CSE202 | DBMS

ElectroBase Management System

EBMS - An Online Electronics Retail Store

Divyajeet Singh

Mehar Khurana

2023

2023

Abstract 2
Stakeholders and their roles....... 3
Entity Relationships 4
Database Schema & Integrity Constraints....................... 5
SQL Queries...................... 6
Project Snapshots 7
User Guide....................... 12
Source Code...................... 13

1

ElectroBase Management System

Abstract

Electronics has always been a booming industry. With the advent of the internet, the industry has seen a massive shift towards online retail. However, it is difficult to keep track of the technical requirements of a store and keep all stakeholders involved and updated.

This is where we come in with **EBMS**, the **ElectroBase Management System**. EBMS is an online retail platform for electronics. It aims to provide a common platform for customers, suppliers, store managers, and delivery agents.

- It is an easy solution for the **customers**, as it aims to provide a diverse catalogue of products to customers. The customers get to choose from a wide range of categories, make changes to their cart, and make secure payments with a method of their choice.
- The **suppliers** get to keep track of their products and edit their description, price, etc. as and when required. The application also manages and generates their sales statistics.
- The database/store managers (**admins**) get assisted in monitoring the transactions and managing the inventory. Based on the requirements, they can add deals or combos on the available products or remove categories from their store.
- EBMS provides a platform for **delivery agents** to manage all orders that have been assigned to them. They can toggle their activity status and view the feedback given to them.

The primary focus of the project is to design an efficient back-end. We aim to create a system that is smooth and easy to use for the customers and easy to manage for the suppliers. The system should support efficient searching through the catalogue and should be able to handle a large number of transactions.

The aim of this project is to bring to life an integrated online retail store for electronics. The project will bring all stakeholders on a common platform and will ensure a smooth and easy-to-use experience for the customers.

Through our project, we attempt to model a real-world problem and provide a solution. Hence, this full-stack project will involve usage of various technologies and tools to create a robust and efficient system. This involves usage of the **Flask Micro-Framework (Python-3)** for front-end development and **MySQL** for back-end development.

Stakeholders and their roles

Stakeholders are the people who will be directly or indirectly affected by the project. They are the people who will be using the system, and hence, their requirements are of utmost importance.

Stakeholder: Customers

The customers are the primary stakeholders who will be interacting with the database. A customer will first have to register and log in. Once done, they can start adding products to their cart, track their orders, and make secure paymments. The application will also allow them to view their transaction history. Some customer-related actions are as follows:

- Browse through the product catalogue
- Add products to cart
- Make secure payments
- Track orders and view transaction history

Stakeholder: Suppliers

Through EBMS, suppliers will be able to supply products with different names, prices, descriptions, etc. to the catalogue. They will also be able to view their sales statistics generated by the application. Some supplier-related actions are as follows:

- Add/remove products to/from the catalogue
- Modify product details like price and quantity
- View sales statistics

Stakeholder: Delivery Agents

The delivery agents will be responsible for delivering the products to the customers. They will be able to view the orders assigned to them and toggle their activity status. Some delivery agent-related actions are as follows:

- View orders assigned to them
- Toggle activity status
- View feedback

Stakeholder: Store Managers (Admins)

The store managers are responsible for handling, maintaining, and regulating our database. They will have the ability to add discounts on products, create combos/deals, remove products from stock, etc. The application will also allow them to keep track of the sales. Some admin-related actions are as follows:

- Remove products from the catalogue
- Add/modify discounts on products
- Create combos/deals on groups of products
- View transaction history

Stakeholder: Electronics Retail Stores

As secondary stakeholders, the other electronics retail stores will not directly interact with the database. However, they will certainly be affected by the project, as it provides a competitive platform for them. Though they are not directly querying the database, they will be indirectly affected by the success of the project.

Stakeholder: Financial Instruments

Financial Instruments include gateways like RazorPay, UPI, Credit/Debit Card providers. The receipts generated by payments could be looked up by them, users, and admins.

Entity Relationships

Entities are the objects that are represented in the database. They are (usually) the real-world objects that are being modelled. How entities interact with each other is governed by the relationships among them. Relational diagrams for the database design can be found in the Progress Report for the project.

