

DBMS PROJECT

COURIER MANAGEMENT

SYSTEM

Abhiraj Banerjee (221210005)

Banoth Tanusri (221210029)

Bathula Venkata Mounika (221210030)

ACKNOWLEDGEMENT

On completion of this project ‘courier management system’, we would like to express our heartfelt gratitude to Dr Shelly Sachdeva, our course instructor for the course on Database Management System, for giving us this project as an opportunity to learn through first-hand work experience and providing her excellent guidance that has helped us throughout.

We would also extend our thankfulness to our Teaching Assistants, Ms. Anuj Kumar, Ms. Purnima Pradeep, Deepanshu Moyal, Aditya Soni for providing their valuable assistance and advice that have played a vital role in completing this project.

This project has been extremely useful for us to learn a lot about Database Management Systems and related topics. Through this project, we learned a lot about the use and implementation of DBMS in our real world. It has not only helped in broadening our horizons but has also provided us exposure to real-life projects and their usefulness.

We thank our course coordinator and faculties for giving us such an amazing and enriching experience.

CONTENTS

1.

INTRODUCTION

In undertaking the development of a courier management system project, despite the prevalence of existing services in the market, several compelling reasons underscore the importance and value of this initiative.

Firstly, the project serves as a robust learning experience, allowing for the practical application of theoretical knowledge in database management systems, software development, and system architecture. This hands-on engagement cultivates a deeper understanding of complex interactions between users, orders, couriers, and databases, enhancing problem-solving skills in real-world scenarios.

Moreover, the development process contributes significantly to skill development. Technical expertise in web development, database design, and system integration is honed, addressing challenges and fostering a more nuanced comprehension of system dynamics. This endeavor is not merely an academic exercise but a tangible addition to one's portfolio, showcasing practical skills to potential employers or clients.

The customization and integration aspects of the project are also noteworthy. Creating a bespoke courier management system allows for tailoring features to specific business needs, presenting a competitive advantage over generic solutions. Furthermore, the project might involve integration with other systems, demonstrating adaptability to diverse technological environments.

Beyond the technical realm, the project facilitates engagement with hypothetical clients or users, fostering the development of communication skills. Understanding and implementing client requirements are crucial facets, preparing individuals for effective collaboration in professional settings.

Importantly, the project sheds light on real-world issues such as security concerns, scalability challenges, or performance optimizations. The identification and resolution of these issues not only contribute to the refinement of the system but also enhance problem-solving abilities.

The educational value of the project extends beyond individual skill development. It can serve as a teaching aid in academic settings, providing students with practical examples that illustrate concepts related to logistics domain knowledge, coding best practices, and system design principles.

In conclusion, the decision to embark on the courier management system project is motivated by a multifaceted array of learning opportunities, skill development, and the potential to showcase capabilities in a dynamic technological landscape. While existing services may fulfil general needs, the creation of a bespoke system aligns with the pursuit of excellence and a deeper understanding of the intricate intersection between technology and logistics.

SYSTEM REQUIREMENTS

Hardware and software requirements are mentioned below:

HARDWARE REQUIREMENTS

* Operating System: Windows 7/8/8.1/10.
* Memory (RAM): 1 GB or above.
* Hard Disk Space: at least 200 MB.  Processor: Intel Pentium 4 or later.

SOFTWARE REQUIREMENTS

* Bootstrap Studio for designing the front-end.
* HTML, CSS for developing frontend.
* MySQL as a back-end query language.
* PHP shall be used as a scripting language.
* Web Browser (Chrome/Firefox, etc.)
* XAMPP as a web host on local server.

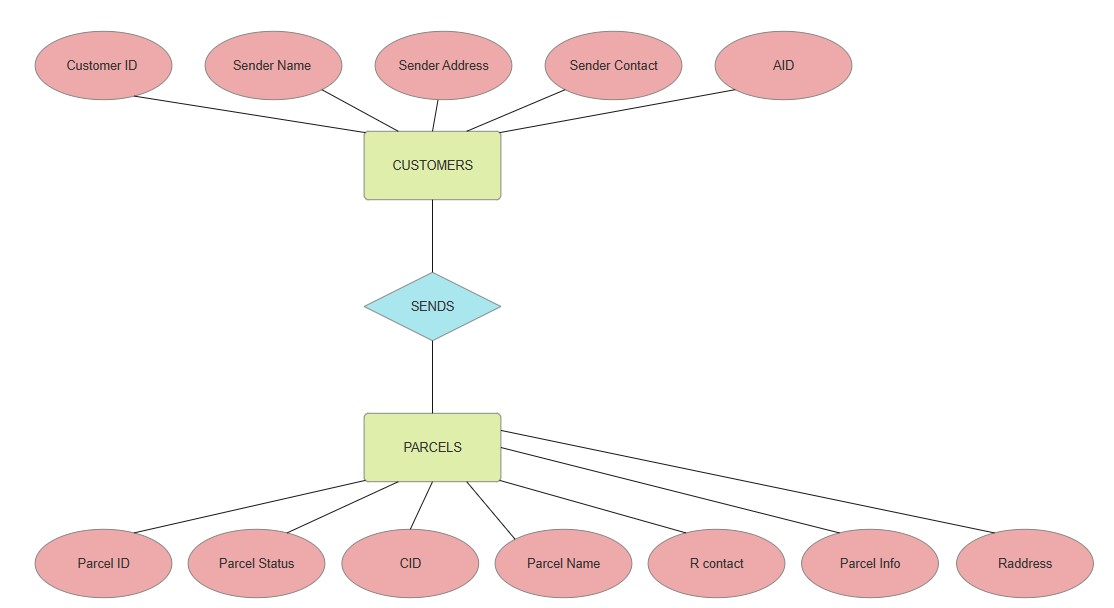
ASSUMPTIONS:

1. There is a single entity "Sender's Address" and "Receiver's Address" for every person and no separate columns for the full name

2.Every person is assumed to have a single contact number in the form of either a phone number or email.

3.Each person is assumed to be able to order as many parcels as he want so every person has a 1:n relation with a parcel.

ER Diagram



Mapping from ER Model to R Model

Customer

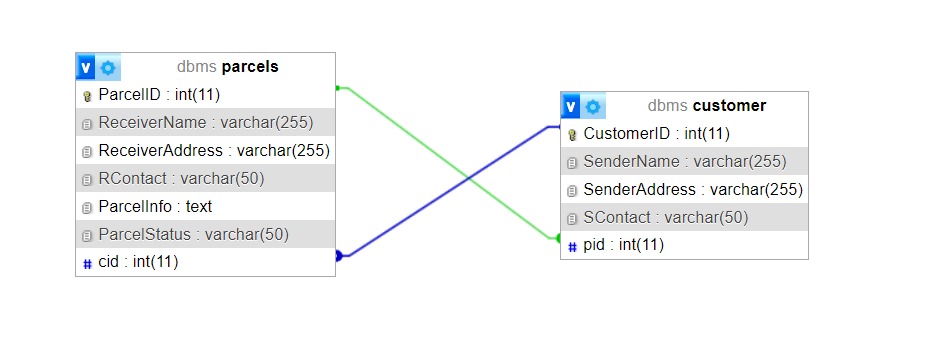
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customer id | Sender name | Sender address | Sender contact | pid |



PARCEL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parcel id | Parcel Status | CID | Parcel Name | Rcontact | Parcel Info | R address |

