Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 5- Issue 1, Paper 13 January 2022

Online Food Ordering System

Sruthi R, *sruthiraghavan2002@gmail.com* .

***Abstract*— This project proposes the Online Food Ordering System. The case study aims to design and develop a database maintaining the records of users.**

Our website provides management of users and their orders along with the payment details and the delivery partners assigned bringing in an ease in the entire functioning of online food ordering systems.

By keeping efficient tracks of who was assigned to deliver which order and what was the order amount along with customer details it first helps keep a track of service and sale, and other, helps to manage complains, refunds and salaries of the deliver agents.

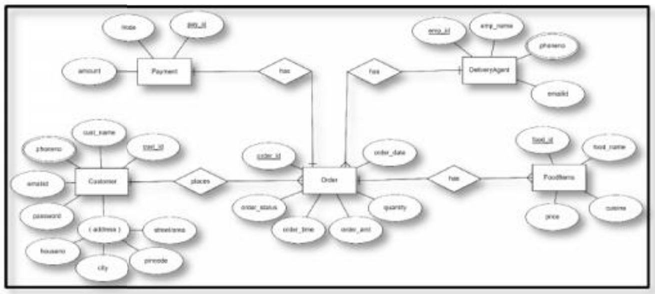
INTRODUCTION

Online food ordering has become a very fast-growing business over the period of time, keeping in mind the lockdown and contact-less activities, people prefer to order food at their places rather than going out and getting in contact with foreign surfaces[1].

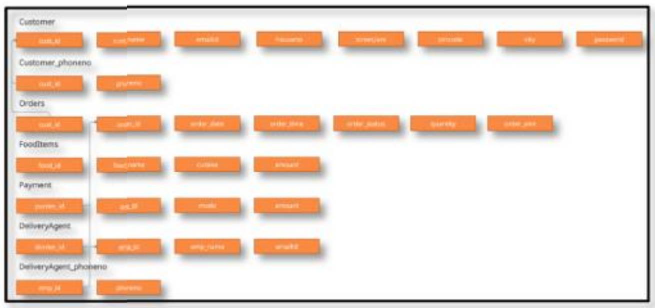
Owing to this, we planed a project to work with the daily records of the orders placed with the customer details along with the payment information connecting it to the deliver partners and their relevant information.

Expanding this, a user-friendly website connected to this system can make it a full-fledged food ordering system with a systematic database at the backend recording all changes and updates..

I. ER DIAGRAM



II. ER TO RELATIONAL MAPPING



III. IMPLEMENTATION

A. *HARDWARE SPECIFICATIONS*

1. Operating System: Windows 10 (64-bit)
2. RAM: 8GB
3. Processor: Intel Core i5 (10th Gen)

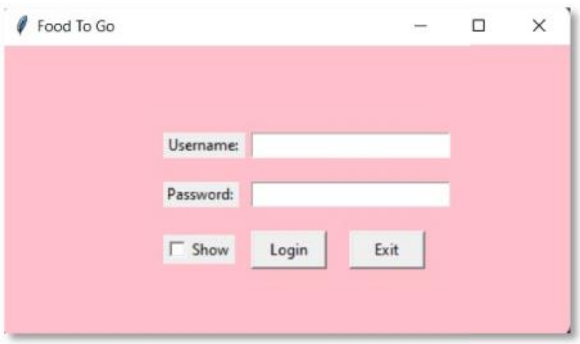
B. *SOFTWARE SPECIFICATIONS*

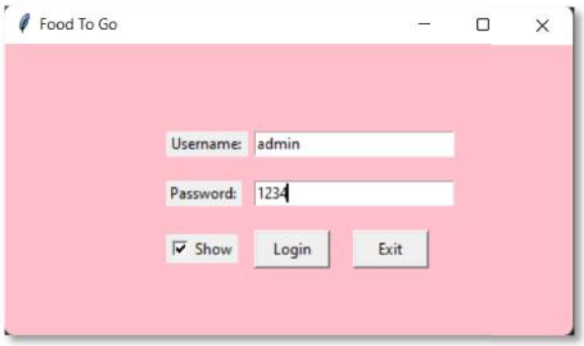
1. Front-end: Python with tkinter module
2. Back-end: SQLite3, MySQL

