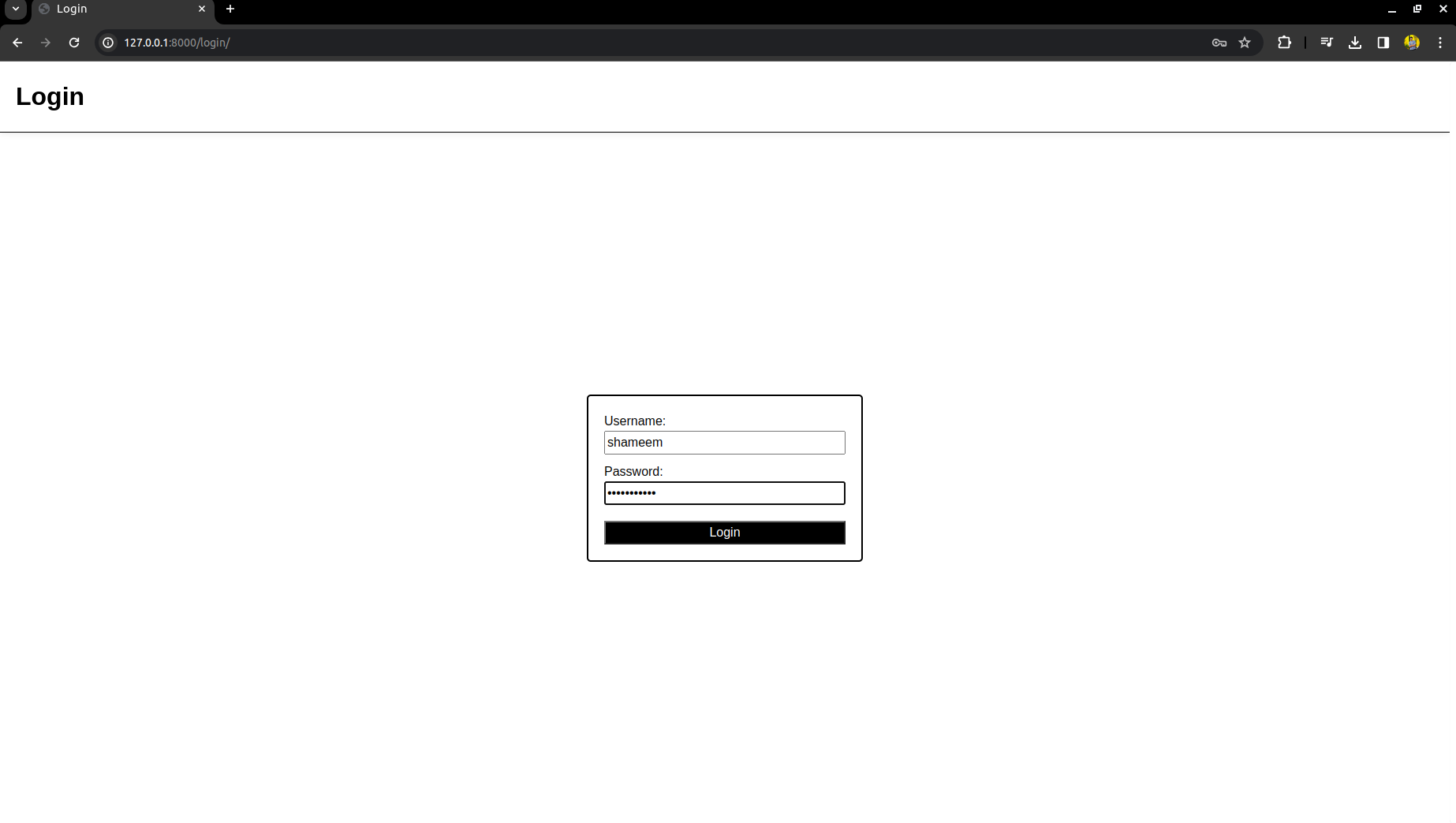
* **Inventory Management :** The admin has the ability to view all products available in the inventory, irrespective of their suppliers.
* **View Supplier Products :** The admin can also view products specific to each supplier. This allows for a detailed overview of products from different suppliers
* **Supplier Details :** The admin can access and view detailed information about each supplier. This includes the supplier's user details, phone number, and address.
* **Purchase Products from Suppliers :** Admins can initiate product purchases from suppliers. This involves adding products to the inventory that are supplied by various suppliers.
* **Report Generation View :** The admin can sort and filter the inventory data to gain insights into specific aspects such as product types, suppliers, and stock quantities
* **CSV Report Generation :** An important feature is the ability to generate a CSV (Comma-Separated Values) file of the inventory report. This enables the admin to export and analyze the inventory data in external use.

**5. Pages**

**5.1 Landing Page :**At the top right corner of the landing page, you'll find a link that directs you to the login page.



