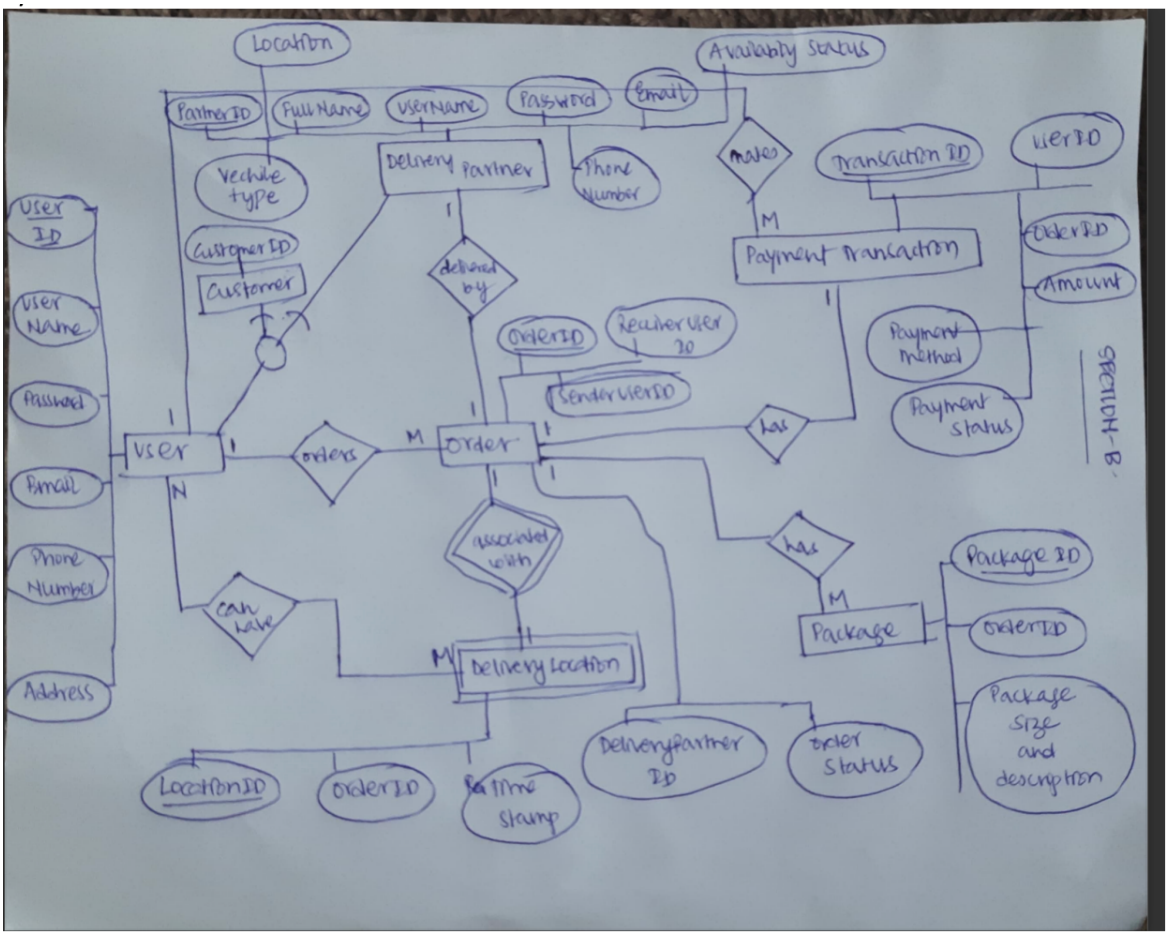
SOWMYA ADAVELLI
A20448281

ONLINE DELIVERY SYSTEM

An online delivery system serves as a secure and user-friendly platform, allowing customers to effortlessly send their desired products to any chosen destination. Upon logging in or signing up, users can choose between "place new order," "view order," and "cancel order" buttons. Customers can choose any one of them to either place a new order, view an already placed order, or cancel the placed order. Customers can enter the package size and description, prompting the system to calculate the associated delivery cost. Delivery partners can also log in or sign up to the system. A delivery partner is then assigned based on their availability, ensuring the safe and timely delivery of the package, ultimately saving customers valuable time. Once the package is successfully delivered, the system seamlessly handles the payment process, providing a convenient and reliable end-to-end experience for users.