IV. EXPERIMENTAL RESULTS DISCUSSION

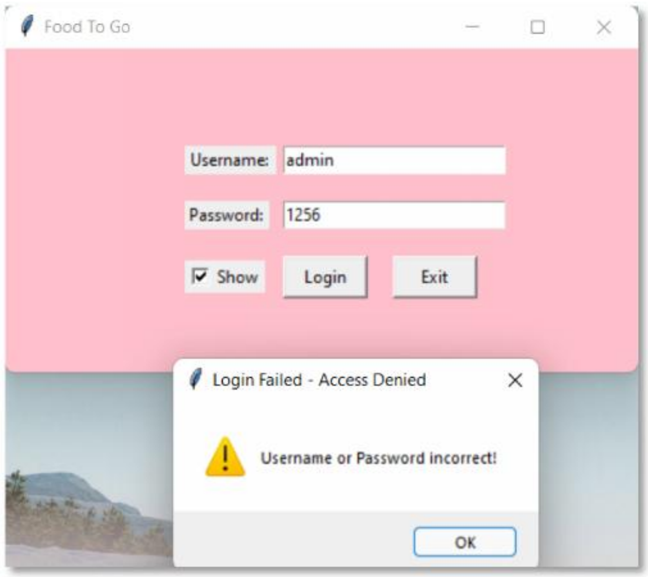
This project includes an admin page connected to a database which keeps records of all the online orders along with their corresponding customers, deliver agents and payment information. It provides an option to add, delete display records.

A. *LOGIN PAGE*





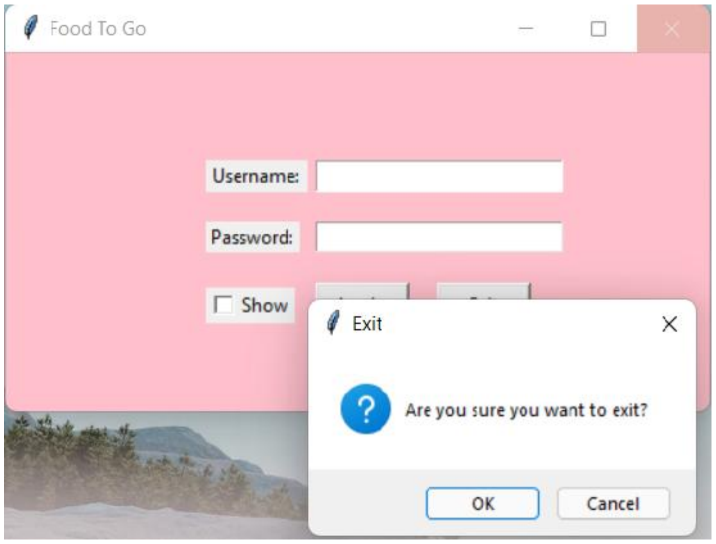
This is the login page which asks for the username and password.



Username or Password incorrect!

On giving wrong username or password, it prompts with an error message window

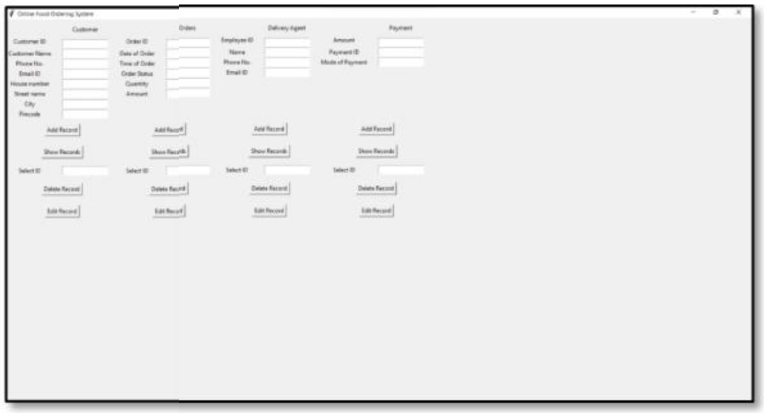
Volume 5- Issue 1, Paper 13 January 2022



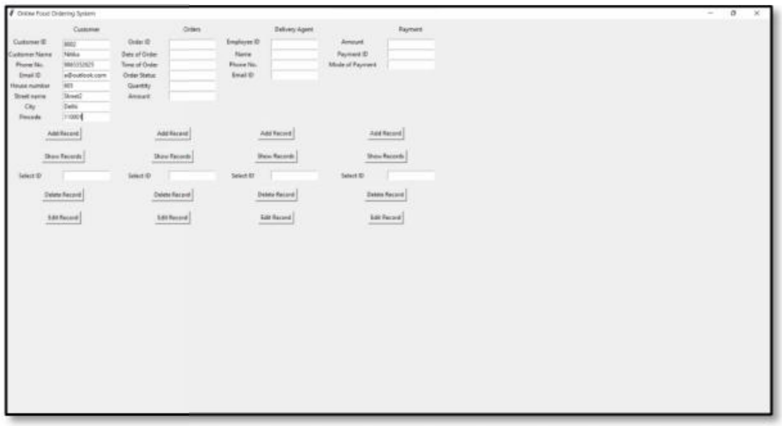
Are you sure you want to exit?

On attempting to close the window it prompts to confirm you exit action.