Entity: `admin`

```
admin( id, username, password )
```

Entity: `customer`

```
customer(
  id, email, password, name(first_name, middle_initial, last_name),
  address(street(street_name, apt_number), city, zip, state, country),
  age, { phone_number }
)
```

Entity: supplier

```
supplier(
   id, email, password, name(first_name, middle_initial, last_name),
   address(street(street_name, apt_number), city, zip, state, country)
)
```

Entity: delivery_agent

```
delivery_agent(
   id, email, password, name(first_name, middle_initial, last_name),
   availability, { phone_number }
)
```

Entity: product

```
product( id, name, supplier_id, price, quantity, product_description )
```

Entity: order

```
order( id, customer_id, da_id, order_date, delivery_date )
```

Entity: (Weak) wallet

```
wallet( id(customer), balance, upi_id )
```

Relationship: product_review(customer, product)

```
product_review( id(customer), id(product), rating, content, review_date )
```

Relationship: da_review(customer, delivery_agent)

```
da_review( id(customer), id(delivery_agent), rating, content, review_date )
```

Relationship: cart(customer, product)

```
cart( id(customer), id(product), quantity )
```

Relationship: order_product(order, product)

```
order_product( id(order), id(product), quantity )
```

Database Schema & Integrity Constraints

Most of the data for the entity tables was generated through Mockaraoo. Mockaroo was used to implement viable constraints on the data being generated. Since the records of the relationship tables are dependent on the records of the entity tables, they were generated using Python scripts.

The complete database schema and generated records, along with the generator, can be found in the `Database-Generation/` directory in the project repository.

SQL Queries

As previously stated, the back-end of the project is handled in MySQL Server; naturally, the queries are written in SQL. The project produced use cases that required complicated and challenging involving the use of subqueries, joins, and aggregate functions. A few examples of such queries are given below.

SQL Query: Embedded OLAP Query

The following query is used to find out the trends in the number of orders on a monthly, quarterly, and yearly basis.

```
SELECT
    YEAR(order.order_date) AS date_year,
    QUARTER(order.order_date) AS date_quarter,
    MONTH(order.order_date) AS date_month,
    COUNT(order.orderID) AS order_count,
    SUM(order_product.price * order_product.quantity) AS revenue
FROM order
JOIN order_product ON order.order_id = order_product.order_id
JOIN product ON order_product.product_id = product.product_id
GROUP BY date_year, date_quarter, date_month WITH ROLLUP
ORDER BY date_year DESC, date_month DESC;
```

SQL Query: Embedded Simple SQL Query

The following query gets a small list of inactive suppliers, i.e. the suppliers who have not supplied any products.

```
SELECT supplier.supplier_id, supplier.email
FROM supplier
WHERE NOT EXISTS (
    SELECT * FROM product WHERE product.supplier_id = supplier.supplier_id
)
LIMIT 10;
```

SQL Query: Embedded Simple SQL Query

The following query fetches a list of top-rated products.

```
SELECT product.product_id, product.name, AVG(product_review.rating) as avg_rating
FROM product_review, product
WHERE product.product_id = product_review.product_id
GROUP BY product.product_id
HAVING avg_rating >= 4.5
ORDER BY avg_rating DESC;
```