ER Diagram :



Above is the ER model for Online Delivery System.

Implementation of Database for Online Delivery System :

Here's a brief description of the tables in the database for an online delivery system:

1. **Customer Table** : This table has all the information regarding a customer . Person who is responsible for placing orders.

    - CustomerID: Unique identifier for each customer.
    - FullName: The full name of the customer.
    - Email: Email address of the customer.
    - Password: Encrypted password for customer login.
    - PhoneNumber: Contact number of the customer.
    - Address: The address of the customer for delivery purposes.
2. **DeliveryPartner Table**: Table has all the information related to a delivery partner who ensures safe delivery of package.
    - PartnerID: Unique identifier for each delivery partner.
    - FullName: The full name of the delivery partner.
    - Email: Email address of the delivery partner.
    - Password: Encrypted password for partner login.
    - PhoneNumber: Contact number of the delivery partner.
    - AvailabilityStatus: Enum value indicating whether the delivery partner is available for assignments.
3. **Orders Table**: Contains all the information related to an order which is placed .
    - OrderID: Unique identifier for each order.
    - CustomerID: Foreign key linking to the Customer table, representing the customer who placed the order.
    - PartnerID: Foreign key linking to the DeliveryPartner table, representing the assigned delivery partner for the order.
    - Location: The delivery location for the order.
    - OrderStatus: The status of the order, such as Picked Up, On the Way or Delivered.
4. **Package Table** : Contains all the package information such as size and description .
    - PackageID: Unique identifier for each package.
    - Size: The size of the package being delivered.
    - Description: Additional details about the package.
    - OrderID: Foreign key linking to the Orders table, indicating the order to which the package is associated.
5. **Payment Table**: Contains all the payment related data.
    - PaymentID: Unique identifier for each payment.

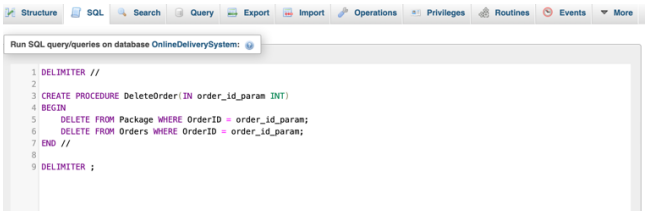
- Amount: The total amount for the order.
- PaymentMethod: The method used for payment, such as credit card or cash.
- PaymentStatus: The status of the payment, whether it is success or fail.
- OrderID: Foreign key linking to the Orders table, connecting the payment to a specific order.

These tables and their respective columns allow for the storage and management of customer and delivery partner details, order information, package specifics, and payment records within the online delivery system.

Trigger : I have created a trigger named "after_customer_delete" to maintain data integrity in the database. When a customer is deleted from the Customer table, this trigger ensures that all associated orders linked to that particular customer are also removed from the Orders table. This process prevents any orphaned records from persisting in the Orders table without a corresponding customer.

View : Created a view by performing an inner join between the Orders table and the Customer table. This view incorporates attributes such as OrderID and Location from the orders table, along with CustomerID and FullName from the customer table. The joining is based on the shared CustomerID in both tables. Subsequently, I utilized the delivery Location extracted from this newly created view to present it on the order summary details page.

Stored Procedure : After the customer clicks on the cancel order option, they are directed to a page where they can enter the order_id. Once the customer provides the order_id, the stored procedure is triggered. It takes the order_id as an input and proceeds to remove the corresponding rows linked to that specific order_id in both the payment and orders tables.



TOOLS AND LANGUAGES USED :

Front End:

Languages: HTML, CSS
Used HTML to create the web page, forms, actions. Added styling using the CSS.

Backend:
Languages: PHP, Javascript
PHP is used to write the logics for the webpages. It is used for server side scripting.