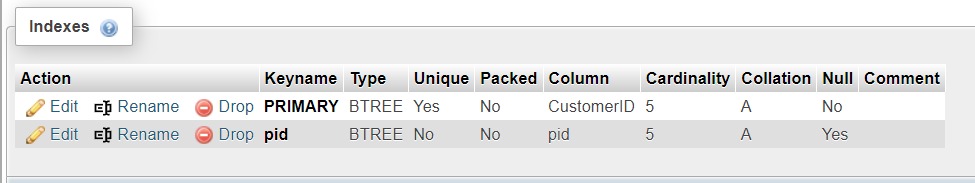
RELATIONAL SCHEMA



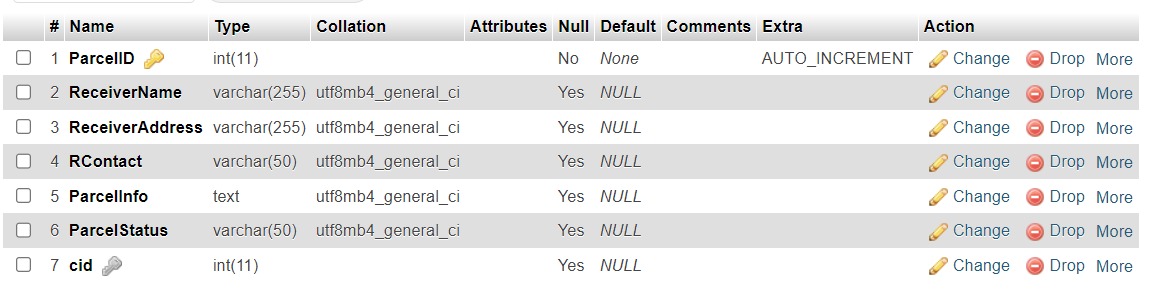
DATA DICTIONARY

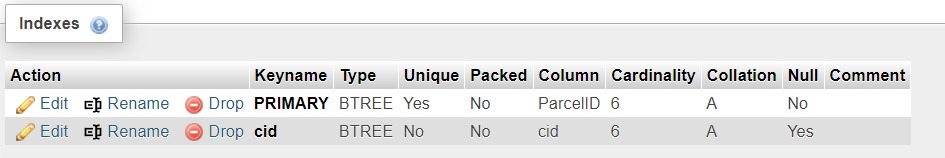
Customers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Links to | Comments | Media(MIME)type |
| CustomerID | Int(11) | No |  |  |  |  |
| SenderName | Varchar(225) | Yes |  |  |  |  |
| SenderAddress | Varchar(225) | Yes |  |  |  |  |
| Scontact | Varchar(50) | Yes |  |  |  |  |
| pid | Int(11) | NO |  |  |  |  |



Parcel





SQL QUERIES

Advanced SQL Queries for creating database

--

--Database: ‘courier management system’

--

-- --------------------------------------------------------------------------------------

CREATE DATABASE mydb;

USE mydb;

CREATE TABLE Parcels (

ParcelD INT(11),

ReceiverName VARCHAR(255),

ReceiverAddress VARCHAR(255),

RContact VARCHAR(50),

Parcelinfo TEXT,

ParcelStatus VARCHAR(50),

PRIMARY KEY (ParcelD)

);

CREATE TABLE Customers (

CustomerID INT(11),

SenderName VARCHAR(255),

SenderAddress VARCHAR(255),

SContact VARCHAR(50),

PRIMARY KEY (CustomerID)

);

ALTER TABLE Parcels

ADD COLUMN CustomerID INT(11);

ALTER TABLE Customers

ADD COLUMN ReceiverID INT(11);

ALTER TABLE Parcels

ADD FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID);

ALTER TABLE Customers

ADD FOREIGN KEY (ReceiverID) REFERENCES Parcels(ReceiverID);

--to insert values into the customer table

INSERT INTO Customer (SenderName, SenderAddress, SContact)

VALUES

('John Doe', '123 Main St', 'john@example.com'),

('Alice Smith', '456 Elm St', 'alice@example.com'),

('Bob Johnson', '789 Oak St', 'bob@example.com'),

('Emma Brown', '101 Pine St', 'emma@example.com'),

('Michael Lee', '555 Cedar St', 'michael@example.com');

--to insert values into the parcel table

INSERT INTO Parcels (ReceiverName, ReceiverAddress, RContact, ParcelInfo, ParcelStatus)

VALUES

('Sarah Johnson', '111 Hill St', 'sarah@example.com', 'Fragile items', 'In Transit'),

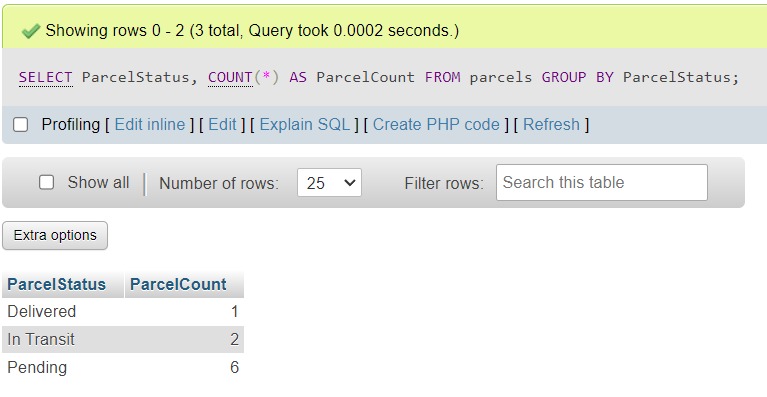
('David Wilson', '222 Valley St', 'david@example.com', 'Electronics', 'Delivered'),

('Olivia Adams', '333 Ridge St', 'olivia@example.com', 'Books', 'Pending'),

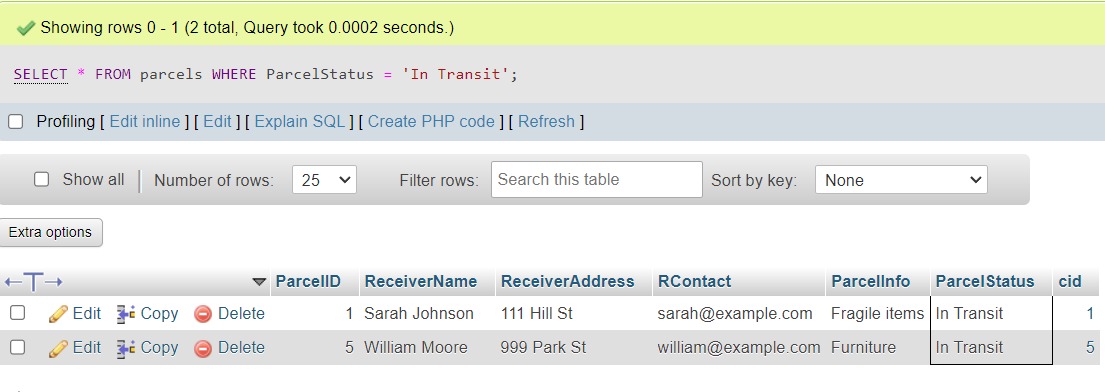
('Sophia Clark', '444 Lake St', 'sophia@example.com', 'Clothing', 'Pending'),

