|  |
| --- |
|  |
| Database Design Project |
| Logistics shinny Dashboard by Falomy Eugene,  Collaborators:  Timothy Sitorus, Mahmoud Ahmad, Ramsey Makhzoumi |

Contents

[I. Introduction: 3](#_Toc89351341)

[II. Scope of the project 3](#_Toc89351342)

[III. Goals of the project: 3](#_Toc89351343)

[IV. ERD Diagram 3](#_Toc89351344)

[V. Normalization 4](#_Toc89351345)

[VI. Dataset Challenges 5](#_Toc89351346)

[VII. URL shiny Application: https://tsitorus.shinyapps.io/LogisticsDashboard/ 5](#_Toc89351347)

[VIII. Shiny Application 5](#_Toc89351348)

[IX. Dashboard 5](#_Toc89351349)

[X. CRUD Functionality ( Create, Read, Update, and Delete) 5](#_Toc89351350)

[XI. Analytics 5](#_Toc89351351)

[XII. Appendix 6](#_Toc89351352)

### Introduction:

In this project, we designed a database for a Third-Party-Logistic firm (also known as a 3PL) called EMAS Logistics, which manages the transportation and logistics aspects of its clients' businesses. The company is headquartered in Dallas, Texas and deals with retailers like Wayfair, HomeDepot, and Ace Hardware to deliver goods to either business-to-business or business-to-consumer customers. Additionally, the business owns its own fleet which offers Truck-Load, Less-Than-Truck-Load, & Van delivery services to their clients at various prices. As we know from the current market situation, having supply chain inefficiencies or delays can cause huge bullwhip effects which essentially hurt both the producer and the consumer. We believe this project offers an effective and clean way for a retailer or 3PL to manage its supply chain operation and mitigate damage/risk from unforeseen circumstances that occur in the supply chain industry all the time.

### Scope of the project

Included in our database are the following categories of information to support the operations of XYZ logistics and its clients:

1. **Order Information:** This includes general information such as Load ID, Customer ID, Order Placement & Expected delivery date, Order Status, Route ID, Vehicle ID, and Driver ID
2. **Driver Information:** This pertains to Driver ID, Drivers First Name, Drivers Last Name, Drivers Phone Number, Driver Cost, and the Truck Number
3. **Distribution Information:** This table has route information such as Route ID, Route Mileage, Route Duration, Route Date, and Route Critical Pull Time
4. **Product Description Information:** This table includes information regarding Products ID, Description, Weight, Color, Category, Price, and Discounts
5. **Order Line Items Information:**  This withholds the respective Load ID, Products ID, and Quantity information
6. **Vehicle Information:** This includes Vehicle ID, Vehicle Description, Vehicle Weight, Vehicle Capacity, Vehicle Fixed Cost, Vehicle Miles per Gallon, and the Vehicle Cost per Mile
7. **Customer Information:** This table speaks has the Customer ID, Last Name, First Name, Address, City, State, Zip Code, Customer Entity Type, Customer Net Terms

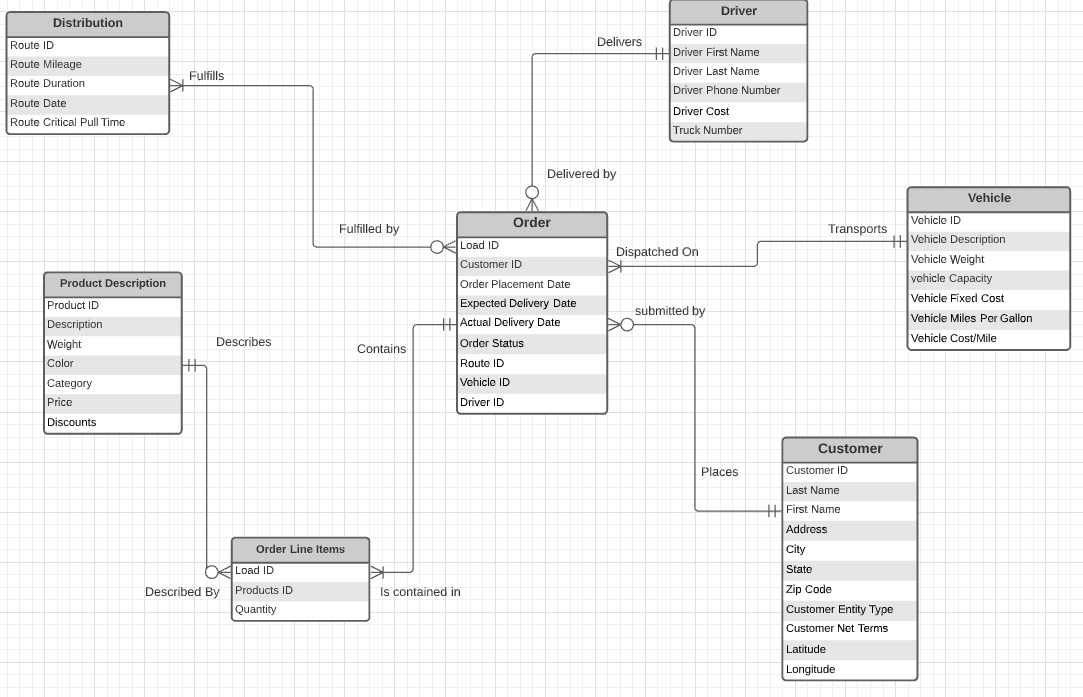
### Goals of the project:

The proposed use of this application is to provide XYZ logistics a platform where they can manage their supply chain operations in an efficient and user-friendly format. This application adds various accessibilities to XYZ logistics, for example, being able to edit order information and track it with an actual generated map or see current or past customer spending. These are just a few of many tools that are provided by the dashboard and application to XYZ logistics. As well, it should be noted that this platform also offers graphs that provide an analysis on various trends within the business and its operations.