Web server and Database : XAMPP , phpMyAdmin and MYSQL
Created tables in MySQL . Added triggers, views and stored procedures to the tables.

TABLES AND BCNF :

1. CUSTOMER :

#	Name	Type
1	CustomerID	varchar(255)
2	FullName	varchar(255)
3	Password	varchar(255)
4	Email	varchar(255)
5	PhoneNumber	varchar(255)
6	Address	varchar(255)

2. DELIVERY PARTNER :

#	Name	Type
1	PartnerID	varchar(255)
2	FullName	varchar(255)
3	Password	varchar(255)
4	Email	varchar(255)
5	PhoneNumber	varchar(255)
6	AvailabilityStatus	enum('Yes', 'No')

3. ORDERS :

#	Name	Type
1	OrderID	varchar(255)
2	OrderStatus	enum('Picked Up', 'On the Way', 'Delivered')
3	CustomerID	varchar(255)
4	PartnerID	varchar(255)
5	Location	varchar(255)

4. PACKAGE :

#	Name	Type
1	PackageID	varchar(255)
2	OrderID	varchar(255)
3	Size	enum('SMALL', 'MEDIUM', 'LARGE')
4	Description	varchar(255)

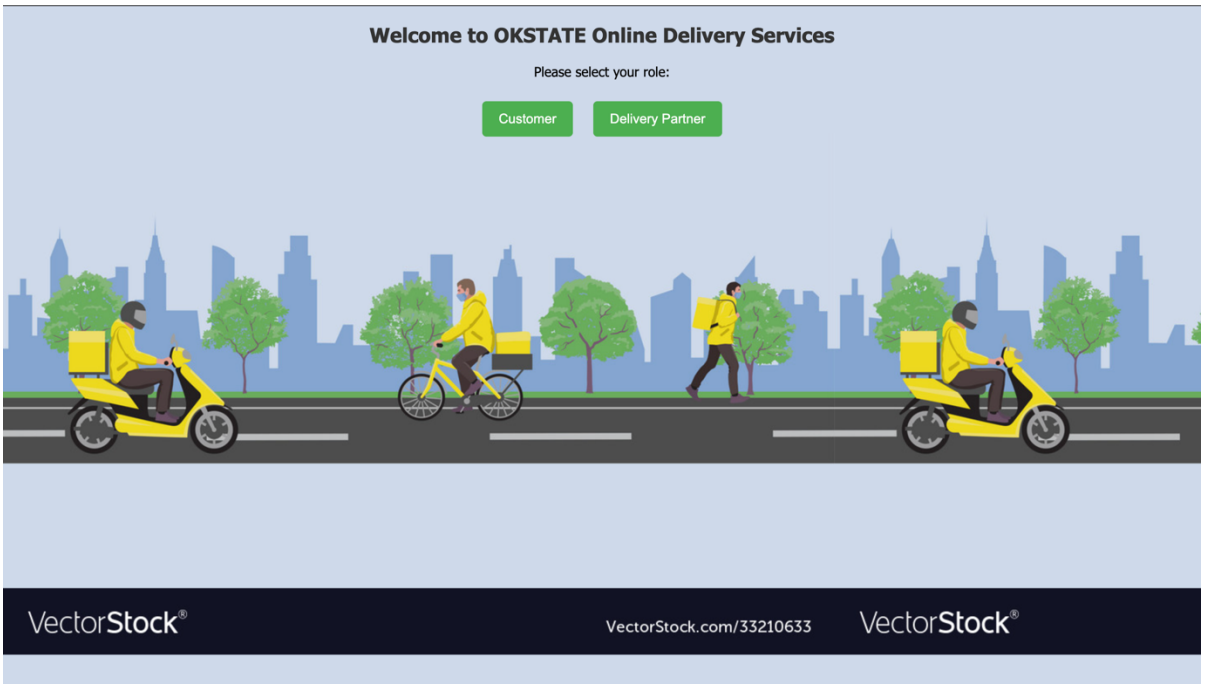
5. PAYMENT :