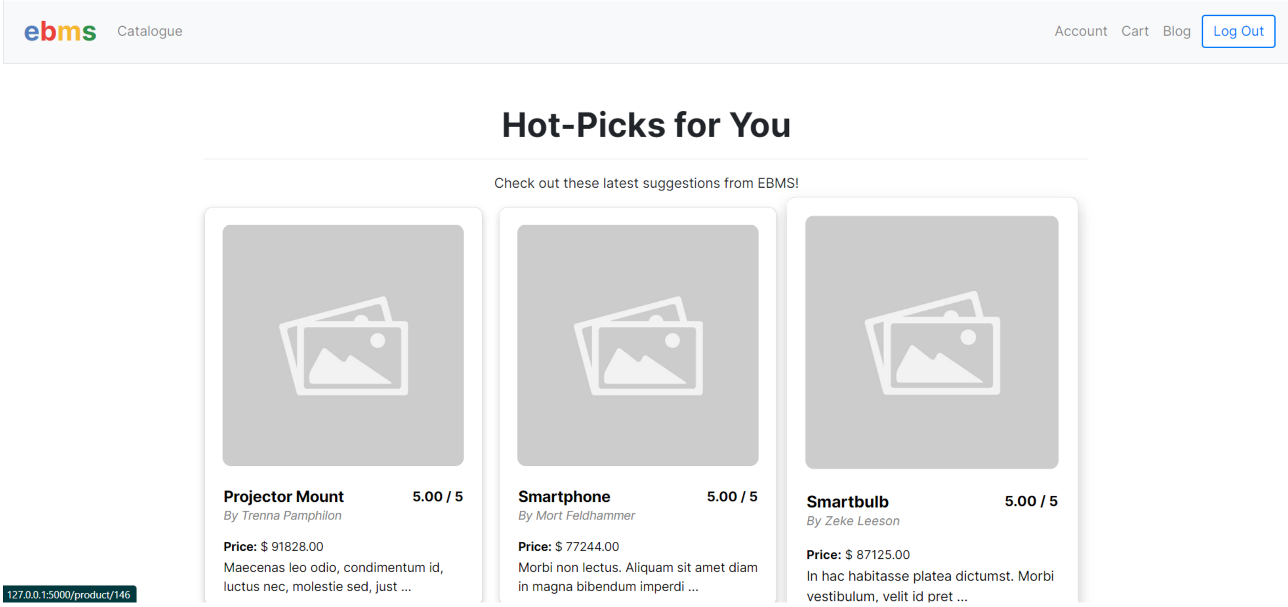
SQL Query: Database Trigger

The following trigger is used to create a wallet for a new customer as soon as they register.

```
DROP TRIGGER IF EXISTS create_wallet;
DELIMITER $$
CREATE TRIGGER create_wallet AFTER INSERT ON customer
FOR EACH ROW
BEGIN
    IF NOT EXISTS (SELECT * FROM wallet WHERE customer_id = NEW.customer_id) THEN
        INSERT INTO wallet (customer_id, balance) VALUES (NEW.customer_id, 0);
    END IF;
END;
$$
DELIMITER ;
```

A comprehensive collection of almost all SQL Queries used in the project can be found in the `SQL/` directory in the project repository.

Project Snapshots



Hot-Picks for You

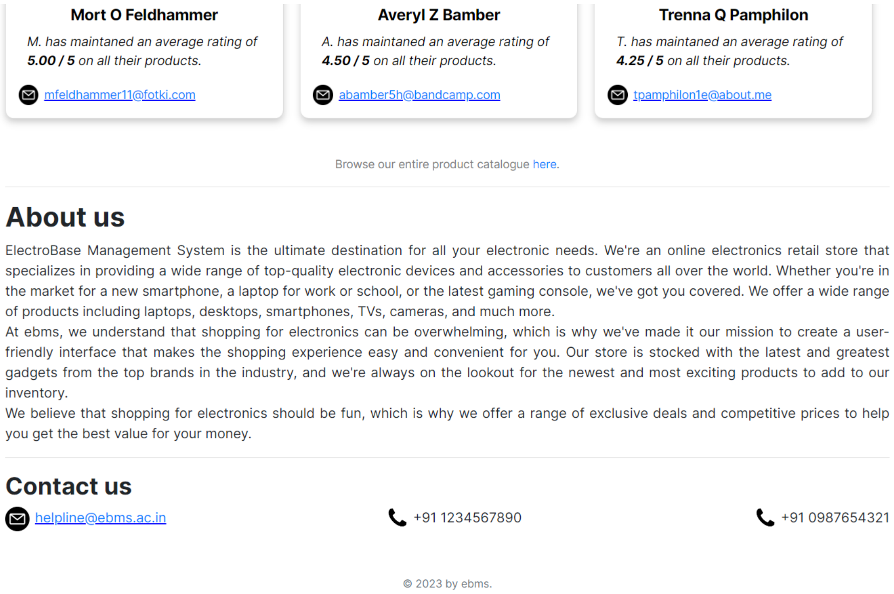
Check out these latest suggestions from EBMS!

Projector Mount **5.00 / 5**
By Trenna Pamphilon
Price: $ 91828.00
Maecenas leo odio, condimentum id, luctus nec, molestie sed, just ...

Smartphone **5.00 / 5**
By Mort Feldhammer
Price: $ 77244.00
Morbi non lectus. Aliquam sit amet diam in magna bibendum imperdi ...

Smartbulb **5.00 / 5**
By Zeke Leeson
Price: $ 87125.00
In hac habitasse platea dictumst. Morbi vestibulum, velit id pret ...

Figure 1.1: Home Page



Mort O Feldhammer

M. has maintaned an average rating of ***5.00 / 5*** *on all their products.*

✉ mfeldhammer11@fotki.com

Averyl Z Bamber

A. has maintaned an average rating of ***4.50 / 5*** *on all their products.*

✉ abamber5h@bandcamp.com

Trenna Q Pamphilon

T. has maintaned an average rating of ***4.25 / 5*** *on all their products.*

✉ tpamphilon1e@about.me

Browse our entire product catalogue here.

About us

ElectroBase Management System is the ultimate destination for all your electronic needs. We're an online electronics retail store that specializes in providing a wide range of top-quality electronic devices and accessories to customers all over the world. Whether you're in the market for a new smartphone, a laptop for work or school, or the latest gaming console, we've got you covered. We offer a wide range of products including laptops, desktops, smartphones, TVs, cameras, and much more.

At ebms, we understand that shopping for electronics can be overwhelming, which is why we've made it our mission to create a user-friendly interface that makes the shopping experience easy and convenient for you. Our store is stocked with the latest and greatest gadgets from the top brands in the industry, and we're always on the lookout for the newest and most exciting products to add to our inventory.

