

Vendor Registration and Profile Management:

Vendor Registration Form:

Elements:

Company name
Contact person name
Contact information (email, phone number)
Business address
Tax identification number
Product/service offerings
Certifications
Industry experience

Email sent to the vendor who signs up

Validation:

Email validation
Tax ID validation
Required fields validation

Approval Workflow:

Workflow:

Submission of registration form triggers an approval process by internal stakeholders (e.g., procurement team, compliance officer).
Approval/rejection notifications are sent to the vendor.

2. Order Management:

Order List:

Filters: Date range, Order number
Status (pending, in progress, fulfilled)

Details:

Items
Quantities
Delivery dates
Status

Order Details:

Drill-down:
Detailed view of each order including items, quantities, delivery addresses, and status.

Order Status Tracking:

Real-time Updates:
Status updates from receipt to delivery.
Notifications for status changes.

3. Invoice and Payment Management:

Invoice Submission:

Form Fields:

Invoice number

Date

Items/services provided

Total amount

Validation:

Invoice number uniqueness

Date validation

Invoice Approval Workflow:

Approvals:

Invoices undergo approval process by finance or accounts payable department.

Approval/rejection notifications are sent to vendors.

Payment Processing:

Integration:

Integration with accounting systems to initiate payments based on approved invoices.

Status Tracking:

Vendors can track payment status in real-time.

Payment History:

View payment history and details.

To facilitate the uploading of goods and materials by vendors and enable them to manage their stock effectively, you can incorporate inventory management functionalities into the system. Here's how it can work:

Goods and Materials Upload:

Product Catalog Management:

Vendors have access to a product catalog management interface where they can:

Add new products or materials to their catalog.

Update existing product details such as descriptions, quantities, and prices.

Remove discontinued products from the catalog.

Upload Process:

Vendors can upload goods and materials to their catalog by providing:

Product name

Description

Quantity available

Unit price

Images

Validation:

Ensure that uploaded products meet validation criteria such as:

Unique product names

Valid quantities (non-negative integers)

Proper formatting for prices

Stock Management:

Upon uploading goods and materials, the system automatically updates the vendor's inventory levels based on the quantities provided.

Vendors can view their current stock levels for each product in real-time.

Stock Management and Fulfillment:

Inventory Dashboard:

Vendors have access to an inventory dashboard displaying:

List of products/materials

Current stock levels

Reorder points (optional)

Low-stock alerts (if applicable)

Stock Updates:

Vendors can manually update stock levels as goods are received, sold, or returned.

Alternatively, integrate the system with barcode scanners or RFID technology for automated stock updates.

Order Fulfillment:

When an order is placed by a customer, vendors receive notifications containing order details.

Vendors can check their inventory dashboard to ensure they have sufficient stock to fulfill the order.

Stock Availability Alerts:

If stock levels are insufficient to fulfill an order, vendors receive alerts or notifications prompting them to replenish stock or adjust order quantities.

Reordering Process:

Vendors can set up automatic reorder points for products/materials.

When stock levels fall below the reorder point, the system generates reorder alerts or purchase requisitions for vendors to restock their inventory.

Integration with Order Management:

The inventory management system seamlessly integrates with the order management module:

When an order is confirmed, stock levels are automatically deducted from the vendor's inventory.

Vendors can track stock deductions and order fulfillment status in real-time.

Table of Contents

Introduction
System Overview
Functional Requirements
Vendor Dashboard
User Interface
Order Management
Payment and Invoicing
Stock Management
Non-Functional Requirements
System Architecture
High-Level Architecture
Detailed Component Design
Database Design
ER Diagram
Table Descriptions
User Interface Design
Wireframes
User Flows
API Design
Endpoint Specifications
Authentication and Authorization
Development and Deployment Plan
Development Milestones
Deployment Strategy
Testing Plan
Unit Testing
Integration Testing
User Acceptance Testing
Security Considerations
Maintenance and Support

1. Introduction

This document provides a comprehensive overview of the Vendor and User Dashboard System designed for managing goods and materials uploads, facilitating user purchases, and handling payments and stock management.

2. System Overview

The system comprises two primary interfaces: the Vendor Dashboard and the User Interface. Vendors can upload goods and manage their inventory, orders, and payments. Users can browse products, make purchases, and process payments, which are reflected in the vendor's order management system.

3. Functional Requirements

Vendor Dashboard

Goods and Material Upload: Vendors can upload product details including name, description, price, quantity, and images.