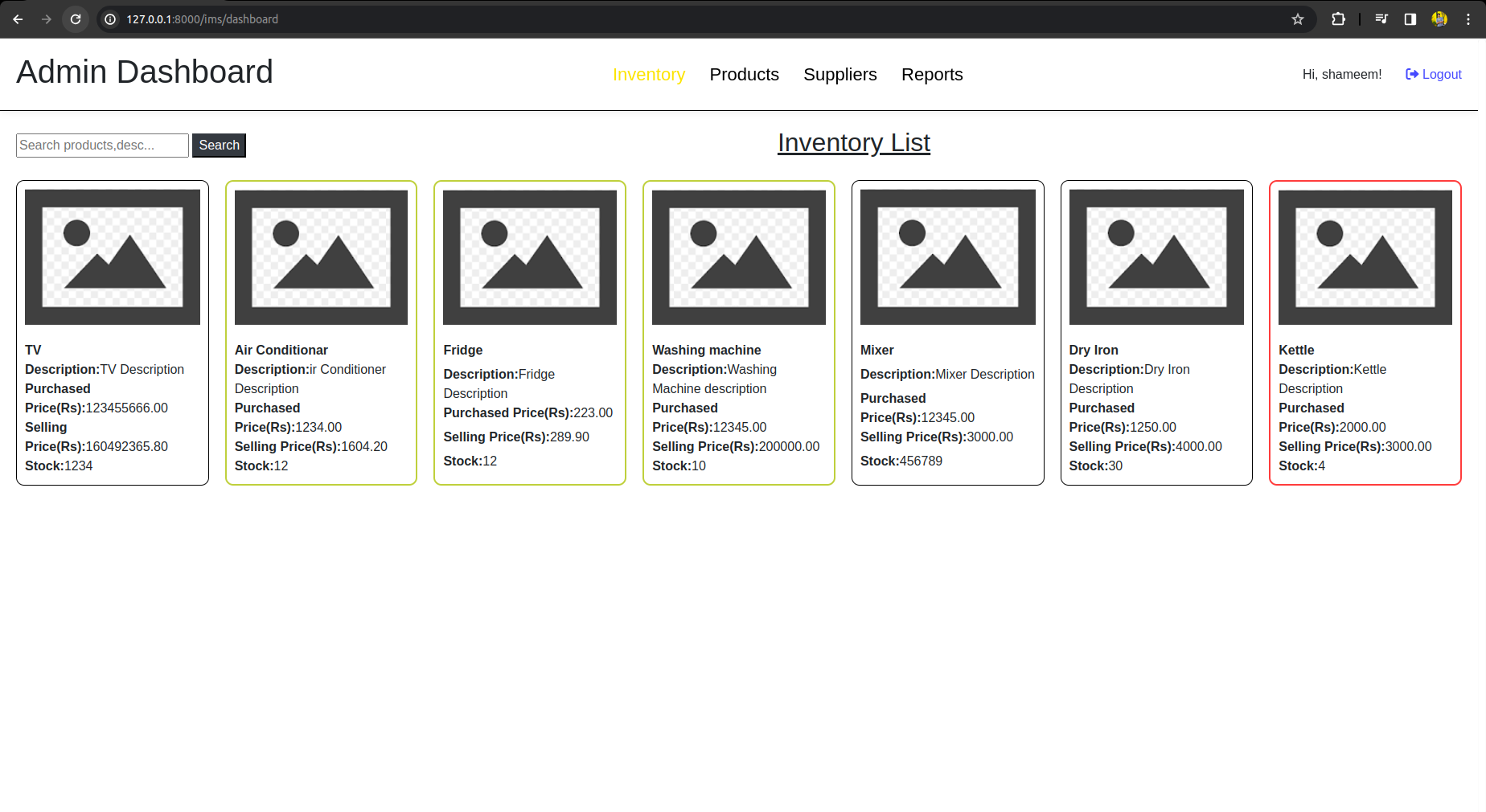
**5.2 Login Page**  
On the login page, you'll find fields for entering your username and password. Upon submitting, any errors will be displayed on the same page. Upon successful login, users will be directed to different pages based on their roles. Administrators will access the admin dashboard, while suppliers will be directed to the supplier dashboard.



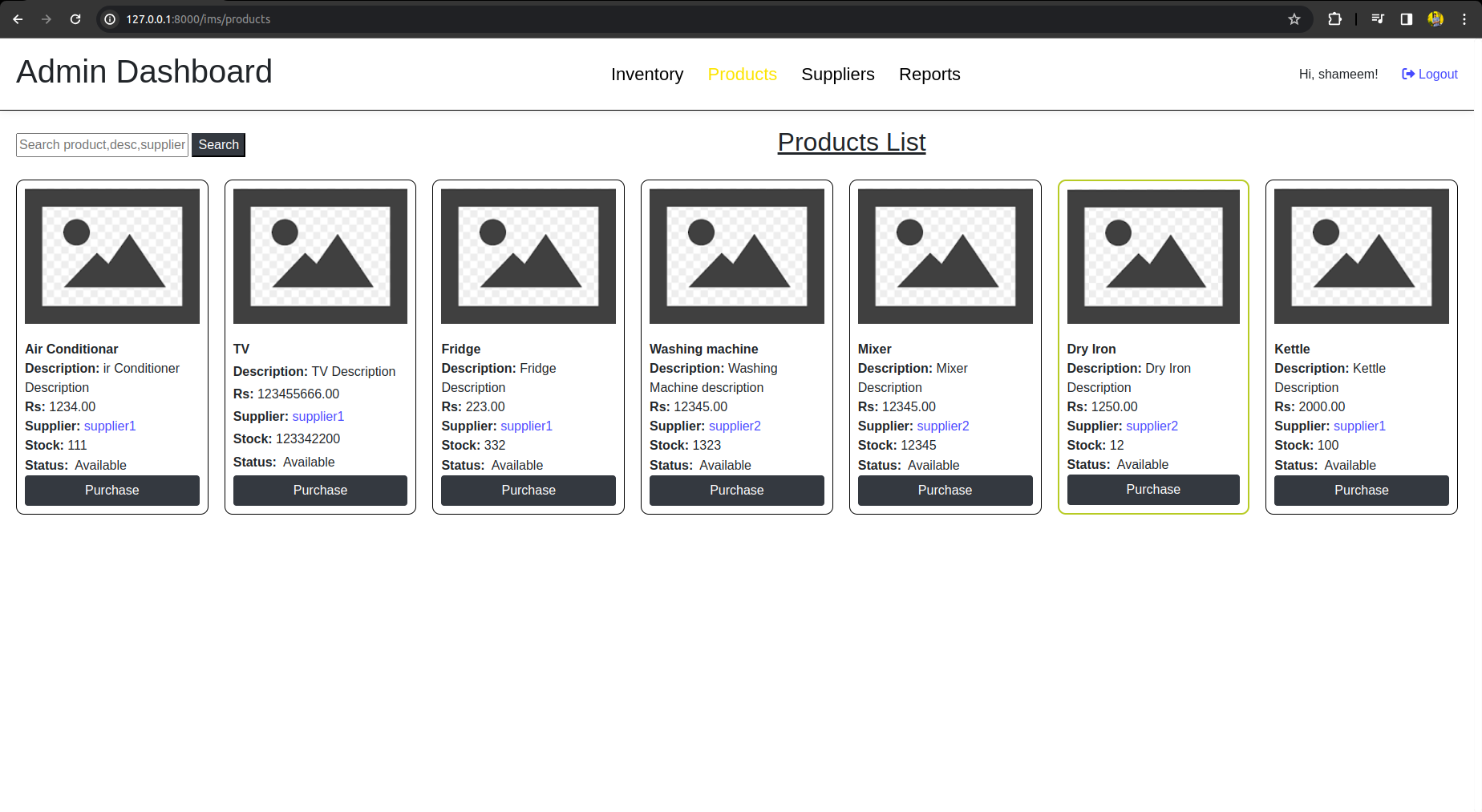
**5.3 Admin Dashboard**

Within the admin dashboard, the homepage features a comprehensive list of inventory products. The dashboard includes a search option, allowing admin

to search based on product name and description. Notably, each product card is equipped with an indicator: a blinking yellow border alerts users when the stock is less than 20, while a red blinking border signifies a stock of 5 or fewer units.