It should also be noted that after successful implementation of this database and dashboard, we have the hopes that perhaps future retailers that deal with XYZ logistics could utilize this dashboard when shifting from outsourcing to inhousing their supply chain & logistical operations.

### ERD Diagram

The picture below is a snapshot of the Entity Relationship Diagram (ERD) of our application. As seen from the picture below, we have a total of 7 tables with relationships described in the picture below. Please refer to the Appendix A to see the data type and constraints. Table is designed as such so that for every order that is created all the different details of that order can be determined. For example, who the customer is or who the driver or even which type of vehicle that order is on can be determined. This is done so we can analyze all the different aspects of the business. We can analyze who takes which order, which customers are spending the most, the different trend of the customers and even the financials behind having the vehicles.

****

### Normalization

When designing the table, we also ensured that the tables that were created were normalized to avoid any redundancy and have data stored logically. We ensured that there were no dependencies on any of the non-primary key attributes, no partial dependencies, and no multivalued attribute.

For our design, the primary key used is the Load ID. Every load Id is unique and is connected to all the different tables. We also have other primary keys such as Driver ID, and customer ID which again ensured the 3rd normal form for our design. As seen from the diagram above, we removed the many to many relationship between Orders and Product description by adding a key table called Order Line Items. This table was created to satisfy the third normal form and not create an illogical database. Last but not least, we also ensured that there were no multivalued attributes in any of the tables to satisfy the first normal form.

### Dataset Challenges

When it came to the dataset, there were a few challenges when it came to the creation and the relationship of the data tables. When it comes to creation, there were challenges in the consistency of the data types. For example there were some tables that created different relational key types. So for the same key, there were tables that were storing it in strings and some as integers. There were also challenges in the relationships as mentioned before because there were many to many relationships that had to be fixed by creating a new table.

### URL shiny Application: <https://tsitorus.shinyapps.io/LogisticsDashboard/>

### Shiny Application

The link above will redirect you to the application. Further screenshots of the application are provided in the appendix found below

### Dashboard

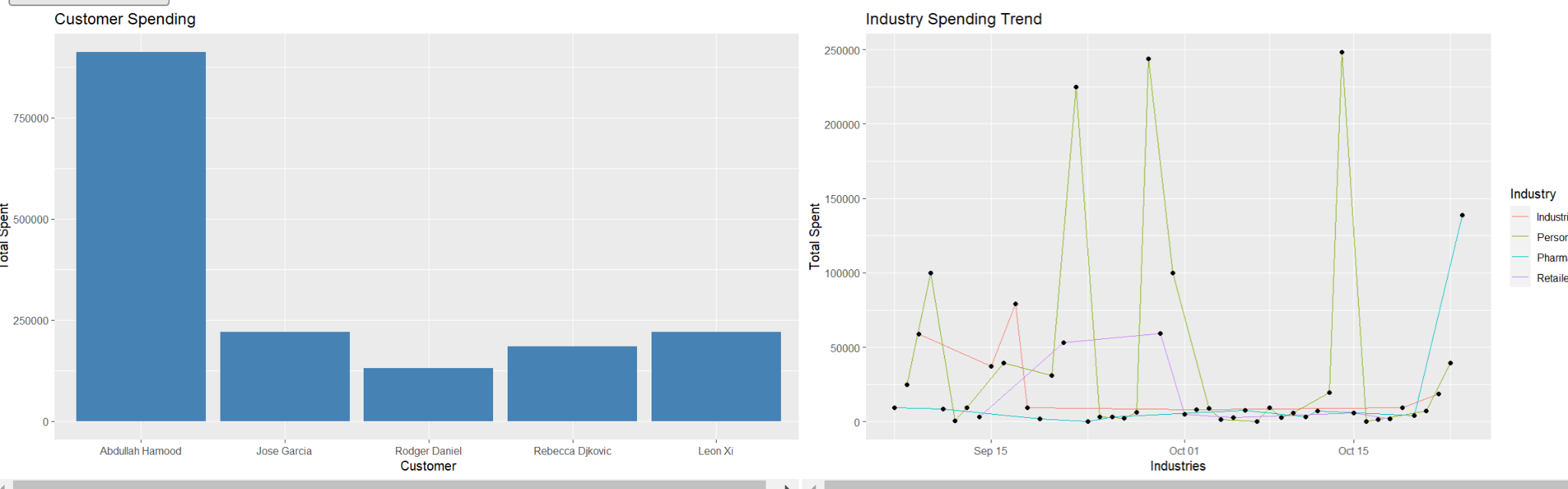
In our dashboard, we designed 6 different tabs, and each tab has different functionalities that aim to solve supply chain related issues. Users can both check and update order status, add and remove vehicle information, track customer spending and industry trend spending. See Appendix for additional information.

### CRUD Functionality ( Create, Read, Update, and Delete)

The (add) **create** function, under add/delete tab, enables users to create new vehicle information. Users can create vehicle information by adding a new vehicle ID, description, weight, capacity, fixed cost, miles per gallon, and cost per mile. Whereas, the **delete** function enables users to delete information about a particular vehicle, and users can only delete vehicle information using the vehicle ID. Moreover, the **read** function, under the welcome tab, helps users see the status of an order using a given load ID. Finally, the **update** function, under the update product tab, enables users to update the order status of a product. For instance, users change the status order of a product from delivered to canceled and so on.

