**Kathmandu University**

**Department of Computer Science and Engineering**

**Dhulikhel, Kavre**

****

**Mini Project Report**

**on**

**“Department Store Management System ”**

**[Code No: COMP 232]**

**(A second year mini project report for partial fulfillment of COMP 232 II/II in Computer Engineering)**

**Submitted by**

Skanda Aryal (06)

Arjun Bhandari (08)

Niranjan Pant (33)

Aseem Regmi (41)

**Submission Date: July 22, 2019**

**Department Store Management System**

**Description**

**Objective**

1. To understand the concept and utilization of database management system
2. To develop a database management system which manages or keeps the record of selling and buying of any type of products in a store.
3. To create a simple react application to perform and demonstrate the basic operations on the created database including insert, delete, update and create.

**System Design**

1. Programming Language : JavaScript
2. Database Management System: MySQL
3. Tools Used: phpMyAdmin, Node.js, ReactJS

**Scenario**

This project has been introduced primarily to manage the department store. The system is useful for the owner of the department store to manage all the details about the products, orders, sales etc.The process of managing and recording data seems very tiring which may also leads to lost of data.Therefore, this project aims to help the owner to maintain the consistency of data . The system is built on React UI Platform and have a user friendly interface that can be used and navigated without any problems.

**Synopsis**

The database represent each Employees with their unique Employee ID, First Name, Last Name, Employee Type ID, Address, Email Address and Phone Number. Each Employee Types are represented by their unique Employee Type ID, Post, Woking Hour and and Monthly Salary of them. The information of customers of the store also needs to be recorded. Customers also have their unique Customer ID, First Name, Last Name, Phone Number, Email Address and their residential Address. Suppliers are also recorded in the database. Each supplier is represented by their unique Supplier ID, Name, Address of the Office and Contact Info. Each product in the department store needs to be categorized so each Product Category in the departmental store has its unique Product Category ID and Name and every Products are represented with their unique ID, Name, Product Category ID, Number of Available Quantities, ID of supplier, Expiry Date and Age Restriction for customers. The customers also have the provision to review the products they have used. Each review is represented by Quality Rating (in percentage), Description, Reviewed Date and ID of reviewed product. An order is given by customers to the department store. Every Orders are represented by their unique Order ID, Status of Order and Ordered Date. Similarly each Ordered Products are represented by Order ID, Product ID and Quantity. The Department Store provides some sort of discount for their customers. Every Voucher have their unique Discount Voucher ID, Discount Percentage and Minimum Billing Amount. Bills are generated after the order is complete. Each Bills include unique Billing ID, Order ID, Discount Voucher ID, Amount To Be Paid and Status of Bill (either Paid or Not Paid). Payments are done by customers and are monitored by employees. Every Payments done are represented with their unique ID, Employee ID, Payment Method, Date, Amount Paid and Customer ID. Receipt is generated after the payment is done. Each Receipt is represented by unique ID, Billing ID and Payment ID. Some products also have warranty so it needs to be recorded separately. Each Issued Warranty can be represented by its unique ID, Warranty Period, Warranty Issued Date, Product ID and Billing ID.

**Schemes**

The staff of the departmental store can add customers, take orders from them, give certain discount to them on the basis of discount vouchers.