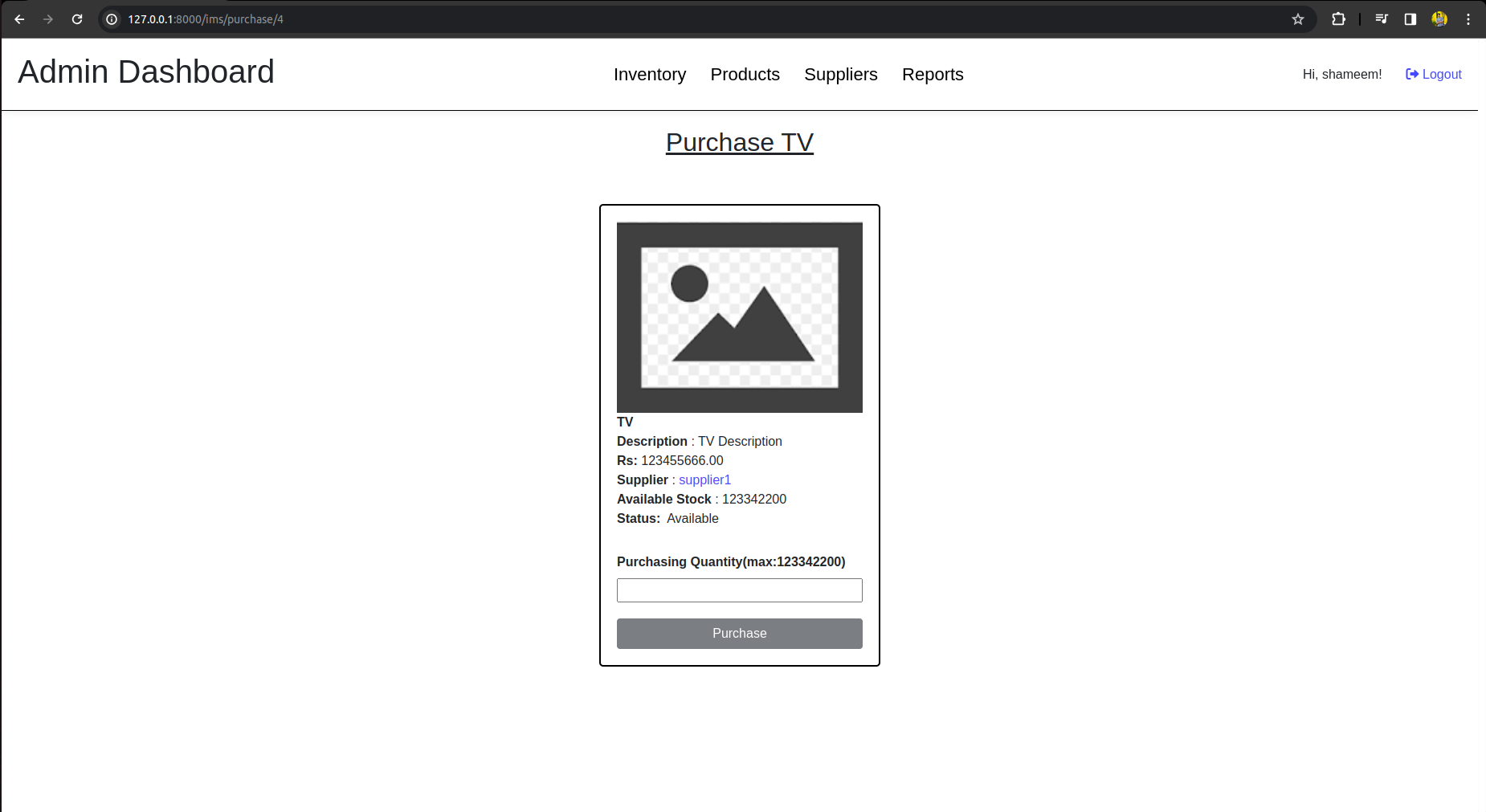
**5.4 Admin Product View**  
The admin has access to a comprehensive view displaying all products sourced from various suppliers. The view incorporates an indicator reflecting stock levels. Additionally, admin can utilize the search option to filter results by product name, description, and supplier. Each product card contains a link to the respective supplier's profile page, and a purchase button is available for each product.



