Mini Project - **Electronics Store Management System**

Group : **Wadeyar**

PRAJWAL S – Manager & Tester

P MANOHARA REDDY – Developer & Database Architect

PRAJWAL M – Developer & Database Architect

Project Statement **: Electronics Store Management System**

It is a multi-user app (Different Admins) that acts as an Audit site for multiple customer bills.

**Frameworks :**

These are the frameworks on which the project is being built.

* **Flask** for application back-end
* **Jinja2** templating, **HTML**, **CSS** and Bootstraps for application front-end
* **MYSQL** for Database Management System

**Role and Privileges**

The platform will have **only one** role with multiple logins:

1. **Admins** are is the superusers of the app and requires no registration

* There will be multiple admins to this application
* The admin login redirects to the admin dashboard
* The admin will manage all the billing and auditing
* The admin can bill the customer on the cash basis and also debt
* All the admin are provided with the clearance of existing customer’s debt
* The admin can change the price of the products i.e both cash and debt price of a product
* The admin is not privileged with adding a new product to the store where he/she can increase the existing stock
* All the admins are privileged to see the business statistics i.e Daywise, Monthwise, Yearwise statistics
* All the admins are privileged to see the details of a specific customer based on their phone number
* All the admins are privileged to add a new worker(other than admins)
* All the admins are privileged to increase or decrease the salary and other details of a specific worker
* None of the admins are privileged to discount for a specific customer, whereas they can change the price of the product
* The changed product price will not be applied to the customers who bought the same product on debt basis previously
* Each product will be having two different prices one for the cash transaction and one another for the debt based

**Terminologies**

**Admin:**

The superuser with maximum control over other products and Customers. Registration is not allowed for the admin: The admin accounts pre-exists in the database and can only be controlled by the developer.

**Customer:**

a customer can either pay by cash which costs him/her less price, but he/she can also purchase the product on debt which cost 15% more the cash transaction price(can be reduced by admin for slow-moving products so as to attract the customers). While clearing the debt 36% of the simple interest is applied on each product from the date of purchase.

A customer is identified by the following attributes

* + 1. Customer name
    2. Phone Number
    3. Product Name
    4. Cost of the product
    5. Date of Purchase

Two separate entities are maintained in the database. i.e for cash based transactions and debt based transactions. Clearance details are sent from the debt entity to the cash entity in the name clearance. i.e details in the debt entity is removed and then sent to the cash entity.

**Stock and Product:**

Each product is already registered by the developer since this is a centralized “Electronic store management system” i.e Electronic Stores in different locations with the single head office like ‘Sony’. None of the admins are privileged to add a new product whereas they are allowed to increase the product count.

A stock is identified by the following attributes

1. Product Name
2. Quantity

A Product is identified by the following attributes

1. Product Name
2. Cash Price
3. Debt Price

**Worker :**

Each Worker is uniquely identified by the following attributes

1. Name
2. Job
3. Phone Number
4. Salary
5. Age

**Access :**

Admin access details are uniquely identified by the following attributes

1. Id – Unique id for every admin
2. Pass – Access code

**Note :**

The provided wireframe & database design is intended **only to illustrate the application's flow** and is **subject to change if it requires based on the our team Decision**.

**Core Functionalities included in the application :**

1. **Admin login**

A login form with fields like username, password etc. A simple login with username and password is implemented. The details are stored in the db. The details fetched with post request is compared with all the entries in the database. if matches then session is initialized otherwise the same page is rendered again.

1. **Cash Transaction**

The Customer name, Phone number, Product Name, Price, Date of purchcase (shown on the webpage no need to remember) is fetched through post request and is sent to the ‘Paid\_Customer\_details’ entity. Price is automatically fetched through the database which is present in the ‘Product\_details’ entity. The date of purchase is fetched from the datetime module of the python application.

1. **Debt Transaction**

The Customer name, Phone number, Product Name, Price, Date of purchcase (shown on the webpage no need to remember) is fetched through post request and is sent to the ‘Debt\_Customer\_details’ entity. Price is automatically fetched through the database which is present in the ‘Product\_details’ entity. Price of a certain product costs 15% more than the cost for cash transaction. The price is subject to change on the demand and stock basis and is completely dependent upon an admin. i.e he/she can change it if it is required. The date of purchase is fetched from the datetime module of the python application.

1. **Debt Clearance**

Customer name, Phone number is fetched from the admin through the clearance page. Once again the application asks for the confirmation by showing how much does the customer need to pay and if again confirmed then the details are sent to the ‘paid\_customer\_details’ entity. The cost is calculated by applying a rate of 36% per annum of the simple interest on each product from its date of purchase. There is no provision for clearing fraction of the amount. i.e if they want to clear complete amount that has to be paid, once paid then the admin is supposed to confirm on the clearance confirmation page.

1. **Business Statistics**

Daywise, Monthwise & Yearwise Statistics is fetched from the database on the request of the admin. A separate page is kept for each individual kind of statistics. Each page shows Total numer of cash transactions, Total numer of debt transactions, Total cash collected, Total debt given, Total amount returned in Clearance on the stipulated period that is fetched through the post request which is again requested by the individual admin.

1. **Stock**

Productwise quantity present in the stock left in the warehouse is fetched from the database. The quantity is automatically decremented by the web application whenever a cash or debt transaction is processed.

1. **Product quantity**

Productwise individual product name, cash price, debt price is fetched by the database. if the quantity is 0 then automatically selling is restricted on it. If an admin tries to sell a product when the quantity is 0 then the server will be crashed.

1. **Add Stock :**

Individual Product quantity is increased on the desire of any individual admin.

1. **Update Product Details**

The price details of the individual products can be increased or decreased on the basis of demand and season. It completely depends on the desire of any individual admin. Before fetching the increased cash price or debt price, the details of the present data is shown to the admin.

1. **Add Employee**

A new employee for a certain store can be added by any individual admin. The employee details comprises of Name, Job, Phone number, salary, age and are stored in database with the entity worker details.

1. **View Employees**

All the employee details are fetched and rendered to any individual admin who requests.

1. **Update Employee**

Any details of a certain worker can be changed by any admin. i.e can increase the salary decrease it.

**Project wireframe :**

[Click here for Figma wireframe](https://www.figma.com/design/jQ0UBu9AOKzKSUKPd7kF4S/Grocery-Web-App?node-id=0-1&t=jzh6qtzbYiu3bXX5-1)

**Database ER Model :**

[Click here for DB ER Model](https://drive.google.com/file/d/1Ezs-Gj0BVmWku3v-01g_nMnQioR-P4Aj/view?usp=sharing)

**Directory structure:**