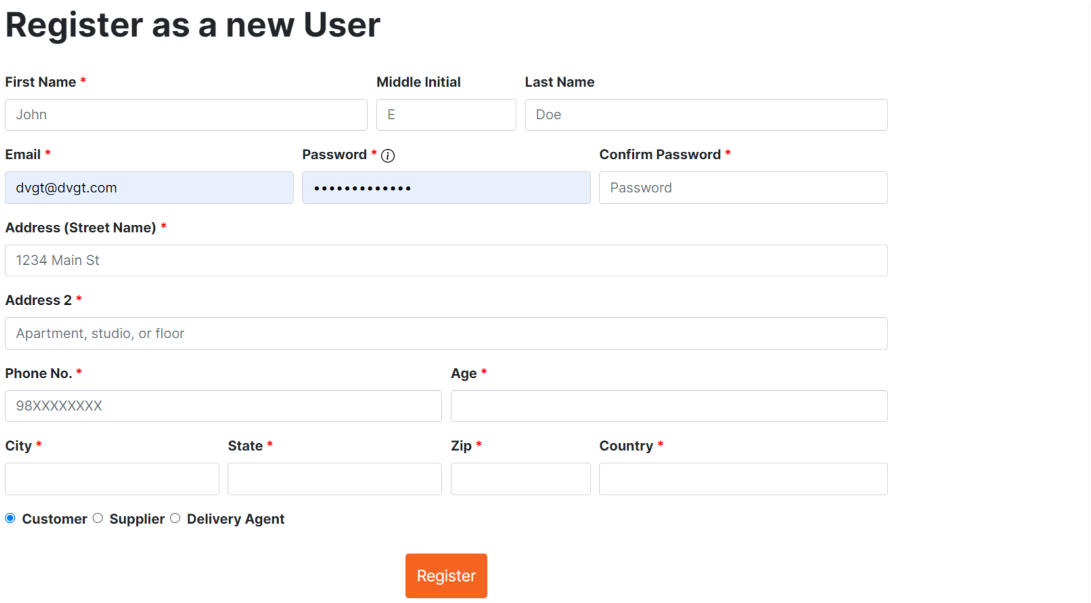
We believe that shopping for electronics should be fun, which is why we offer a range of exclusive deals and competitive prices to help you get the best value for your money.

Contact us

✉ helpline@ebms.ac.in ☎ +91 1234567890 ☎ +91 0987654321

© 2023 by ebms.

Figure 1.2: About Us Page (Home Page Footer)