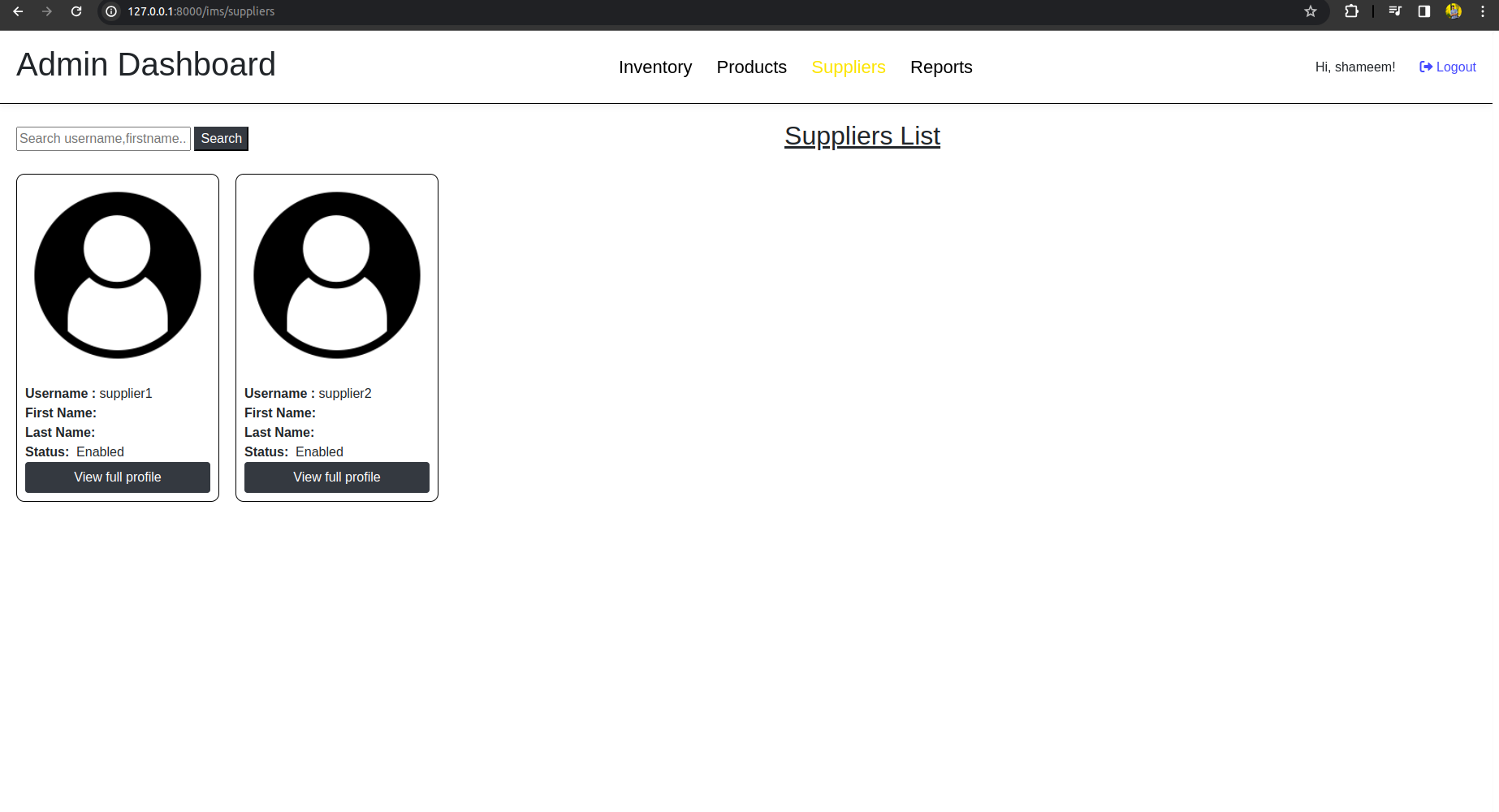
**5.5 Admin Purchase Product**

Upon clicking the purchase button, admin will be directed to the Purchase page, where they can initiate the buying process for the specific product. Several validations are in place, including ensuring that the purchase count is less than or equal to the available count. Additionally, a 30% increment from the buying price is applied in our pricing logic implemented in the backend.

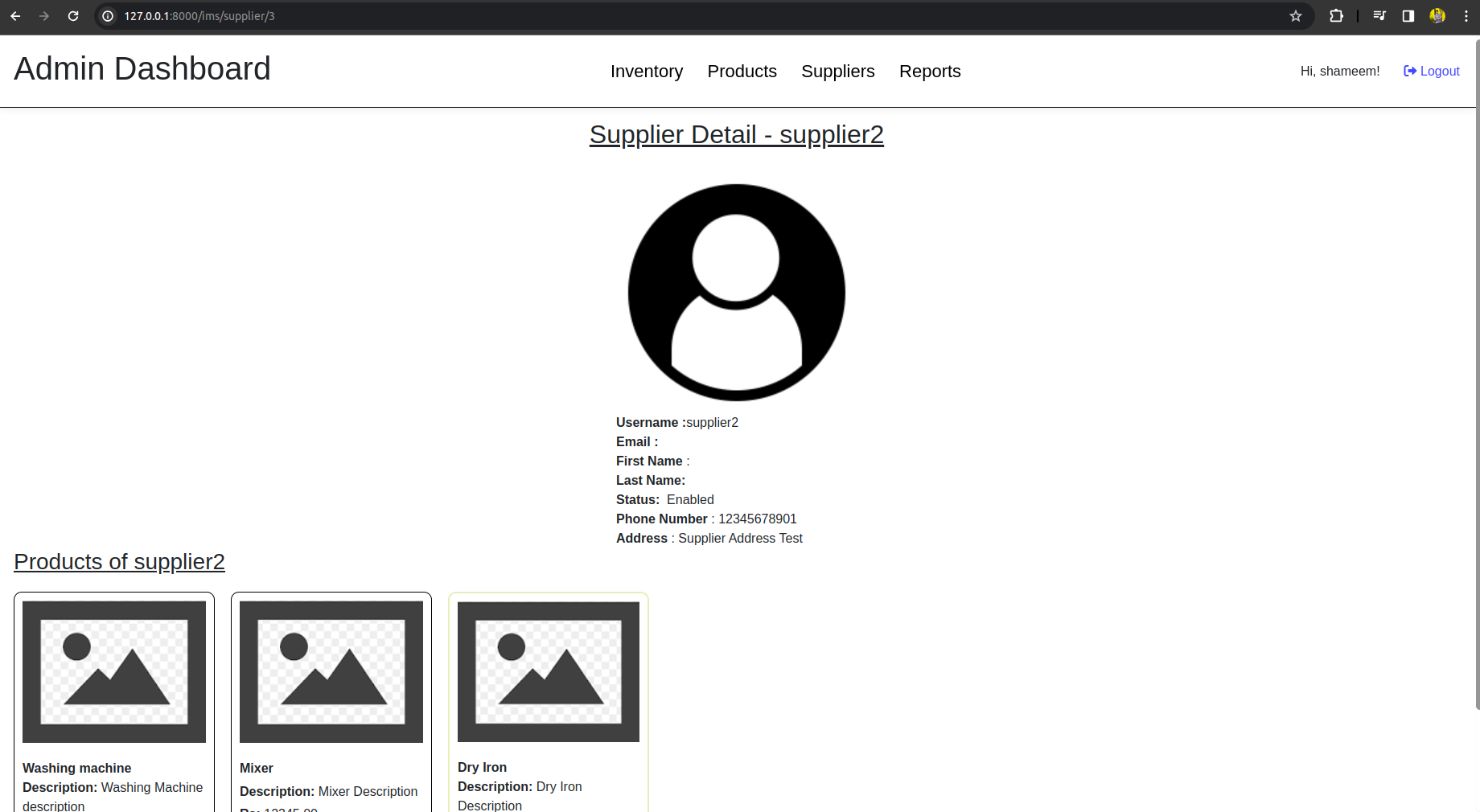
The Purchase page also features a link to the supplier's detailed page. Upon completing the purchase, the product is added to the Inventory table. If the product is already present in the table, the stock count is incremented; otherwise, a new entry is created. Simultaneously, the count is subtracted from the Product table. These validations are integral to the seamless and accurate handling of the purchase process.