**Entities and their attributes:**

1. employee\_types: Consists information about various types of employees.
   * Id
   * Post
   * Working hour
   * Monthly salary
2. employee\_info: Information of the working employees
   * Email
   * Phone\_no
   * First\_name
   * Last\_name
   * Residential\_address
3. Employees
   * Id
   * Email
   * Employee\_id
4. Customer Info: Consists the required information of all the customers.
   * Email
   * First Name
   * Last Name
   * Phone Number
   * Residential Address
5. Customer : Consists the information of customer.
   * Id
   * Email
6. Supplier\_info: Consists the information of suppliers.
   * Email
   * Name
   * Office\_address
   * Phone\_number
7. Suppliers: Contains additional information of suppliers.
   * Id
   * Email
8. Product\_catagory: Contains the information about products type in the store.
   * Id
   * Category
9. Products : Contains the information about the products available in store.
   * Id
   * Product\_name
   * Product\_category\_id
   * Available\_quantities
   * Price
   * Discounted\_price
   * Supplier\_id
   * Date\_of\_expire
   * Minimum\_age\_required
10. Reviews: Contains the reviews given by customers to the products.
    * Customer\_id
    * Product\_id
    * Review\_percentage
    * Review\_description
    * Reviewed date
11. Orders: Contains information about the orders given by the customers.
    * Customer\_id
    * Order status
    * Ordered date
12. Ordered\_products: Contains the information of ordered products.
    * Order\_id
    * Product\_id
    * Quantity
13. Contains the information about billing,discount rate.
    * Id
    * Discount\_rate
    * Minimum\_billing\_amount
14. Discount: It contains the information about the discount made to the customer.
    * Id
    * Voucher\_id
    * Customer\_id
15. Bills: Contains the information of the bills payable by the customers.
    * Order\_id
    * Voucher\_id
    * Bill\_status
16. Payments: Contains the information of payments by the customer.
    * Id
    * Employee\_id
    * Payment\_method
    * Payment\_time
    * Amount\_paid
    * Customer\_id
17. Receipts: Contains the information of the receipts given to the customers
    * Billing\_id
    * Payment\_id
18. Warranties: Contains the information of product warranty.
    * Id
    * Warranty\_start
    * Warranty\_end
    * Product\_id
    * Bill\_id

**Normalization**

**1NF**

There are no multivalued attributes in any one of the tables. So, the database is already in 1NF form.

**2NF**

In entities employees, customer and supplier, a few attributes were partially dependent on email key attribute so the tables were broken down into (supplier\_info and supplier), (customer\_info and customers) and (employee\_info and employees). Hence, the database is reduced to 2NF form.

**3NF**

There are no dependencies of attributes upon other non key attributes of the sam table. So, the database is reduced to 3NF.

**Primary Keys**

The underlined attributes in each entity are the primary keys of the corresponding entities. Primary Keys uniquely identifies each entity.

**Foreign Keys**

The dotted line attributes are the foreign keys used to link the parent and child tables. Foreign Keys are essential to show relationship between two tables.

**Weak Entity**

In our database, the entity **ordered\_products** is weak entity as it has no primary key and depends upon order\_id to be uniquely identified. The order\_id can be used to determine the ordered\_products associated with specific order\_id.

**Join**

SQL Join Clause combines columns from one or more tables in a relational database.

**Examples of join used in our database :**

* **Three Entities in a single join clause (payments, employees, customers)**

SELECT payments.\*, employees.id, customers.id FROM

payments INNER JOIN employees

ON payments.employee\_id = employees.id

INNER JOIN customers ON payments.customer\_id = customers.id ORDER BY payments.id

* **A simple Inner Join Involving Two Entities**

SELECT ordered\_products.\*, orders.order\_id FROM ordered\_products INNER JOIN orders