B. *MAIN PAGE*



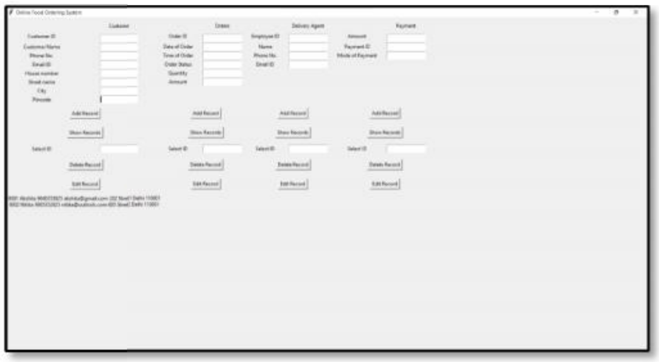
This is the first window that appears on attempting a successful login. We can add, delete, see records.



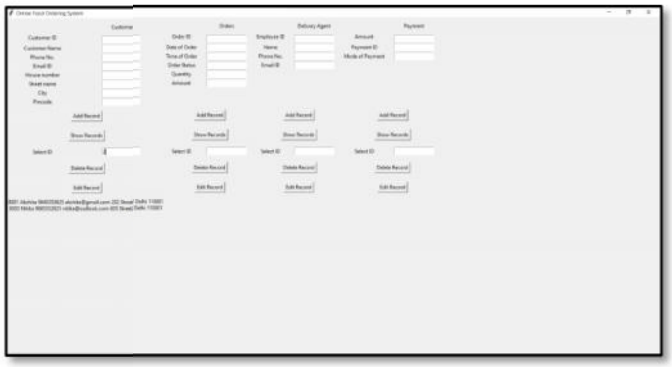
Customer ID 8002
Customer Name Nikita
Phone No. 9861152621
Email ID xx@outlook.com
House number 603
Street name Street2
City Delhi
Pincode 110001

This adds a record in the customer table that can be verified using Show Records button.

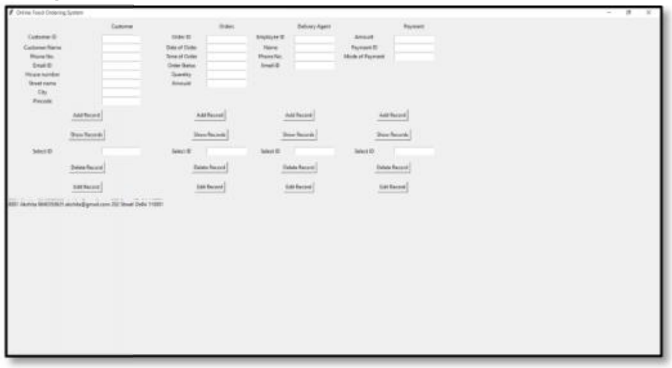
Volume 5- Issue 1, Paper 13 January 2022



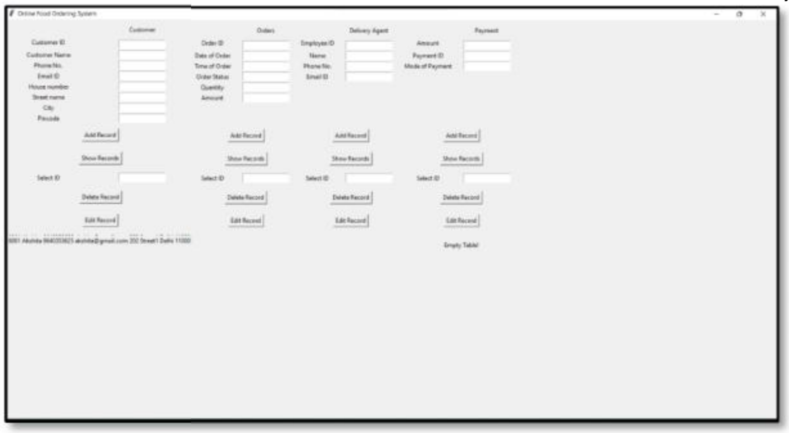
In order to delete a record, I shall enter the 'Record ID' in the Select ID text box. Record ID is like serial number of the record, so to delete the second record, I shall enter 2 in the text box and the second entry gets deleted.



After clicking on Delete Record, I again click on Show Record and,



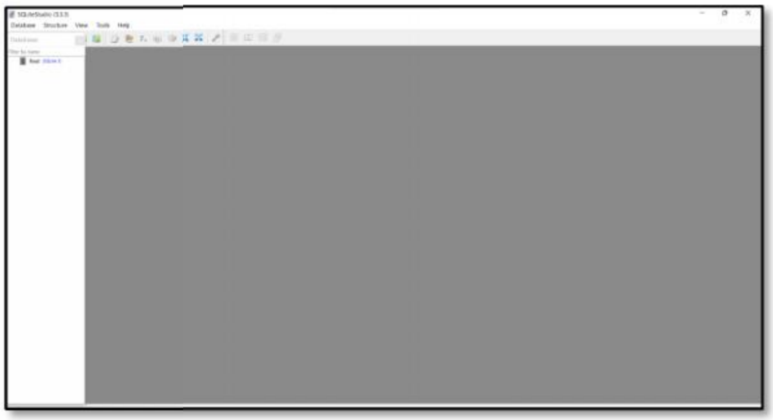
For a table, that doesn't have any entry, we shall get an 'Empty table!' prompt on clicking Show Record