#	Name	Type
1	**PaymentID**	varchar(255)
2	**Amount**	double
3	**PaymentMethod**	enum('Card', 'Cash')
4	**PaymentStatus**	enum('Success', 'Fail')
5	**OrderID**	varchar(255)

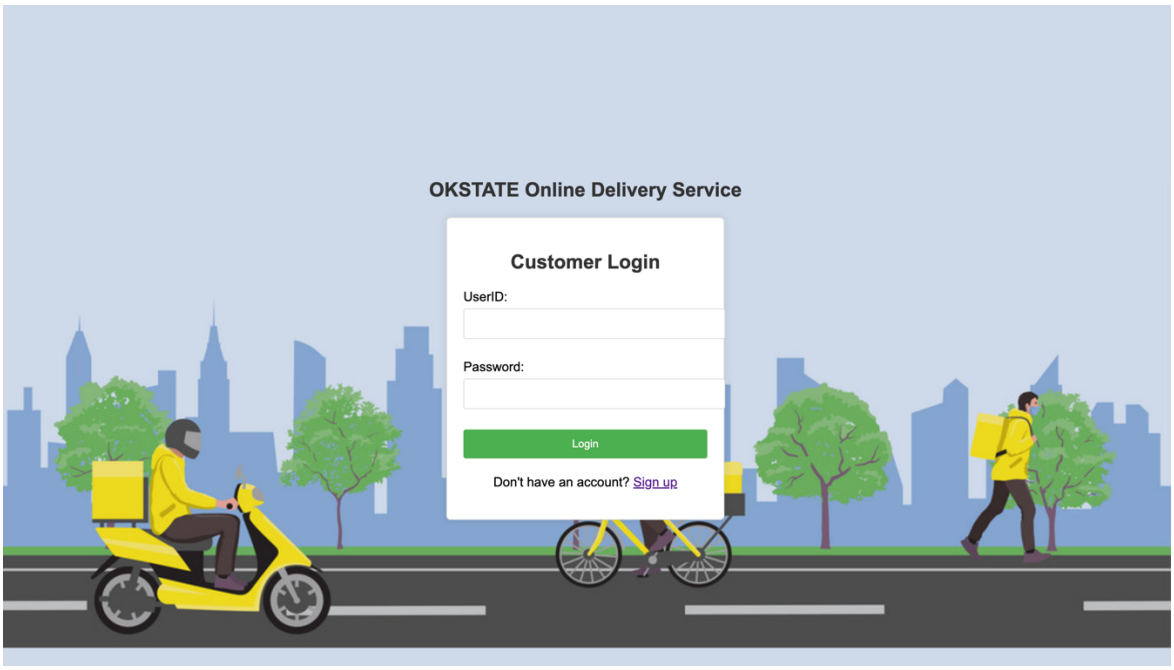
• **1NF** : The five tables mentioned above are in the first normal form (1NF) because they do not contain multi-valued attributes.
• **2NF** : All tables are in the second normal form (2NF) because no non-prime attribute can be derived from any proper subset of the candidate key. This indicates that data is organized efficiently.
• **3NF** : All tables are in the third normal form (3NF) because there are no transitive dependencies present. All columns can be retrieved directly from the primary key, contributing to data integrity.
• **BCNF** : Initially we had DeliveryLocation table (present in ER diagram) , there wasn't any use of this so deleted that table and added location attribute instead in Orders table . And also removed CustomerID from Payment table .
After these changes tables were in BCNF.