ON ordered\_productsorder\_id = orders.id   
ORDER BY ordered\_products.id

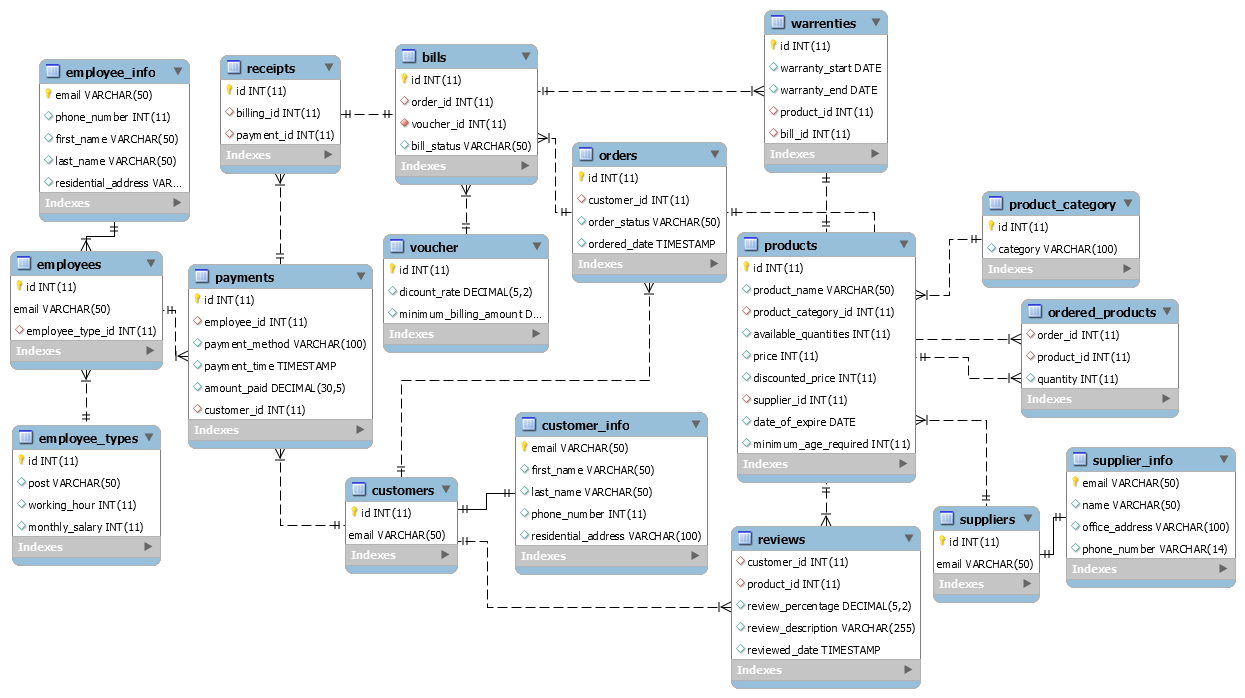
* **A join Query With Like Operator used for searching rows by attributes**

SELECT customers.id, customers.email, customer\_info.\* FROM customers INNER JOIN customer\_info

ON customers.email = customer\_info.email

WHERE customer\_info.first\_name LIKE '%${nameLikeText}%' OR customer\_info.last\_name LIKE '%${nameLikeText}%'

