<<ONLINE SHOPPING MANAGEMENT SYSTEM>>

18CSC209J - Database Management System and Cloud Integration Services

Mini Project Report
Submitted by

Student Name : ADITI JAIN
[Reg. No.: RA2111028010011]
B.Tech. CSE - <<CLOUD COMPUTING>>

Student Name : KUMARI HARSHITA
[Reg. No.: RA2111028010060]
B.Tech. CSE - <<CLOUD COMPUTING>>

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

DEPARTMENT OF NETWORKING AND COMMUNICATION
SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section 3 of UGC Act, 1956)
S.R.M. NAGAR, KATTANKULATHUR – 603 203
KANCHEEPURAM DISTRICT

MAY 2023

TABLE OF CONTENTS

BONAFIDE

This is to certify that **18CSC209J - DATABASE MANAGEMENT SYSTEM AND CLOUD INTEGRATION SERVICES LABORATORY Mini Project report** titled " **ONLINE SHOPPING MANAGEMENT SYSTEM** "is the bonafide work of

Student Name : ADITI JAIN (Reg no :RA2111028010011),

Student Name : KUMARI HARSHITA (Reg no : RA2111028010060)

who undertook the task of completing the project within the allotted time.

SIGNATURE	**SIGNATURE**
Dr.	**Dr. Annapurani Panaiyappan K**
Assistant Professor Department of Networking and Communication SRM Institute of Science and Technology	Professor and Head Department of Networking and Communication SRM Institute of Science and Technology

ABSTRACT

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform.

This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favourite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as FlipKart or ebay. Since the application is available in the Smartphone it is easily accessible and always available.

INTRODUCTION

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform. Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favourite shop.

1.1 PROJECT OBJECTIVE:

The objective of the project is to make an application in android platform to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view.

1.2 PROJECT OVERVIEW:

The central concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store. The information pertaining to the products are stores on an RDBMS at the server side (store). The Server process the customers and the items are shipped to the address submitted by them. The application was designed into two modules first is for the customers who wish to buy the articles. Second is for the storekeepers who maintains and updates the information pertaining to the articles and those of the customers.

PROJECT SCOPE:

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24*7 and a home delivery system which can make customers happy. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flipKart or ebay. Since the application is available in the Smartphone it is easily accessible and always available.

PROBLEM STATEMENT

the statement problem may include the following details:

Scope: What areas of the organization's operations need to be managed through the online management system? This may include inventory management, sales management customer relationship management, financial management, etc.

User requirements: Who will be using the system and what are their requirements? This may include employees, managers, customers, and suppliers. Each group may have different requirements and access levels.

Security: How can the system ensure the security and privacy of user data? This may include secure login systems, encryption, data backups, and access control.

Efficiency: How can the system ensure that operations are efficient and streamlined? This may include automated processes, integration with other systems, and real-time reporting.

Scalability: How can the system be designed to accommodate growth and expansion of the organization?

Cost: What is the budget for developing and maintaining the system? What are the cost-benefit trade-offs of different approaches to development and implementation?

.

MODULE DESCRIPTION

Home Page module : This is the first page of website. Home page redirects different pages. About us, Faq page and contact us pages are static pages in the website.

Customer Account: To create new customer account customer needs to register to the system. Existing users can login to the system. Customer can update his profile and he can change his password after the login.

Purchase cart: This will show added list in the cart.

Payment panel: This allows user to make payment. After the payment the system generates receipt.

Supplier account: Administrator can verify supplier account. After verification supplier or vendor can sell products through online.

Sell product: Selling product done by supplier. Suppliers can add product, view product set discount, expiry date, etc.

View Ordered products: Suppliers can view product order list done by customer.

Products List module: This view display all product lists. Customer can search and view products and they can filter products by category. Seller can view only his products. Admin can monitor all products and he has option to delete the product.

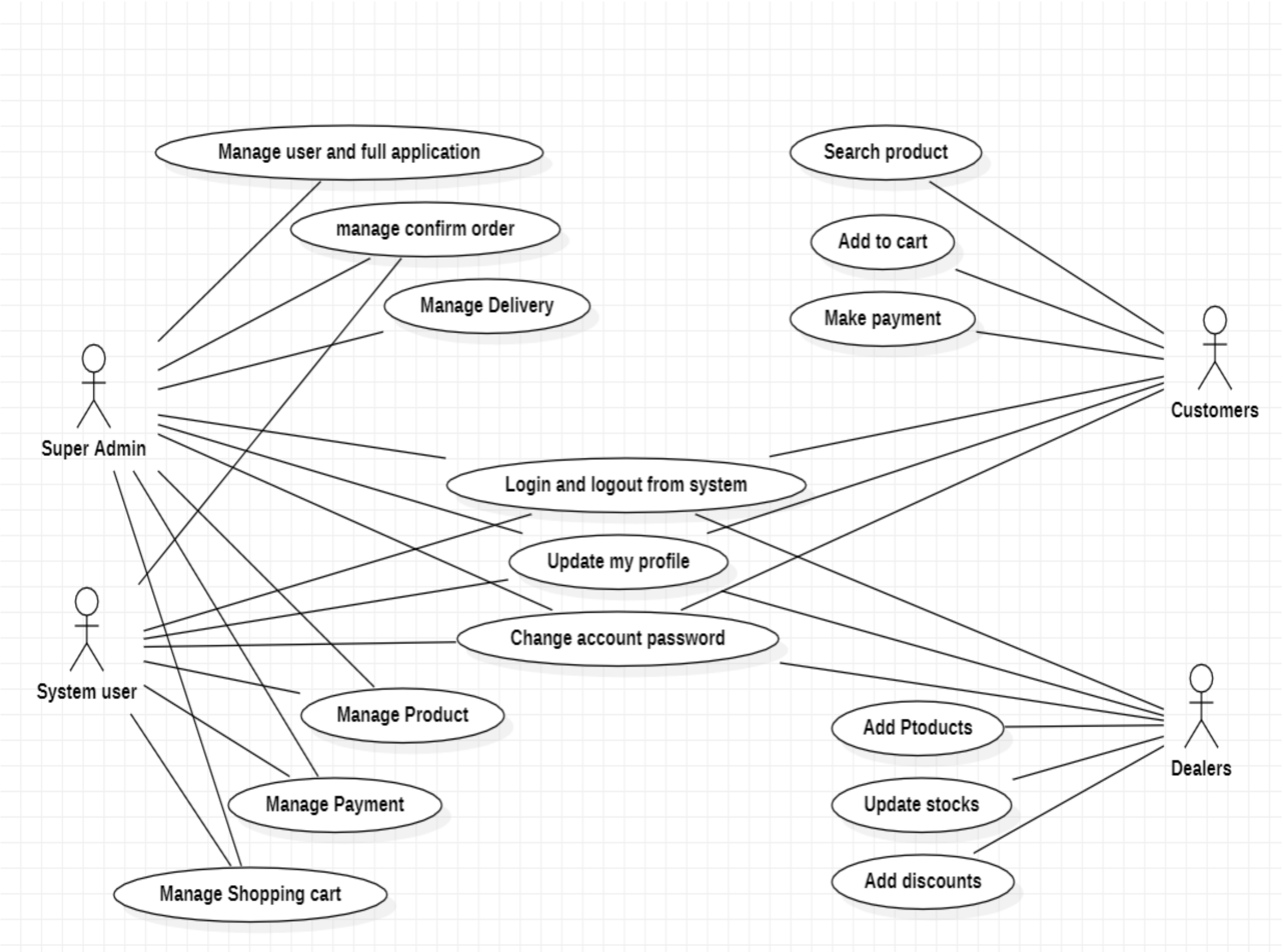
Dashboard module: This component is for administrator to manage settings part of the website. Admin is the superuser having full authority of the website.