APPENDIX :

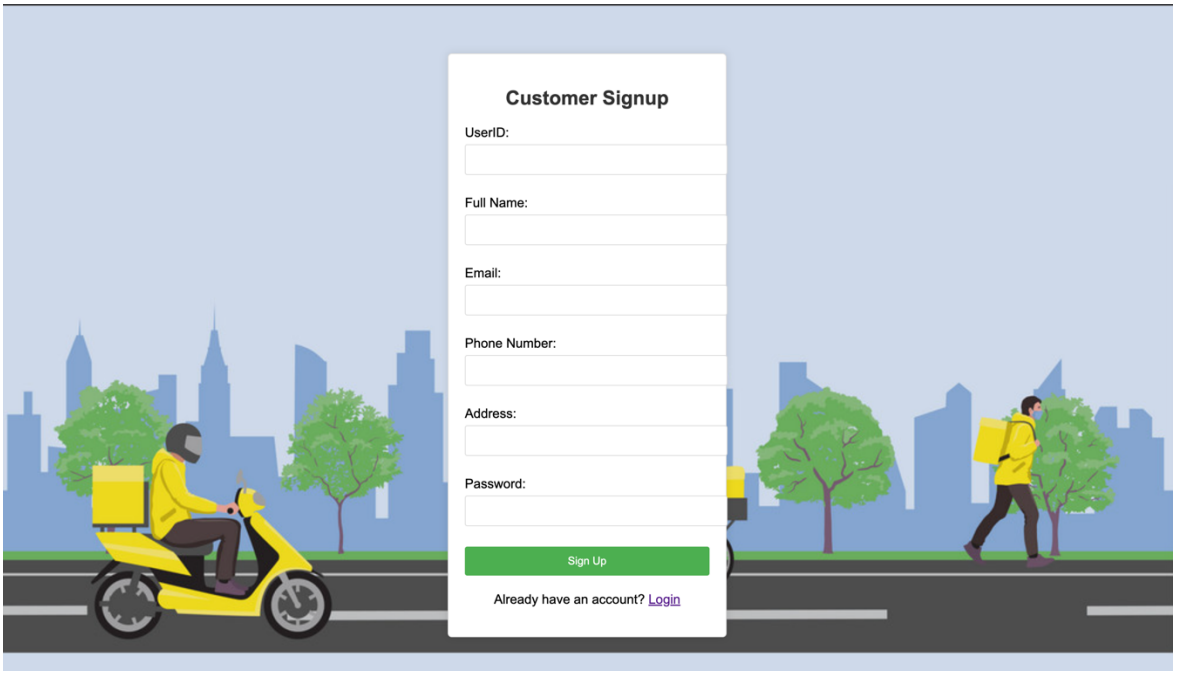
This is the homepage of the online delivery system. It features two buttons, each leading to separate login pages customized for either customers or delivery partners, depending on your role.



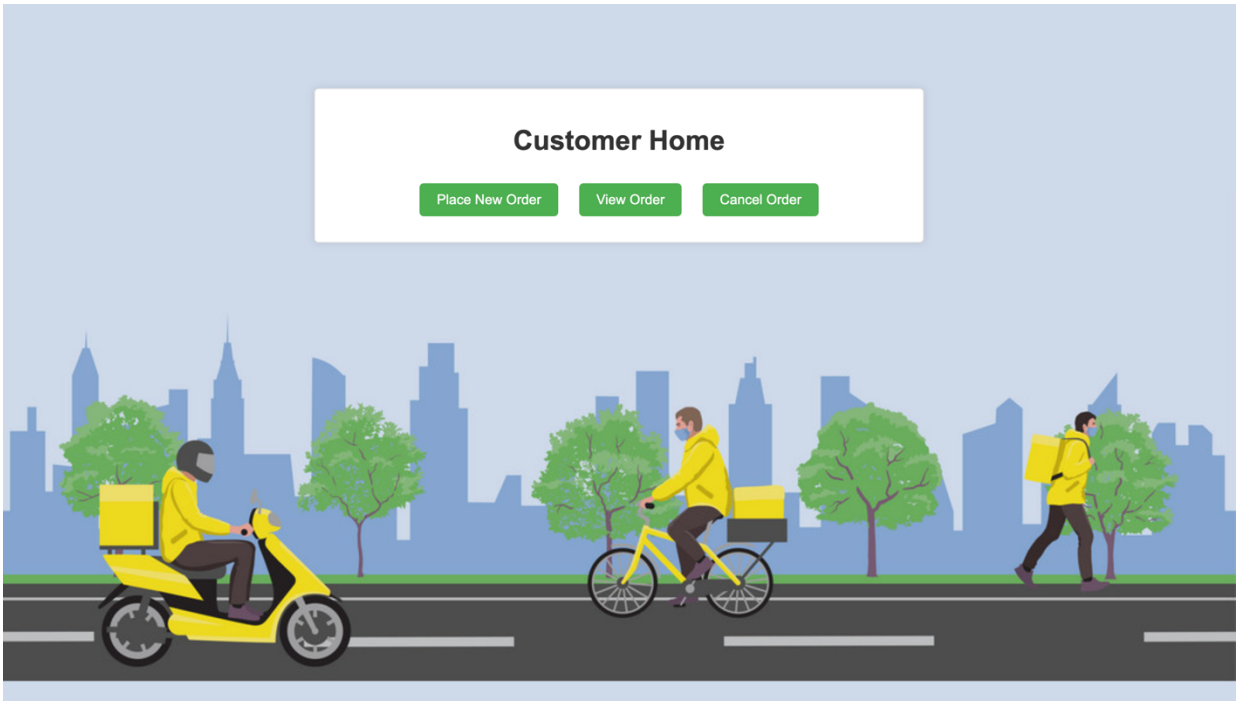
This is the customer login page where the customer provides their user ID and password to access the system.