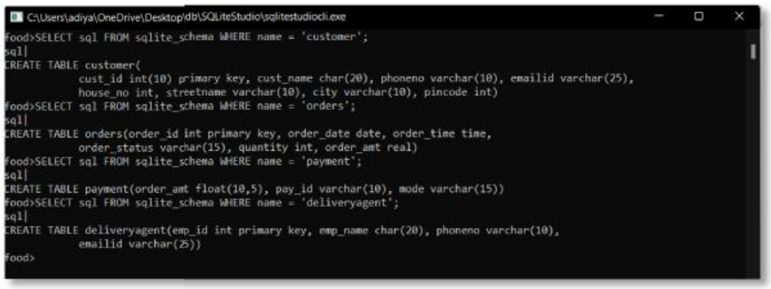
```
0001 Akshita 9840253625 akshita@gmail.com 202 Street1 Delhi 11000
```

V. APPENDIX I

The database



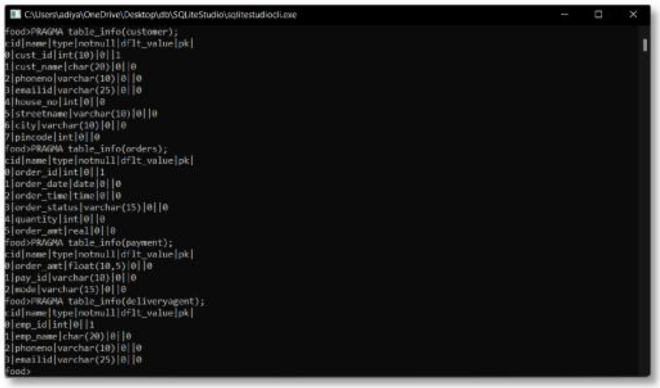
The table create queries,



```
C:\Users\adiya\OneDrive\Desktop\db\SQLiteStudio\sqlitestudiocli.exe
food>SELECT sql FROM sqlite_schema WHERE name = 'customer';
sql|
CREATE TABLE customer(
          cust_id int(10) primary key, cust_name char(20), phoneno varchar(10), emailid varchar(25),
          house_no int, streetname varchar(10), city varchar(10), pincode int)
food>SELECT sql FROM sqlite_schema WHERE name = 'orders';
sql|
CREATE TABLE orders(order_id int primary key, order_date date, order_time time,
          order_status varchar(15), quantity int, order_amt real)
food>SELECT sql FROM sqlite_schema WHERE name = 'payment';
sql|
CREATE TABLE payment(order_amt float(10,5), pay_id varchar(10), mode varchar(15))
food>SELECT sql FROM sqlite_schema WHERE name = 'deliveryagent';
sql|
CREATE TABLE deliveryagent(emp_id int primary key, emp_name char(20), phoneno varchar(10),
          emailid varchar(25))
food>
```

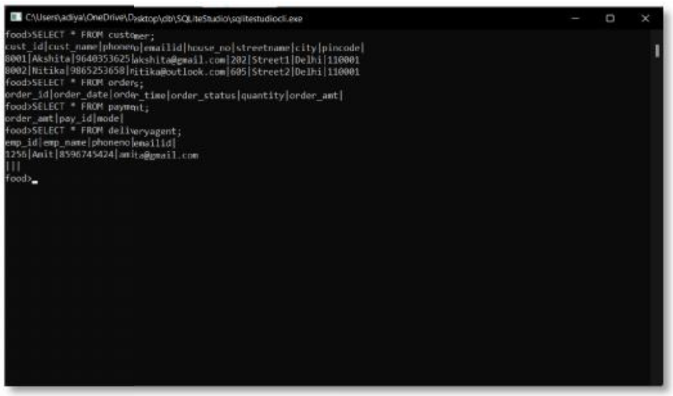
The table structures,

Volume 5- Issue 1, Paper 13 January 2022



```
C:\Users\adiya\OneDrive\Desktop\db\SQLiteStudio\sqlitestudiocli.exe
food>PRAGMA table_info(customer);
cid|name|type|notnull|dflt_value|pk|
0|cust_id|int(10)|0||1
1|cust_name|char(20)|0||0
2|phoneno|varchar(10)|0||0
3|emailid|varchar(25)|0||0
4|house_no|int|0||0
5|streetname|varchar(10)|0||0
6|city|varchar(10)|0||0
7|pincode|int|0||0
food>PRAGMA table_info(orders);
cid|name|type|notnull|dflt_value|pk|
0|order_id|int|0||1
1|order_date|date|0||0
2|order_time|time|0||0
3|order_status|varchar(15)|0||0
4|quantity|int|0||0
5|order_amt|real|0||0
food>PRAGMA table_info(payment);
cid|name|type|notnull|dflt_value|pk|
0|order_amt|float(10,5)|0||0
1|pay_id|varchar(10)|0||0
2|mode|varchar(15)|0||0
food>PRAGMA table_info(deliveryagent);
cid|name|type|notnull|dflt_value|pk|
0|emp_id|int|0||1
1|emp_name|char(20)|0||0
2|phoneno|varchar(10)|0||0
3|emailid|varchar(25)|0||0
food>
```