('William Moore', '999 Park St', 'william@example.com', 'Furniture', 'Delivered');

--SELECT ParcelStatus, COUNT(\*) AS ParcelCount FROM parcels GROUP BY ParcelStatus;



--SELECT \* FROM parcels WHERE ParcelStatus = 'In Transit';

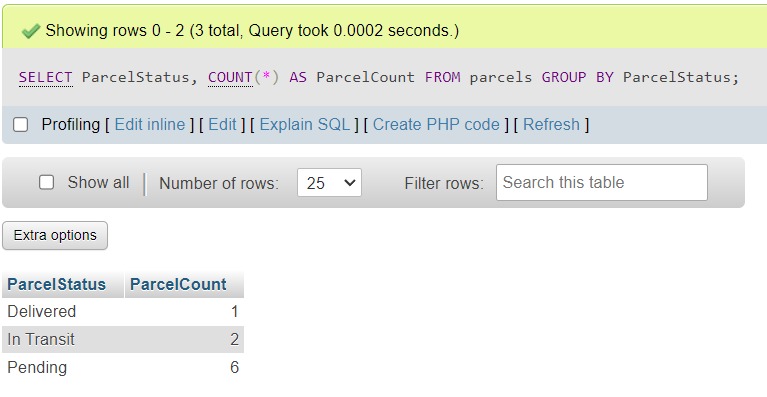


--SELECT SenderName, ReceiverName, COUNT(\*) AS TotalParcels

FROM customer

INNER JOIN parcels ON customer.CustomerID = parcels.CustomerID

GROUP BY SenderName, ReceiverName;

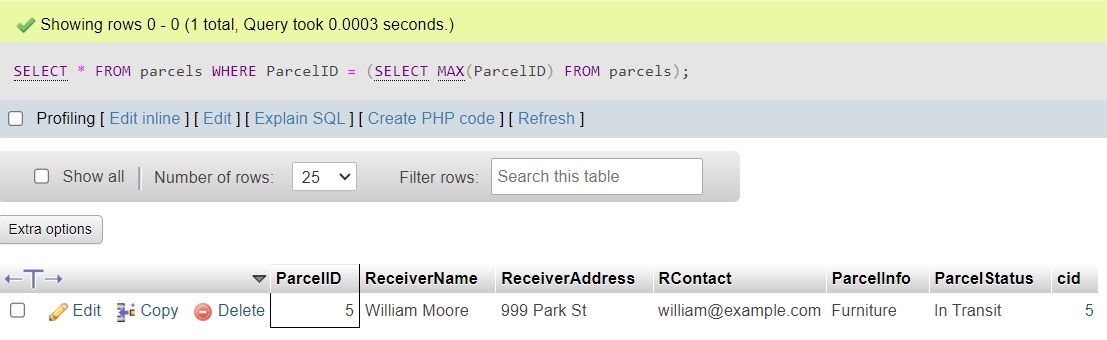


--Find the parcel with the highest ParcelID and display its details:

SELECT \*

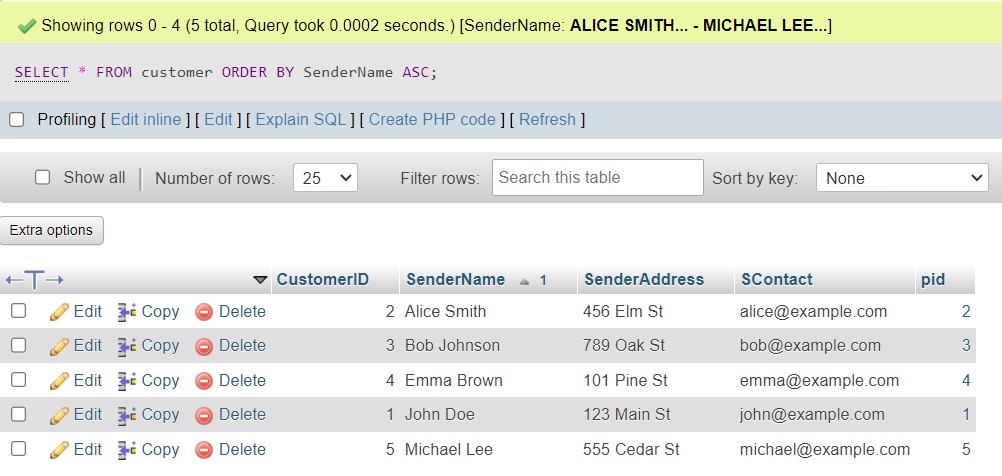
FROM parcels

WHERE Parcel ID = (SELECT MAX(ParcelID) FROM parcels);



--List customers sorted by their SenderNames in ascending order:

--SELECT \* FROM customer ORDER BY SenderName ASC;



--List the customers who have sent more parcels than the average number of parcels sent:

SELECT c.SenderName, COUNT(p.ParcelID) AS SentParcelsCount

FROM customer c

LEFT JOIN parcels p ON c.CustomerID = p.cid

GROUP BY c.SenderName

HAVING COUNT(p.ParcelID) > (

SELECT AVG(cnt)

FROM (

SELECT COUNT(p2.ParcelID) AS cnt

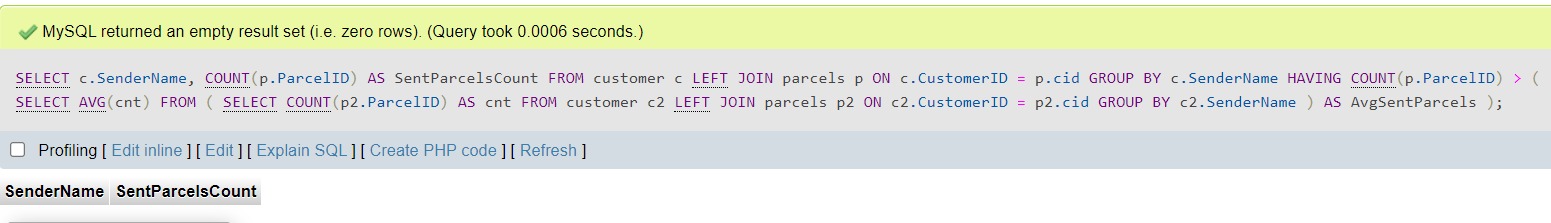
FROM customer c2

LEFT JOIN parcels p2 ON c2.CustomerID = p2.cid

GROUP BY c2.SenderName

) AS AvgSentParcels

);



--Find the customers with the most sent parcels and display the count:

SELECT SenderName, COUNT(\*) AS SentParcelsCount

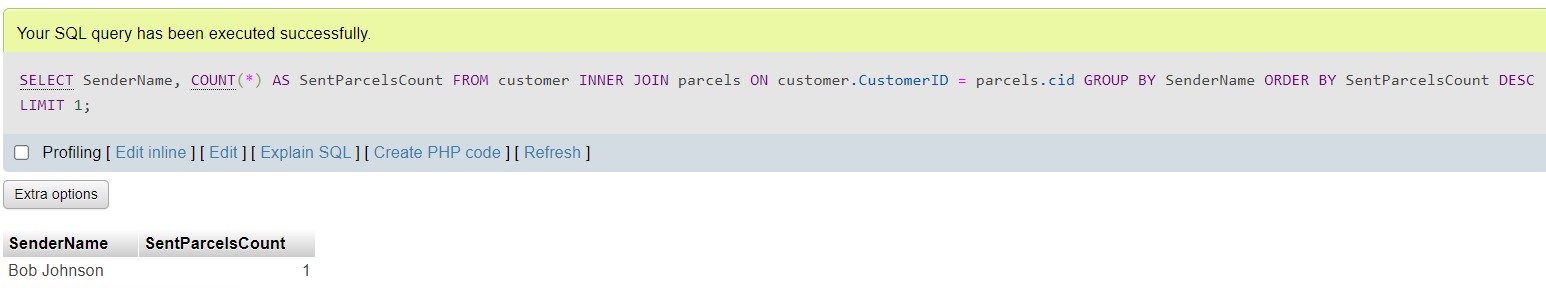
FROM customer

INNER JOIN parcels ON customer.CustomerID = parcels.CustomerID

GROUP BY SenderName

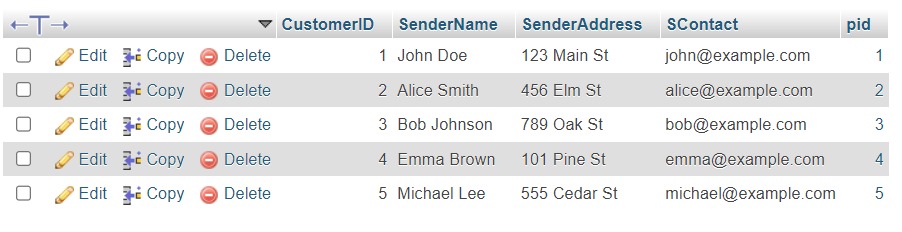
ORDER BY SentParcelsCount DESC

LIMIT 1;

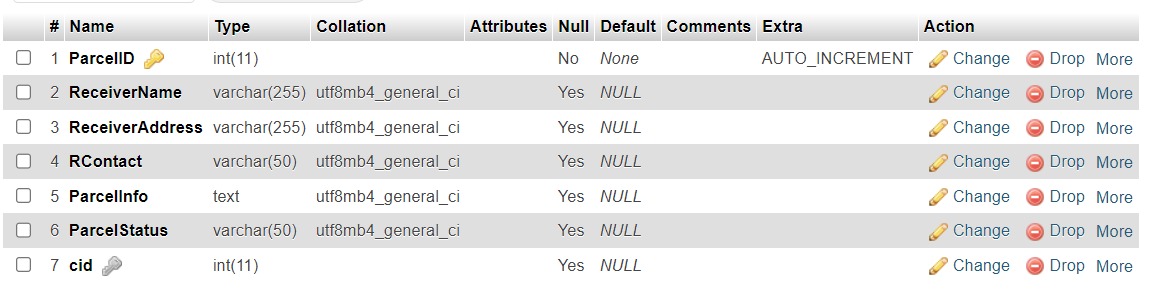


POPULATED TABLES:

CUSTOMER:

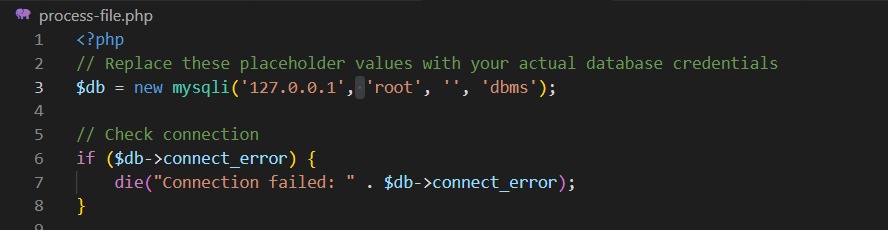


Parcel:



DB CONNECTIVITY

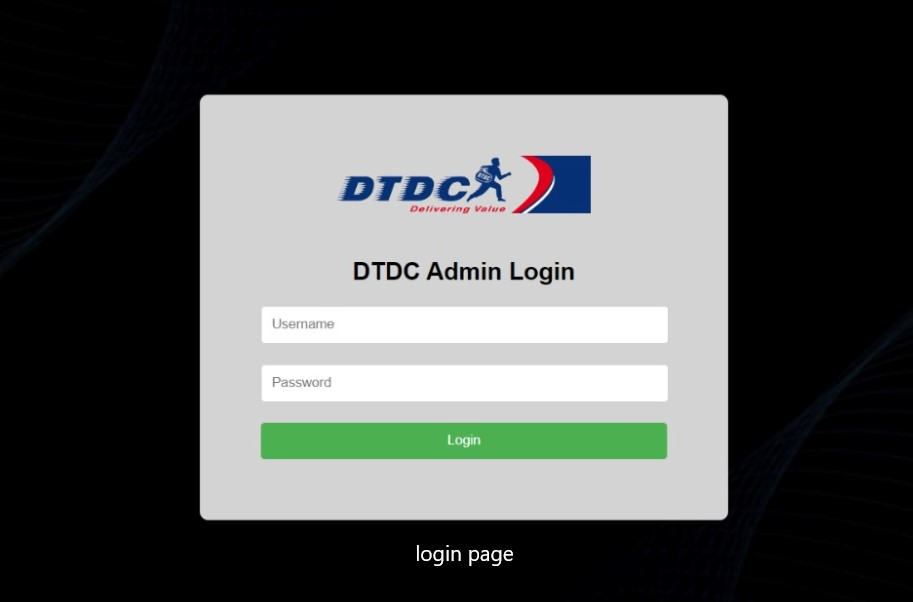
The database has been connected to the frontend user interface using PHP as the scripting language. The code for establishing the connection is as follows:



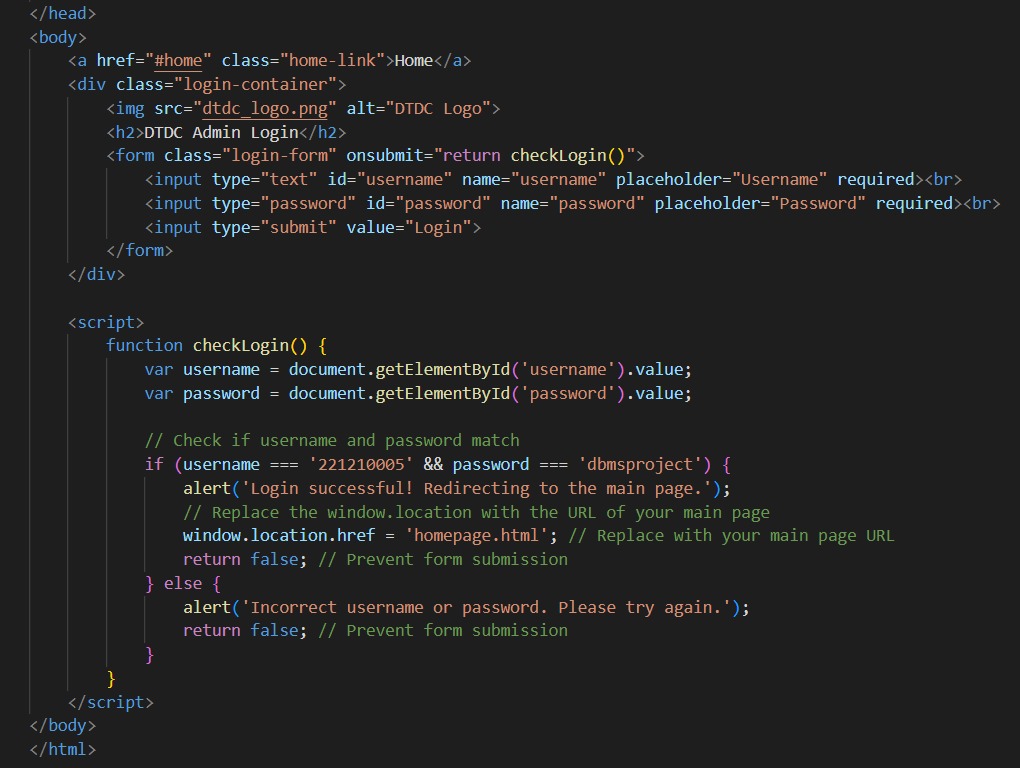
The above code serves to establish the connection between the backend (database) and frontend (user interface). The PHP codes corresponding to each functionality being implemented are mentioned along with the respective functionality in the following pages.