**5.6 Admin Supplier List View**  
Within the admin supplier list view, a compilation of registered suppliers is presented. The interface incorporates a search functionality, allowing adminto search based on supplier username and first name. Moreover, a "View Full Profile" button is available for each supplier, enabling administrators to access comprehensive details about the respective supplier.



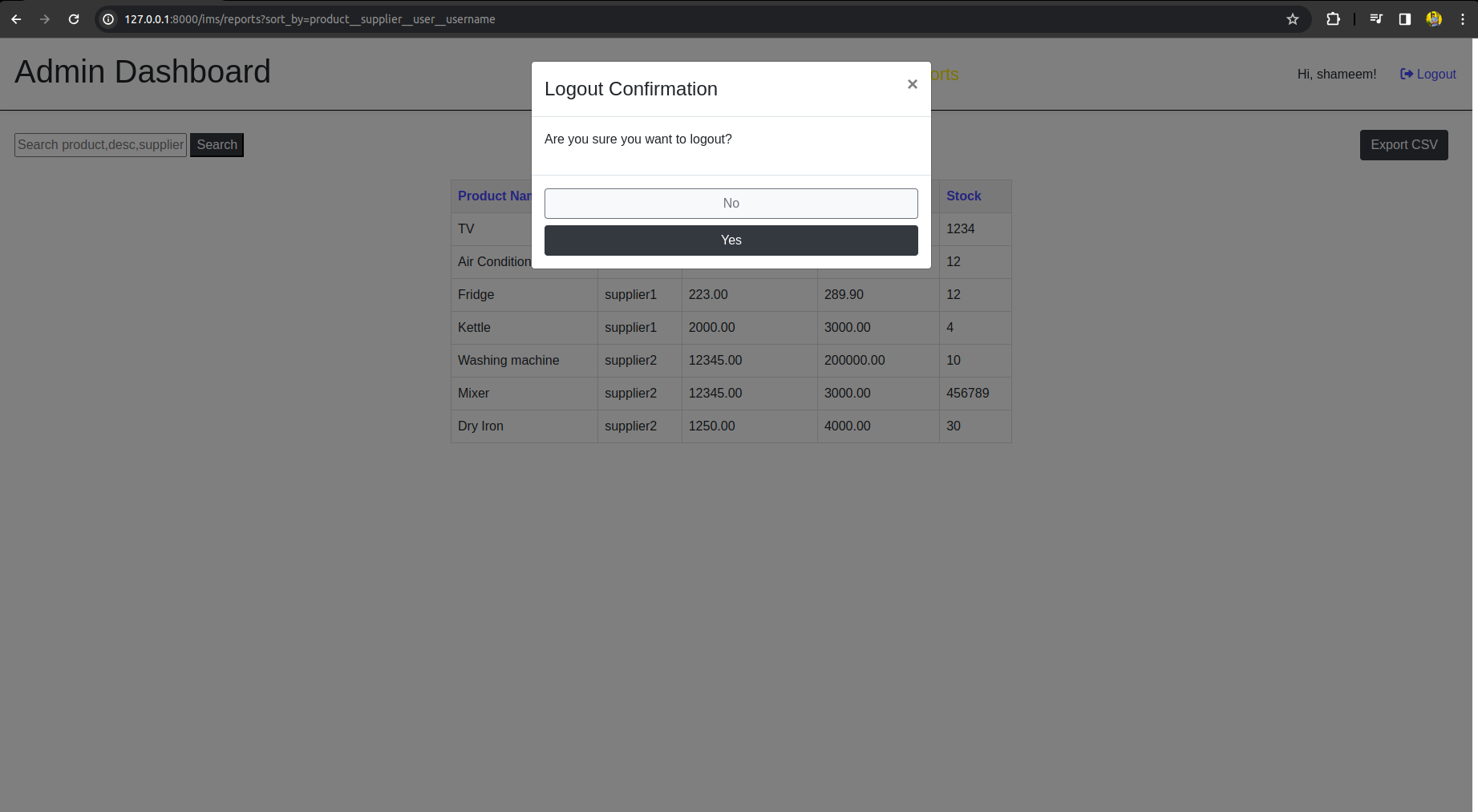
**5.7 Supplier Details Page**  
In the Supplier Details section, administrators have access to comprehensive information about a specific supplier. This includes detailed personal and account-related information. Additionally, there is a product listing specific to that particular supplier, offering a comprehensive view of the products associated with their account.



**5.8 Admin Inventory Report**   
On the Inventory Report Page, administrators have visibility into comprehensive details of all inventory products. The page features a versatile search option allowing the administrator to search using product name, product description, and supplier name. Furthermore, for enhanced usability, there is a sorting mechanism implemented—each table heading is a clickable link, enabling sorting of the list in ascending order based on the selected column when clicked.