Category and sub category: Category and subcategories used to stores product types. Only admin can add and edit or delete the category details.

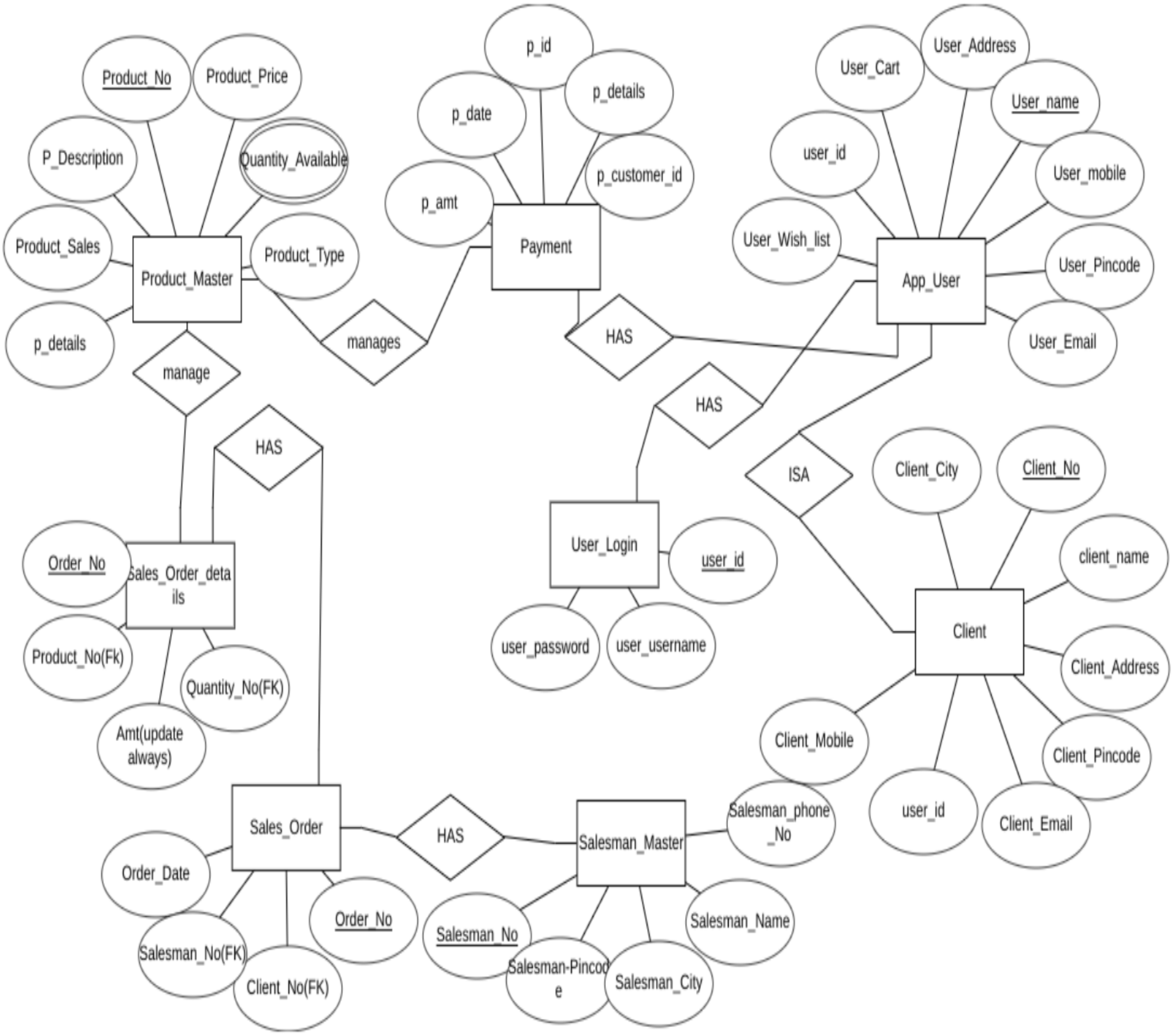
Message Module: Customer can communicate with admin through message features.

Report module: This module displays order report, product report, customer report, etc.

USE CASE DIAGRAM



ER DIAGRAM



DETAILS OF TABLE:

Details of tables:

- 1)Product_master

  - A.p_details -[FK]payment details, This attribute is common between payment and product master table.

    - B.product_sales- Information of product sales
    - C.product_description -detailed information of product which are there to sale

  - D.product_no -[PK]product has given a unique id i.e number .user ordered some specific
    a product which is named by some product number
  - E.product_price -prices of products
  - F.quantity_available -multivalued attribute - this is a multivalued attribute which describes available quantities.
  - G.product_type -Types of product.categories like women, men,etc.

- 2.sales_order_details - A.order_no-[PK]user orders some specific product and it has some order number.

  - B.product_no- [FK]product has given a unique id i.e number. user ordered some specific
    a product which is named by some product number
  - C.amt-price of product
  - D.quantity_no- how many quantities are ordered by the user is indicated by quantity number
- 3.Sales_order -
  - A.order_date-date on which user ordered product
  - B.salesman_no-[FK]salesman who will deliver order to user
  - C.client_no- [FK]user number,id ,etc.
  - D.order_no-[PK]user orders some specific product and it has some order number.

