**High-Level Design (HLD)**

**1. Supplier Registration Form**

**Overview**

The Supplier Registration module allows suppliers to enter their business details and submit necessary documents for verification.

**Key Components**

* **Frontend:** Registration form with real-time validation feedback.
* **Backend:** Handles form submission, document storage, and status updates.
* **Database:** Stores supplier details and documents securely.
* **Notification System:** Sends confirmation messages to suppliers upon submission.

**Workflow**

1. The supplier accesses the registration form.
2. Enters company details, uploads required documents, and submits the form.
3. Backend validates input and securely stores data.
4. Supplier receives a confirmation message.
5. Data is sent to administrators for verification.

**2. Supplier Profile Verification**

**Overview**

Administrators review and verify supplier profiles to ensure legitimacy.

**Key Components**

* **Frontend:** Supplier verification dashboard for administrators.
* **Backend:** Fetches pending profiles, verifies documents, and updates the supplier list.
* **Database:** Stores verification status (Pending, Verified, Rejected).
* **Notification System:** Sends approval/rejection messages to suppliers.

**Workflow**

1. Admins log in and access a list of pending supplier profiles.
2. Review uploaded documents and supplier details.
3. Choose **Verify** (profile gets listed) or **Reject** (requires rejection reason).
4. System updates verification status in the database.
5. Supplier receives an email/notification about the decision.

**3. Purchase Order Creation**

**Overview**

Procurement managers create, manage, and track purchase orders electronically.

**Key Components**

* **Frontend:** Purchase Order creation form and tracking dashboard.
* **Backend:** Processes purchase order submissions and manages order statuses.
* **Database:** Stores purchase order details, supplier links, and status updates.
* **Notification System:** Sends confirmation emails/alerts upon creation and updates.

**Workflow**

1. Procurement manager selects a supplier and enters order details.
2. Saves the order as a draft or submits it for approval.
3. Submitted orders are stored in the database with status tracking.
4. Confirmation notifications are sent.
5. Managers track progress through the dashboard (Pending, Approved, Completed).

**Low-Level Design (LLD)**

**1. Supplier Registration Form**

**Frontend Components**

* **SupplierRegistrationForm.jsx**
  + Fields: Company Name, Contact Info, Address, Document Upload
  + Validation logic for required fields
  + API integration for submission

**Backend Endpoints**

* POST /api/suppliers/register
  + Validates form input
  + Stores supplier details in the database
  + Sends confirmation response
* GET /api/suppliers/{id}
  + Fetch supplier details for verification

**Database Schema (Supplier Table)**

CREATE TABLE suppliers (

supplier\_id INT PRIMARY KEY AUTO\_INCREMENT,

company\_name VARCHAR(255) NOT NULL,

contact\_email VARCHAR(255) UNIQUE NOT NULL,

address TEXT NOT NULL,

documents JSON NOT NULL,

status ENUM('Pending', 'Verified', 'Rejected') DEFAULT 'Pending'

);

**2. Supplier Profile Verification**

**Frontend Components**

* **AdminDashboard.jsx**
  + Displays a list of pending supplier profiles
  + "Verify" and "Reject" buttons with a modal for rejection reasons

**Backend Endpoints**

* GET /api/admin/suppliers/pending
  + Fetches all suppliers with "Pending" status
* POST /api/admin/suppliers/verify/{id}
  + Updates supplier status to "Verified"
* POST /api/admin/suppliers/reject/{id}
  + Updates status to "Rejected" with a reason

**Database Schema Updates**

ALTER TABLE suppliers ADD COLUMN rejection\_reason TEXT NULL;

**3. Purchase Order Creation**

**Frontend Components**

* **PurchaseOrderForm.jsx**
  + Fields: Supplier Selection, Items, Quantity, Cost
  + Status tracking UI for submitted orders

**Backend Endpoints**

* POST /api/purchase\_orders
  + Creates a new purchase order
* GET /api/purchase\_orders
  + Fetches all orders with status filtering
* PUT /api/purchase\_orders/{id}
  + Updates order status

**Database Schema (Purchase Order Table)**

CREATE TABLE purchase\_orders (

order\_id INT PRIMARY KEY AUTO\_INCREMENT,

supplier\_id INT NOT NULL,

items JSON NOT NULL,

total\_cost DECIMAL(10,2) NOT NULL,

status ENUM('Pending', 'Approved', 'Completed') DEFAULT 'Pending',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (supplier\_id) REFERENCES suppliers(supplier\_id)

);

**Conclusion**

* **HLD** outlines the overall system architecture, workflows, and main components.
* **LLD** details API endpoints, frontend components, and database schemas.

Would you like to add more functionalities or refine any part of this design? 🚀