Functionalities

1.The Login page ,it is accessible only for the Admin

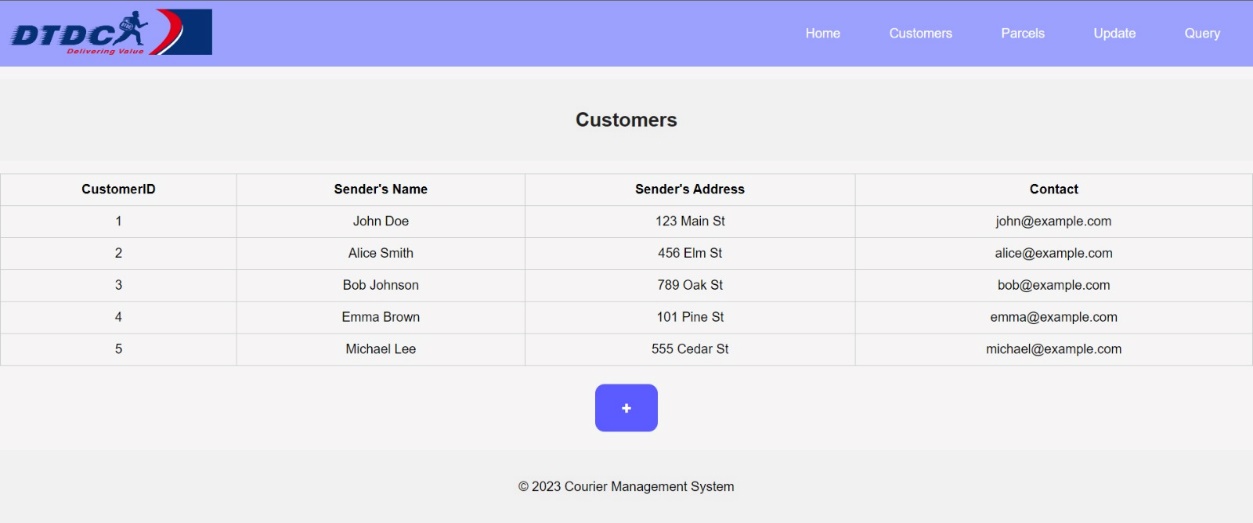


Login page code

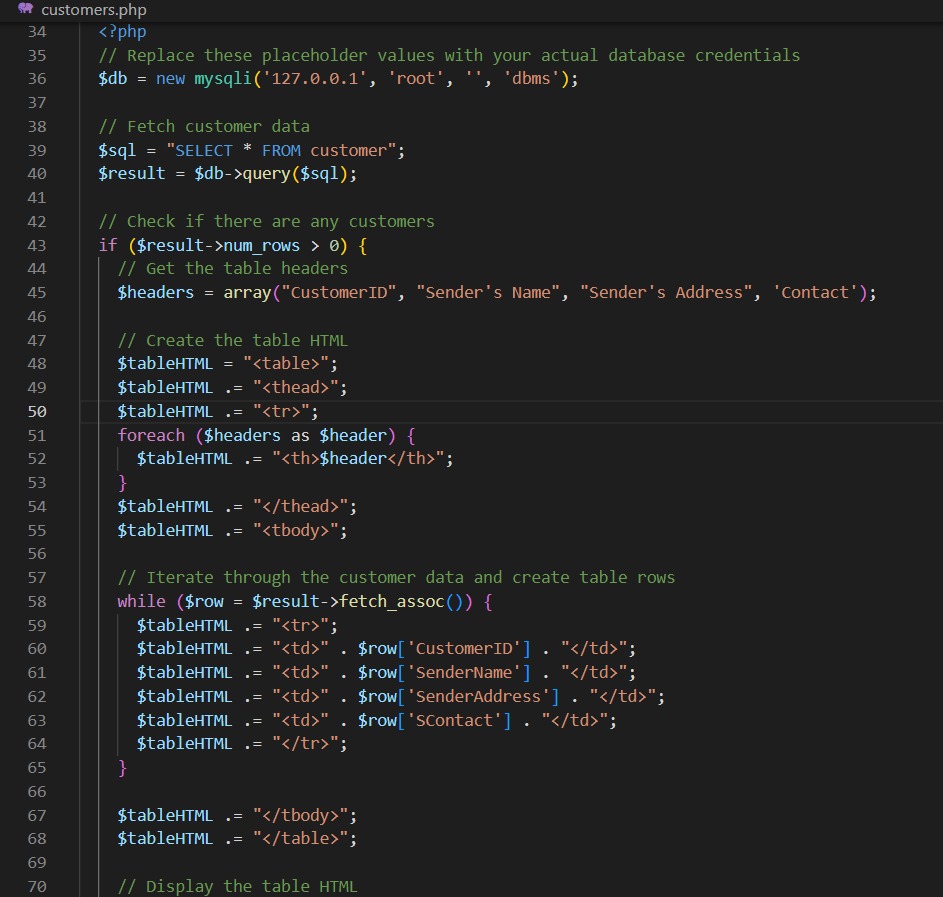


2.There are 5 things when you login,Home,Customers,Parcels,Update,Query.