- 4.salesman_master

  - A.salesman_no-[PK]salesman unique id
  - B.salesman_pincode-pincode id of salesman
  - C.salesman_city-city of salesman
  - D.salesman_name-name of salesman

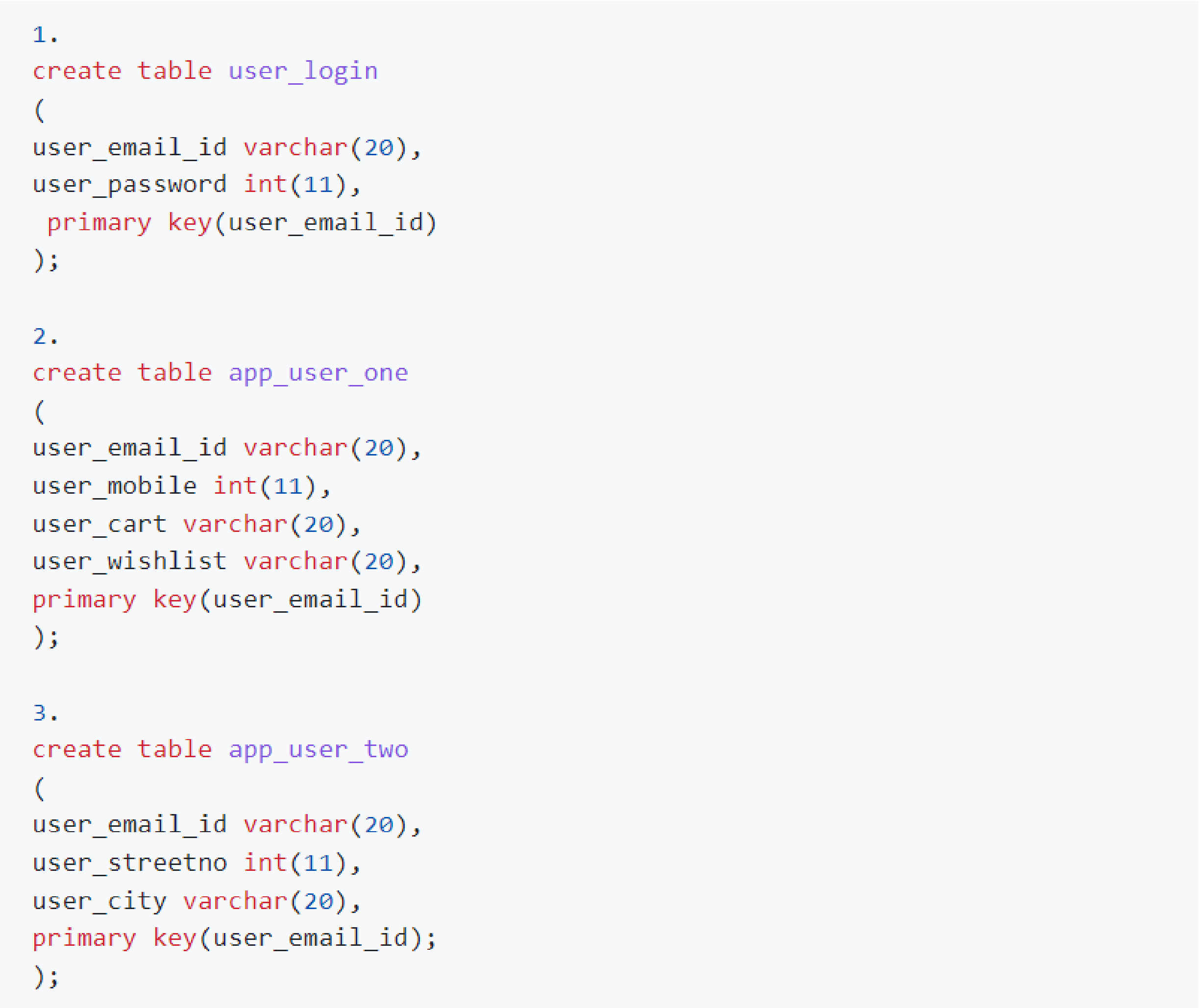
  - E.salesman_phone_no-salesman phone number

- 5.client-
  - A.client_mobile-mobile number of client
  - B.user_id-[FK]user id of client to log in to web/app
  - C.client_email-email id of client for creation of workspace for user
  - D.client_pincode-pincode of client,address requirement
  - E.client_address-address of client for delivery purpose
  - F.client_name-name of client
  - G.client_no-[PK]unique id of client which is primary key of this table
  - H.client_city-city of client for address purpose for delivery.

- 6.App_user -
  - A.user_wish_list-user wants to wish to buy this product
  - B.user_id-[FK]user id of client/user user id is necessary for log in to the web/app
  - C.user_cart-user added product to buy the product
  - D.user_address-address of user
  - E.user_name-name of user for log in to the app/website
  - F.user_mobile-mobile number of user
  - G.user_pincode-pincode of user for location purpose.
  - H.user_email-email address of the user for login purpose.

- 7.user_log_in
  - A.user_password-password of the user for security purposes.
  - B.user_username-username of user for login purpose