**ER DIAGRAM**



Fig; ER Diagram

**Screenshots of Application Front End**

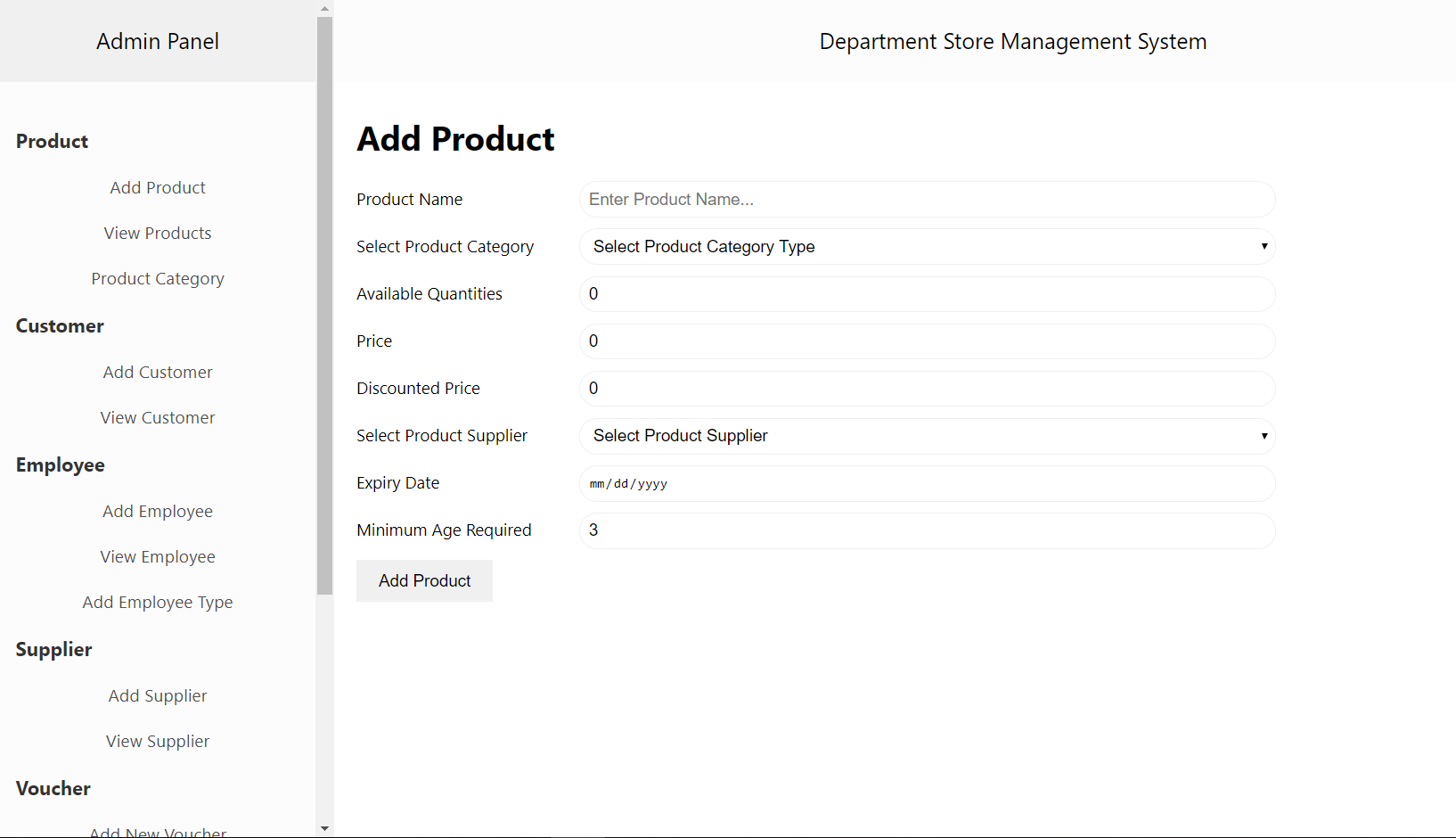


Fig: App-1

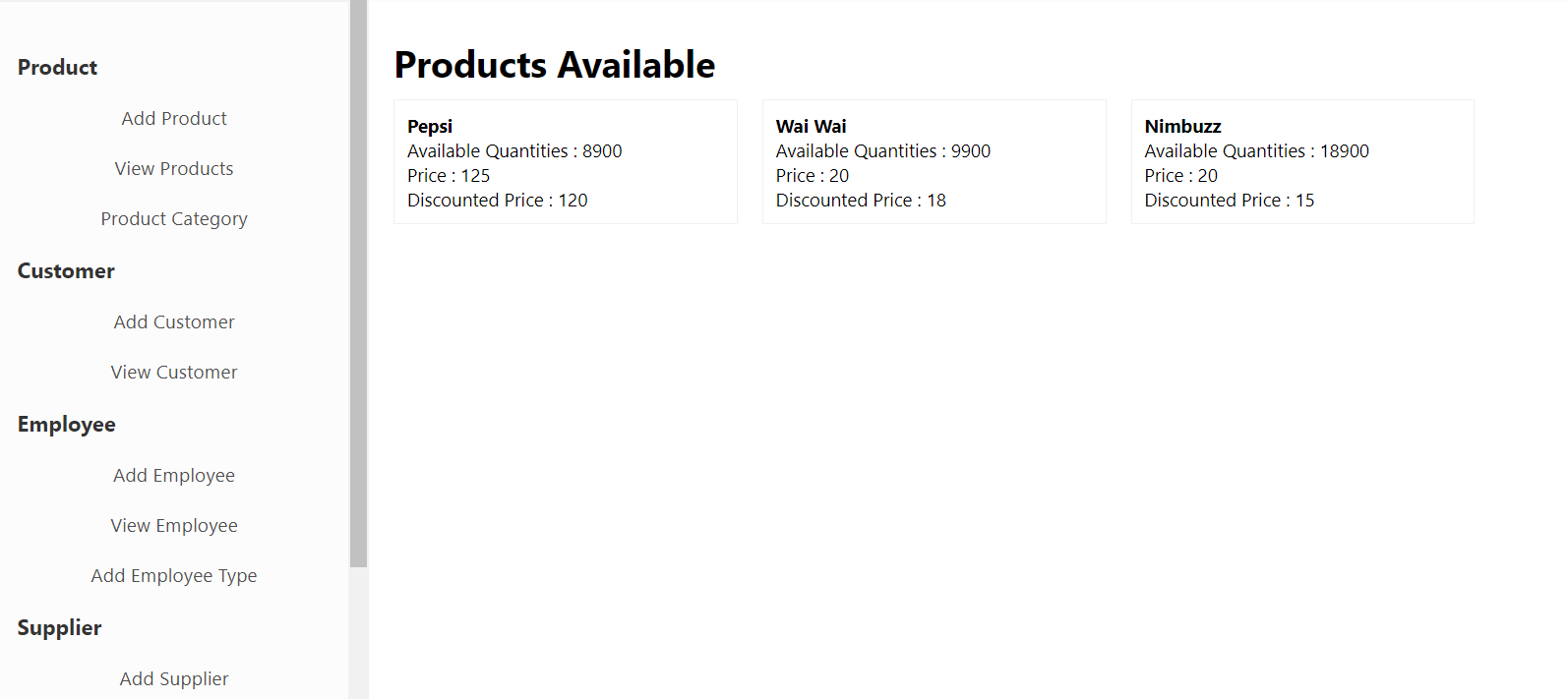


