Project Name: SneakerHeads
PG-DAC Sep-2021

Documentation On

# “SneakerHeads”

Submitted By

**Group No: 20**


- Prathamesh Ranshur. 210943120073
- Rohit Kale. 210943120082
- Jagruti Borse. 210943120022
- Shahezad Khan. 210943120088


**UNDER THE GUIDENCE OF:**

Mrs. Harshita Maheshwari
Faculty Member

Infoway Technologies, Pune.

## ACKNOWLEDGEMENT

The successful completion of a project is not an individual's effort. It is an outcome of cumulative efforts of all group members, each having its own importance to the objective.

This section is a value of thanks and gratitude towards all members for contributed in their own special way towards the completion of the project. For their invaluable comments and suggestions, we wish to thank all group members. We owe our gratitude and appreciation to our guide Mrs. Harshita Maheshwari whose suggestions and encouragement helped us to co-ordinate our project. We would also like to express our gratitude towards Infoway faculty and management for their Extended Support. Our thanks and appreciations also go to all project members for developing the project.

## 1. Introduction:

Sneakers are an integral part of our dressing-up routine, and you can go from formal to party, simple to stylish, casual to trendy in minutes by simply switching your shoes. A huge variety and design of sneaker shoes are now available in the market for various activities. The value of a good pair of shoes is constant. Online shopping is the process preferred by many these days. This sneaker shopping website is a good collection of sneakers for men and women paired with extremely profitable offers and deals to choose from this website. Thus, this online sneaker shopping will relieve people by providing access for purchasing, renting exclusive collection sneakers.

The Purpose of the project is to build an application program to provide a platform to rent expensive shoes at an affordable range that can be convenient for the customer and also manage the sales of the shoes.

### 1.1. Problem Definition:

Nowadays, the common people cannot afford the costly exclusive collections of shoes. So, our proposed website is for these people to make these expensive shoes available on a rental basis. The customer can book for the unaffordable shoe and can collect it from the vendor store on rent for a specific period with the advance deposit for the safety purpose of the product. In order to increase online shopping, customer online shopping behavior while shopping online should be given priority. The Current shopping System is critical to set up online shops, customers to browse through the shops, and a system administrator to approve and reject requests for new shops and maintain lists of shop categories. This is a small scale project for Online Sneakers shopping System. The basic idea is that the customer can buy sneakers from anywhere during any time by using their email and password through which they have been logged in. Some sneakers are too expensive for getting a hands-on, so we have provided a feature of buying a sneaker through rental system which is not present in existing systems.

### 1.2. Objective of Project :

- To develop an easy-to-use web-based interface where users can search for products, view a complete description of the products for renting or ordering.
- It provides an easy and convenient way to search for products specific to their needs.
- List a set of products based on the search and the user can filter the list based on various parameters.
- A user can view the specification of the product with the image.
- The objective is to book the shoe from the platform as per the customer demand.

### 1.3. Users:

- Admin:

- Log in to the system.
- View the list of all registered customers and shops.
- Add, remove, edit and update the details of the product
- Maintain a record of the sales.

- Customer:

- Log in to the system
- Select the shop for rent/buy the product
- Add/remove the product from the cart
- View the product into the catalog
- Book the product
- Book time slot for rent

### 1.4 Characteristics:

1. A sleek and user friendly interface
2. A buying as well as rental website portal
3. Featured display of products
4. FAQs section
5. Easy accessible contact form
6. Discounts.

## 2. Feasibility study:

### 2.1. Technical feasibility:

**Definition:** A buying and renting website of exclusive sneakers.
**Objective:** The main objective of the website to make the high cost sneakers available to customer on the rental basis.
**Product:** Admin add the product.
Customer will register and login to system for rent/purchase of the sneakers. After adding the product to cart customer can make the payment and logout.

# 3. Analysis

## 3.1. Existing System

Traditionally if anyone wanted to buy a sneaker, he/ she would have to physically go a store and buy a sneaker which was good for older days when technology was not that improved but as time passed Online Systems were developed which consisted of an online application for buying sneakers without physically going to the place.

## 3.2. Proposed System

A website where the customer can afford the exclusive high cost sneakers on the rental basis on a daily charges.

## 3.3. SDLC Model:

The meaning of Agile is swift or versatile." Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

## 3.4. Software Requirement Specification:

1. OS: Windows 7 & above

2. Browser: Google Chrome latest.

3. IDE: STS/Eclipse, VS code, MySQL workbench.

4. Network speed: 50kbps minimum

## 3.5. Hardware Requirement Specification:

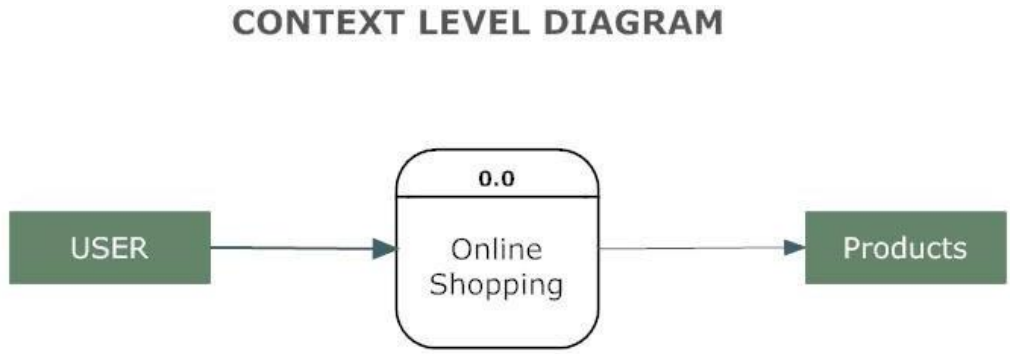
1. Processor**:** Intel i5 & above/ Ryzen 2 & above.