### Analytics

Our logistic app performs analytics functions that help users analyze the spending trends within different industries such as industrial, pharmaceutical, and retailers. We created a line chart of total spending and industries colored by industries. As users enter a range of dates, they can see industry spending patterns, which we believe can be useful from a managerial perspective. Also, we created a bar chart that shows the patterns of customers within a particular date range. Finally, we created a word cloud that shows the number of loads each driver has taken. From a managerial perspective, we believe this can be helpful to know which driver is overworking and how the loads can be evenly distributed.



### Appendix

* 1. **Appendix A: Table Description and constraints**

**Order Line Items:**

|  |  |  |
| --- | --- | --- |
| **Column Name:** | **Data Type:** | **Description:** |
| Load ID | Varchar(50) | Load ID which is the ID for each order. Unique for every order |
| ProductsID | Varchar(50) | The ID for each product that is unique |
| Quantity | Int | Quantity of each products purchases |

**Distribution:**

|  |  |  |
| --- | --- | --- |
| **Column Name:** | **Data Type:** | **Description:** |
| RouteID | Varchar(50) | Unique ID for every route |
| RouteMileage | Int | Total mules for the route to be completed |
| RouteDuration | Decimal(5,2) | Time taken for the route to be completed |
| RouteArea | varchar(50) | Area of delivery |
| RouteCPT | time | Time route needs to depart |

**Product Description:**

|  |  |  |
| --- | --- | --- |
| **Column Name:** | **Data Type:** | **Description:** |
| ProductsID | Varchar(50) | Unique ID for every product |
| Description | Varchar(250) | Brief description of each product |
| Weight | Float | Weight of product |
| Color | Varchar(250) | Color of product |
| Category | Varchar(250) | Category product belongs to |
| Price | Float | Retail price of product |
| Discounts | Decimal(5,2) | Discounts if any of the product |

**Customer**:

|  |  |  |
| --- | --- | --- |
| **Column Name:** | **Data Type:** | **Description:** |
| Load Id | Varchar (50) | Load ID which is the ID for each order. Unique for every order |
| Customer ID | Varchar (50) | Unique ID for every Customer |
| Last Name | Varchar(50) | Customer Last Name |
| First Name | Varchar(50) | Customer First name |
| Address | Varchar(50) | Customer Address |
| City | Varchar(50) | Customer City |
| State | Varchar(50) | Customer State |
| Zip Code | Float | Customer zip code |
| CustomerEntitiy | Varchar(50) | Customer industry |
| CustomerNetTerms | Int | Customer net terms |
| Latitude | Float | Latitude of address |
| Longitude | Float | Longitude of address |

**Orders:**

|  |  |  |
| --- | --- | --- |
| **Column Name:** | **Data Type:** | **Description** |
| Load ID | Varchar (50) | Load ID which is the ID for each order. Unique for every order |
| Customer ID | Varchar(50) | Unique ID for every Customer |
| Order Date | Date | Date when order is placed |
| Expected Delivery Date | Date | When delivery is expected |
| Actual Delivery Date | Date | Actual delivery date of the order |
| OrderStatus | Varchar(50) | Status of the order |
| RouteID | Varchar(50) | Unique ID for every route |
| VehicleID | Varchar(50) | Unique ID for every vehicle |
| DriverID | Varchar(50) | Unique ID for every Driver |

**Vehicle:**

|  |  |  |
| --- | --- | --- |
| **Column Name:** | **Data Type:** | **Description** |
| VehicleID | Varchar(50) | Unique Vehicle Identification |
| VehicleDesc | Varchar(255) | The Description of each vehicle |
| VehicleWeight | Int | The weight of each vehicle measure in pounds |
| VehicleCapacity | Int | The capacity of each vehicle in terms of amount of load |
| VehicleFixedCost | Decimal(5,2) | Fixed  maintenance cost for each vehicle |
| VehicleMPG | Decimal(5,2) | Vehicle Miles per Gallon |
| VehicleCPM | Decimal(5,2) | Vehicle’s cost per mile |

**Driver:**

|  |  |  |
| --- | --- | --- |
| **Column Name:** | **Data Type:** | **Description** |
| DriverID | Varchar(50) | Unique ID  so that drivers can be easily identified |
| DriverFirstName | Varchar(50) | Driver’s First Name |
| DriverLastName | Varchar(50) | Driver’s last Name |
| DriverPhNumber | Float | Driver’s Phone Number |
| DLClass | Varchar(50) | Driver’s Licence |
| DriverCost | Decimal(5,2) | The cost for each driver |
| TruckNumber | Int | Unique ID for each truck |