Fig: App -2

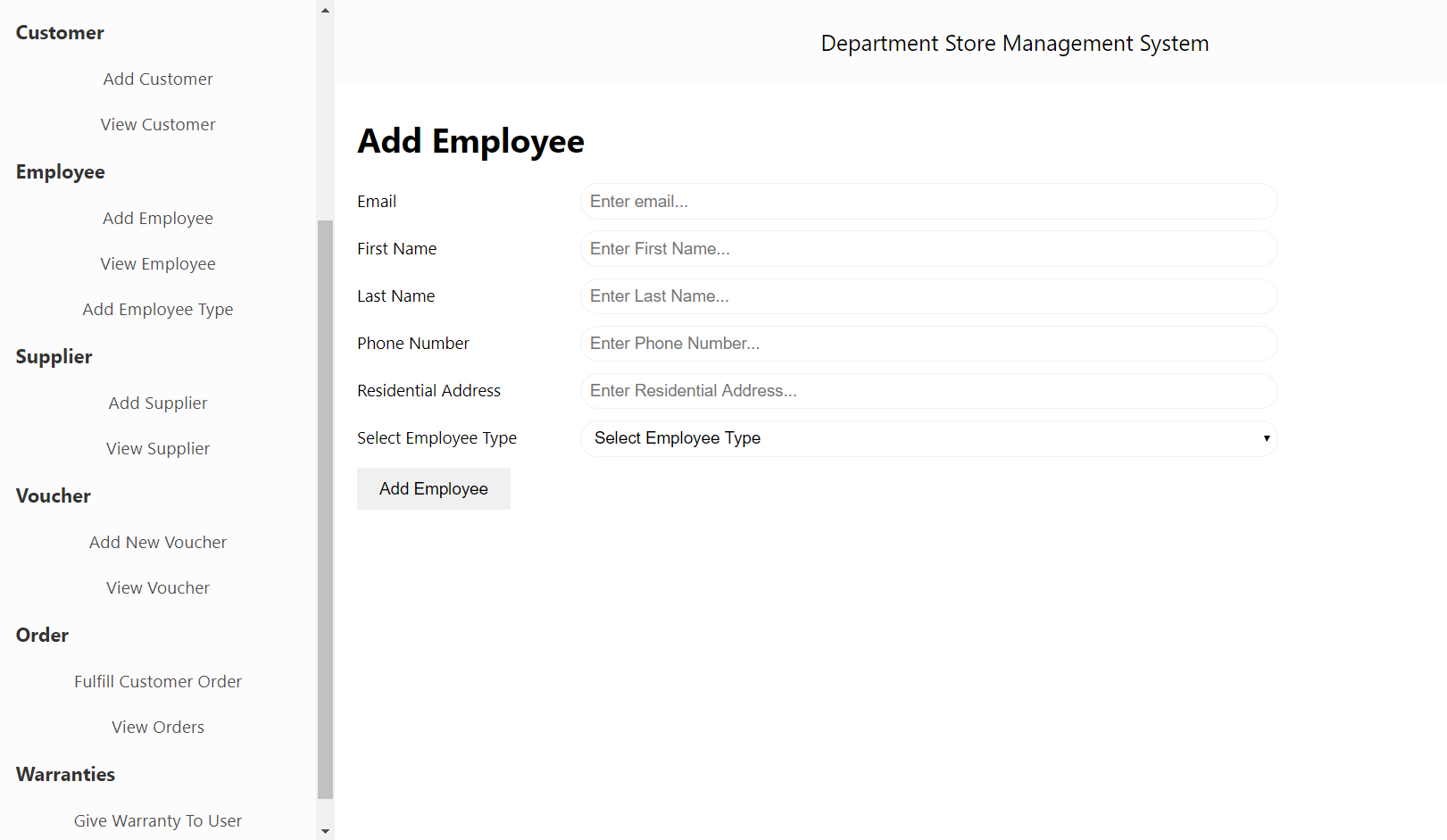


Fig: App -3

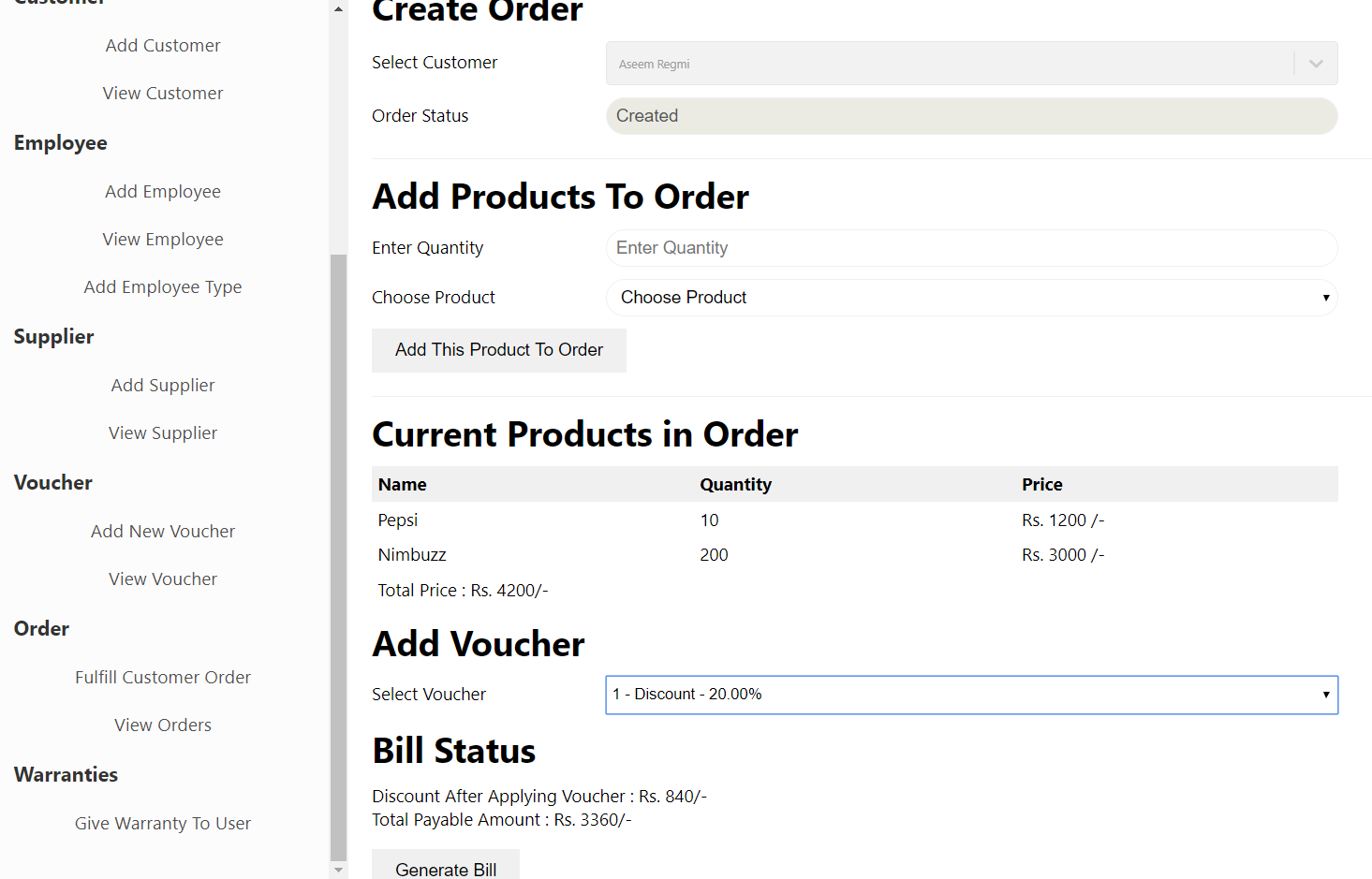