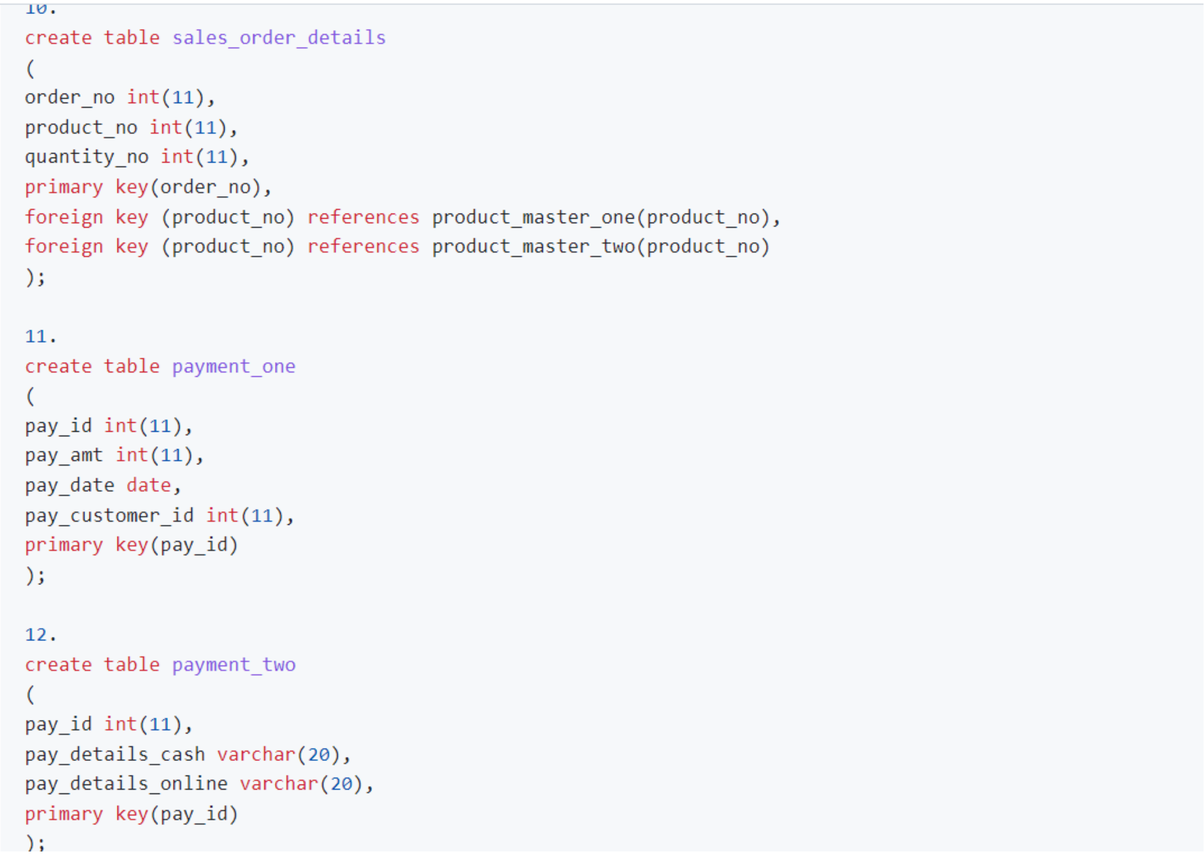
DATABASE CREATION USING DDL AND DML



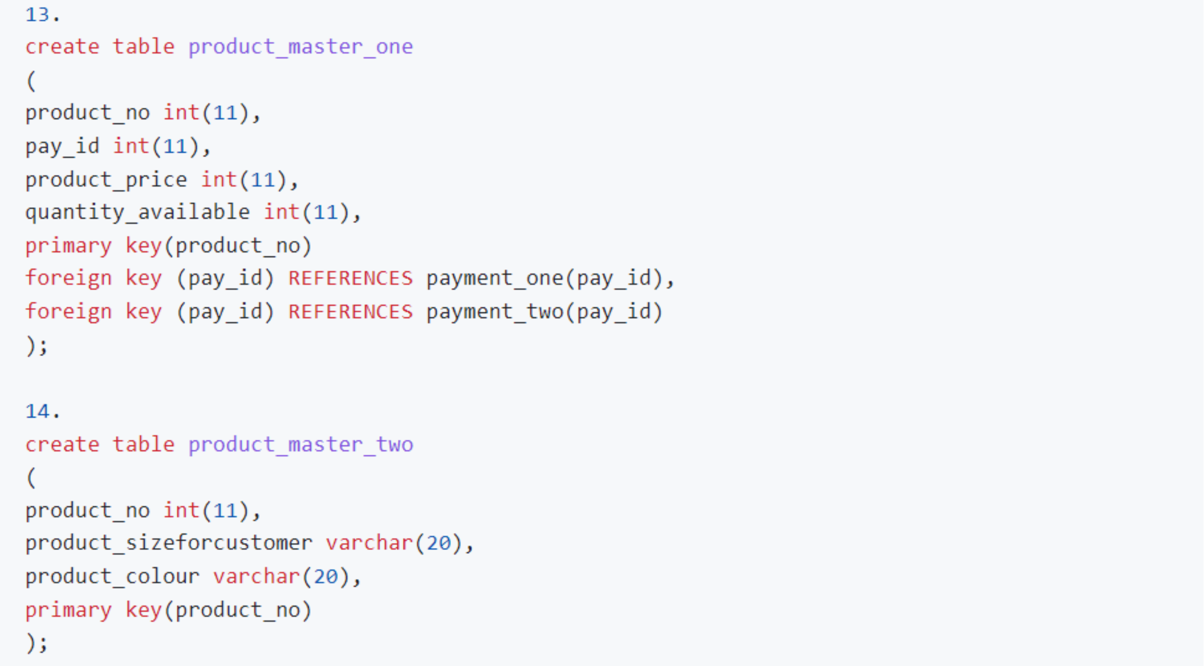
```
1.
create table user_login
(
user_email_id varchar(20),
user_password int(11),
 primary key(user_email_id)
);

2.
create table app_user_one
(
user_email_id varchar(20),
user_mobile int(11),
user_cart varchar(20),
user_wishlist varchar(20),
primary key(user_email_id)
);

3.
create table app_user_two
(
user_email_id varchar(20),
user_streetno int(11),
user_city varchar(20),
primary key(user_email_id);
);
```

```
4.
create table app_user_three(
user_email_id varchar(20),
user_state varchar(20),
user_pincode int(11),
primary key(user_email_id)
);

5.
create table client_one
(
client_no int(11),
client_name varchar(20),
client_email varchar(20),
client_mobile int(11),
user_email_id varchar(20),
 primary key(client_no),
foreign key(user_email_id) references app_user_one(user_email_id),
foreign key(user_email_id) references app_user_two(user_email_id),
foreign key(user_email_id) references app_user_three(user_email_id),
 foreign key(user_email_id) references app_user_one(user_email_id)
);

6.
create table client_two
(
client_no int(11),
client_streetno int(11),
primary key(client_no)
);
```

```
7.
create table client_three
(
client_no int(11),
client_pincode int(11)
);

8.
 create table salesman_master
(
salesman_no int(11),
salesman_name varchar(20),
salesman_phoneno int(11),
salesman_pincode int(11),
primary key(salesman_no);
);

9.
create table sales_order
(
order_no int(11),
order_date date,
salesman_no int(11),
client_no int(11),
foreign key (order_no) references sales_order_details(order_no),
foreign key (client_no) references client_one(client_no),
foreign key (salesman_no) references salesman_master(salesman_no),
foreign key (order_no) references sales_order_details(order_no)
);
```

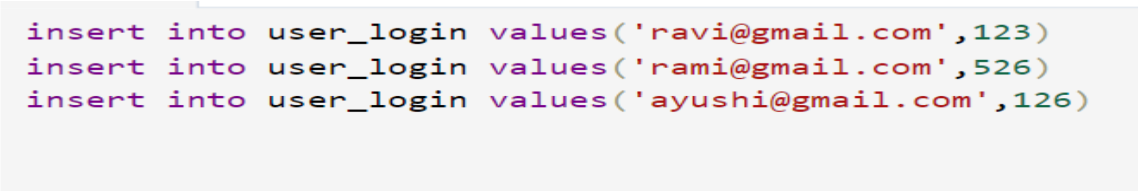


```
10.
create table sales_order_details
(
order_no int(11),
product_no int(11),
quantity_no int(11),
primary key(order_no),
foreign key (product_no) references product_master_one(product_no),
foreign key (product_no) references product_master_two(product_no)
);

11.
create table payment_one
(
pay_id int(11),
pay_amt int(11),
pay_date date,
pay_customer_id int(11),
primary key(pay_id)
);

12.
create table payment_two
(
pay_id int(11),
pay_details_cash varchar(20),
pay_details_online varchar(20),
primary key(pay_id)
);
```

