**Description**

This is a vendor management system task that involves creating, listing, updating, and deleting vendor details based on user input. Using the Django framework, I created a vendor app to display these details. Users can create vendors by providing various arguments, which are then stored in the backend database.This task was assigned by FATMUG.

**Plan**

* Start and Planning:Begin by comprehensively understanding the project scope and establishing the foundational structure.
* Vendor Setup: Create a robust system for effectively managing vendor information and thoroughly test its functionality.
* Purchase Order Management: Develop purchase order forms and verify their functionality to ensure smooth transaction processes.
* Performance Evaluation: Implement a scoring mechanism for vendors and dynamically calculate their performance scores.
* Historical Tracking: Integrate a feature for tracking vendor performance history over time, enabling better decision-making.
* Testing Phase: Conduct rigorous testing to identify and rectify any issues or bugs present in the system.
* Deployment and Maintenance: Launch the system online for user accessibility and utilization.Continuously monitor the system, address any issues promptly, and provide ongoing support to users.

**Vendor Profile Management**

* POST /api/vendors/: Create a new vendor.
* GET /api/vendors/: List all vendors.
* PUT /api/vendors/{vendor\_id}/: Update a vendor's details.
* DELETE /api/vendors/{vendor\_id}/: Delete a vendor.
* UPDATE /api/vendors/{vendor\_id}/: Update a vendor.
* POST /api/contact/:Contact Us

**Purchase Order Tracking**

* POST /api/purchase\_orders/: Create a new order.
* GET /api/purchase\_orders/: List all orders.
* PUT /api/purchase\_orders/{vendor\_id}/: Update the order's details.
* DELETE /api/purchase\_orders/{vendor\_id}/: Delete an order.
* UPDATE /api/purchase\_orders/{vendor\_id}/: Update an order.

**Tech Components**

Stack: Python3, Django, dbsqlite3

Server: localhost:8000

**Installations**

pip install django

pip install djangorestframework

pip install -r requirements.txt

python manage.py makemigrations

python manage.py migrate

python manage.py createsuperuser

python manage.py runserver

http://localhost:8000/vendors/ --for vendor list

http://localhost:8000/admin --for admin

**Screenshots**

1. **Homepage**: The initial landing page of the website.

2. **Vendor Creation**: Allows users to input basic details to create a vendor profile.

3. **Vendor List Display**: After adding a vendor, users can view a list of all vendors.

4. **Vendor Details Update**: Users can update existing vendor details, and these changes are immediately reflected in the vendor list.

5. **Vendor Deletion**: Provides functionality to delete vendor details, with the changes reflected in the vendor list.

6. **Contact Us Form**: A form where users can submit inquiries or messages to the website administrators.

7. **Administration Page**:

- Historical Performances: A section displaying historical performance data.

- Purchase Orders: Information related to purchase orders.

- Vendors: A section dedicated to managing vendor-related information.

8. **Vendors List in Administration**: Displays a list of vendors within the administration page, likely for management and oversight purposes.