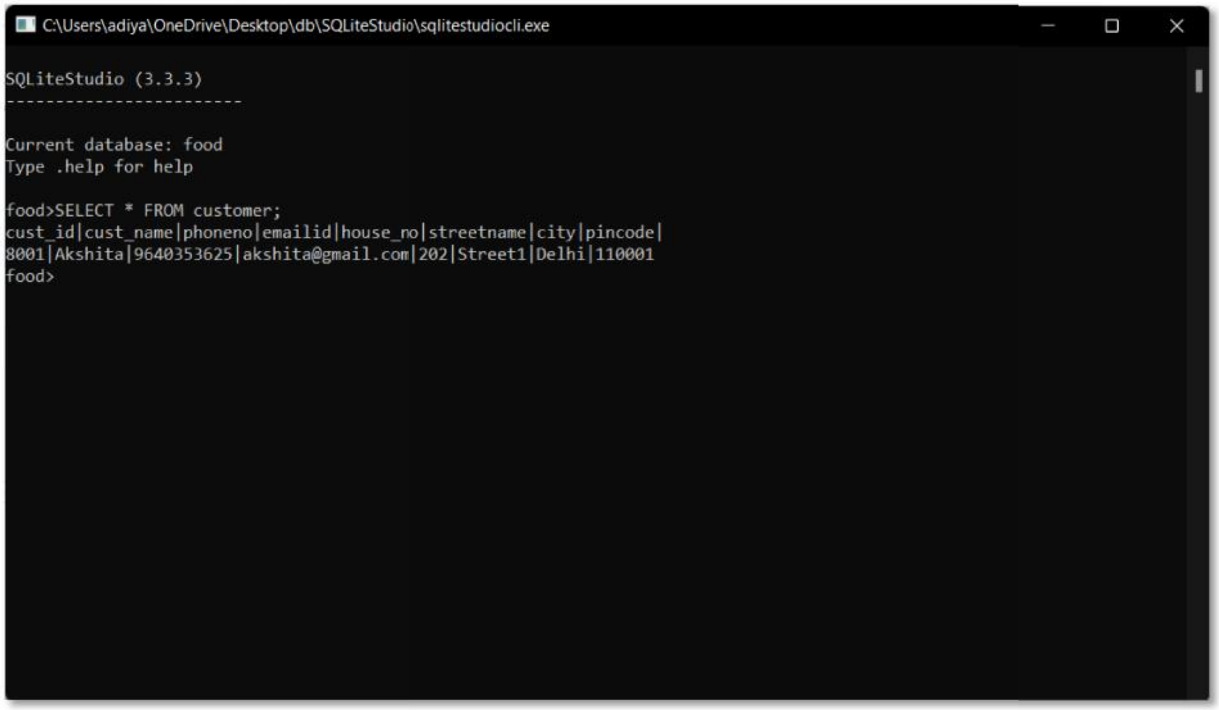
The SELECT * FROM queries for each table,



```
C:\Users\adiya\OneDrive\Desktop\db\SQLiteStudio\sqlitestudiocli.exe
food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8001|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
8002|Nitika|9865253658|nitika@outlook.com|605|Street2|Delhi|110001
food>SELECT * FROM orders;
order_id|order_date|order_time|order_status|quantity|order_amt|
food>SELECT * FROM payment;
order_amt|pay_id|mode|
food>SELECT * FROM deliveryagent;
emp_id|emp_name|phoneno|emailid|
1256|Amit|8596745424|amita@gmail.com
|||
food>
```

VI. APPENDIX II

As I add a record in the 'customer' table and show records on the front end, it simultaneously gets added in the database Previously, database had

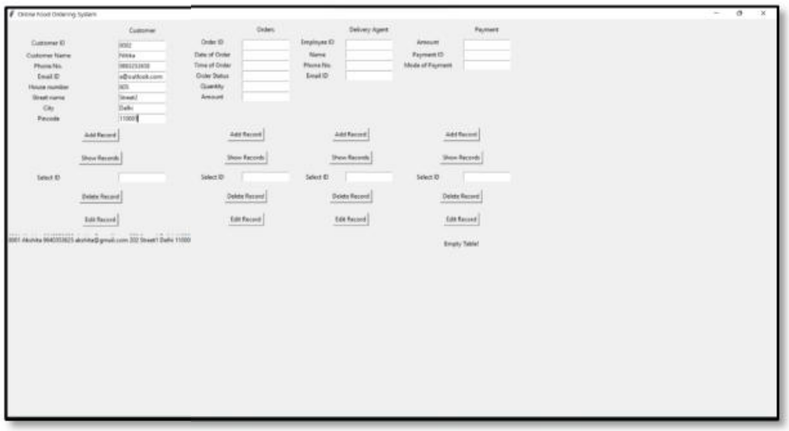


```
C:\Users\adiya\OneDrive\Desktop\db\SQLiteStudio\sqlitestudiocli.exe

SQLiteStudio (3.3.3)
------------------------

Current database: food
Type .help for help

food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8001|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
food>
```