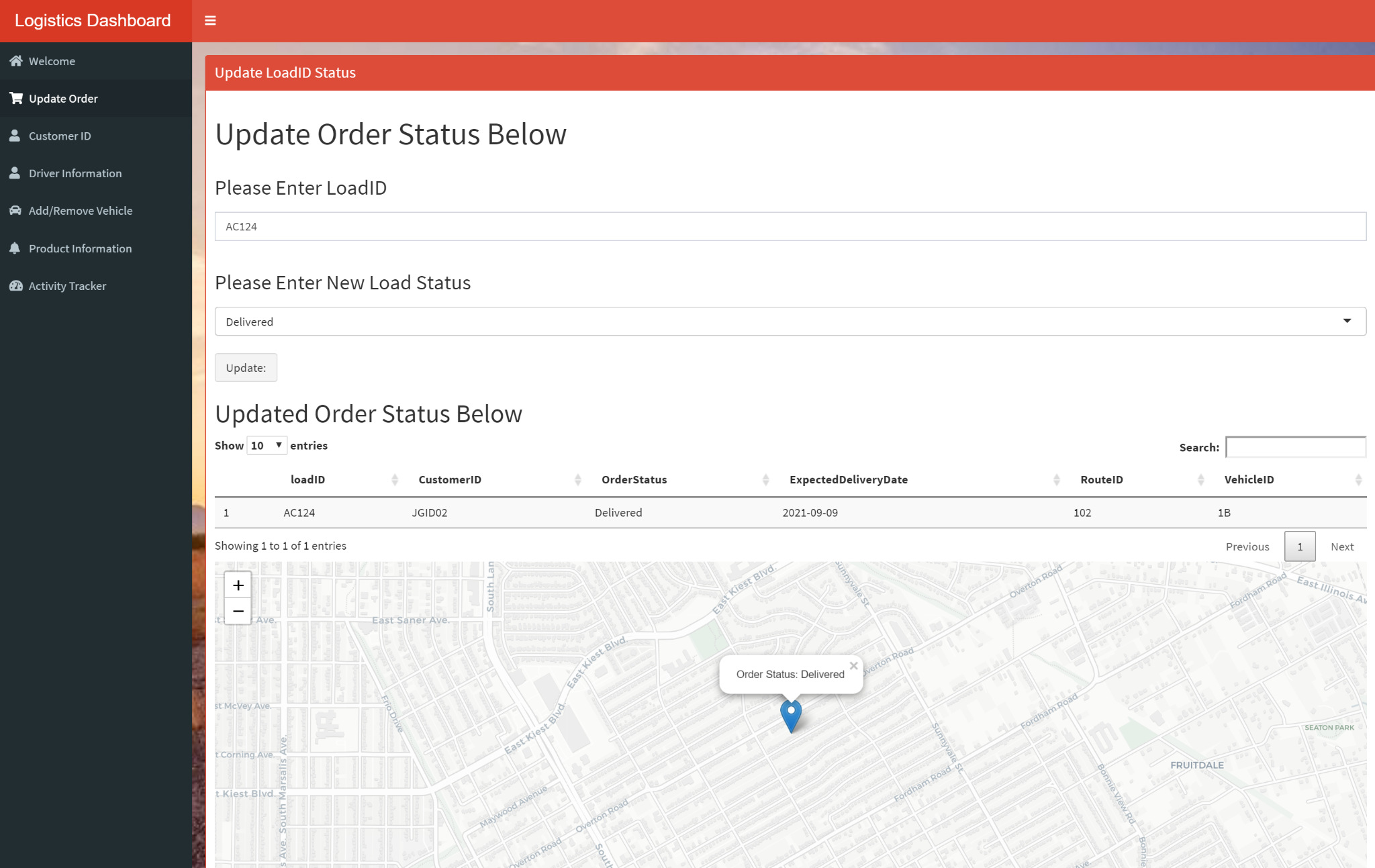
* 1. **Appendix B: Datatable relationships**
* Customer places 0 to many Orders
* Orders submitted by 1 customer
* Orders Dispatched on 1 Vehicle
* Vehicle transports 1 to many Order
* Orders Delivered by 1 Driver
* Driver Delivers 0 to many Orders
* Orders Fulfilled by 1 to many distribution
* Distribution fulfills 0 to many Orders
* Order Contains 1 to many order line items
* Order Line items is contained in 1 order
* Order Line Items Described by 1 Product Description
* Product Description Describes 0 to many Order line Items
  1. **Appendix C: Screenshots of the application**

**Welcome Tab:** The tab shows the home page of our application



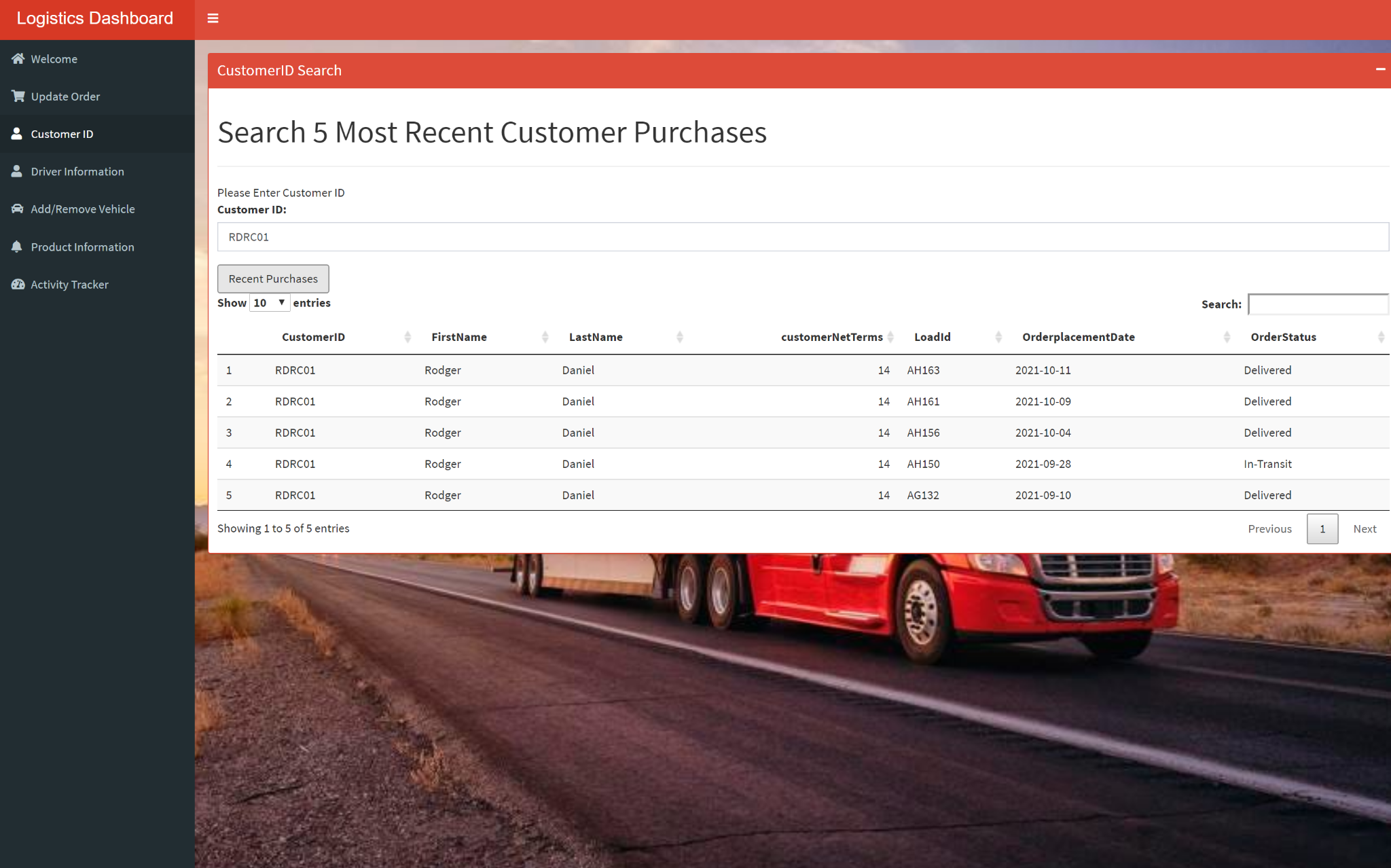
Users can enter the load ID to see order status. This fits the requirement of the **read function** of the CRUD operations.