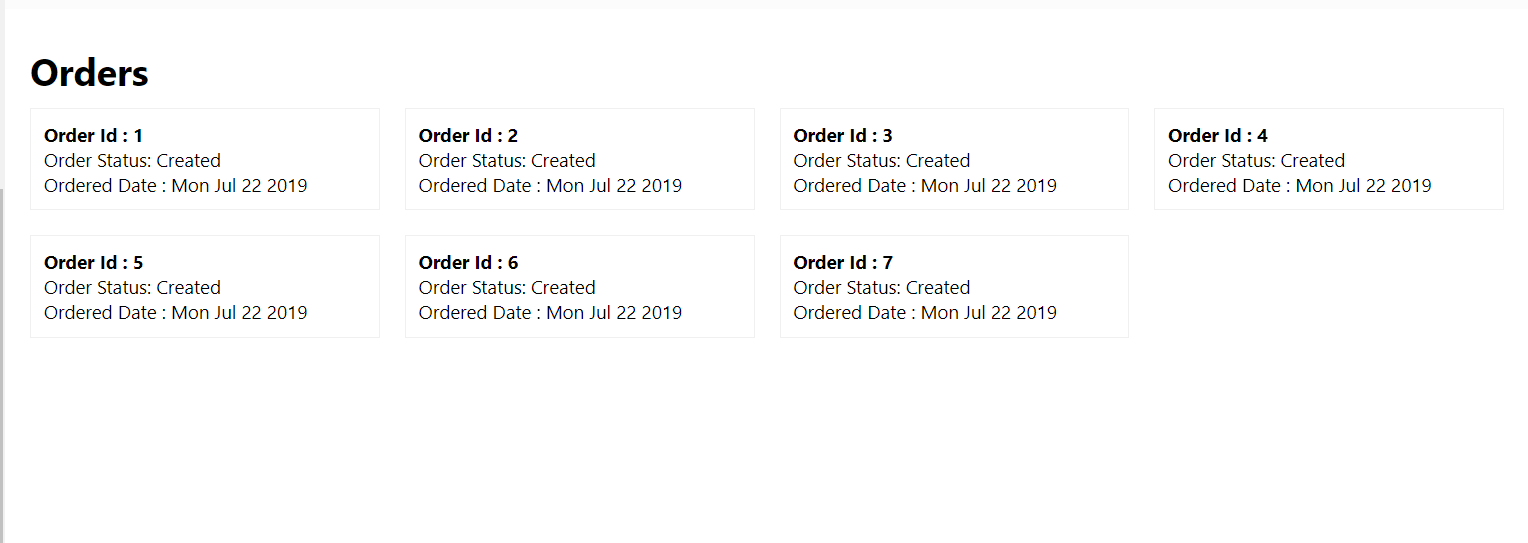


Fig: App - 4



Fig; App -5