Register as a new User

First Name * John

Middle Initial E

Last Name Doe

Email * dvgt@dvgt.com

Password * ⓘ •••••••••••••

Confirm Password * Password

Address (Street Name) * 1234 Main St

Address 2 * Apartment, studio, or floor

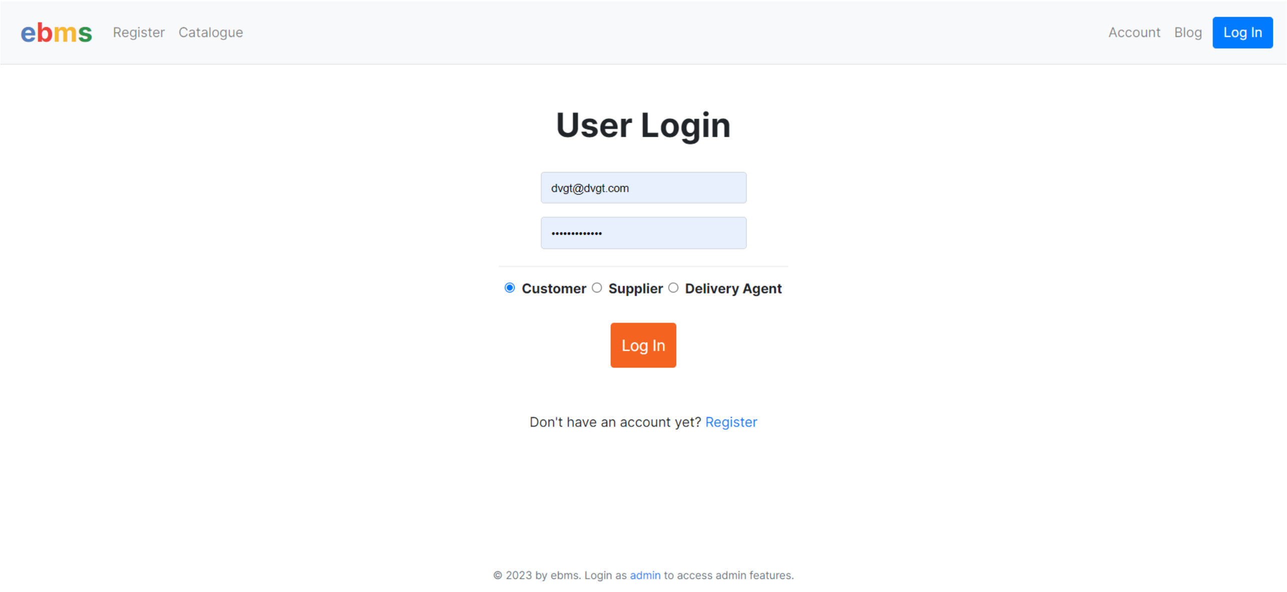
Phone No. * 98XXXXXXXX

Age *

City * **State *** **Zip *** **Country ***

● **Customer** ○ **Supplier** ○ **Delivery Agent**

Figure 1.3: Register Page



User Login

dvgt@dvgt.com

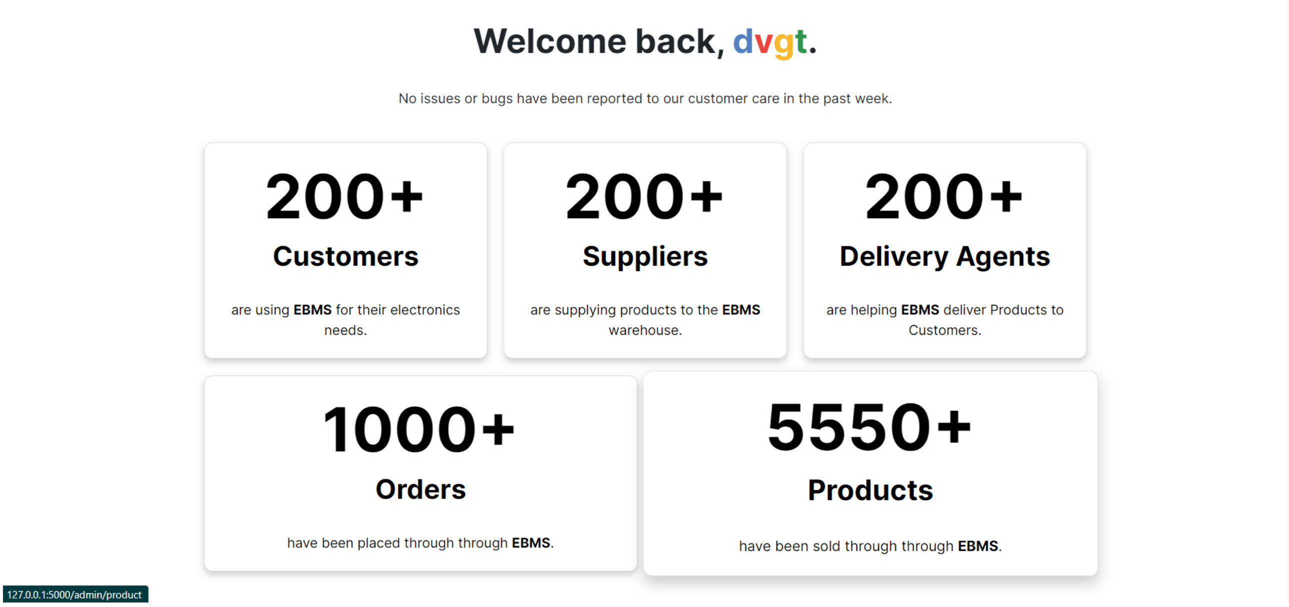
•••••••••••••

● Customer ○ Supplier ○ Delivery Agent

Don't have an account yet? Register

© 2023 by ebms. Login as admin to access admin features.

Figure 1.4: Login Page



Welcome back, dvgt.

No issues or bugs have been reported to our customer care in the past week.

200+

Customers

are using **EBMS** for their electronics needs.

200+

Suppliers

are supplying products to the **EBMS** warehouse.

200+

Delivery Agents

are helping **EBMS** deliver Products to Customers.