Order Management: Vendors can view and manage orders placed by users.

Invoice and Payment Management: Vendors can generate invoices and view payment statuses.

Stock Management: Vendors can track stock levels and update inventory.

User Interface

Product Browsing: Users can view and search for products.

Purchase and Payment: Users can select products, add them to a cart, and proceed to checkout using various payment methods.

Order Tracking: Users can view the status of their orders.

Order Management

Order Processing: Orders placed by users are tracked and managed by vendors.

Status Updates: Order statuses are updated by vendors and reflected in the user interface.

Payment and Invoicing

Payment Gateway Integration: Integration with payment gateways for processing payments.

Invoice Generation: Automatic generation of invoices upon successful payment.

Stock Management

Inventory Tracking: Real-time tracking of stock levels.

Stock Alerts: Notifications for low stock levels.

4. Non-Functional Requirements

Scalability: The system should handle increasing numbers of users and vendors.

Performance: Fast response times for all user interactions.

Security: Secure handling of user and vendor data, especially payment information.

Usability: Intuitive interfaces for both vendors and users.

Reliability: High availability and minimal downtime.

5. System Architecture

High-Level Architecture

Frontend: User Interface (React.js), Vendor Dashboard (React.js)

Backend: REST API (Node.js/Express.js)

Database: SQL (MySQL/PostgreSQL)

Authentication: JWT-based authentication

Detailed Component Design

Frontend Components: Login, Product Listing, Product Detail, Cart, Checkout, Order History, Vendor Upload, Vendor Orders, Vendor Stock.

Backend Services: User Service, Product Service, Order Service, Payment Service, Stock Service.

6. Database Design

ER Diagram

[Include ER Diagram Image]

Table Descriptions

Users: user_id, username, password, email, role

Products: product_id, name, description, price, quantity, vendor_id

Orders: order_id, user_id, product_id, quantity, total_price, status

Payments: payment_id, order_id, user_id, amount, status

Invoices: invoice_id, order_id, user_id, amount, generated_at

Stock: stock_id, product_id, quantity

7. User Interface Design

Wireframes

[Include Wireframe Images]

User Flows

User Registration/Login: New user registration and existing user login flow.

Product Purchase: Browsing products, adding to cart, and completing purchase.

Vendor Upload: Steps for vendors to upload and manage products.

8. API Design

Endpoint Specifications

/api/v1/auth/register: POST, User Registration

/api/v1/auth/login: POST, User Login

/api/v1/products: GET, List Products

/api/v1/products/

: GET, Product Details

/api/v1/cart: POST, Add to Cart

/api/v1/orders: POST, Create Order

/api/v1/payments: POST, Process Payment

/api/v1/vendors/products: POST, Upload Product

/api/v1/vendors/orders: GET, List Vendor Orders

Authentication and Authorization

JWT: Used for securing endpoints and ensuring only authenticated users can access certain functionalities.

9. Development and Deployment Plan

Development Milestones

Phase 1: Basic User Interface and Vendor Dashboard Setup

Phase 2: Backend API Development

Phase 3: Integration and Testing

Phase 4: Deployment and User Testing

Deployment Strategy

Environment Setup: Development, Staging, and Production environments.

CI/CD Pipeline: Automated testing and deployment pipeline using tools like Jenkins or GitHub Actions.

10. Testing Plan

Unit Testing

Tools: Jest, Mocha

Scope: Testing individual components and functions.

Integration Testing

Tools: Postman, Newman

Scope: Testing the interaction between different components and services.

User Acceptance Testing

Scope: Ensuring the system meets the requirements and works as expected for end-users.

11. Security Considerations

Data Encryption: Encrypt sensitive data in transit and at rest.

Authentication: Secure authentication mechanisms using JWT.

Access Control: Role-based access control for different functionalities.

12. Maintenance and Support

Documentation: Comprehensive documentation for end-users and developers.

Support: Dedicated support team for handling issues and updates.

Updates: Regular updates and patches to ensure system security and performance.

User Interface Forms and Form Fields

1. Product Browsing

Product Search Form

Search Bar:

Field Name: search_query

Type: text

Placeholder: Search for products...

Category Filter:

Field Name: category

Type: dropdown

Options: All Categories, Electronics, Clothing, Books, Home & Kitchen, etc.