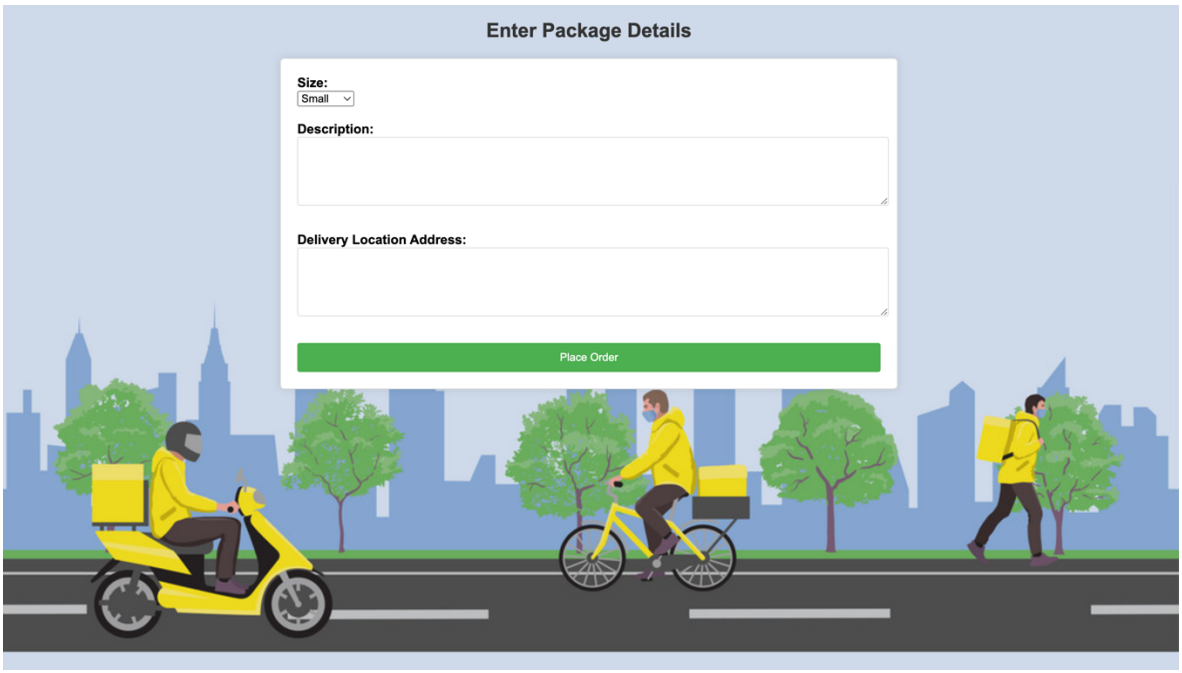
If it is the customer's initial login, they are required to sign up by entering their personal details. Upon successful sign-up, they are redirected to the login page to access the system using the appropriate credentials.