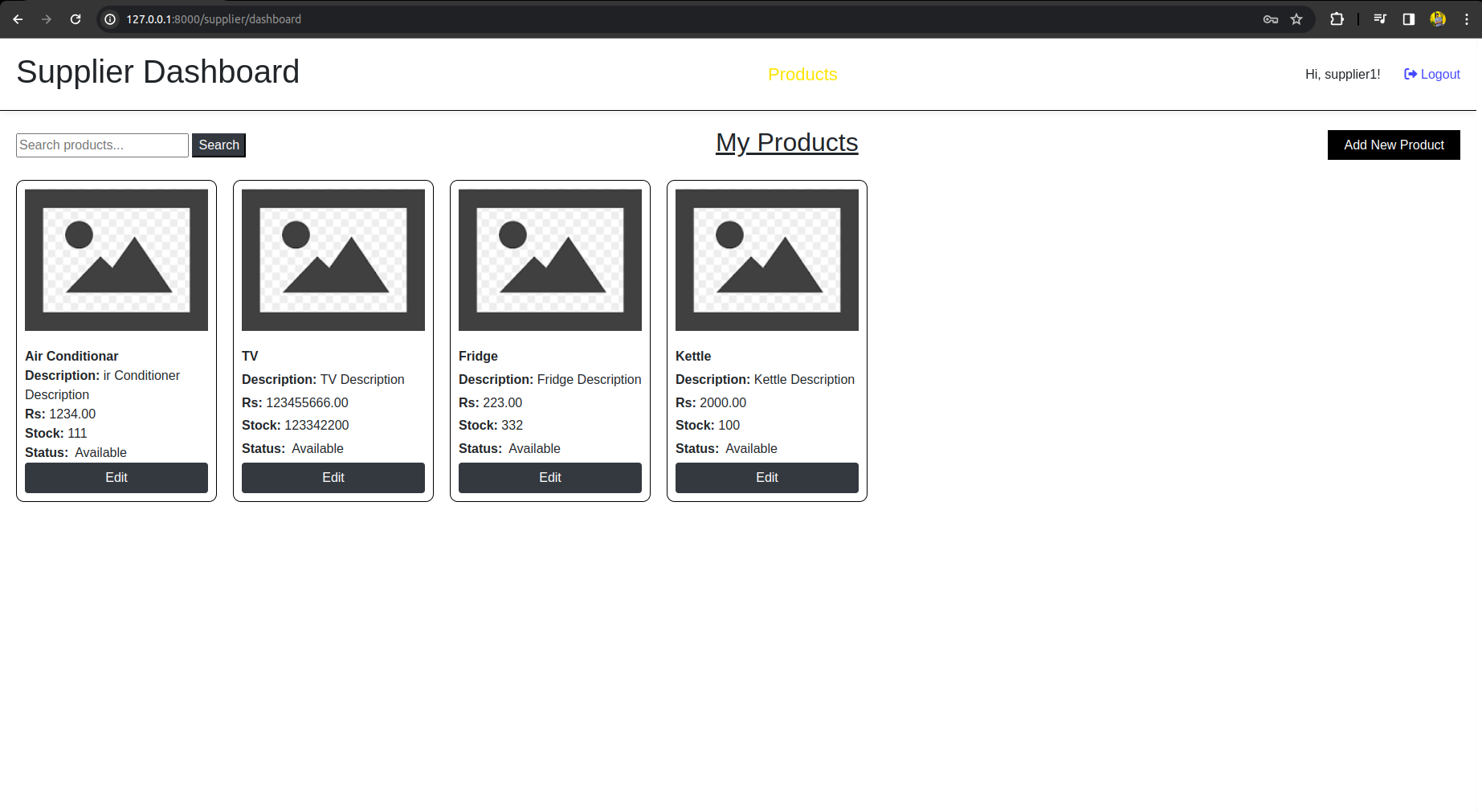
In addition to the search and sorting capabilities, an "**Export CSV**" button is conveniently located at the top right. This functionality empowers administrators to generate CSV files containing either the complete details of the inventory or filtered information based on the current search criteria.

**5.9 Logout**  
In the top-right corner, there is a universal "Logout" option visible to all logged-in users. Upon clicking this option, a confirmation modal appears, ensuring that the user intends to log out before proceeding. This additional step adds a layer of security and prevents accidental logouts.