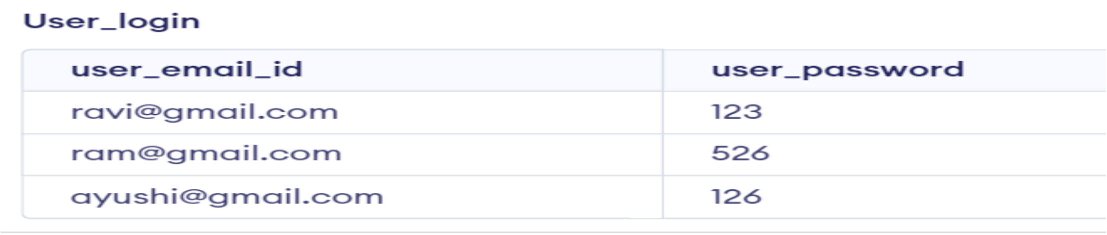


```
13.
create table product_master_one
(
product_no int(11),
pay_id int(11),
product_price int(11),
quantity_available int(11),
primary key(product_no)
foreign key (pay_id) REFERENCES payment_one(pay_id),
foreign key (pay_id) REFERENCES payment_two(pay_id)
);

14.
create table product_master_two
(
product_no int(11),
product_sizeforcustomer varchar(20),
product_colour varchar(20),
primary key(product_no)
);
```

INSERTING VALUES AND OUTPUT

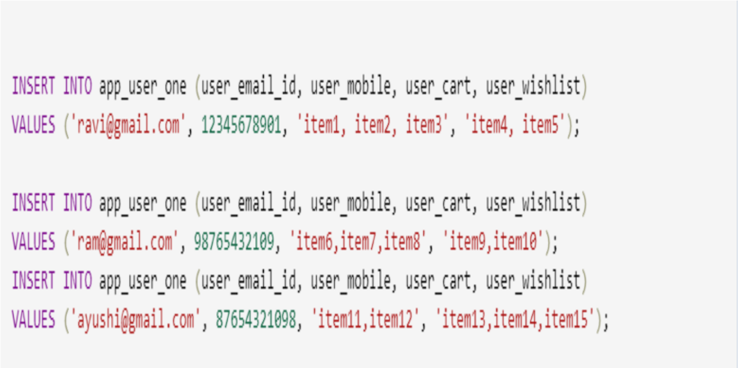


```
insert into user_login values('ravi@gmail.com',123)
insert into user_login values('rami@gmail.com',526)
insert into user_login values('ayushi@gmail.com',126)
```

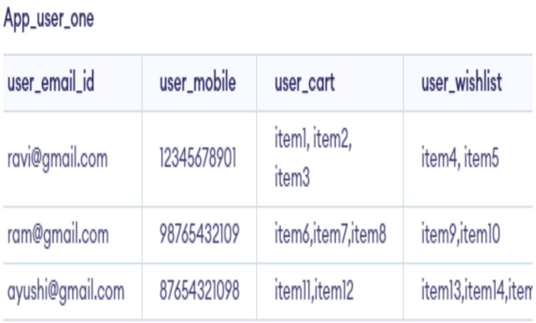


User_login

user_email_id	user_password
ravi@gmail.com	123
ram@gmail.com	526
ayushi@gmail.com	126

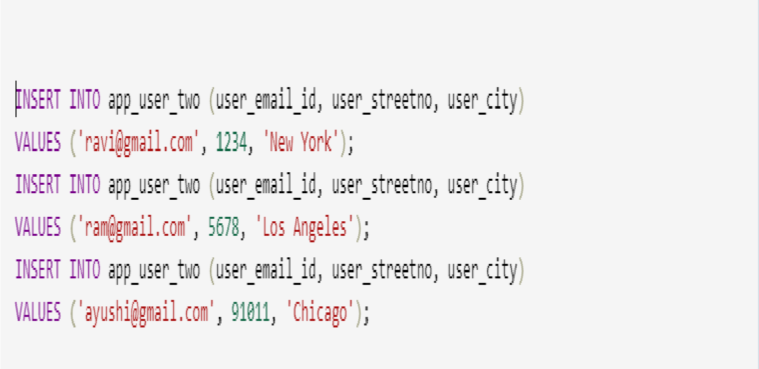


```
INSERT INTO app_user_one (user_email_id, user_mobile, user_cart, user_wishlist)
VALUES ('ravi@gmail.com', 12345678901, 'item1, item2, item3', 'item4, item5');

INSERT INTO app_user_one (user_email_id, user_mobile, user_cart, user_wishlist)
VALUES ('ram@gmail.com', 98765432109, 'item6,item7,item8', 'item9,item10');
INSERT INTO app_user_one (user_email_id, user_mobile, user_cart, user_wishlist)
VALUES ('ayushi@gmail.com', 87654321098, 'item11,item12', 'item13,item14,item15');
```

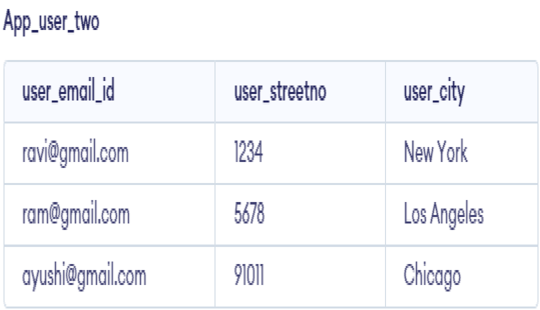


App_user_one

user_email_id	user_mobile	user_cart	user_wishlist
ravi@gmail.com	12345678901	item1, item2, item3	item4, item5
ram@gmail.com	98765432109	item6,item7,item8	item9,item10
ayushi@gmail.com	87654321098	item11,item12	item13,item14,iten

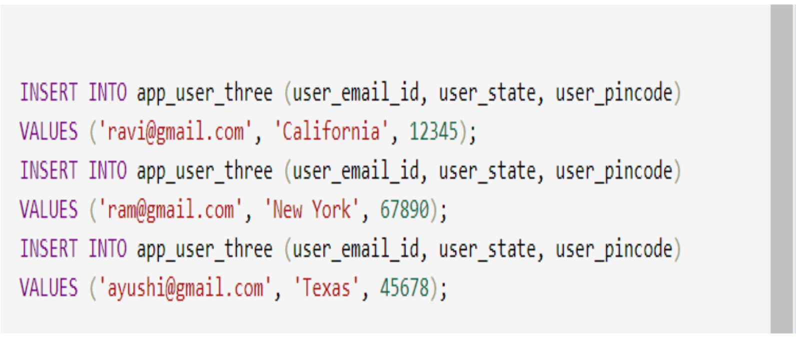


```
INSERT INTO app_user_two (user_email_id, user_streetno, user_city)
VALUES ('ravi@gmail.com', 1234, 'New York');
INSERT INTO app_user_two (user_email_id, user_streetno, user_city)
VALUES ('ram@gmail.com', 5678, 'Los Angeles');
INSERT INTO app_user_two (user_email_id, user_streetno, user_city)
VALUES ('ayushi@gmail.com', 91011, 'Chicago');
```