1000+

Orders

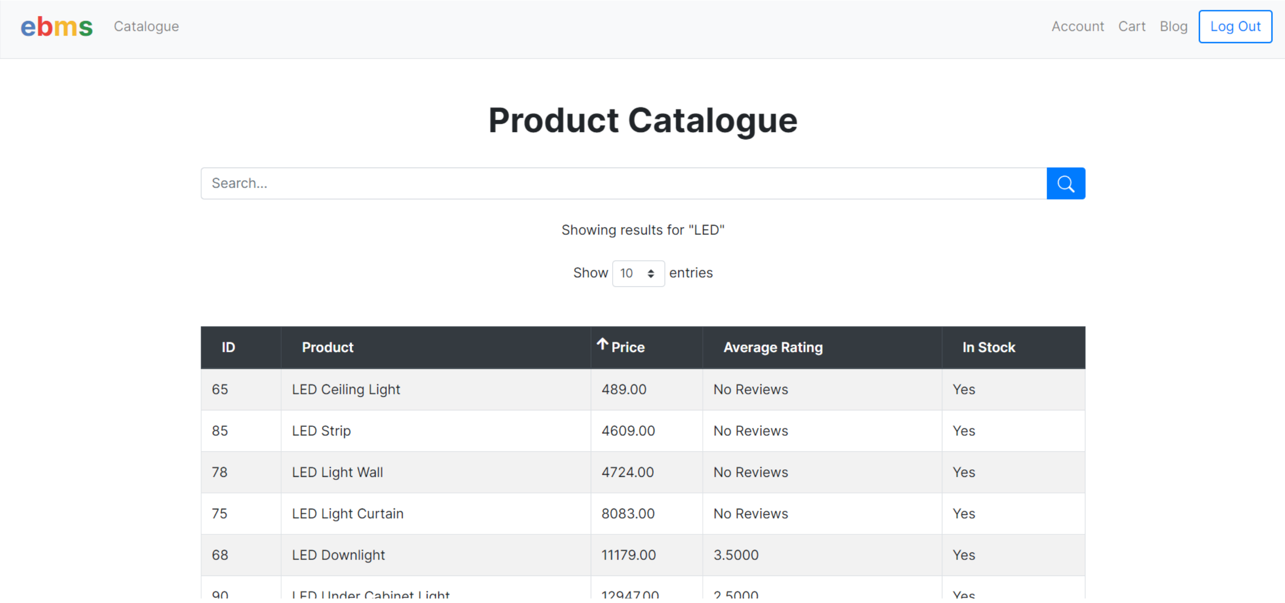
have been placed through through **EBMS**.

5550+

Products

have been sold through through **EBMS**.

Figure 1.5: Admin Dashboard



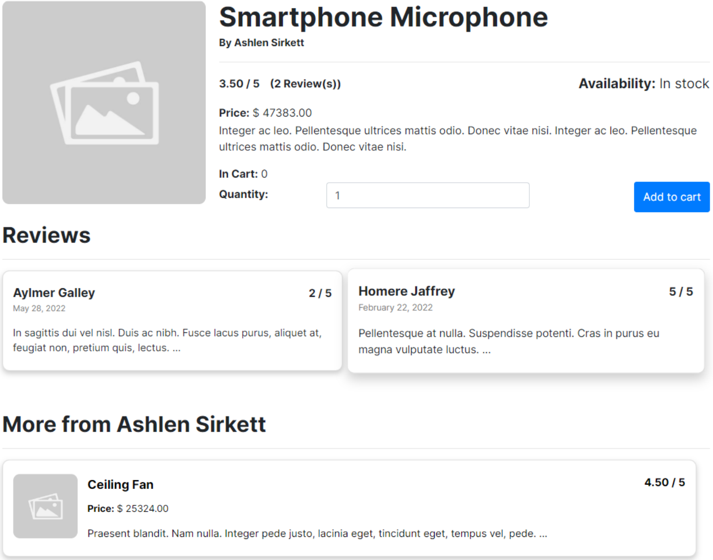
Product Catalogue

Showing results for "LED"

Show 10 entries

ID	Product	↑ Price	Average Rating	In Stock
65	LED Ceiling Light	489.00	No Reviews	Yes
85	LED Strip	4609.00	No Reviews	Yes
78	LED Light Wall	4724.00	No Reviews	Yes
75	LED Light Curtain	8083.00	No Reviews	Yes
68	LED Downlight	11179.00	3.5000	Yes
90	LED Under Cabinet Light	12947.00	2.5000	Yes

Figure 1.6: Product Catalogue



Smartphone Microphone

By Ashlen Sirkett

3.50 / 5 (2 Review(s))

Availability: In stock

Price: $ 47383.00

Integer ac leo. Pellentesque ultrices mattis odio. Donec vitae nisi. Integer ac leo. Pellentesque ultrices mattis odio. Donec vitae nisi.