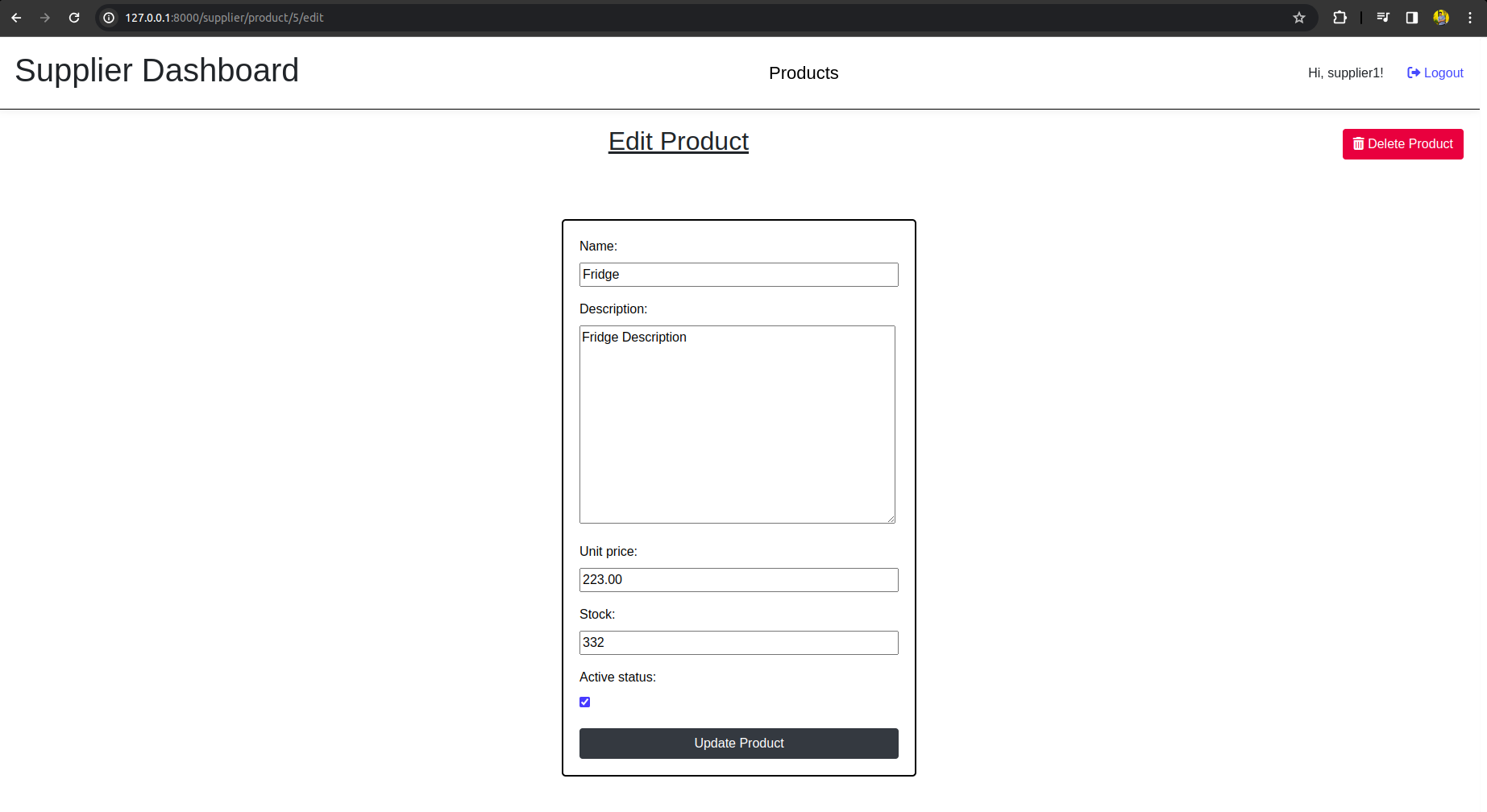


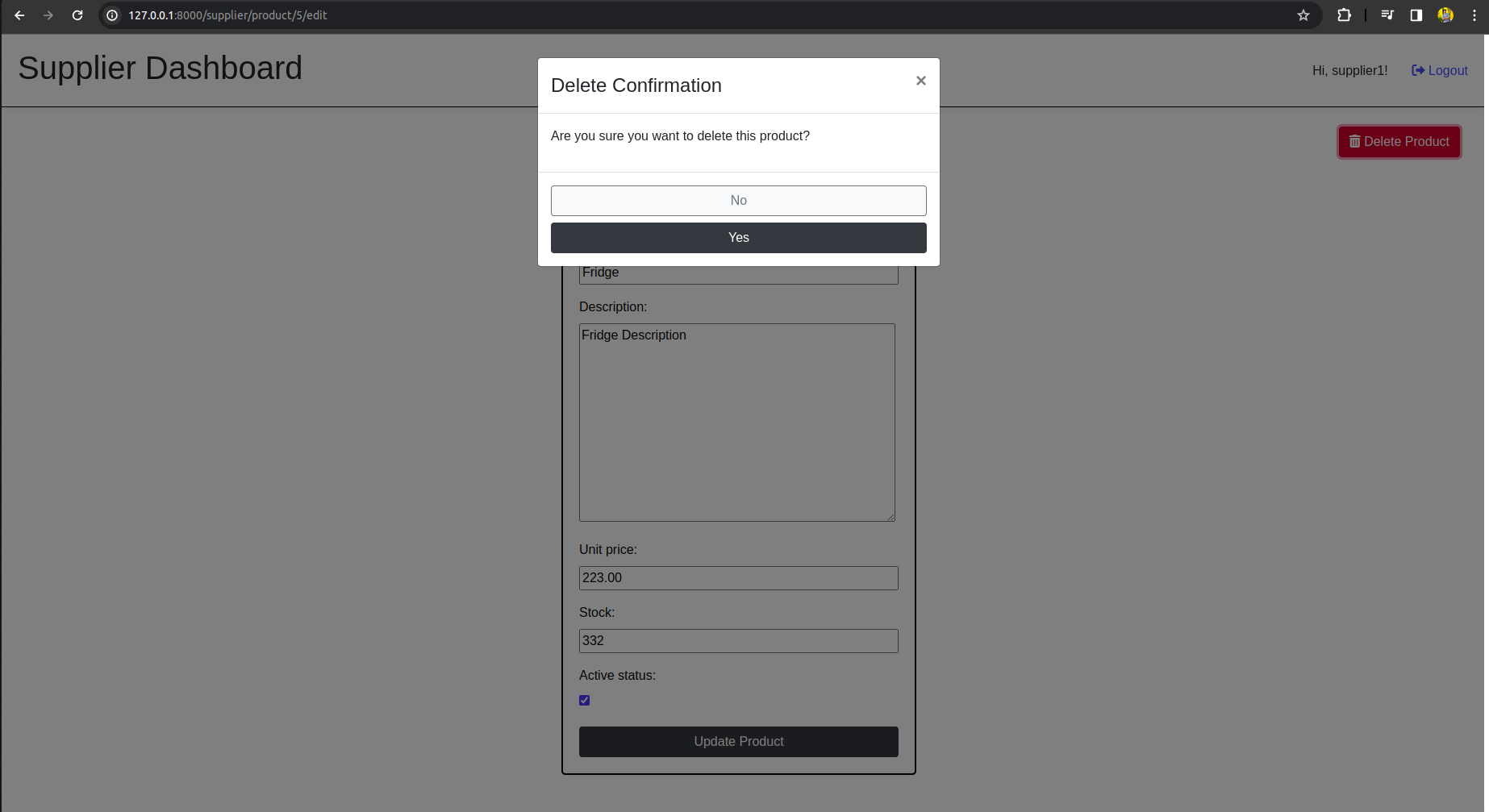
**5.10 Supplier Dashboard**  
Within the Supplier Dashboard, suppliers have a comprehensive overview of their products. Each product entry includes an "Edit" button for quick redirection to its edit page, facilitating efficient management. Additionally, a prominent "Add New" button is available for the convenient addition of new products.

To enhance usability, a search functionality is seamlessly integrated, enabling suppliers to search against product names and descriptions. Furthermore, a dynamic blinking effect is incorporated, drawing attention to products with stock levels that require notice. This ensures that suppliers can easily identify and address products needing attention directly from their dashboard.