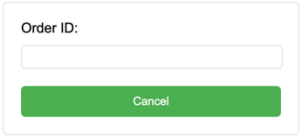
Upon successful login, the customer is directed to a home page designed for customers. This page features three buttons: "place new order," "view order," and "cancel order." The customer can select any of these options based on their preference.



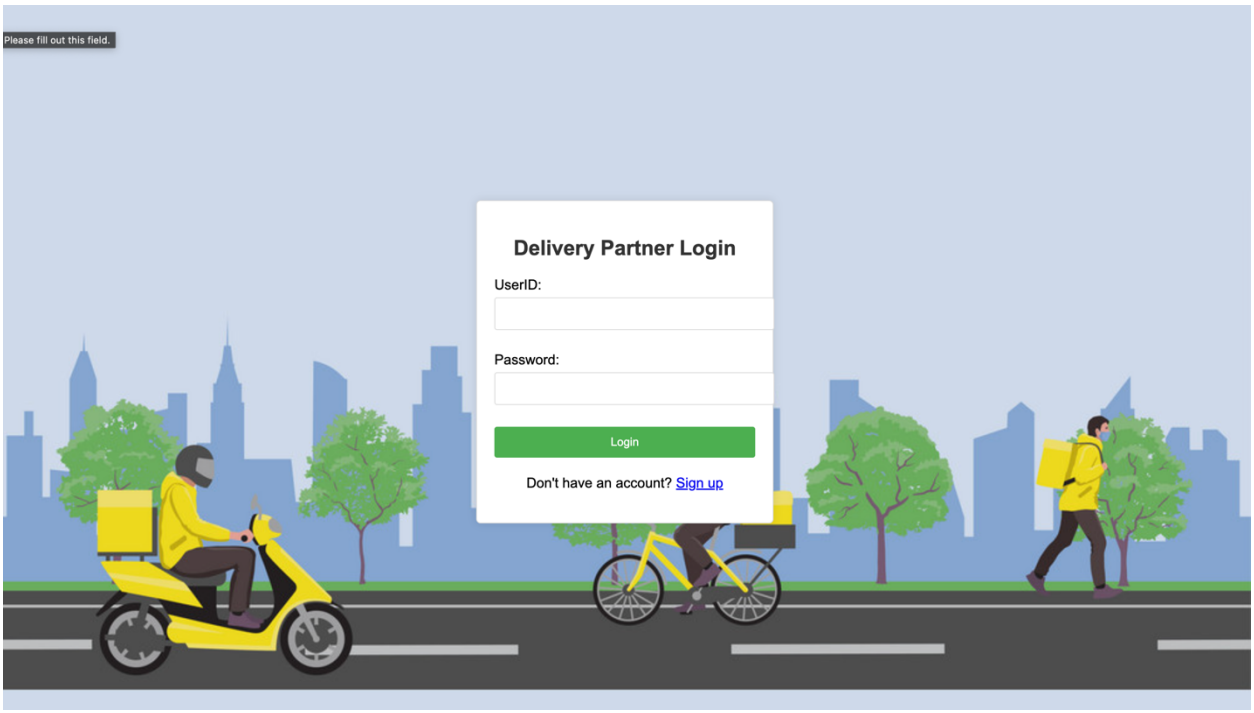
Upon choosing "place new order," you'll be directed to a page where you can input the package size, description, and delivery location address. After providing the required details, you can proceed by selecting "place order." Once the order is successfully placed, a subsequent page will present a comprehensive summary of all the order-related information.



Upon clicking the cancel order button on the customer home page, you will be redirected to the aforementioned page where you can input the order_id. Subsequently, the order associated with the provided order_id will be canceled.



This is the delivery partner login page where the delivery partner provides their user ID and password to access the system.



If it is the delivery partner's initial login, they are required to sign up by entering their personal details. Upon successful sign-up, they are redirected to the login page to access the system using the appropriate credentials.