**Update Oder Tab:**



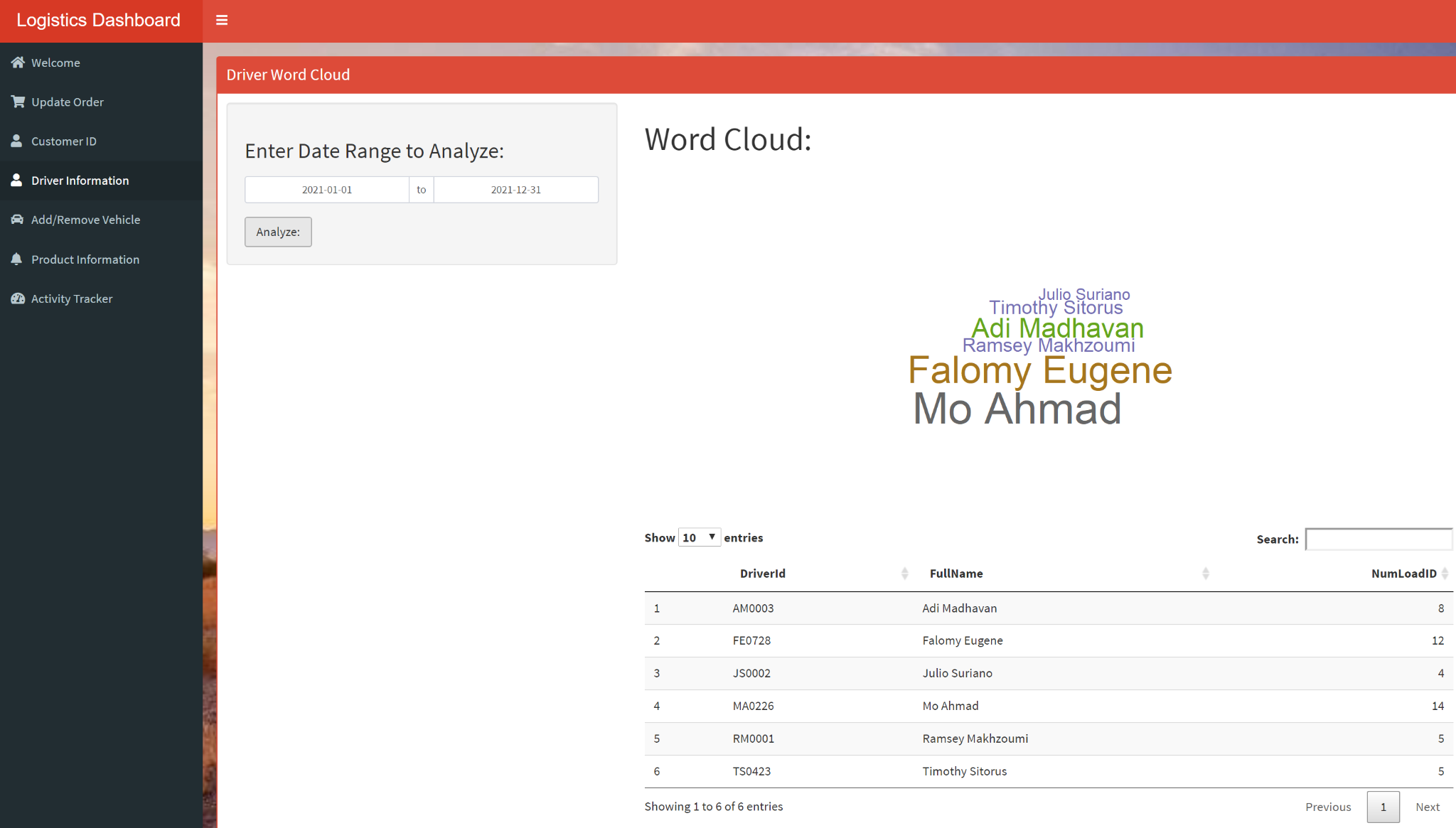
Users can enter the load ID to update the status of an order, and it will also show a map of where the order is located. This utilizes the **update function** of the CRUD operation.