2. RAM: Minimum 2 GB

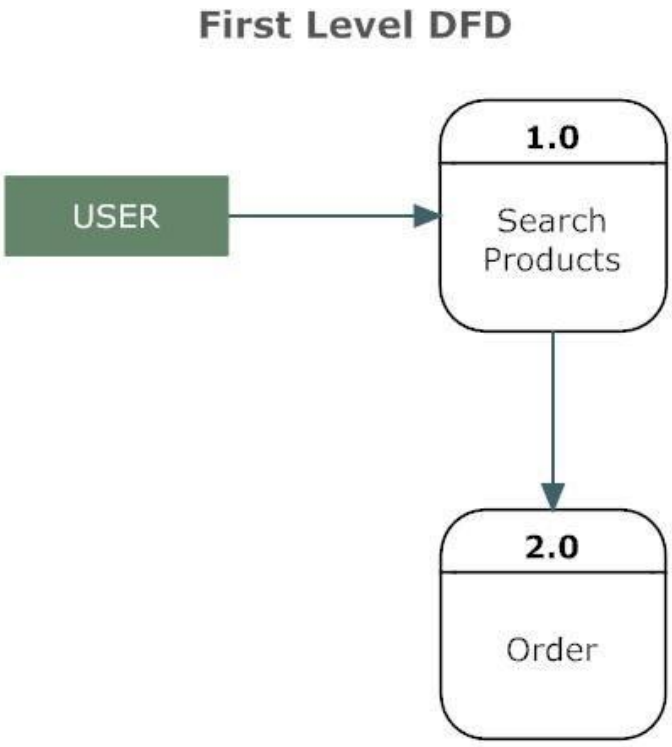
3. Hard disk: 200 GB minimum

**Data Flow Diagram (DFD):**

Level 0:

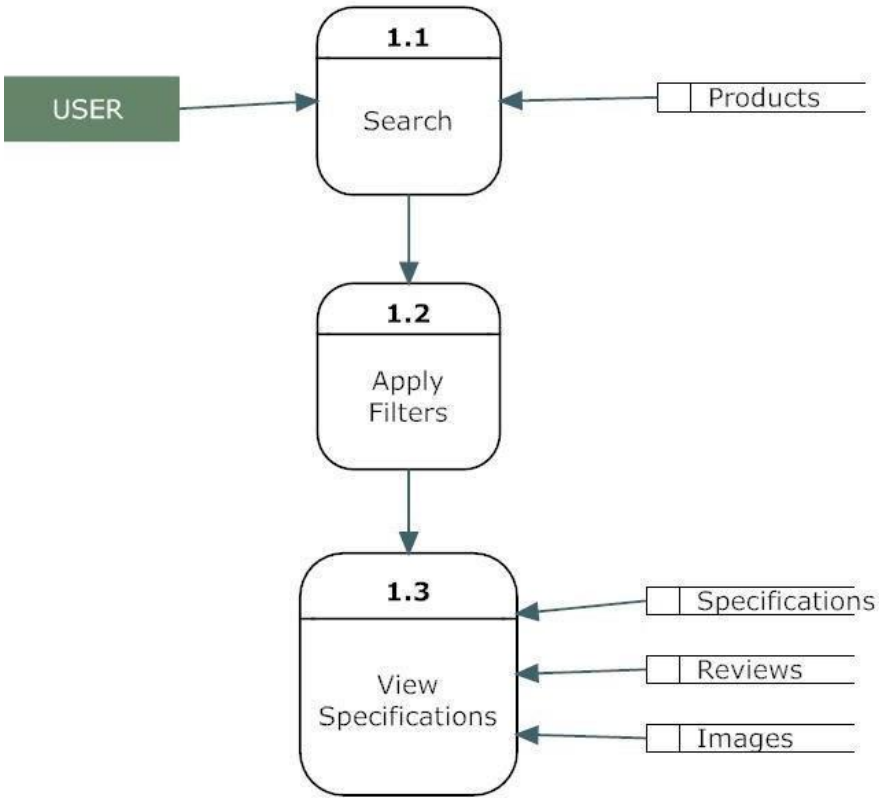



Level 1:

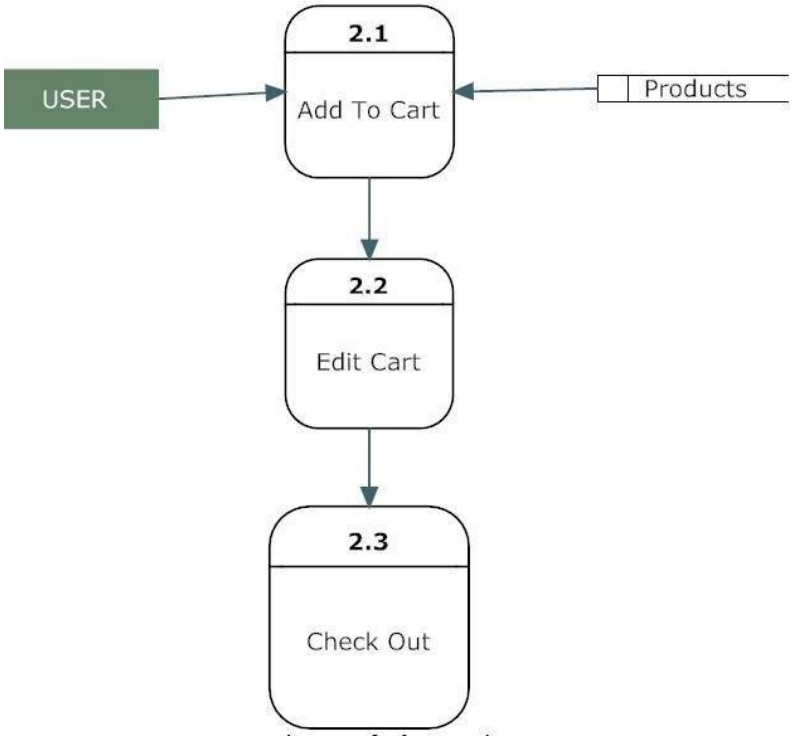


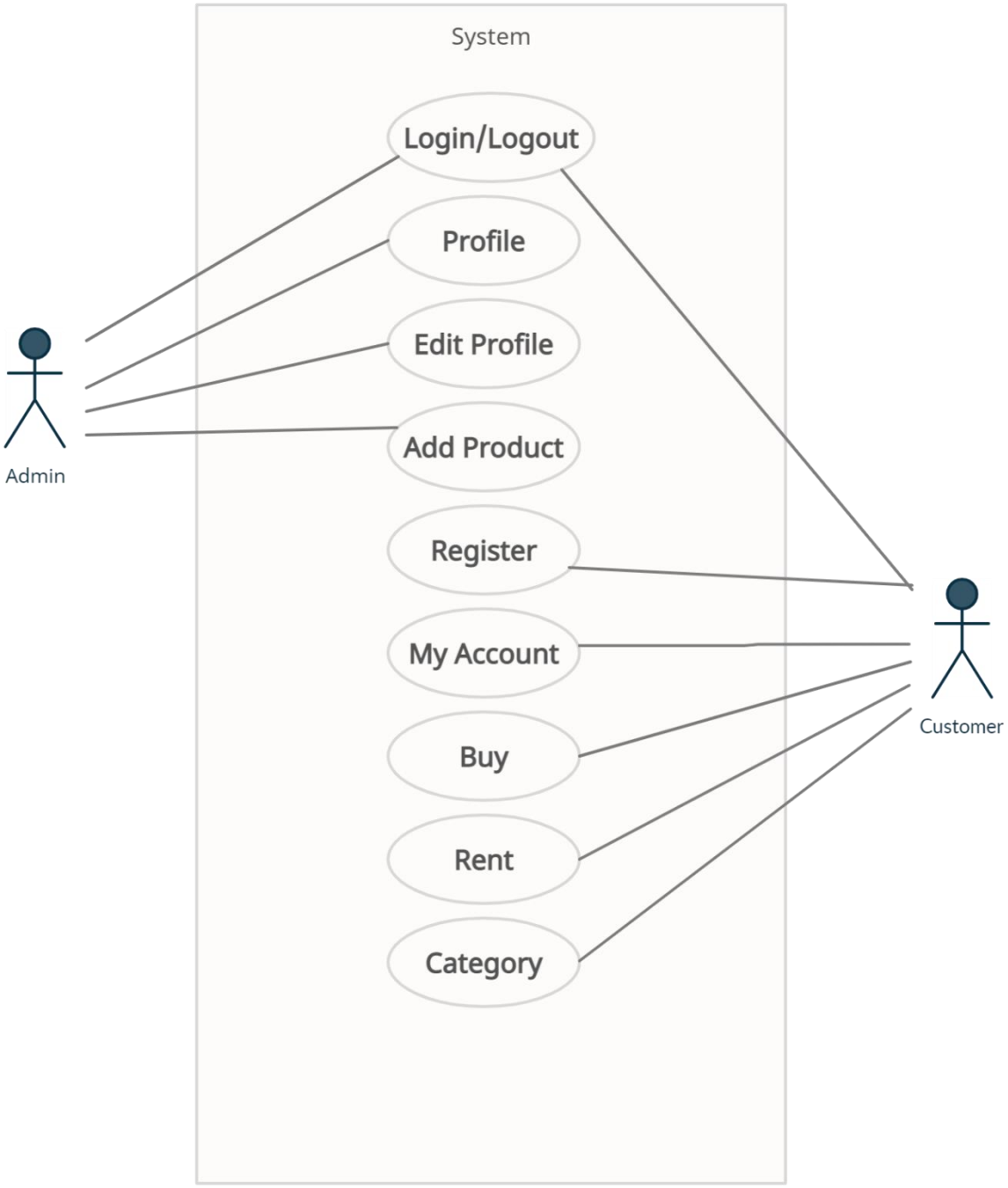

Level 2:

## SECOND LEVEL DFD

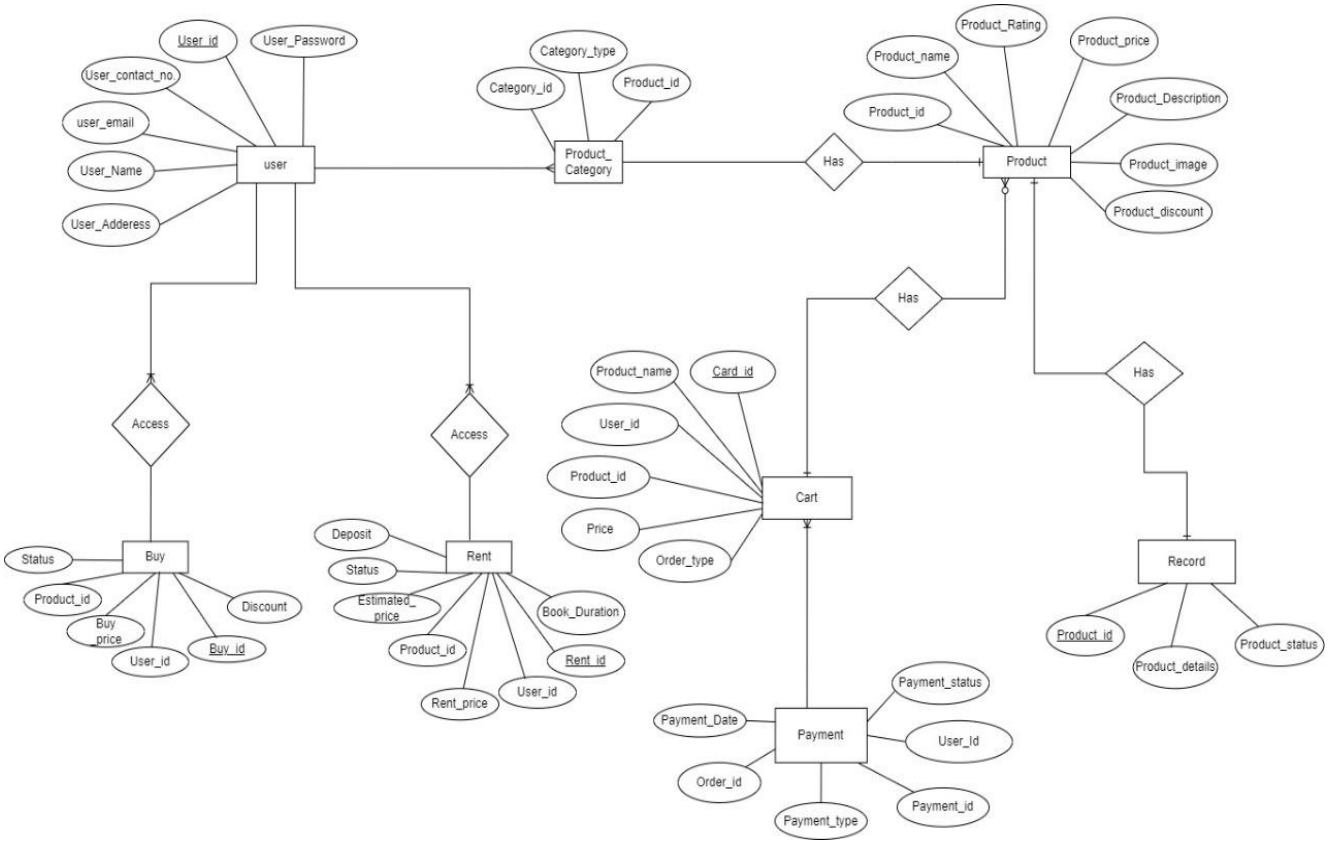


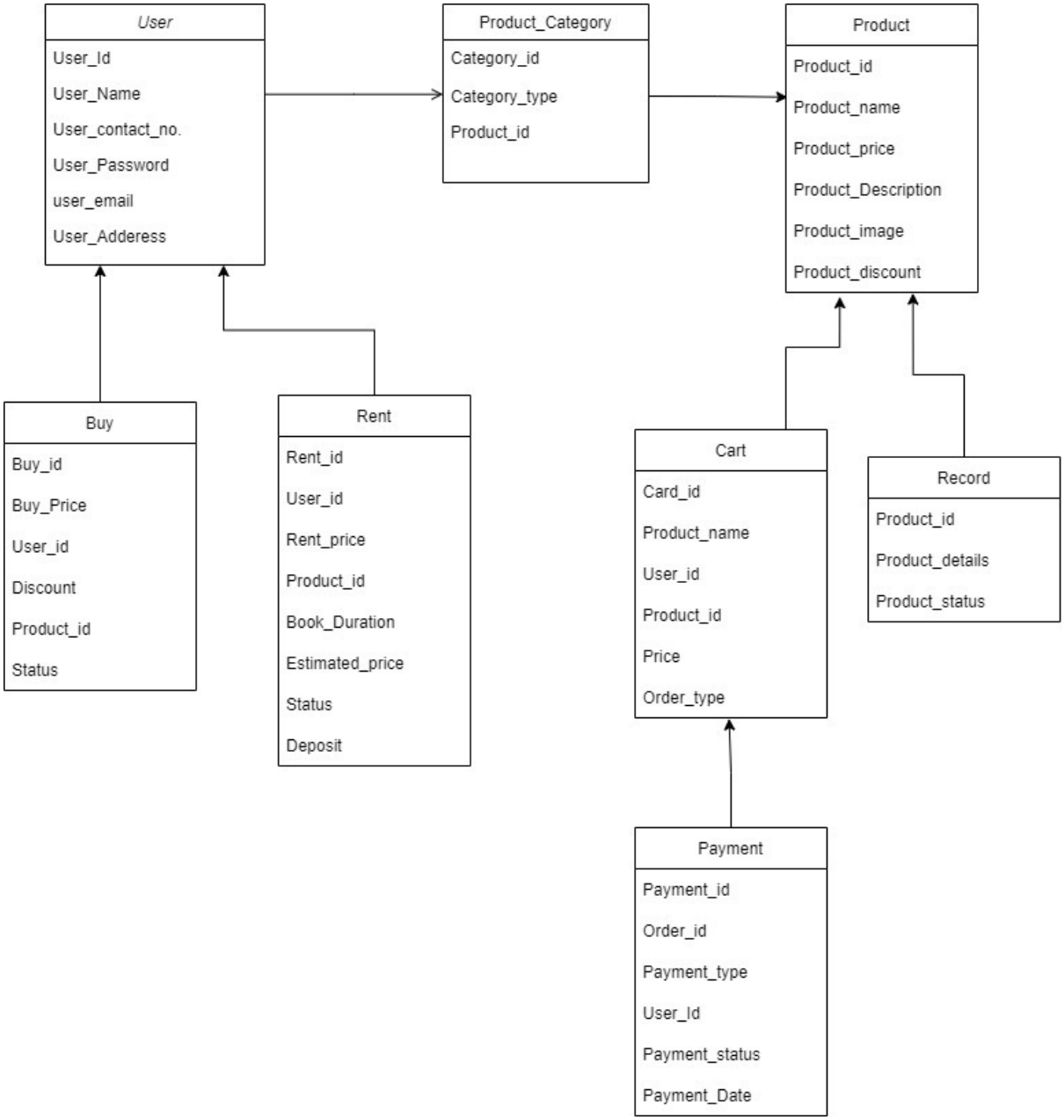

## SECOND LEVEL DFD






USE CASE DIAGRAM


User_id
User_Password
User_contact_no
user_email
User_Name
User_Adderess
user
Category_type
Category_id
Product_id
Product_
Category
Has
Product_Rating
Product_name
Product_price
Product_id
Product_Description
Product
Product_image
Product_discount
Has
Access
Access
Has
Product_name
Card_id
User_id
Product_id
Price
Order_type
Cart
Status
Buy
Product_id
Buy
price
User_id
Buy_id
Discount
Deposit
Status
Estimated_
price
Product_id
Rent_price
Rent
Book_Duration
Rent_id
User_id
Record
Product_id
Product_details
Product_status
Payment_status
Payment_Date
Payment
User_Id
Order_id
Payment_type
Payment_id


User
User_Id
User_Name
User_contact_no.
User_Password
user_email
User_Adderess
Product_Category
Category_id
Category_type
Product_id
Product
Product_id
Product_name
Product_price
Product_Description
Product_image
Product_discount
Buy
Buy_id
Buy_Price
User_id
Discount
Product_id
Status
Rent
Rent_id
User_id
Rent_price
Product_id
Book_Duration
Estimated_price
Status
Deposit
Cart
Card_id
Product_name
User_id
Product_id
Price
Order_type
Record
Product_id
Product_details
Product_status
Payment
Payment_id
Order_id
Payment_type
User_Id
Payment_status
Payment_Date

## 5. Implementation

5.1. Modules:

**Admin:**

- Log in to the system
- View the list of all registered customers and shops.
- Add, remove, edit and update the details of the product
- Maintain a record of the sales.

**Customer:**

- Customer can register and login using credentials.
- He/she can view the sneakers and can buy/Rent the sneakers.
- Add/remove the product from the cart.
- Book the product.
- Book time slot for rent.
- Customers will collect the product from the shop

5.2. Module description:

- Product Management: This product management is to manage the product brand-wise. All products will be managed by the admin and customers will be able to see the product. Admin can add, view, and delete the products.
- Admin Management: The admin will take the deposit for the renting of the product from the customer after the booking of the desired product and then confirm the order making it available for the customer.
- Customer Management: The customer can register and log in/log out to the system using credentials. They can view product details, can add products to the cart, and rent their product. They can also buy second-hand sneakers or rent sneakers for a specific period.
- Order Management: Customers can perform update operations on the order as per requirement. Details of product stored, order processed, shipping, and return processing are provided in this management.
- Cart Management: Cart management shows the products that have been added to the cart. The management also shows an order summary and lets the customer add or remove the product.

- Rent Management: For the renting of the product, the customer must first make the deposit payment in advance for booking the product. The product will then bind to the customer id and be available for the customer.