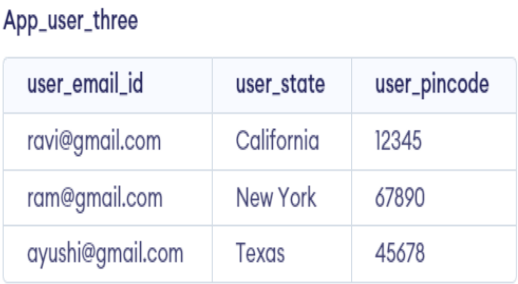


App_user_two

user_email_id	user_streetno	user_city
ravi@gmail.com	1234	New York
ram@gmail.com	5678	Los Angeles
ayushi@gmail.com	91011	Chicago

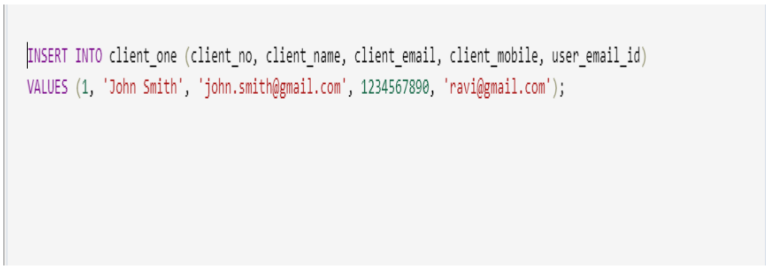


```
INSERT INTO app_user_three (user_email_id, user_state, user_pincode)
VALUES ('ravi@gmail.com', 'California', 12345);
INSERT INTO app_user_three (user_email_id, user_state, user_pincode)
VALUES ('ram@gmail.com', 'New York', 67890);
INSERT INTO app_user_three (user_email_id, user_state, user_pincode)
VALUES ('ayushi@gmail.com', 'Texas', 45678);
```

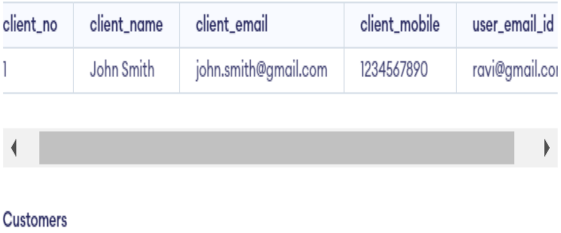


App_user_three

user_email_id	user_state	user_pincode
ravi@gmail.com	California	12345
ram@gmail.com	New York	67890
ayushi@gmail.com	Texas	45678

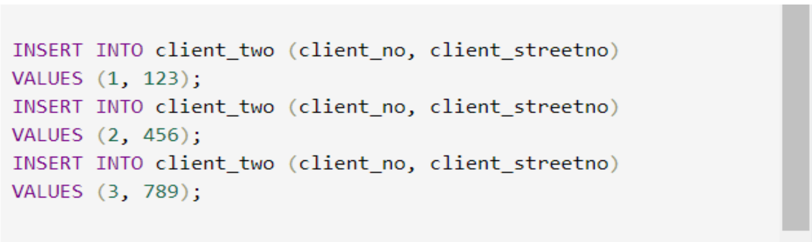


```
INSERT INTO client_one (client_no, client_name, client_email, client_mobile, user_email_id)
VALUES (1, 'John Smith', 'john.smith@gmail.com', 1234567890, 'ravi@gmail.com');
```

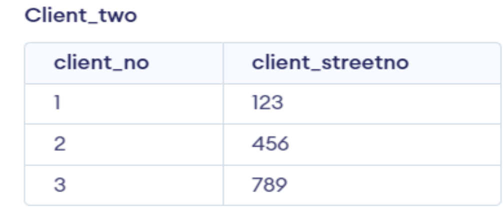


client_no	client_name	client_email	client_mobile	user_email_id
1	John Smith	john.smith@gmail.com	1234567890	ravi@gmail.co

Customers

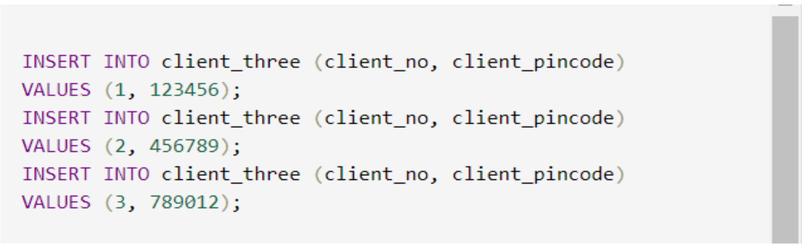


```
INSERT INTO client_two (client_no, client_streetno)
VALUES (1, 123);
INSERT INTO client_two (client_no, client_streetno)
VALUES (2, 456);
INSERT INTO client_two (client_no, client_streetno)
VALUES (3, 789);
```

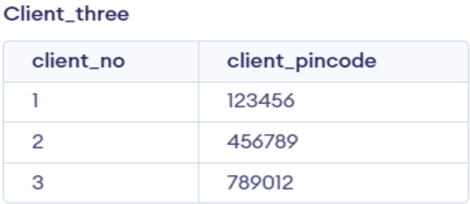


Client_two

client_no	client_streetno
1	123
2	456
3	789

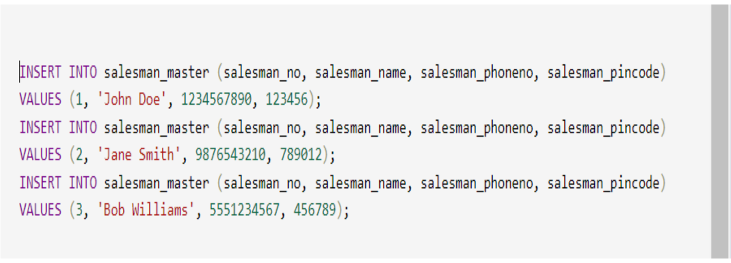


```
INSERT INTO client_three (client_no, client_pincode)
VALUES (1, 123456);
INSERT INTO client_three (client_no, client_pincode)
VALUES (2, 456789);
INSERT INTO client_three (client_no, client_pincode)
VALUES (3, 789012);
```