**Customer ID Tab:**



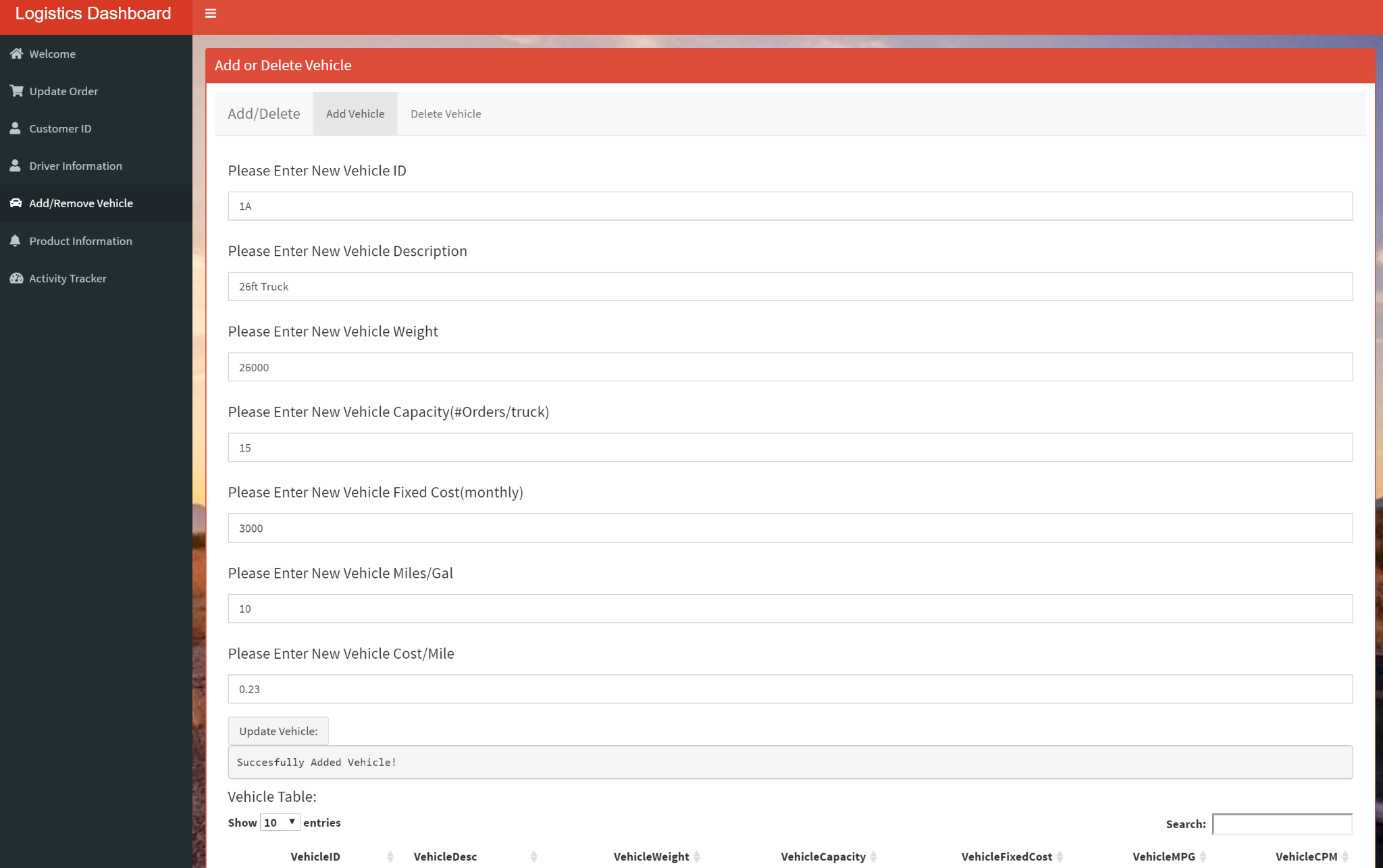
Users can enter customer ID to see the 5 most recent purchases and status for a particular customer.

