Database Application Proposal for Online Grocery Store, HealthyGreens

Introduction

Nowadays, more and more businesses are moving online to expand their customer base and reach a wider audience. E-commerce platforms offer convenience for customers allowing them to shop from anywhere anytime, offering a wide range of products, and quick scheduled deliveries. For companies operation online, the use of reliable technologies is crucial. This proposal outlines the development of a database application built to improve the user experience of an online grocery store. The proposed database application will enhance the functionality and efficiency of the online grocery store, ensuring better service and long-term success.

Objective

The main goal of this database application is to create a reliable and user-friendly system that manages all aspects of an online grocery store. This includes customer and order management, product inventory, and payment handling. The primary goal of the application is to optimize business operation, reduce errors, and possibly provide valuable insights through data analytics.

Target Users

The database application will serve the following user groups: individuals who shop for groceries online (customers), staff responsible for overseeing operations (managers, customer service support), personal responsible for collection of orders (personal shoppers).

Value Proposition

The database application will offer the following benefits: enhanced efficiency, improved customer experience, security, scalability, and comprehensive analytics. Firstly, all processes will be automated, reducing manual work and minimizing the number of errors. Secondly, the application will provide user-friendly shopping experience with updates on order and payment status, inventory levels, collector performance. Furthermore, the analytics and reports will provide valuable business insights to improve decision making. And lastly, the data will be securely storing according to compliance with industry standards to protect sensitive information.

Application Features and Description

The application will use various technologies and tools. The main programming languages are JavaScript, Python, and SQL. The front end will be implemented with React.js, the backend – with Django. SQL Server will be used for RDBMS.

Some of the major application features will be customer registration, customer management, product catalog management, order placement and processing, payment processing, reporting and analytics, and others.

Tools and Resources

The application will leverage various technologies and tools. React.js will be used for fronted development to design dynamic interactive user interface; Django will be used for backend development to build scaled and secure backend; the application will use Microsoft SQL Server as a database management system to ensure robust data storage and retrieval. The hosting provider for the application will be Microsoft Azure for its reliability and scalability.