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

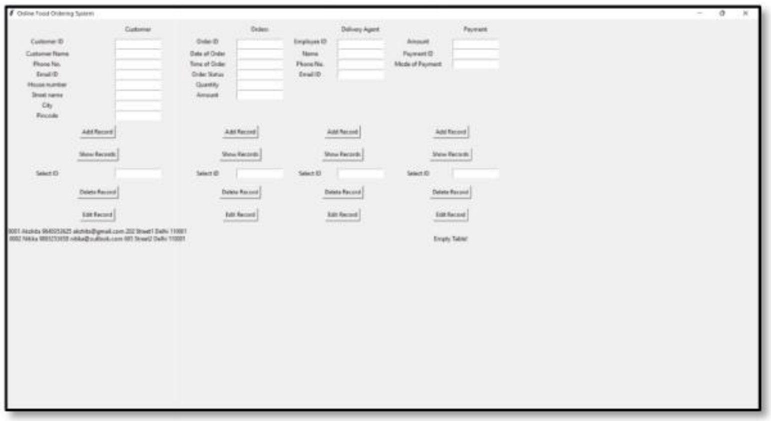
Volume 5- Issue 1, Paper 13 January 2022

Adding



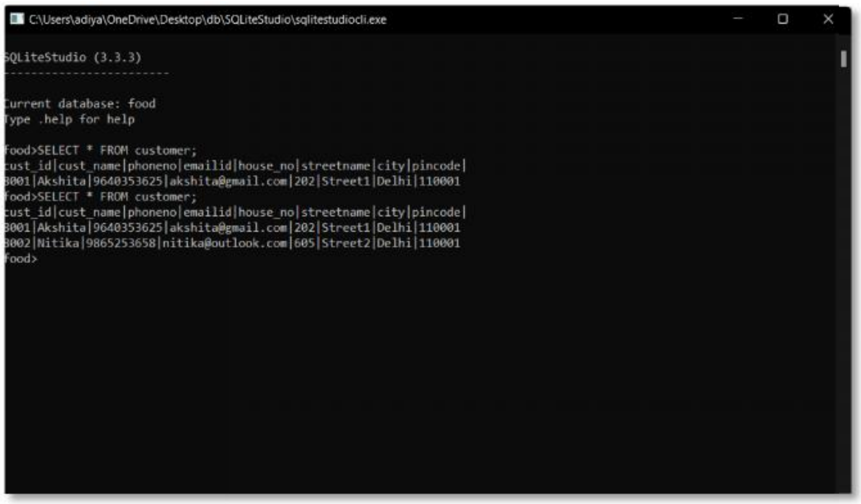
```
Customer ID       8002
Customer Name     Nitika
Phone No.         9865253658
Email ID          ni@outlook.com
House number      605
Street name       Street2
City              Delhi
Pincode           110001

Order ID  Date of Order  Time of Order  Order Status  Quantity  Amount
Employee ID  Name  Phone No.  Email ID
Amount  Payment ID  Mode of Payment
Select ID

8001 Akshita 9640353625 akshita@gmail.com 202 Street1 Delhi 11000

Empty Table!
```

Showing



```
Customer ID  Customer Name  Phone No.  Email ID  House number  Street name  City  Pincode
Order ID  Date of Order  Time of Order  Order Status  Quantity  Amount
Employee ID  Name  Phone No.  Email ID
Amount  Payment ID  Mode of Payment
Select ID

8001 Akshita 9640353625 akshita@gmail.com 202 Street1 Delhi 110001
8002 Nitika 9865253658 nitika@outlook.com 605 Street2 Delhi 110001

Empty Table!
```