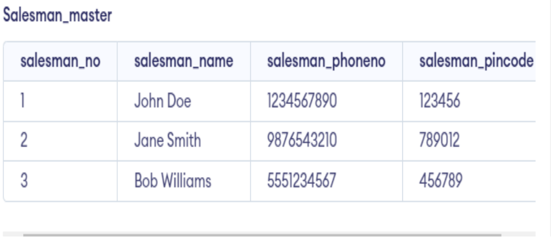


Client_three

client_no	client_pincode
1	123456
2	456789
3	789012

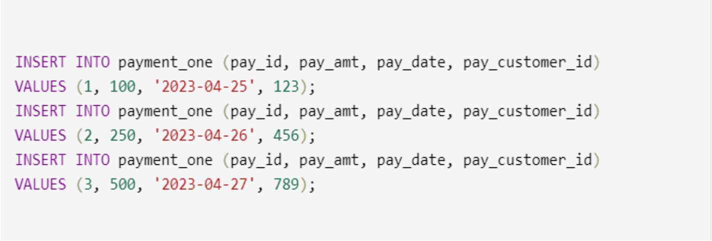


```
INSERT INTO salesman_master (salesman_no, salesman_name, salesman_phoneno, salesman_pincode)
VALUES (1, 'John Doe', 1234567890, 123456);
INSERT INTO salesman_master (salesman_no, salesman_name, salesman_phoneno, salesman_pincode)
VALUES (2, 'Jane Smith', 9876543210, 789012);
INSERT INTO salesman_master (salesman_no, salesman_name, salesman_phoneno, salesman_pincode)
VALUES (3, 'Bob Williams', 5551234567, 456789);
```

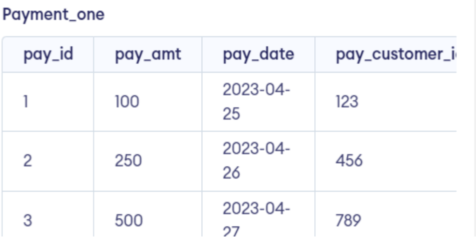


Salesman_master

salesman_no	salesman_name	salesman_phoneno	salesman_pincode
1	John Doe	1234567890	123456
2	Jane Smith	9876543210	789012
3	Bob Williams	5551234567	456789

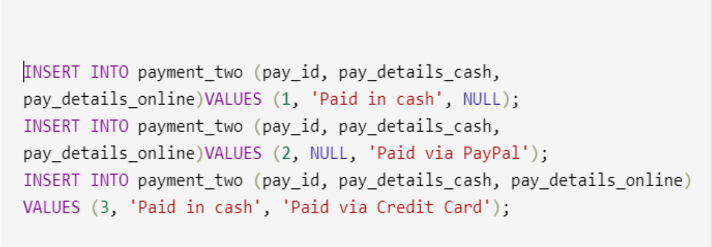


```
INSERT INTO payment_one (pay_id, pay_amt, pay_date, pay_customer_id)
VALUES (1, 100, '2023-04-25', 123);
INSERT INTO payment_one (pay_id, pay_amt, pay_date, pay_customer_id)
VALUES (2, 250, '2023-04-26', 456);
INSERT INTO payment_one (pay_id, pay_amt, pay_date, pay_customer_id)
VALUES (3, 500, '2023-04-27', 789);
```

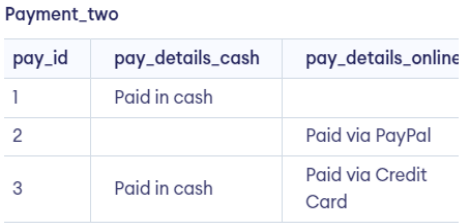


Payment_one

pay_id	pay_amt	pay_date	pay_customer_i
1	100	2023-04-25	123
2	250	2023-04-26	456
3	500	2023-04-27	789

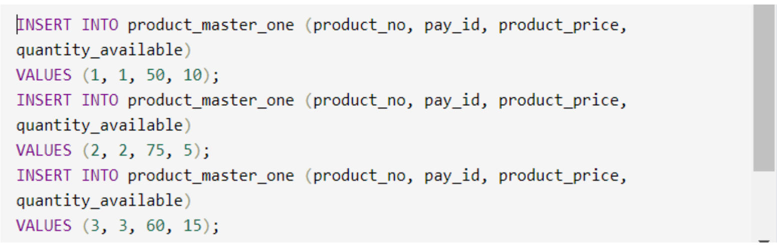


```
INSERT INTO payment_two (pay_id, pay_details_cash,
pay_details_online)VALUES (1, 'Paid in cash', NULL);
INSERT INTO payment_two (pay_id, pay_details_cash,
pay_details_online)VALUES (2, NULL, 'Paid via PayPal');
INSERT INTO payment_two (pay_id, pay_details_cash, pay_details_online)
VALUES (3, 'Paid in cash', 'Paid via Credit Card');
```

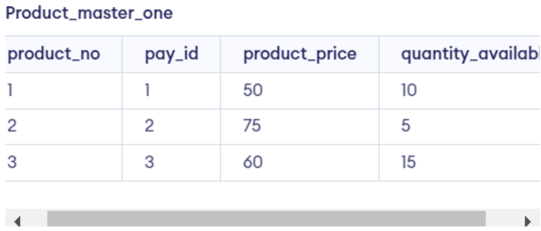


Payment_two

pay_id	pay_details_cash	pay_details_online
1	Paid in cash	
2		Paid via PayPal
3	Paid in cash	Paid via Credit Card



```
INSERT INTO product_master_one (product_no, pay_id, product_price,
quantity_available)
VALUES (1, 1, 50, 10);
INSERT INTO product_master_one (product_no, pay_id, product_price,
quantity_available)
VALUES (2, 2, 75, 5);
INSERT INTO product_master_one (product_no, pay_id, product_price,
quantity_available)
VALUES (3, 3, 60, 15);
```



Product_master_one

product_no	pay_id	product_price	quantity_availabl
1	1	50	10
2	2	75	5
3	3	60	15

NORMALIZATION OF DATABASE

USING 1NF,2NF,3NF:

1.Product_master(product_no,product_size,product_description,product_type,quantity_available p_details)

1NF-

product_master_one(product_no,p_id,product_price,quantity_available)

product_master_two(product_no,product_sizeforcustomer,product_colour)

2NF-

product_master_one(product_no,p_id,product_price,quantity_available)

product_master_two(product_no,product_sizeforcustomer,product_colour)

3NF-

product_master_one(product_no,p_id,product_price,quantity_available)

product_master_two(product_no,product_sizeforcustomer,product_colour)

2.payment(p_amt,p_date,p_id,p_details,p_customer_id)

1NF-

payment_one(p_id,p_amt,p_date,p_customer_id

payment_two(p_id,p_details_cash,p_details_online)

2NF-

payment_one(p_id,p_amt,p_date,p_customer_id)

payment_two(p_id,p_details_cash,p_details_online)

3NF-

payment_one(p_id,p_amt,p_date,p_customer_id)

payment_two(p_id,p_details_cash,p_details_online)

3.sales_order_details(order_no,product_no,quantit_no)

1NF-

sales_order_details(order_no,product_no,quantit_no)

2NF-

sales_order_details(order_no,product_no,quantit_no)

3NF-

sales_order_details(order_no,product_no,quantit_no)

4.sales_order(order_no,order_date,salesman_no,client-no)

1NF-

sales_order(order_no,order_date,salesman_no,client-no)

2NF-

sales_order(order_no,order_date,salesman_no,client-no)