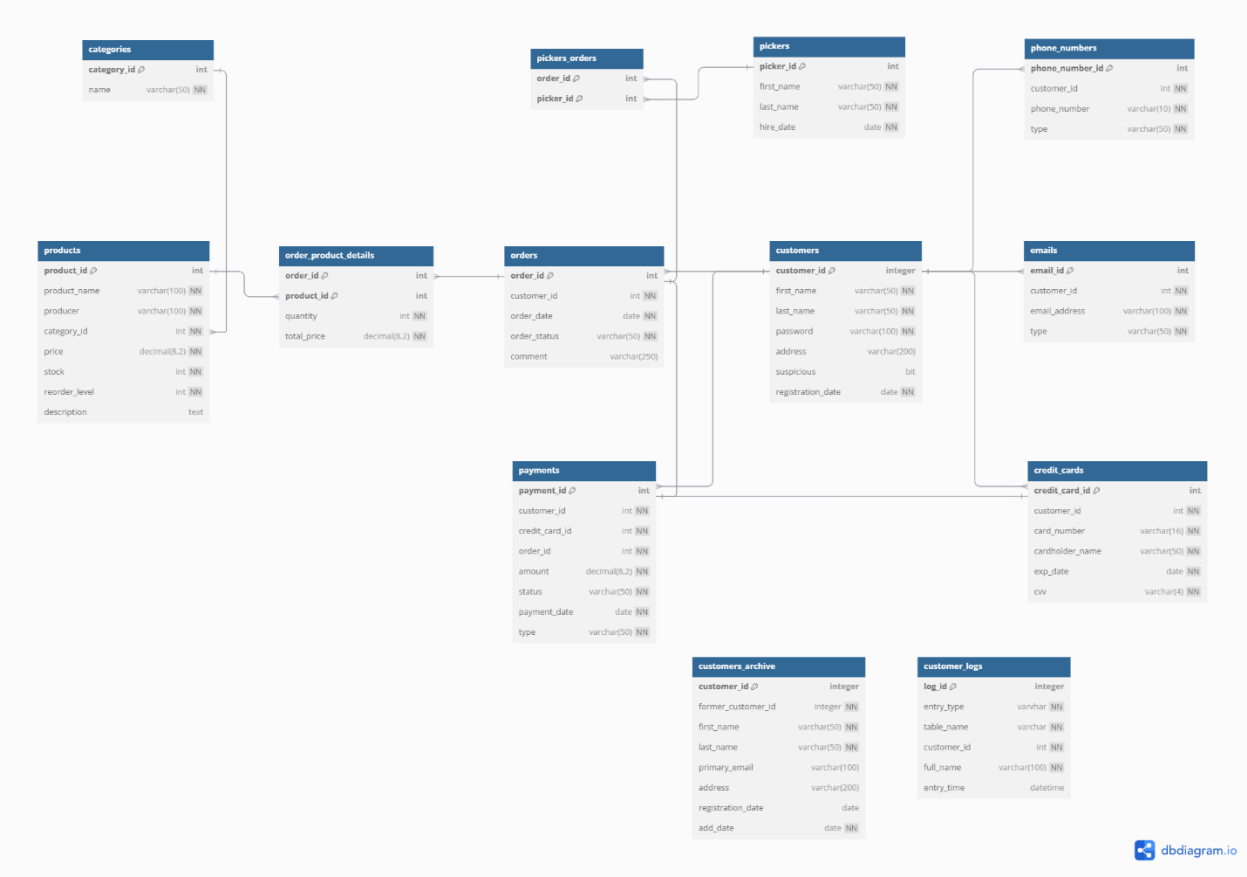
Challenges

The major challenges for this application will be ensuring that customers' personal data is stored securely, handling increasing volumes of data and transactions as the business grows, integration with existing systems, training of employees to use the application, and bug handling.

Expected ROI

By switching to the newly implemented database application, we are expecting to reduce operational costs by at least 10%, achieved through automation of various processes. Improved customer experience and higher customer retentions will lead to repeat purchases, increasing revenue by 5 – 12 %. Comprehensive analytics provided by the application will allow more effective business strategies. While scalability might be challenging, the application will allow to support business growth without significant additional investments.

Therefore, the implementation of the database application will drive business growth and provide significant return on investments.


categories
category_id int
name varchar(50) NN
pickers_orders
order_id int
picker_id int
pickers
picker_id int
first_name varchar(50) NN
last_name varchar(50) NN
hire_date date NN
phone_numbers
phone_number_id int
customer_id int NN
phone_number varchar(10) NN
type varchar(50) NN
products
product_id int
product_name varchar(100) NN
producer varchar(100) NN
category_id int NN
price decimal(8,2) NN
stock int NN
reorder_level int NN
description text
order_product_details
order_id int
product_id int
quantity int NN
total_price decimal(8,2) NN
orders
order_id int
customer_id int NN
order_date date NN
order_status varchar(50) NN
comment varchar(250)
customers
customer_id integer
first_name varchar(50) NN
last_name varchar(50) NN
password varchar(100) NN
address varchar(200)
suspicious bit
registration_date date NN
emails
email_id int
customer_id int NN
email_address varchar(100) NN
type varchar(50) NN
payments
payment_id int
customer_id int NN
credit_card_id int NN
order_id int NN
amount decimal(8,2) NN
status varchar(50) NN
payment_date date NN
type varchar(50) NN
credit_cards
credit_card_id int
customer_id int NN
card_number varchar(16) NN
cardholder_name varchar(50) NN
exp_date date NN
cvv varchar(4) NN
customers_archive
customer_id integer
former_customer_id integer NN
first_name varchar(50) NN
last_name varchar(50) NN
primary_email varchar(100)
address varchar(200)
registration_date date
add_date date NN
customer_logs
log_id integer
entry_type varvhar NN
table_name varchar NN
customer_id int NN
full_name varchar(100) NN
entry_time datetime
dbdiagram.io