Price Range Filter:

Field Name: price_range

Type: range

Min: 0

Max: 1000

Step: 10

Sort By:

Field Name: sort_by

Type: dropdown

Options: Relevance, Price: Low to High, Price: High to Low, Newest Arrivals

2. Purchase and Payment

Product Details Page

Quantity Selector:

Field Name: quantity

Type: number

Min: 1

Default: 1

Add to Cart Button:

Field Name: add_to_cart

Type: button

Label: Add to Cart

Shopping Cart Page

Cart Items List:

Fields: product_name, product_image, quantity, price, subtotal

Quantity Updater:

Field Name: quantity

Type: number

Min: 1

Remove Item Button:

Field Name: remove_item

Type: button

Label: Remove

Total Price:

Field Name: total_price

Type: label

Checkout Form

Shipping Information:

Name:

Field Name: shipping_name

Type: text

Address:

Field Name: shipping_address

Type: text

City:

Field Name: shipping_city

Type: text

State:

Field Name: shipping_state

Type: text

Postal Code:

Field Name: shipping_postal_code

Type: text

Country:

Field Name: shipping_country

Type: text

Phone Number:

Field Name: shipping_phone

Type: text

Billing Information (if different from Shipping):

Checkbox to indicate same as shipping:

Field Name: same_as_shipping

Type: checkbox

Name:

Field Name: billing_name

Type: text

Address:

Field Name: billing_address
Type: text
City:
Field Name: billing_city
Type: text
State:
Field Name: billing_state
Type: text
Postal Code:
Field Name: billing_postal_code
Type: text
Country:
Field Name: billing_country
Type: text
Phone Number:
Field Name: billing_phone
Type: text

Payment Information

Credit/Debit Card Information:

Card Number:

Field Name: card_number
Type: text

Cardholder Name:

Field Name: cardholder_name
Type: text

Expiration Date:

Field Name: expiration_date
Type: text

CVV:

Field Name: cvv
Type: text

Alternative Payment Methods:

PayPal:

Field Name: paypal_email
Type: text

Other Methods: As per integration

Submit Order Button:

Field Name: submit_order
Type: button
Label: Place Order

3. Order Tracking

Order Tracking Form

Order ID:

Field Name: order_id

Type: text

Placeholder: Enter your order ID

Email:

Field Name: email

Type: email

Placeholder: Enter your email

Order Status Display

Order Details:

Fields: order_id, product_name, quantity, price, order_date

Shipping Status:

Field Name: shipping_status

Type: label

Values: Processing, Shipped, Out for Delivery, Delivered

Database Design

1. Product Browsing

Products Table

product_id: INT, Primary Key, Auto-increment

name: VARCHAR(255), Not Null

description: TEXT, Not Null

price: DECIMAL(10, 2), Not Null

quantity: INT, Not Null

category_id: INT, Foreign Key (references categories.category_id)

image_url: VARCHAR(255), Not Null

created_at: TIMESTAMP, Default: CURRENT_TIMESTAMP

updated_at: TIMESTAMP, Default: CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP

Categories Table

category_id: INT, Primary Key, Auto-increment

name: VARCHAR(255), Not Null

2. Purchase and Payment

Users Table

user_id: INT, Primary Key, Auto-increment

username: VARCHAR(255), Not Null, Unique

password: VARCHAR(255), Not Null

email: VARCHAR(255), Not Null, Unique

created_at: TIMESTAMP, Default: CURRENT_TIMESTAMP

updated_at: TIMESTAMP, Default: CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP

Orders Table

order_id: INT, Primary Key, Auto-increment
user_id: INT, Foreign Key (references users.user_id)
total_price: DECIMAL(10, 2), Not Null
status: VARCHAR(50), Default: 'Processing'
shipping_name: VARCHAR(255), Not Null
shipping_address: TEXT, Not Null
shipping_city: VARCHAR(255), Not Null
shipping_state: VARCHAR(255), Not Null
shipping_postal_code: VARCHAR(20), Not Null
shipping_country: VARCHAR(255), Not Null
shipping_phone: VARCHAR(20), Not Null
billing_name: VARCHAR(255), Not Null
billing_address: TEXT, Not Null
billing_city: VARCHAR(255), Not Null
billing_state: VARCHAR(255), Not Null
billing_postal_code: VARCHAR(20), Not Null
billing_country: VARCHAR(255), Not Null
billing_phone: VARCHAR(20), Not Null
payment_method: VARCHAR(50), Not Null
created_at: TIMESTAMP, Default: CURRENT_TIMESTAMP
updated_at: TIMESTAMP, Default: CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP

Order Items Table

order_item_id: INT, Primary Key, Auto-increment
order_id: INT, Foreign Key (references orders.order_id)
product_id: INT, Foreign Key (references products.product_id)
quantity: INT, Not Null
price: DECIMAL(10, 2), Not Null

Payments Table

payment_id: INT, Primary Key, Auto-increment
order_id: INT, Foreign Key (references orders.order_id)
amount: DECIMAL(10, 2), Not Null
payment_status: VARCHAR(50), Default: 'Pending'
payment_method: VARCHAR(50), Not Null
transaction_id: VARCHAR(255), Unique
created_at: TIMESTAMP, Default: CURRENT_TIMESTAMP

3. Order Tracking

Order Tracking Fields

order_id: Referencing the Orders table to fetch the details
user_id: Referencing the Users table for user-specific orders
status: From the Orders table to track the current status of the order
shipping_status: Included in the Orders table to indicate various stages such as 'Processing', 'Shipped', 'Out for Delivery', 'Delivered'

Goods and Material Upload

Form Fields:

Product Name:

Input type: Text

Placeholder: "Enter product name"

Product Description:

Input type: Text area

Placeholder: "Enter product description"

Price:

Input type: Number

Placeholder: "Enter product price"

Quantity:

Input type: Number

Placeholder: "Enter product quantity"

Images:

Input type: File (multiple)

Placeholder: "Upload product images"

Order Management

Form Fields:

Order ID:

Input type: Text (Read-only)

Placeholder: "Order ID"

Customer Name:

Input type: Text (Read-only)

Placeholder: "Customer name"

Product Details:

Input type: Text area (Read-only)

Placeholder: "Product details"

Order Date:

Input type: Date (Read-only)

Placeholder: "Order date"

Order Status:

Input type: Dropdown

Options: ["Pending", "Processing", "Shipped", "Delivered", "Cancelled"]

Tracking Number:

Input type: Text

Placeholder: "Enter tracking number"

Invoice and Payment Management

Form Fields:

Invoice ID:

Input type: Text (Read-only)

Placeholder: "Invoice ID"

Order ID:
Input type: Text (Read-only)
Placeholder: "Order ID"
Customer Name:
Input type: Text (Read-only)
Placeholder: "Customer name"
Product Details:
Input type: Text area (Read-only)
Placeholder: "Product details"
Total Amount:
Input type: Number (Read-only)
Placeholder: "Total amount"
Payment Status:
Input type: Dropdown
Options: ["Pending", "Paid", "Failed"]

Stock Management

Form Fields:

Product ID:
Input type: Text (Read-only)
Placeholder: "Product ID"
Product Name:
Input type: Text (Read-only)
Placeholder: "Product name"
Current Stock Level:
Input type: Number (Read-only)
Placeholder: "Current stock level"
Update Stock Quantity:
Input type: Number
Placeholder: "Enter quantity to add/subtract"
Stock Status:
Input type: Dropdown
Options: ["In Stock", "Out of Stock"]

Vendor Dashboard

1. Vendor Dashboard

1.1 Goods and Material Upload

Product Details Upload: Vendors can upload details of their products through a form. The form includes fields for:

Name: The name of the product.

Description: A detailed description of the product, including specifications and features.

Price: The cost of the product.

Quantity: The number of units available for sale.

Images: Vendors can upload multiple images to showcase the product from different angles.

Process Flow:

Vendor logs into the dashboard.
Navigates to the 'Goods and Material Upload' section.
Fills in the product details in the provided form.
Uploads images and submits the form.
The product is added to the inventory and is visible to users for browsing.

1.2 Order Management

View and Manage Orders: Vendors can access a comprehensive list of orders placed by users, including details such as:

Order ID: Unique identifier for each order.
Product Details: Information about the products ordered.
Customer Information: Details of the customer who placed the order.
Order Status: Current status of the order (e.g., pending, processed, shipped, delivered).

Process Flow:

Vendor logs into the dashboard.
Navigates to the 'Order Management' section.
Views the list of orders and their details.
Updates the status of orders as they are processed (e.g., from 'pending' to 'processed').
Communicates with customers if needed (e.g., for order confirmation or shipping details).