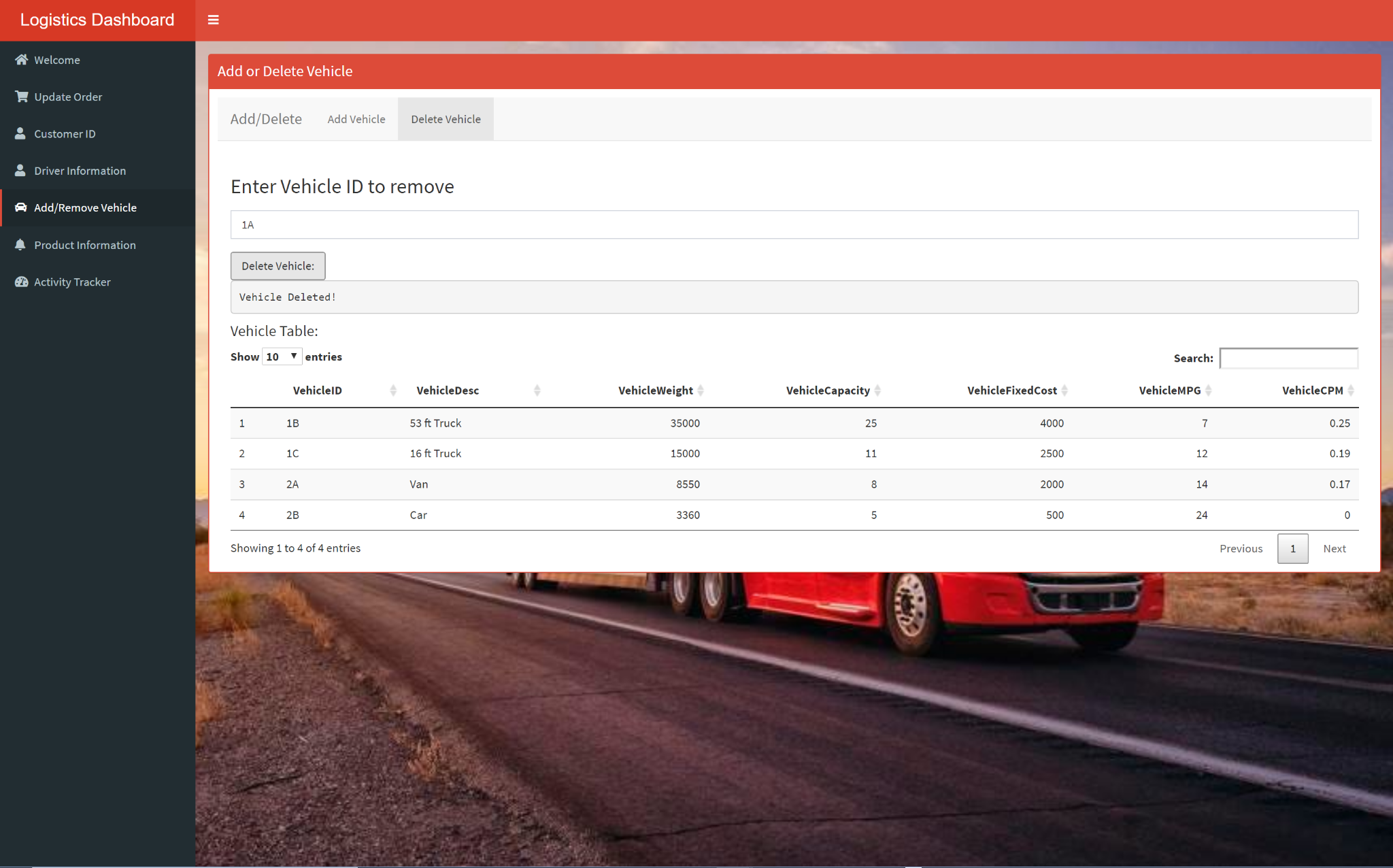
**Driver Information Tab:**



Users can choose a date range, then click Analyze, to see the number of loads each driver has taken. The bigger the names appear, the bigger the more the loads for the driver in question. This utilized the **Analytics function** of the CRUD operations.