In Cart: 0

Quantity: 1

Reviews

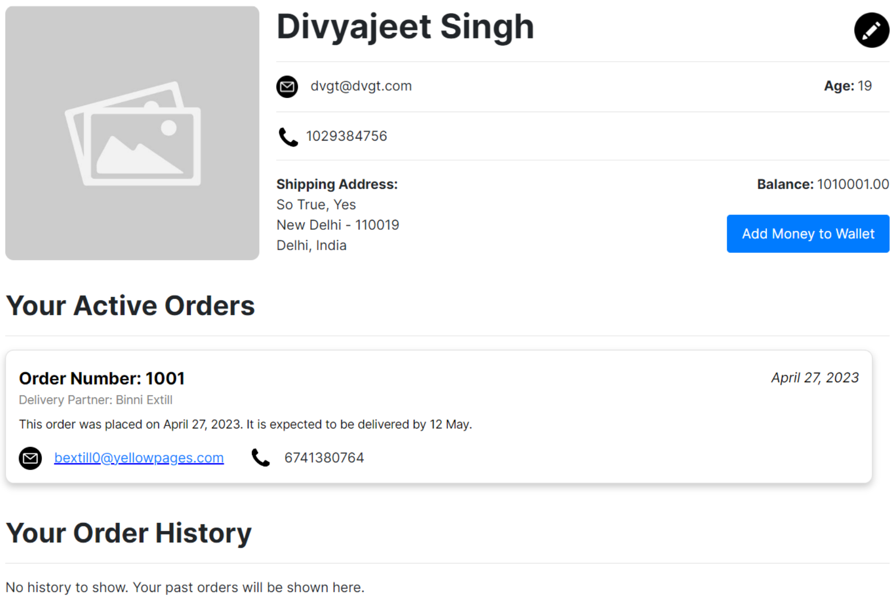
Aylmer Galley 2 / 5
May 28, 2022
In sagittis dui vel nisl. Duis ac nibh. Fusce lacus purus, aliquet at, feugiat non, pretium quis, lectus. ...

Homere Jaffrey 5 / 5
February 22, 2022
Pellentesque at nulla. Suspendisse potenti. Cras in purus eu magna vulputate luctus. ...

More from Ashlen Sirkett

Ceiling Fan 4.50 / 5
Price: $ 25324.00
Praesent blandit. Nam nulla. Integer pede justo, lacinia eget, tincidunt eget, tempus vel, pede. ...

Figure 1.7: Product Details



Divyajeet Singh

dvgt@dvgt.com

Age: 19

1029384756

Shipping Address:
So True, Yes
New Delhi - 110019
Delhi, India

Balance: 1010001.00

Your Active Orders

Order Number: 1001

April 27, 2023

Delivery Partner: Binni Extill

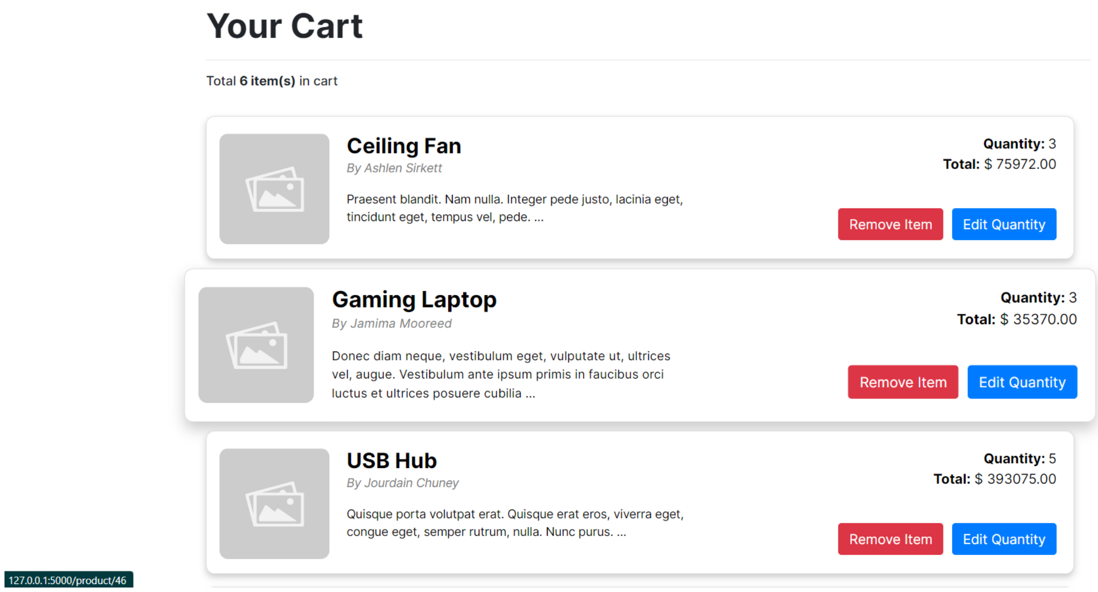
This order was placed on April 27, 2023. It is expected to be delivered by 12 May.

bextill0@yellowpages.com 6741380764

Your Order History

No history to show. Your past orders will be shown here.

Figure 1.8: Customer Profile Page



Your Cart

Total **6 item(s)** in cart

Ceiling Fan

By Ashlen Sirkett

Praesent blandit. Nam nulla. Integer pede justo, lacinia eget, tincidunt eget, tempus vel, pede. ...