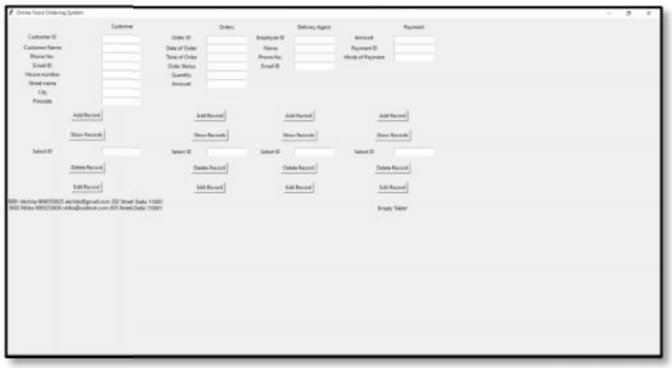
Then the database,



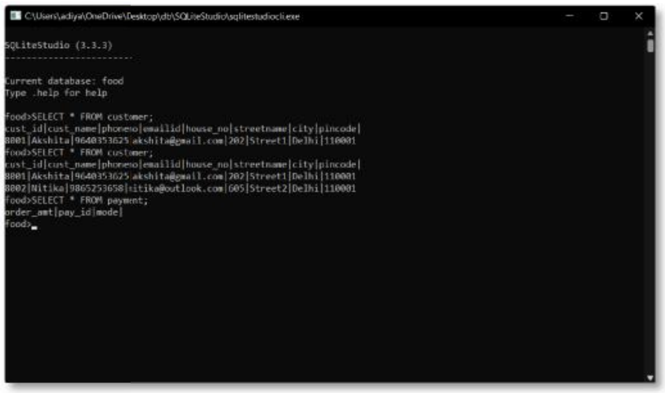
```
C:\Users\adiya\OneDrive\Desktop\db\SQLiteStudio\sqlitestudiocli.exe

SQLiteStudio (3.3.3)
------------------------

Current database: food
Type .help for help

food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8001|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8001|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
8002|Nitika|9865253658|nitika@outlook.com|605|Street2|Delhi|110001
food>
```

For an empty table,

Volume 5- Issue 1, Paper 13 January 2022

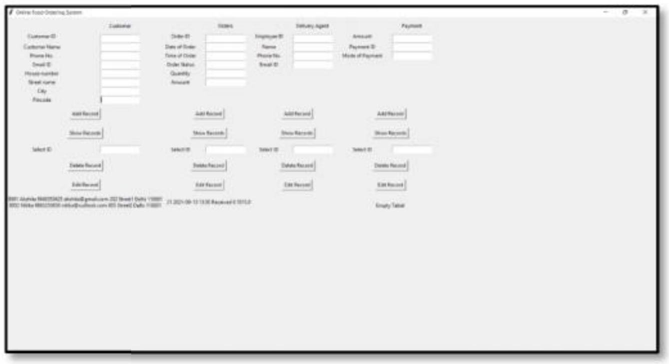


The database,

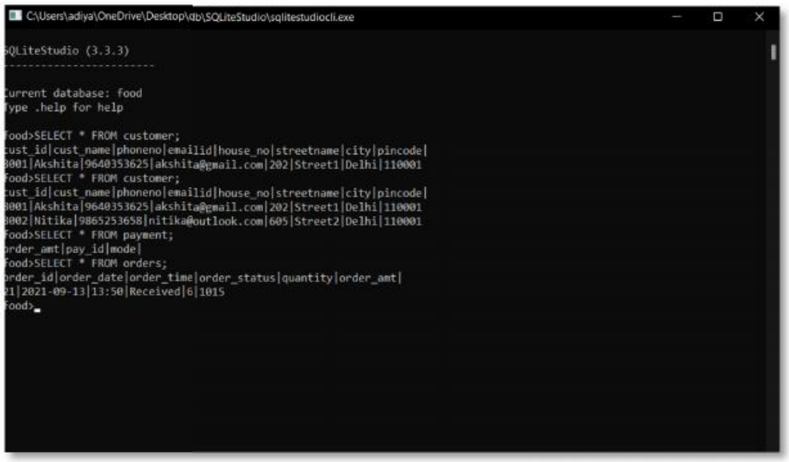


```
C:\Users\adiya\OneDrive\Desktop\db\SQLiteStudio\sqlitestudiocli.exe

SQLiteStudio (3.3.3)
------------------------

Current database: food
Type .help for help

food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8801|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8801|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
8802|Nitika|9865253658|nitika@outlook.com|605|Street2|Delhi|110001
food>SELECT * FROM payment;
order_amt|pay_id|mode|
food>
```

For the deletion of records, we first show records in the ‘order’ table, then we delete one record from the order table and see the corresponding change in the database.