5.3. Introduction of technologies used:

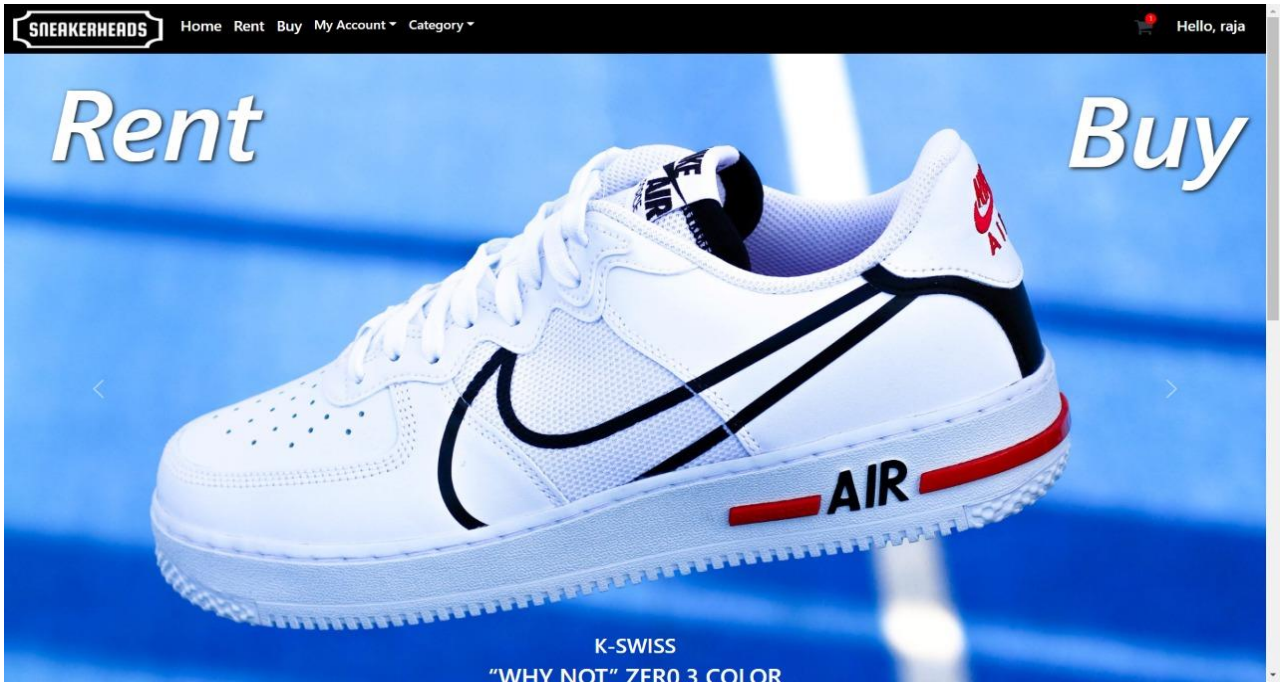
- React: ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front end library responsible only for the view layer of the application.
  A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The components are the heart of all React applications. These Components can be nested with other components to allow complex applications to be built of simple building blocks. ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.
- MySQL: MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company**.** MySQL follows the working of Client-Server Architecture. This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services. Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched. The process of MySQL environment is the same as the client-server model.
- Spring: The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments. Core technologies: dependency injection, events, resources, i18n, validation, data binding, type conversion, AOP, mock objects, Test Context framework, Spring MVC, Test, WebTestClient. Data Access: transactions, DAO support, JDBC, ORM, Marshalling XML.
- Spring Boot: Spring Boot is a Spring module that provides the RAD (Rapid Application Development) feature to the Spring framework. Spring Boot is a project that is built on the top of the Spring Framework. It provides an easier and faster way to set up, configure, and run both simple and web-based applications. It is a Spring module that provides the RAD (*Rapid Application Development*) feature to the Spring Framework. It is used to create a stand-alone Spring-based application that you can just run because it needs minimal Spring configuration. In short, Spring Boot is the combination of Spring Framework and Embedded Servers**.**
- Testing: Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

API repository Easily store, catalog, and collaborate around all your API artifacts on one central platform. Postman can store and manage API specifications, documentation, workflow recipes, test cases and results, metrics, and everything else related to APIs.
The Postman platform includes a comprehensive set of tools that help accelerate the API lifecycle—from design, testing, documentation, and mocking to the sharing and discoverability of your APIs.