Quantity: 3
Total: $ 75972.00

Gaming Laptop

By Jamima Mooreed

Donec diam neque, vestibulum eget, vulputate ut, ultrices vel, augue. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia ...

Quantity: 3
Total: $ 35370.00

USB Hub

By Jourdain Chuney

Quisque porta volutpat erat. Quisque erat eros, viverra eget, congue eget, semper rutrum, nulla. Nunc purus. ...

Quantity: 5
Total: $ 393075.00

Figure 1.9: Customer's Cart

User Guide

The website is currently hosted on a developmental server, while the database is hosted on a local machine. The website can be accessed at `http://127.0.0.1:5000/` only when the local server is running.

Navigation through the application

The front-end of the application is inspired from top online retailers in the real-market. The user interface is designed to be as smooth and intuitive as possible. Key principles like "*Consistency*", "*Accessibility*", and "*Keep it simple, stupid*" were kept in mind while creating the UI.

Although there remain some minor bugs and unimplemented features, the application is fully functional and can be used by all stakeholders.

Stakeholder: Store Managers (Admins)

Creating an admin account is possible only through the intervention of the developers. An admin must log in to their panel by clicking on the `Log in as Admin` button shown in the footer. The admin dashboard holds the power to manage the entire business, from each individual product to the entire supply chain. Using the dashboard, the admin gets access to aggregates such as trends in orders, best-selling products, demographics, etc.

Stakeholder: Customers

To use EBMS to its full potential, a customer must create an account using the `Log In` button located in the naviagation bar. From there, the `Create an account` button can be used to create an account.

After logging in, the customer can select products to add to their cart through the catalogue. The customer can also write reviews for the products that they have purchased. They can also add balance to their EBMS wallet, which can be used to pay for orders. They can view their personal information, their order history, and the current active (undelivered) orders on their profile page, which can be accessed through the `Account` button on the navigation bar.

Stakeholder: Suppliers

Suppliers can create an account following similar steps as mentioned above.

The supplier dashboard allows the supplier to view their products. They can add new products, update existing products, and delete products from their dashboard. Suppliers also get access to the reviews and ratings of their products. They can view and edit their personal information such as their warehouse address thorugh the `Account` button. EBMS also generates sales reports for the supplier, which can be accessed (and possibly in the future, downloaded) from the dashboard.

Stakeholder: Delivery Agents

Delivery agents must also create an account to use EBMS.

Delivery agents can find their active orders on their dashboard. From there, they can mark orders as delivered. They are also allowed access to their previous orders, and the details of each order, if required. They can also view and edit their personal information such as their contact number through the `Account` button. Delivery agents have ratings and reviews associated with them, which can also be viewed on their dashboard.

Source Code

The progress report for the project can be found at `Report/Progress-Report/main.pdf` in the repository. The source code for the project, inclusive of reports and database, can be found at `https://github.com/divyajeettt/EBMS`.

Dependencies

ElectroBase Management System is written completely in `Python 3.11.0` by the authors. The front-end is written in `HTML-4`, `CSS-3`, and `JavaScript`. The handler for the front-end is written in `Flask`, a popular web micro-framework for Python. The back-end is handled using `MySQL Server`, a popular relational database management system.

Setup: Dependencies

To use EBMS on a local machine, at least the following dependencies must be installed:

- A `Python 3.9.0` interpreter
  - `Flask 2.3.0`
  - `Flask-Session 0.4.0`
  - `mysql-connector-python 8.0.33`
- A `MySQL Server 8.0.26` distribution

Setup: Installation & Setup

Assuming you have Python and MySQL Server ready on a machine, you may follow the steps below to setup EBMS:

- Clone the repository and navigate to the directory.
- Create `front-end/app/.env` with your database credentials. A sample file is provided at `front-end/app/.sample_env`.
- To setup the database, through a MySQL Command Line Client, run:
  - `SOURCE Database-Generation/database-schema.sql` to create the database.
  - (Optional) `SOURCE Database-Generation/data-population.sql` to populate the database with sample data.
- To install the other Python dependencies, run `pip install -r requirements.txt`.

Setup: Run the Application

To run the application, run the command `python -m front-end/run.py` from the root directory of the repository.

Disclaimer

The authors claim to have written the entire codebase from scratch themselves, with proper attribution to the resources used. The authors do not claim any responsibility for the use of this project in any other context, even if it results in damage or data loss.

The project is currently not as secure as it should be, and is not intended for production use. It is also not intended for commercial use, and will be guarded by the MIT License. For example, the website is not protected against SQL injection attacks.

EBMS is a course project for CSE202: Fundamentals of Database Management Systems (Winter 2023) at IIIT-Delhi.