When you click on the customers,



Code part 1

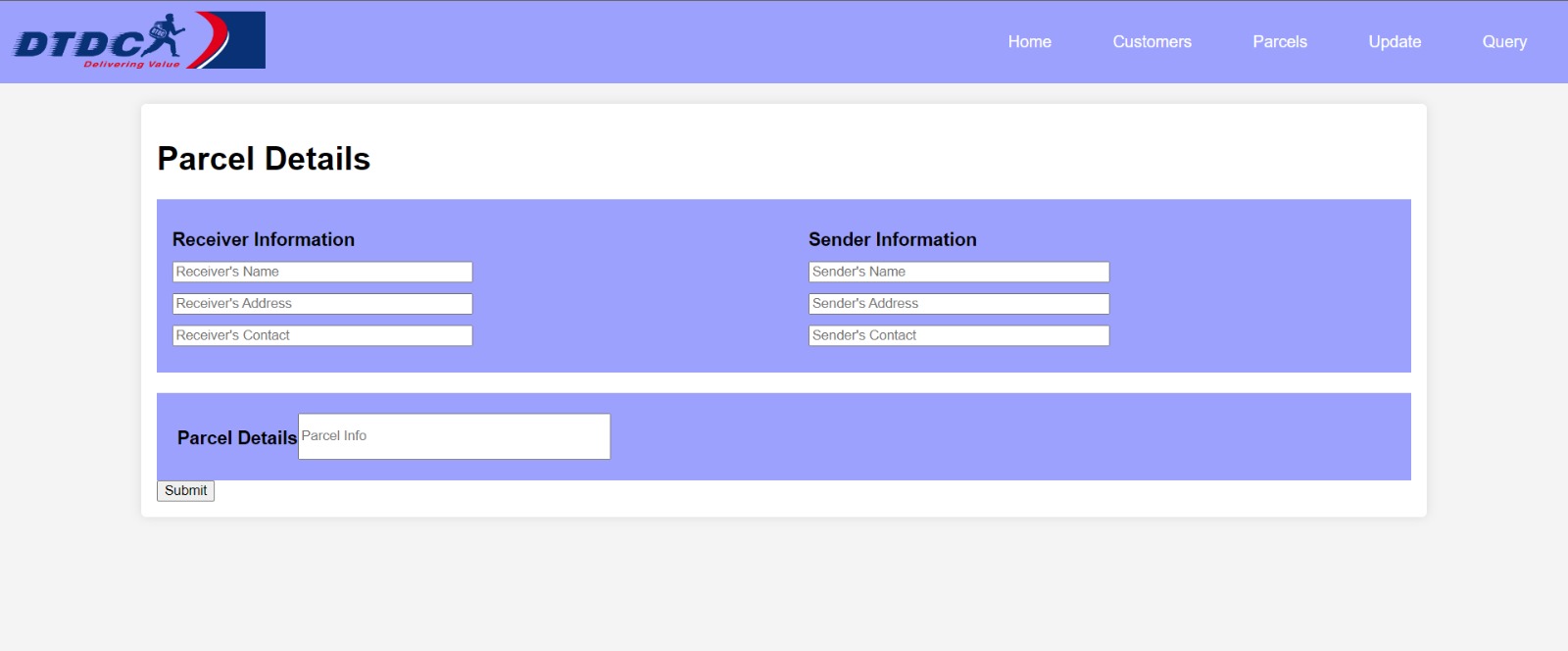


Code part 2

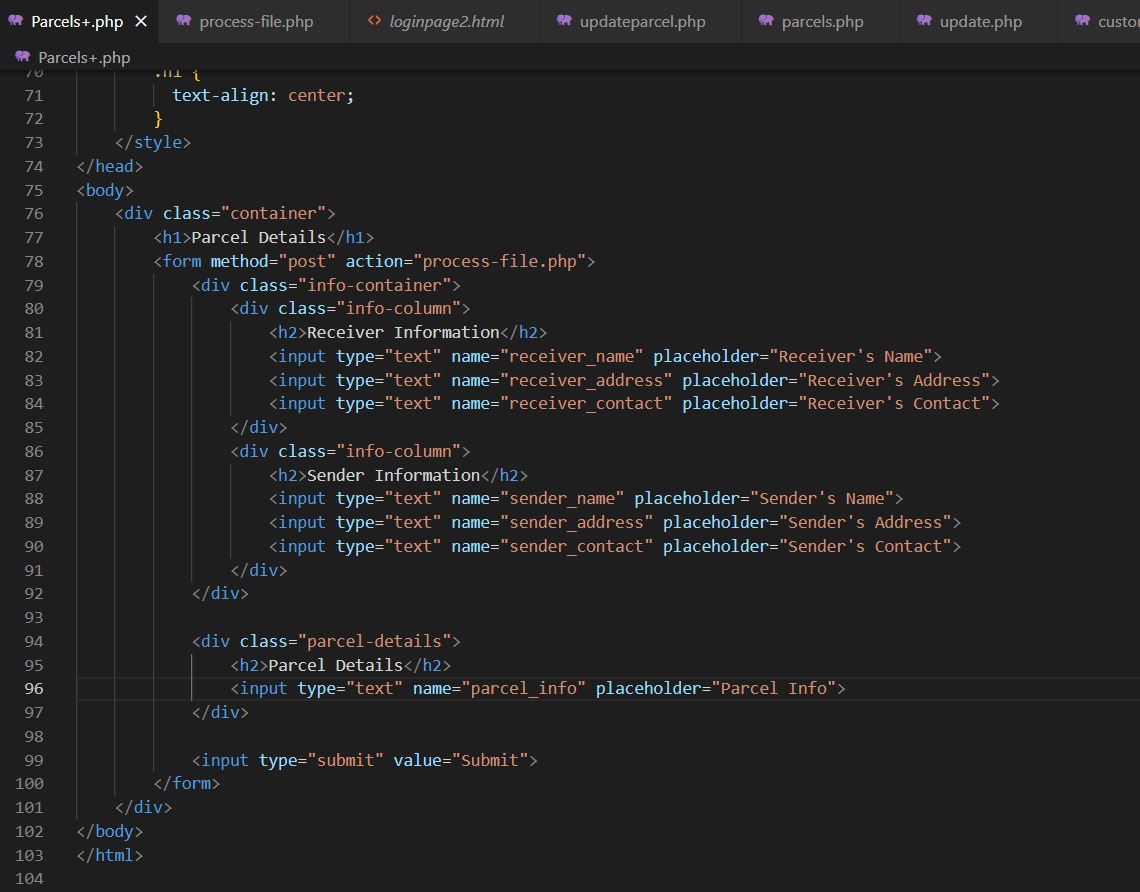


The customers table have 5 columns with customer id, sender’s name, sender’s address, contact details. The customer id gets generated automatically here.

3. the + button in customers page takes us here



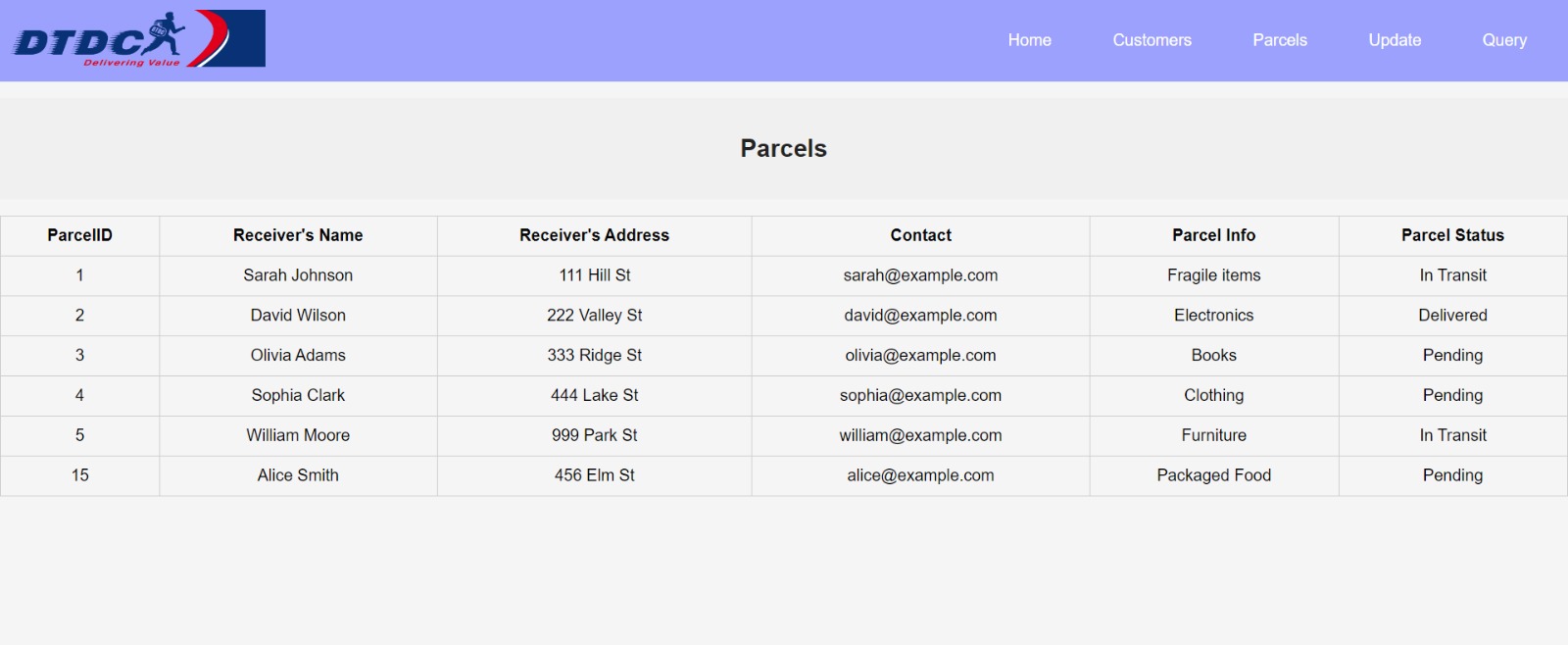
Html code for this:



4.this is the connection and input codes of the previous image

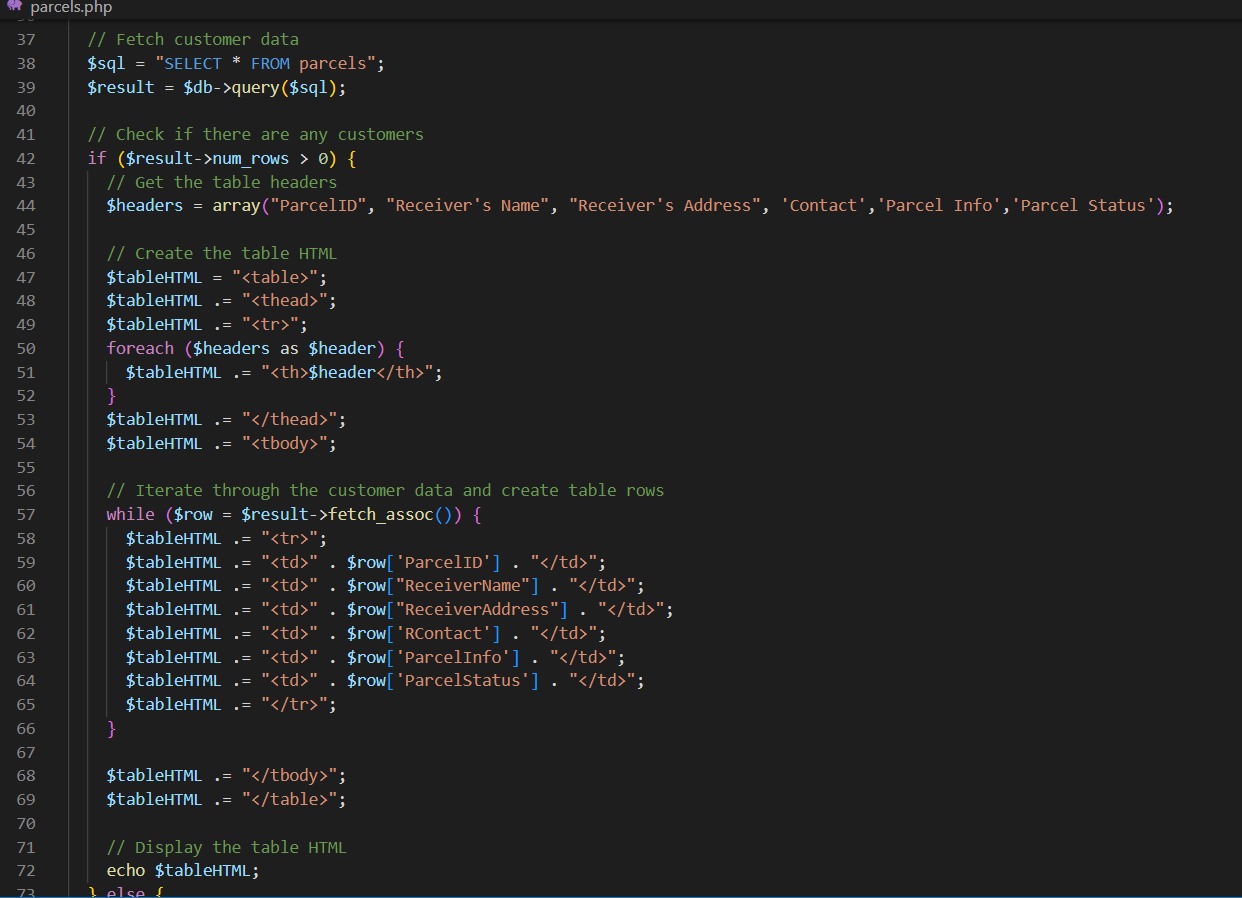


5.When you Click On the parcels above,



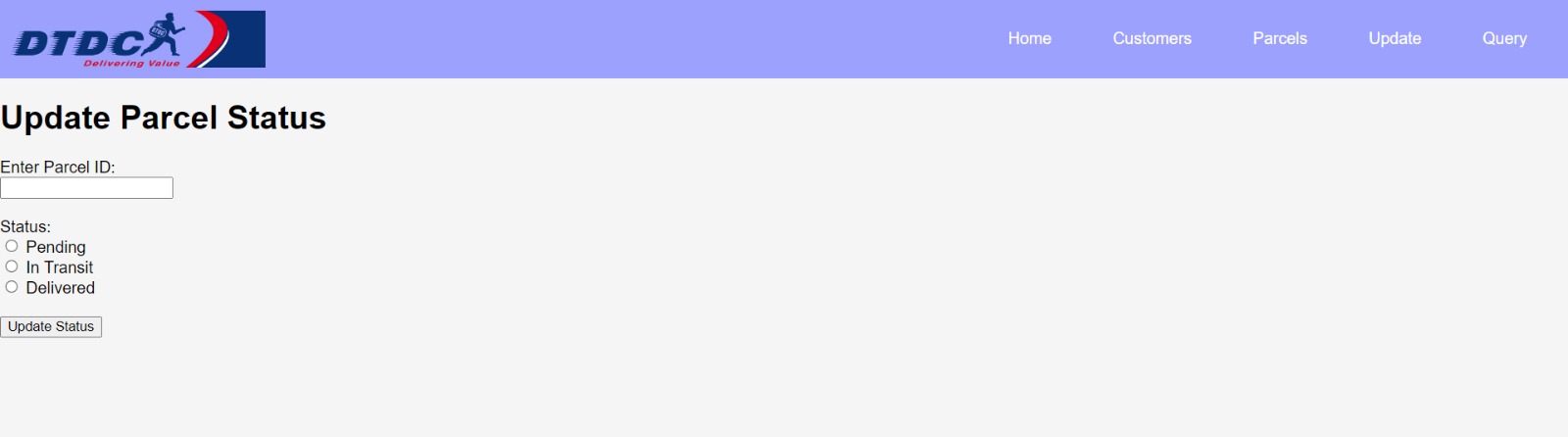
All the details regarding the parcels and receiver’s gets displayed.