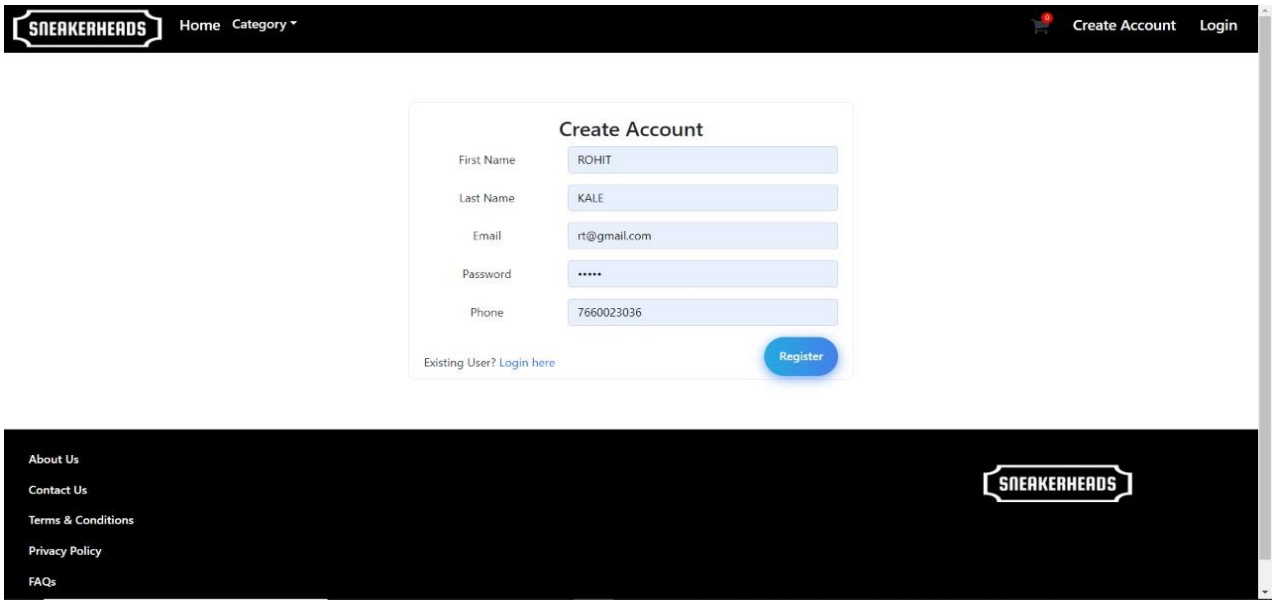
## 6. Test cases

| Test Case ID | Test Case Objective | Prerequisites | Steps | Input Data | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|---|---|
| TC_01 | Check the customer with registration | Text field should be enabled. | 1. Enter valid values in the required fields.<br>2. Click the Register button. | 1. Name="Rohit"<br>2.Lastname="Kale"<br>3.EmailID= "rohit@gmail.com"<br>4.Pass="Rohit123"<br>5.Phone=78454545 | User Should register on the website | Customer registration successful | PASS |
| TC_02 | Test the admin as pervalid login credentials | Text field should be enabled. | 1. Enter valid values in the required fields.<br>2. Click the login button. | 1.Username= "admin@gmail.com "<br>2.Pass="qwerty123 " | Admin should login to the website | Admin login successful | PASS |
| TC_03 | Test the admin as per invalid login credentials | Text field should be enabled. | 1. Enter valid values in the required fields.<br>2. Click the login button. | 1.Username= "admin@gmail.com "<br>2.Pass="qwerty12" | Invalid credentials | Invalid credentials | FAIL |
| TC_04 | Test the customer as per valid login credentials | Text field should be enabled. | 1. Enter valid values in the required fields.<br>2. Click the login button. | 1.Username= "rohit@gmail.com"<br>2.Pass="rohit123" | Customer should login to the website | Customer login successful | PASS |
| TC_05 | Usability | All links should be enabled for navigation | Have user click on the various links on the page /website | Click on the link from user login | Links will redirect user to another web page according to link address | Links worling successfully | PASS |

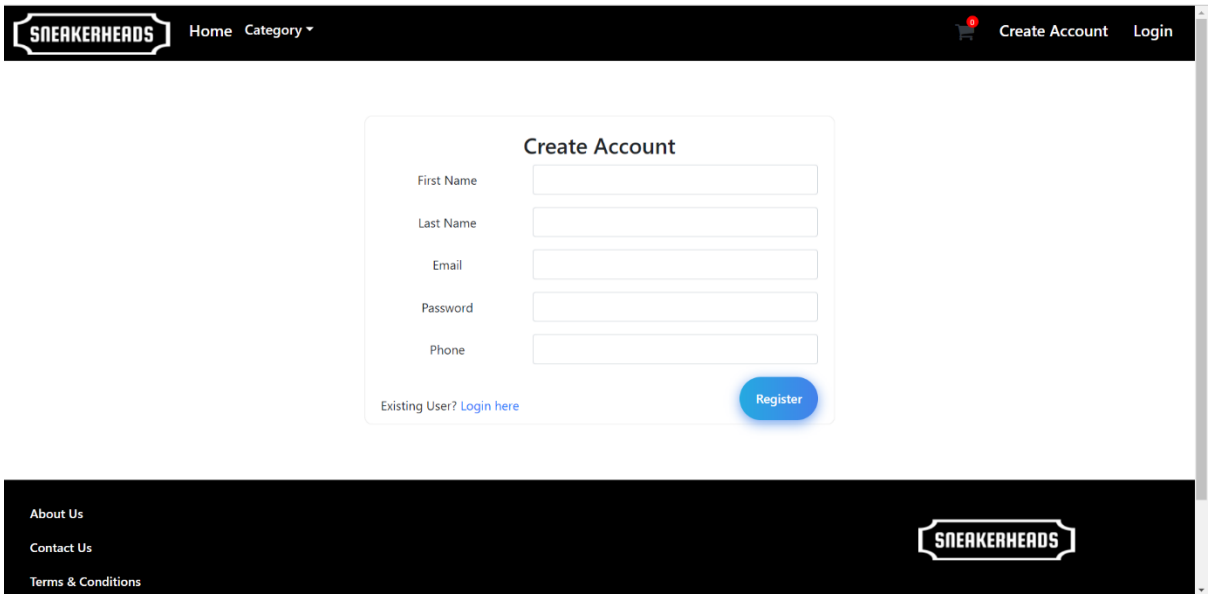
## 7. Screenshot of Webpages:



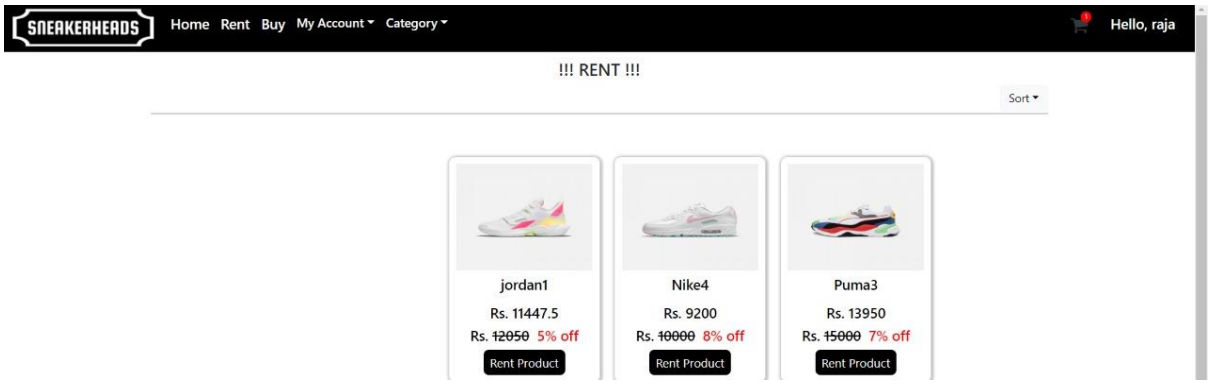

HOME PAGE



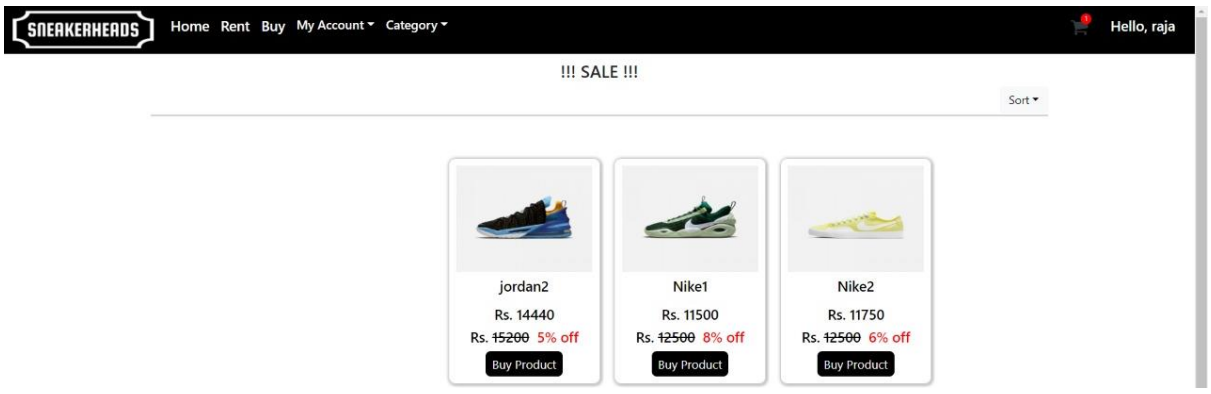

ADMIN PAGE




Customer Registration




RENT PAGE




BUY PAGE

## 8. Conclusion:

This proposed system for the buying and renting of the sneakers makes the customer to have the product at an affordable range and experience the most exclusive collection of the sneakers. This varied sneakers are in high demand among the people which gives a boost to the demand in these sneakers.

## 9. Future Enhancements:

- Accessories can be added.
- Kids category can be included.
- Realiable payments methods with securites can be included.

## 10. Bibliography:


https://www.w3schools.com/react/default.asp
https://www.w3schools.com/html/default.asp
https://www.w3schools.com/js/default.asp
https://www.w3schools.com/css/default.asp
https://www.javatpoint.com/mysql-tutorial
https://www.javatpoint.com/java-tutorial
https://www.javatpoint.com/spring-tutorial
https://www.javatpoint.com/spring-boot-tutorial

**User:**

| Field | Type | Null | key | Default | Extra |
|---|---|---|---|---|---|
| User_Id | Int | No | PRI | NULL | auto_increment |
| User_email | Varchar(20) | | | | |
| User_FirstName | Varchar(10) | | | | |
| User_LastName | Varchar(10) | | | | |
| User_Password | Varchar(10) | | | | |
| User_Address | Varchar(30) | | | | |
| User_Contact_no | Long | | | | |

**Buy:**

| Field | Type | Null | key | Default | Extra |
|---|---|---|---|---|---|
| Buy_id | int | NO | PRI | NULL | auto increment |
| Status | Varchar(50) | | | | |
| Buy_Price | Varchar(20) | | | | |
| User_id | int | FK | | | |
| Prodcut_id | int | FK | | | |
| Discount | int | | | | |

**Product:**

| Field | Type | Null | key | Default | Extra |
|---|---|---|---|---|---|
| Product_id | int | NO | PRI | NULL | auto_Increment |
| Product_Name | Varchar(10) | | | | |
| Product_Description | Varchar(50) | | | | |
| Discount | int | | | | |
| Product_Image | varbinary | | | | |
| Product_Price | int | | | | |
| Product_Rating | Varchar(10) | | | | |

**Rent:**

| Field | Type | Null | key | Default | Extra |
|---|---|---|---|---|---|
| Rent_Id | int | NO | PRI | NULL | auto_Increment |
| Rent_price | Float | | | | |
| Product_id | int | | FK | | |
| Estimated_price | Float | | | | |
| Status | Varchar(10) | | | | |
| Deposit | Float | | | | |
| Book_Duration | int | | | | |
| User_Id | int | | | | |

**Payment:**

| Field | Type | Null | key | Default | Extra |
|---|---|---|---|---|---|
| Payment_id | int | No | PRI | NULL | Auto increment |
| User_Id | int | | FK | | |
| Payment_Type | Varchar(10) | | | | |
| Payment_Date | Date | | | | |
| Payment_Status | Varchar(10) | | | | |
| Order_Id | Varchar(10) | | | | |

**Cart:**

| Field | Type | Null | key | Default | Extra |
|---|---|---|---|---|---|
| Cart_Id | int | No | PRI | NULL | auto increment |
| Product_Name | Varchar(10) | | | | |
| Product_id | int | | FK | | |
| Order_type | Varchar(10) | | | | |
| User_id | int | | FK | | |
| Price | Float | | | | |

**Product Category:**

| Field | Type | Null | key | Default | Extra |
|---|---|---|---|---|---|
| Category_id | Int | No | PRI | NULL | auto increment |
| Category_Type | Varchar(10) | | | | |
| Product_id | Int | | FK | | |

**Record:**

| Field | Type | Null | key | Default | Extra |
|---|---|---|---|---|---|
| Product_id | int | No | PRI | NULL | auto_increment |
| Product_details | Varchar(10) | | | | |
| Product_status | Varchar(10) | | | | |