1.3 Invoice and Payment Management

Generate Invoices: Vendors can generate invoices for each order, which include:

Invoice Number: Unique identifier for the invoice.
Order Details: Summary of the order.
Billing Information: Customer's billing details.
Payment Status: Indicates whether the payment has been received.
View Payment Statuses: Vendors can track the payment status of each order (e.g., pending, paid, refunded).

Process Flow:

Vendor logs into the dashboard.
Navigates to the 'Invoice and Payment Management' section.
Selects an order and generates an invoice.
Tracks payment status and updates the system accordingly.
Sends invoice to the customer via email or makes it available for download.

1.4 Stock Management

Track Stock Levels: Vendors can monitor the quantity of products in their inventory in real-time.

Update Inventory: Vendors can update the stock levels when new inventory is added or when stock is sold.

Process Flow:

Vendor logs into the dashboard.
Navigates to the 'Stock Management' section.
Views current stock levels for all products.
Updates stock levels manually or through batch upload.
Sets alerts for low stock levels to receive notifications when stock needs replenishment.

2. User Interface

2.1 Product Browsing

View Products: Users can browse products uploaded by vendors. Each product listing includes:

Product Name: The name of the product.
Image: A thumbnail image of the product.
Price: The cost of the product.
Short Description: A brief overview of the product.
Search and Filter Options: Users can search for products by name, category, price range, and other filters.

Process Flow:

User navigates to the product browsing page.
Uses search and filter options to find desired products.
Clicks on a product to view detailed information.

2.2 Purchase and Payment

Select Products: Users can select products and add them to a shopping cart.

Checkout Process:

Review Cart: Users review the items in their cart.
Enter Shipping Information: Users provide shipping details.
Select Payment Method: Users choose from various payment methods (e.g., credit card, PayPal).
Confirm Order: Users review the order summary and confirm the purchase.

Process Flow:

User selects a product and adds it to the cart.
Navigates to the cart and reviews the items.
Proceeds to checkout and enters shipping information.
Selects a payment method and completes the payment process.
Receives an order confirmation and invoice.

2.3 Order Tracking

View Order Status: Users can track the status of their orders through the 'Order History' section.

Order Details: Each order includes details such as:

Order ID: Unique identifier for the order.

Product Details: Information about the ordered products.

Shipping Status: Current status of the shipment (e.g., pending, shipped, delivered).

Process Flow:

User logs into their account.

Navigates to the 'Order History' section.

Selects an order to view detailed status and tracking information.

3. Order Management

3.1 Order Processing

Track Orders: Orders placed by users are tracked in the vendor's dashboard.

Update Order Status: Vendors update the status of orders as they are processed, which is reflected in the user's order history.

Process Flow:

Vendor receives a new order notification.

Navigates to the 'Order Management' section and views the order details.

Processes the order and updates the status (e.g., from 'pending' to 'shipped').

User receives updates on the order status.

3.2 Status Updates

Real-Time Updates: Order statuses are updated in real-time, providing users with current information about their order.

Process Flow:

Vendor updates the order status in the dashboard.

The updated status is reflected in the user's order history.

User receives a notification about the status change.

4. Payment and Invoicing

4.1 Payment Gateway Integration

Secure Payment Processing: Integration with trusted payment gateways (e.g., Gpay, PhonePay) to process payments securely.

Multiple Payment Methods: Users can choose from various payment methods during checkout.

Process Flow:

User selects a payment method during checkout.
Payment is processed through the integrated payment gateway.
User and vendor receive payment confirmation.

4.2 Invoice Generation

Automatic Invoicing: Invoices are automatically generated upon successful payment and sent to the user.

Process Flow:

User completes the payment.
An invoice is generated and sent to the user via email or made available for download.
Vendor can view the generated invoice in the dashboard.

5. Stock Management**5.1 Inventory Tracking**

Real-Time Tracking: Vendors can monitor stock levels in real-time, ensuring accurate inventory management.

Process Flow:

Vendor logs into the dashboard.
Navigates to the 'Stock Management' section to view current stock levels.
Updates inventory when new stock is added or when stock is sold.

5.2 Stock Alerts

Low Stock Notifications: Vendors receive alerts when stock levels fall below a predefined threshold, prompting them to reorder.

Process Flow:

Vendor sets low stock threshold for each product.
System monitors stock levels and triggers an alert when levels fall below the threshold.
Vendor receives the alert and takes action to replenish stock.
This detailed documentation outlines the functionalities and workflows for the Vendor Dashboard and User Interface on the AECearth platform, ensuring a comprehensive understanding of how these features operate and interact with users and vendors.