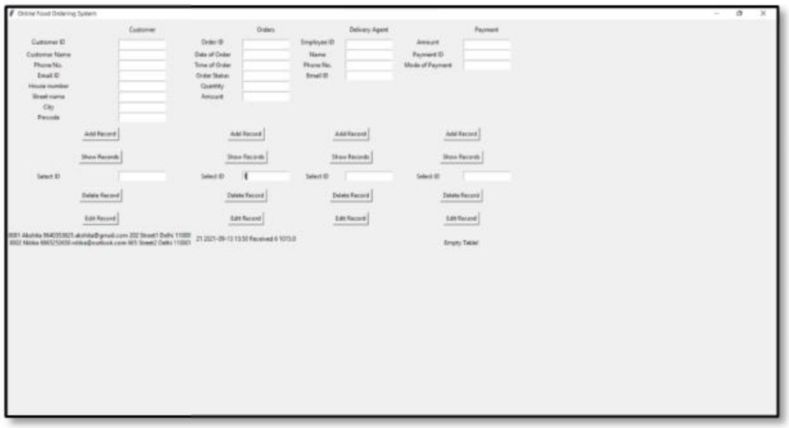


Volume 5- Issue 1, Paper 13 January 2022


C:\Users\adiya\OneDrive\Desktop\db\SQLiteStudio\sqlitestudiocli.exe

```
SQLiteStudio (3.3.3)
------------------------

Current database: food
Type .help for help

food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8001|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8001|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
8002|Nitika|9865253658|nitika@outlook.com|605|Street2|Delhi|110001
food>SELECT * FROM payment;
order_amt|pay_id|mode|
food>SELECT * FROM orders;
order_id|order_date|order_time|order_status|quantity|order_amt|
21|2021-09-13|13:50|Received|6|1015
food>_
```

We delete the first record, record ID = 1,

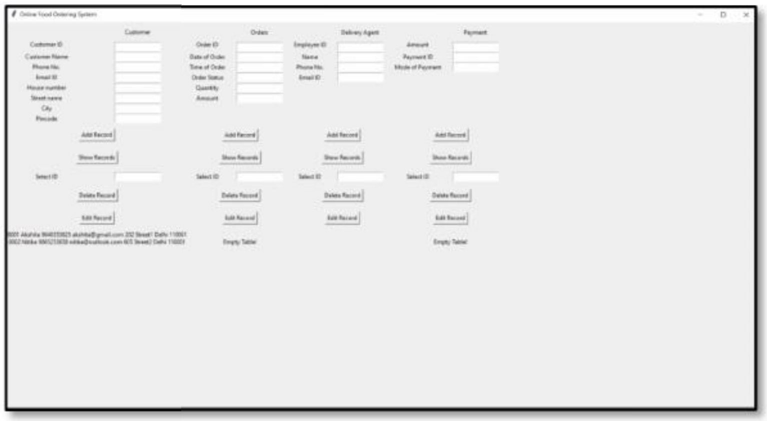


```
8001 Akshita 9640353625 akshita@gmail.com 202 Street1 Delhi 110001
8002 Nitika 9865253658 nitika@outlook.com 605 Street2 Delhi 110001
```

```
21 2021-09-13 13:50 Received 6 1015.0
```

```
Empty Table!
```

After deleting,



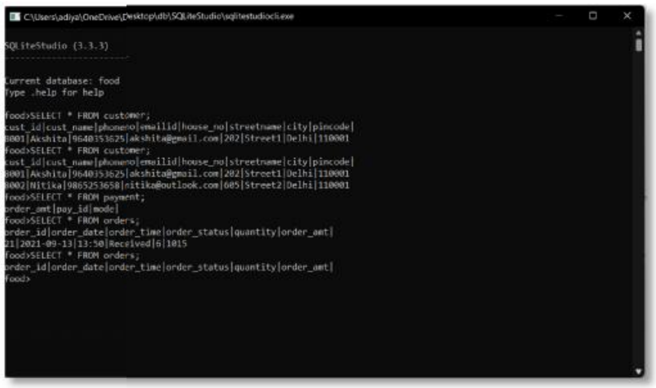
```
8001 Akshita 9640353625 akshita@gmail.com 202 Street1 Delhi 110001
8002 Nitika 9865253658 nitika@outlook.com 605 Street2 Delhi 110001
```

```
Empty Table!
```

```
Empty Table!
```

The database

Volume 5- Issue 1, Paper 13 January 2022



```
C:\Users\adiya\OneDrive\Desktop\db\SQLiteStudio\sqlitestudiocli.exe

SQLiteStudio (3.3.3)
------------------------

Current database: food
Type .help for help

food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8001|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
food>SELECT * FROM customer;
cust_id|cust_name|phoneno|emailid|house_no|streetname|city|pincode|
8001|Akshita|9640353625|akshita@gmail.com|202|Street1|Delhi|110001
8002|Nitika|9865253658|nitika@outlook.com|605|Street2|Delhi|110001
food>SELECT * FROM payment;
order_amt|pay_id|mode|
food>SELECT * FROM orders;
order_id|order_date|order_time|order_status|quantity|order_amt|
21|2021-09-13|13:50|Received|6|1015
food>SELECT * FROM orders;
order_id|order_date|order_time|order_status|quantity|order_amt|
food>
```

VII. CONCLUSION AND FUTURE WORK

Owing to the increasing demands on such food ordering websites and applications, a management system to keep a track in a systematic manner can be called as for the need of the hour. With the knowledge we had and gaining more knowledge in the same stream, we designed this front end connected with the backend which we hope when connected to a full-fledged user-interface along with admin interface shall bear fruitful and efficient results on regular uses.

REFERENCES

[1]. Cristina-Edina Domokos, Barna Séra, Károly Simon
Lajos Kovács, Tas-Béla Szakács, “ Netfood: A Software System for Food Ordering and Delivery”, presented at the 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)