Code for it:



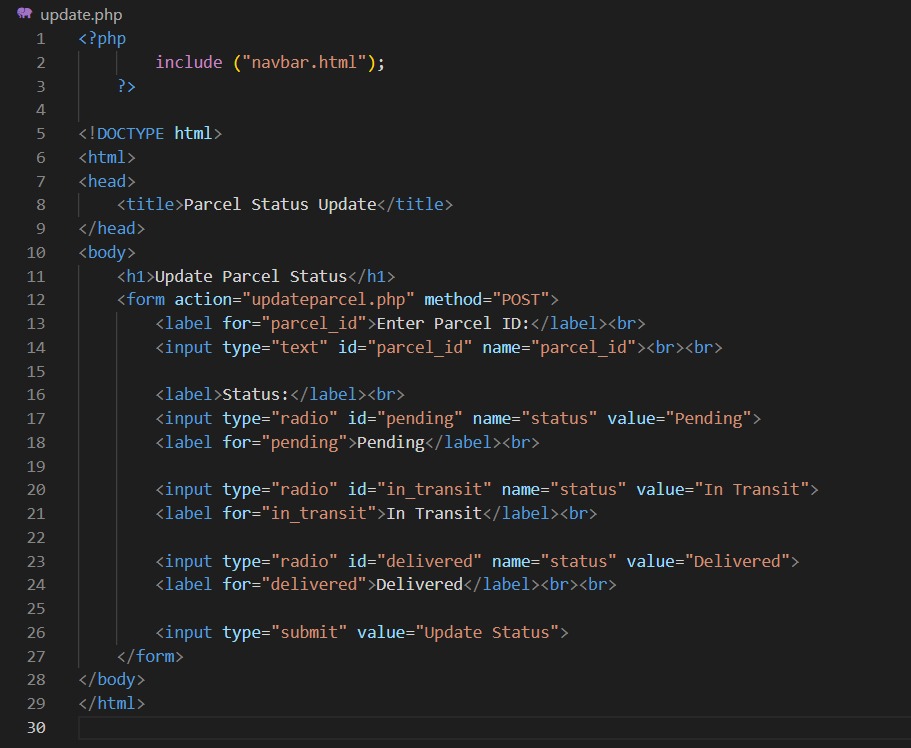


7.When you Click on the update button

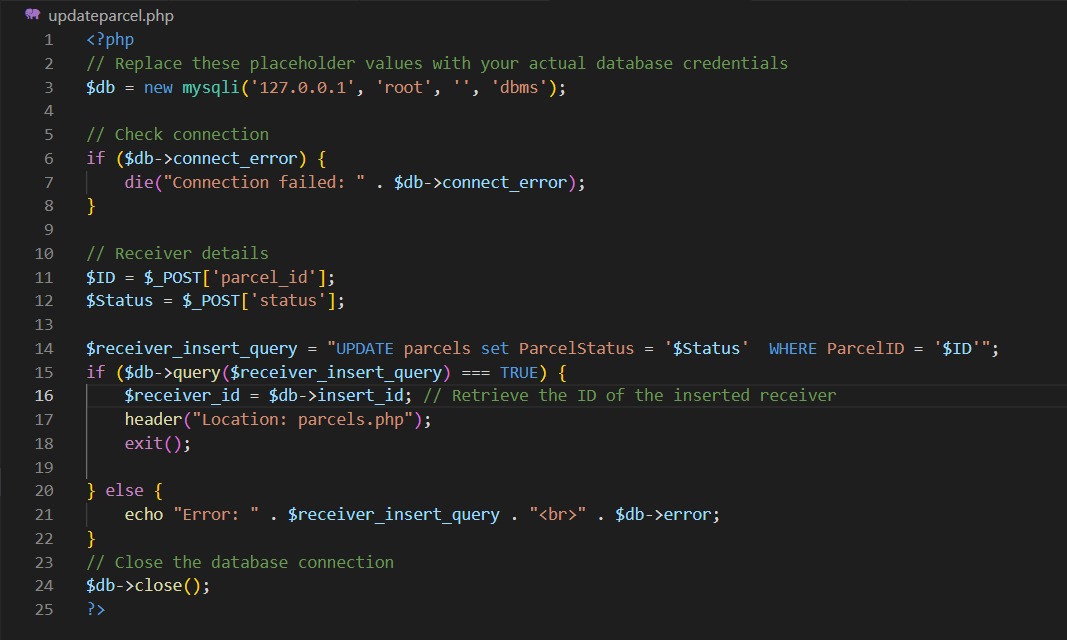


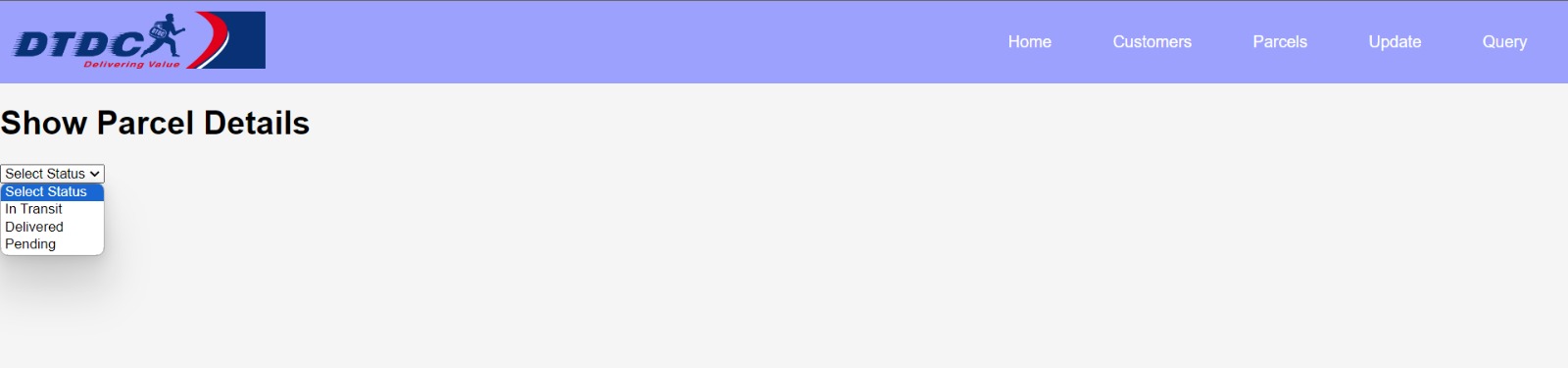
Here you can update the status of the parcel if it is in pending or transit or delivered.

Code:



Update backend code:





8.When you click on the query button:





BIBLIOGRAPHY

* ¬ Fundamentals of DATABASE SYSTEMS -Elmasri, Navathe
* Database Management Systems -Ramakrishnan, Gehrke
* [WWW.YOUTUBE.COM](http://WWW.YOUTUBE.COM)
* [WWW.CHATGPT.AI](http://WWW.CHATGPT.AI)
* DOCS.PHYMYADMIN.NET