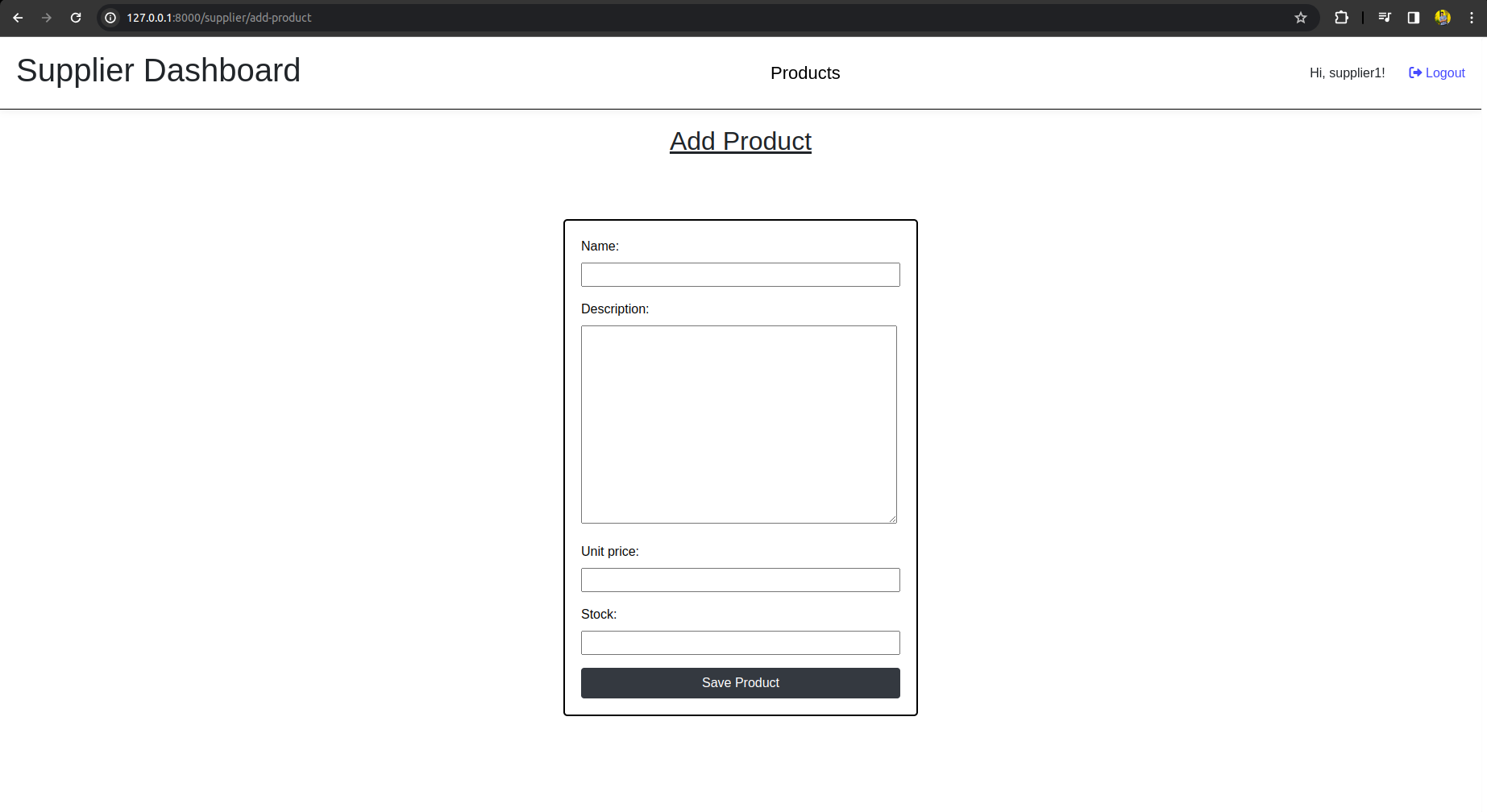


**5.11 Edit Product**In the product editing view for suppliers, they can currently edit only their own products. Additionally, a 'Delete' button is provided, which currently deletes the product entry. However, please note that a future modification is planned to implement a soft delete mechanism instead. This will involve changing the product's status flag to indicate its inactive state, allowing us to retain crucial information for future reference. Furthermore, a confirmation modal has been incorporated to ensure accuracy in deletions and prevent accidental removals.





**5.12 Add Product**  
The Supplier Dashboard includes an "Add Product" view, providing suppliers with a straightforward interface to add new products. In this view, suppliers can easily input basic details to create and incorporate new products into Product table



**6. TestCase :**

**1. TestSupplierModel**

**Purpose:** Tests the Supplier model to ensure proper data validation and constraints.

Test Cases:

* **test\_instance**: Verifies the creation of a Supplier instance with valid data.
* **test\_unique\_user\_is\_enforced**: Ensures that creating two suppliers with the same user raises an IntegrityError.
* **test\_unique\_phone\_number**: Checks that creating a supplier with an existing phone number raises an IntegrityError.
* **test\_phone\_number\_length:** Validates that creating a supplier with an invalid phone number length raises a ValidationError.

**2. TestProductModel**

**Purpose:** Tests the Product model to validate its data requirements and constraints.

Test Cases:

* **test\_instance:** Verifies the creation of a Product instance with valid data.
* **test\_name\_required:** Ensures that creating a product without a name raises a ValidationError.
* **test\_negative\_unit\_price**: Checks that creating a product with a negative unit price raises a ValidationError.
* **test\_negative\_stock\_quantity**: Validates that creating a product with a negative stock quantity raises an IntegrityError.

**3. TestInventoryModel**

**Purpose:** Tests the Inventory model to ensure proper creation and constraints.

Test Cases:

* **test\_instance:** Validates the creation of an Inventory instance with valid data.
* **test\_negative\_selling\_unit\_price:** Ensures that creating an inventory with a negative selling unit price raises a ValidationError.
* **test\_negative\_stock\_quantity:** Checks that creating an inventory with a negative stock quantity raises an IntegrityError.
* **test\_unique\_product:** Verifies that creating two inventories with the same product raises an IntegrityError.

**4. CustomLoginViewTest**

**Purpose**: Tests the login functionality and redirection based on user roles.

Test Cases:

* **test\_login\_redirect\_admin**: Checks if the admin user is redirected to the admin dashboard.
* **test\_login\_redirect\_supplier**: Ensures that the supplier user is redirected to the supplier dashboard.
* **test\_login\_invalid\_credentials**: Validates that login fails with invalid username or password.

**5. AdminDashboardViewTest**

**Purpose:** Tests the admin dashboard view, including context data and access control.

Test Cases:

* **test\_get\_context\_data\_no\_search**: Ensures the correct population of context data without a search query.
* **test\_get\_context\_data\_with\_search**: Validates the context is correctly filtered with a search query.
* **test\_access\_denied\_for\_non\_admin**: Ensures access is denied for non-admin users.
* **test\_unauthorized\_access**: Checks if access is denied for non-logged users.

**6. SupplierDashboardViewTest**

**Purpose:** Tests the supplier dashboard view, including context data and access control.

Test Cases:

* **test\_get\_context\_data\_no\_search**: Ensures the correct population of context data without a search query.
* **test\_get\_context\_data\_with\_search**: Validates the context is correctly filtered with a search query.
* **test\_access\_denied\_for\_non\_supplier:** Ensures access is denied for non-supplier users.

**7. AddProductViewTest**

**Purpose**: Tests the add product view, focusing on access control and product creation.

Test Cases:

* **test\_access\_denied\_for\_anonymous\_user:** Ensures access is denied for an anonymous user.
* **test\_access\_granted\_for\_logged\_in\_supplier**: Checks if access is granted for a logged-in supplier.
* **test\_product\_creation\_success**: Ensures that a product is successfully created when the necessary data is provided.

**7.Conclusion**

In conclusion,This is a simple inventory management system, developed using Django, incorporates two user roles: Admin and Supplier.