Introduction to Ego Rides and the Application

Ego Rides Rental services Pvt Ltd is a business offering both scooter rentals and sales to urban users for a sustainable, efficient, and reasonably priced transportation scooter rental and sales services. It offers electric kick-scooter rentals either for short-term use or the sale of scooters for long-term use. The business targets city dwellers, tourists, and scooter enthusiasts who values pocket environmentally friendly travel options.
The application developed for Ego Rides is specifically designed to manage sales leveraging an Excel front-end linked to an Access database to make the business run smoothly. This application includes VBA code that automates some of the key processes, including generating three main worksheets.
Orders Worksheet: This is where a list of customer sales are populated through a VBA subroutine. This will dynamically pull data from the database into the spreadsheet. It will have the ability to color-coded order statuses such as pending, shipped, delivered.
Order Form: A user-friendly form for the input of new sales orders from employees. All data validation checks are included, like product and customer exist, etc. Upon submission, this form will update the data in the database.
Customer Orders Sheet: This is the detailed view of orders a particular customer placed for easy referencing of the customer information, the status of the order, and product details involved. It will be populated through VBA subroutine from the table Customers and Orders.
This solution incorporates VBA code that simplifies scooter sales management, enhancing the general business workflow while maintaining an easy-to-use Excel front-end for business users.

The Access Database and Queries

The Ego Rides database is developed to assist sales operations. This database has been developed in Microsoft Access software and it contains multiple tables that are interconnected through the relationships. The database stores the data on customers, scooter inventory, orders and payments. Here is a sample of the tables and queries used in the Ego Rides database.

Customers Table: This table contains information about the customers who uses Ego Rides services. Each customer has a unique CustomerID that is linked to orders in the Orders Table.

Field Name	Data Type	Description
CustomerID	AutoNumber (Primary Key)	Unique identifier for each customer
Name	Text	Customer's name
Address	Text	Customer's address

Scooters Table: This table stores information about the scooters that are in stock. Each scooter has a unique ScooterID.

Field Name	Data Type	Description
ScooterID	AutoNumber (Primary Key)	Unique identifier for each scooter

Field Name	Data Type	Description
Model	Text	Model of the scooter
Price	Currency	Price of the scooter
Stock	Number	Number of scooters available for rent
Brand	Text	Brand of the scooter
Color	Text	Color of the scooter

Orders Table: This table tracks all orders placed by customers. It contains references to CustomerID and ScooterID, forming the relationships between the customer and the scooter purchased.

Field Name	Data Type	Description
OrderID	AutoNumber (Primary key)	Unique identifier for each order
CustomerID	Number (Foreign Key)	Reference to CustomerID in Customers Table
ScooterID	Number (Foreign Key)	Reference to ScooterID in Scooters Table
Quantity	Number	Number of scooters ordered
OrderDate	Date/Time	Date and time when the order was placed
TotalPrice	Currency	Total price of the order
Status	Text	Status of the order (e.g - Pending, Shipped, Delivered)

Queries in the Database:

Order List
This query retrieves order details including OrderID, Customer Name, Scooter Model, Quantity, Order Date, Total Price, and Status by joining the Orders, Customers, and Scooters tables.

Orders for Each Scooter Model
This query calculates the total quantity ordered and the total revenue for each scooter model. It joins the Orders table with the Scooters table to group the data by scooter model.

Revenue by Customer
This query calculates the total revenue generated by each customer. It joins the Orders and Customers tables and groups the result by Customer Name.

These queries provides insights to the business of Ego Rides by tracking individual orders, understanding sales performance of scooters, and analysing customer revenue contributions.

Front-End: Design and Business Usage of the Excel Sheet

This Ego Rides Excel front-end is designed to handle orders, customer interaction, and inventory. The Excel sheet connects to the Access database with the help of VBA queries and it contains four main worksheets, as follows:

Orders Worksheet:
This contains the details of the orders placed such as OrderID, Customer Name, Scooter Model, Quantity, Order Date, Total Price, and Status. The data populates through the subroutine GenerateOrderData(), which retrieves data from the Orders Table in Access.

Sales by Product Worksheet:
This sheet provides a pivot table to summarize total sales and quantity sold per scooter model to track product performance of the business.

Order Form Worksheet:
This form provides the facility for the user to enter new orders with the scooter model, quantity required, and details of the customer. The InsertNewProduct() subroutine will insert the order into the database once all the data is validated. This sheet also contains a button with data validation which reduces the human error.

Customer Orders Worksheet:
This sheet shows the customer-specific order details and outstanding orders. It is populated using the subroutine GetCustomerDetailsAndOutstandingOrders(), which retrieves customer data and outstanding orders.

VBA Middleware - Description of the VBA Code :

The VBA code explanation includes three basic subroutines interacting with an Access database, automating such an activity that involves the moving and data handling in Excel.

Since I was getting lot expression mismatch error while executing the codes, I have included **'CStr'** while Inner Joining queries to convert the value's data type to a string data type, so the return type of this function is a string.

Sub GenerateOrderData():
This subroutine is written to query information from three tables in Access, Orders, Customers, and Scooters for orders. It will write in the Orders WorkSheet within Excel with fields from OrderID to OrderStatus. In preparation for reading records, this subroutine has been programmed to clear this work area for newer information to come in.

Sub GetCustomerDetailsAndOutstandingOrders() :
This subroutine fetches the details of a customer whose name is taken as input. It queries the Customers table for customer information and the Orders Table for pending orders. It then displays the customer details and their outstanding orders in the Customer Orders Worksheet. If no orders are pending, it informs the user via a pop-up message box.

Sub InsertNewProduct():
This subroutine will allow the user to add new scooter products for both sales and rentals into the Scooters Table within the Access database. It pulls data from the Order Form Worksheet-

scooter model, price, stock and sets a new record in the Scooters Table. Before inserting the data, it will check for its validity to make sure all fields are properly filled.

Conclusion: Scaling the Ego Rides Application for a Real-World Business

The Ego Rides application developed on Excel, VBA, and an Access database can be upscaled for commercial purposes by improving features and integrating better tools. Moving all the data onto SQL Server or MySQL databases would let the company to access the data real-time across locations and increase the volumes of transactions.

The Excel workbooks can be upgraded with more advanced dashboards and reports to provide deeper insight into sales, customer preferences and inventory levels. Automated stock updates, payment processing system and detailed customer segmentation are some of the features that could make the operations seamless.

The VBA subroutines can be enhanced technically to do even better validation, advanced error handling and fuller automation to minimise the amount of manual intervention required. Also, the web interface or mobile application developed will enable the renting of scooters, ordering, and online payments helps engaging customers much better.

More sophisticated inventory management would enable real-time tracking of scooter availability and automatic order updates. As the customer base and data grow, the system would be able to handle large datasets without sacrificing performance or accuracy.

Once fully scaled, the Ego Rides application could serve as a comprehensive scooter rental and sales platform, meeting the demands of a broader market and supporting the business's growth objectives.

GitHub Link: https://github.com/Chandu89073/Ego-Rides.git