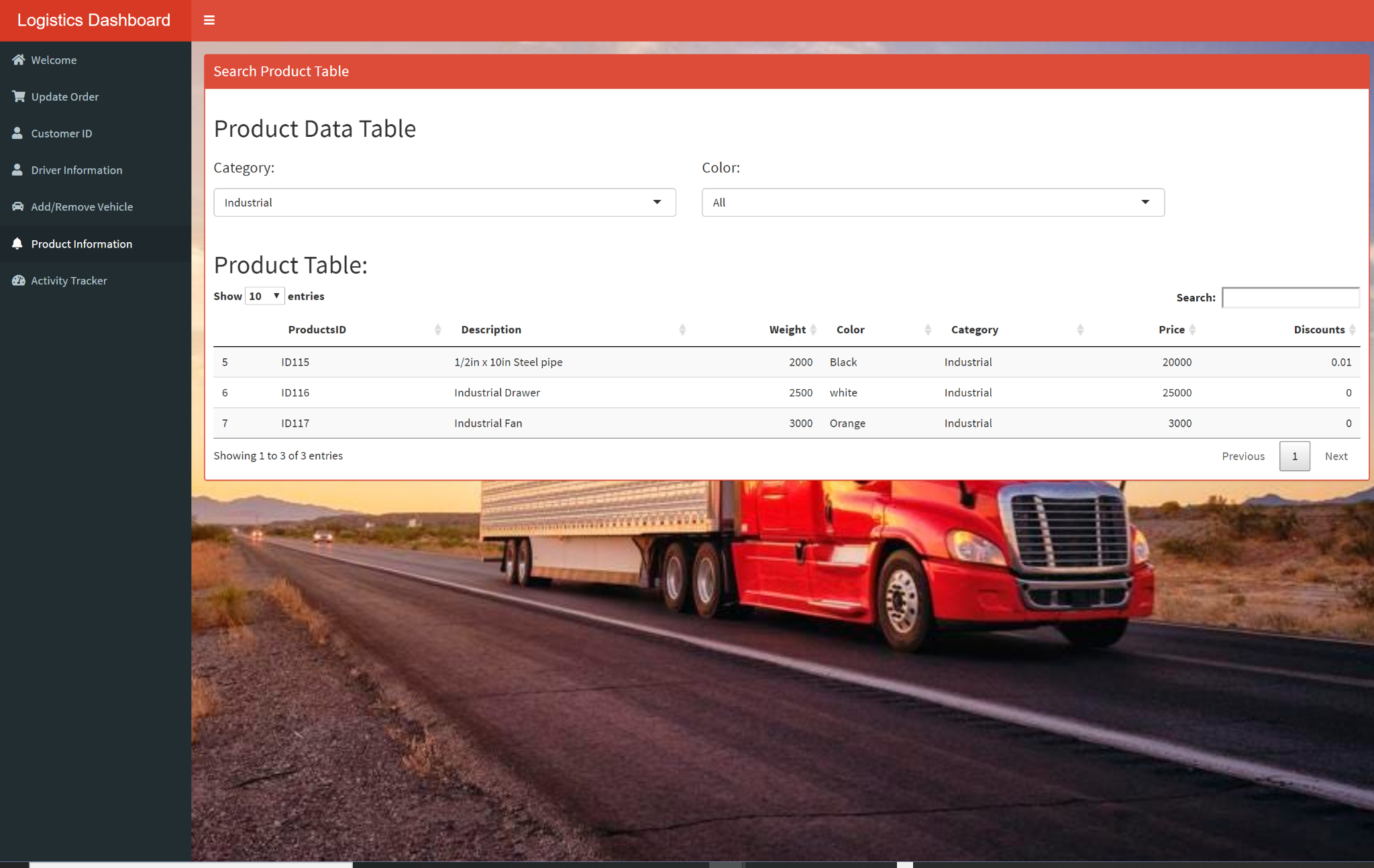
**Add/Remove vehicle Tab:**





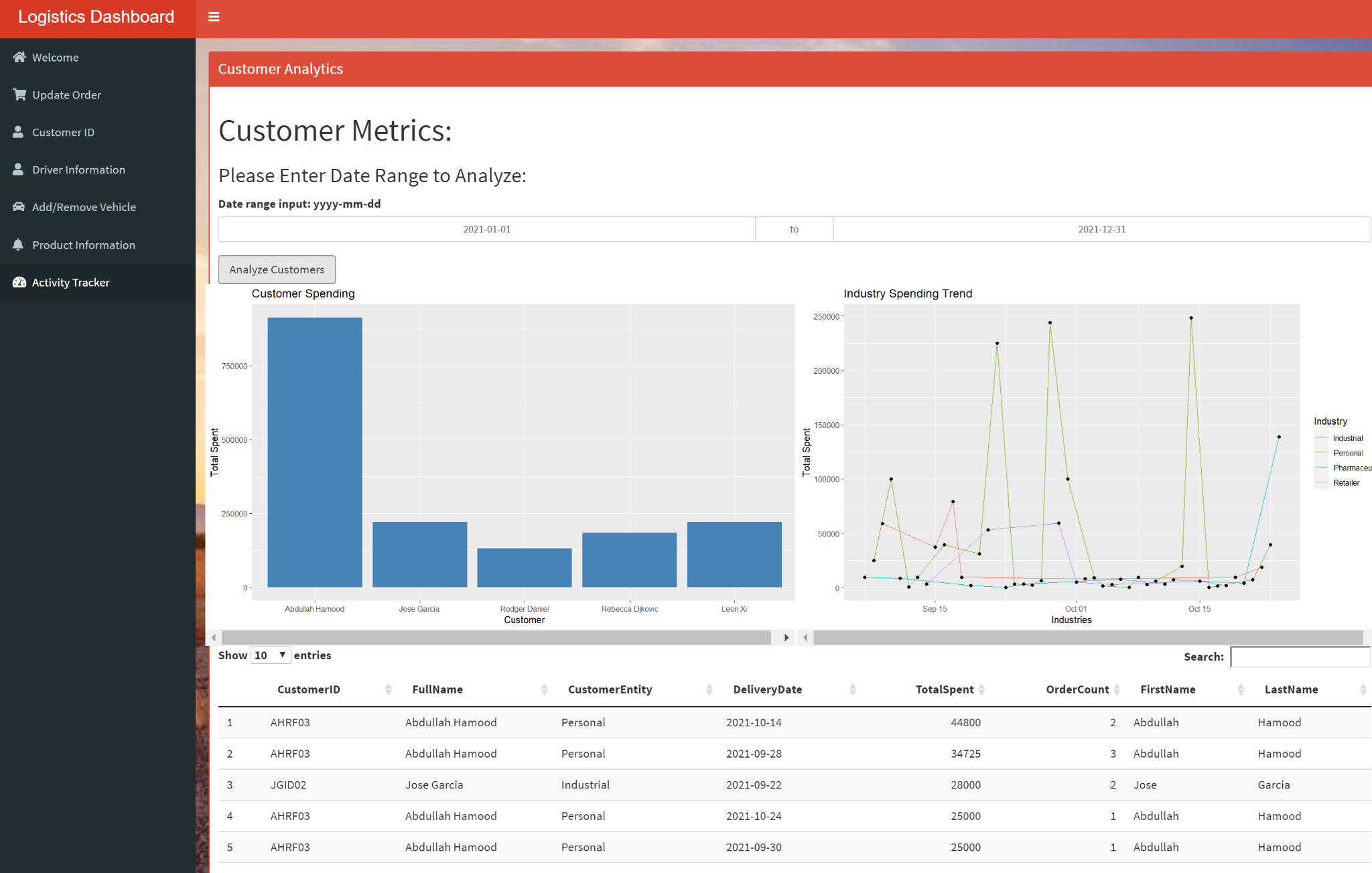
Users can both add and delete vehicles. This utilizes the **add and delete function** in the CRUD operation.

**Production Information Tab:**



Users can choose a category to see the information such as price, discounts, about products from the chosen category

**Activity Tracker Tab:**



Users can choose a range date to see analysis that is made based on industry trends, and which customer has spent the most. This utilizes the **Analytics function** in the CRUD Operation.