3NF-

sales_order(order_no,order_date,salesman_no,client-no)

5.salesman_master(salesman_no,salesman_name,salesman_phoneno,salesman_address)

1NF-

salesman_master(salesman_no,salesman_name,salesman_phoneno,salesman_pincode)

2NF-

salesman_master(salesman_no,salesman_name,salesman_phoneno,salesman_pincode)

3NF-

salesman_master(salesman_no,salesman_name,salesman_phoneno,salesman_pincode)

6.client(client_no,client_name,client_address,client_city,client_mobile,client_email,client_pincode,user_email_id)

1NF-

client_one(client_no,client_name,client_email,client_mobile,user_email_id)

client_two(client_no,client_streetno,client_city,client_state,client_pincode)

2NF-

client_one(client_no,client_name,client_email,client_mobile,user_email_id)

client_two(client_no,client_streetno,client_city,client_state,client_pincode)

3NF-

client_one(client_no,client_name,client_email,client_mobile, user_email_id)

client_two(client_no,client_streetno,client_city)

client_three(client_no,client_state,client_pincode)

7.app_user(user_email_id,user_pincode,user_user_mobile,user_address,user_cart,user_wishlist)

1NF-

app_user_one(user_email_id,user_mobile,user_cart,user_wishlist)

app_user_two(user_email_id,user_streetno,user_city,user_state,user_pincode)

2NF-

app_user_one(user_email_id,user_mobile,user_cart,user_wishlist)

app_user_two(user_email_id,user_streetno,user_city,user_state,user_pincode)

3NF-

app_user_one(user_email_id,user_mobile,user_cart,user_wishlist)

app_user_two(user_email_id,user_streetno,user_city)

app_user_three(user_email_id,user_state,user_pincode)

8.user_login(user-email_id,user_password)

1NF-

user_login(user-email_id,user_password)

2NF-

user_login(user-email_id,user_password)

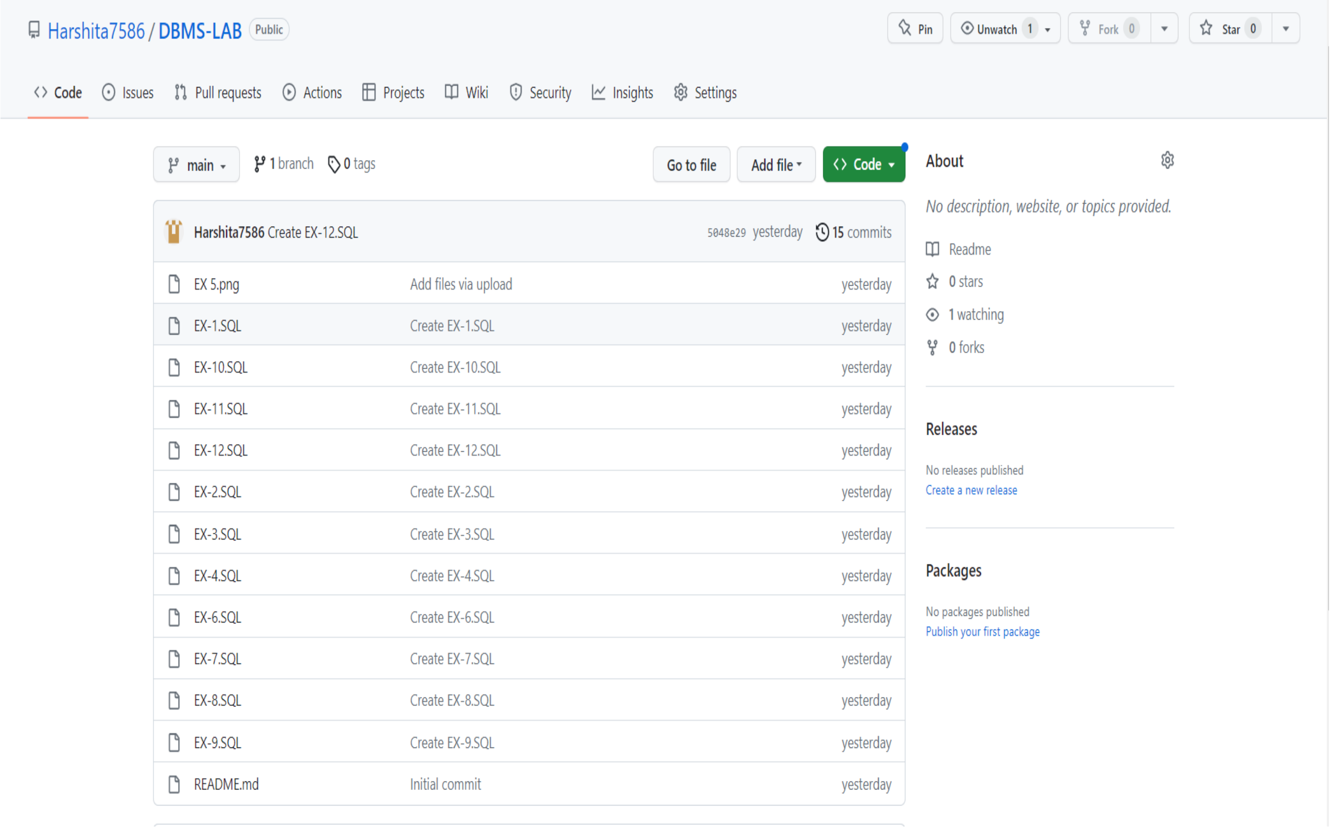
3NF-

user_login(user-email_id,user_password)

CONCLUSION

he ‘Online Shopping’ is designed to provide a web based application that would make searching, viewing and selection of a product easier. The search engine provides an easy and convenient way to search for products where a user can Search for a product interactively and the search engine would refine the products available based on the user’s input. The user can then view the complete specification of each product. They can also view the product reviews and also write their own reviews. Use of Ajax components would make the application interactive and prevents annoying post backs. Its drag and drop feature would make it easy to use.

APPENDIX-I SCREENSHOT



Harshita7586 Create EX-12.SQL 5048e29 yesterday 15 commits

EX 5.png	Add files via upload	yesterday
EX-1.SQL	Create EX-1.SQL	yesterday
EX-10.SQL	Create EX-10.SQL	yesterday
EX-11.SQL	Create EX-11.SQL	yesterday
EX-12.SQL	Create EX-12.SQL	yesterday
EX-2.SQL	Create EX-2.SQL	yesterday
EX-3.SQL	Create EX-3.SQL	yesterday
EX-4.SQL	Create EX-4.SQL	yesterday
EX-6.SQL	Create EX-6.SQL	yesterday
EX-7.SQL	Create EX-7.SQL	yesterday
EX-8.SQL	Create EX-8.SQL	yesterday
EX-9.SQL	Create EX-9.SQL	yesterday
README.md	Initial commit	yesterday

About

No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Releases

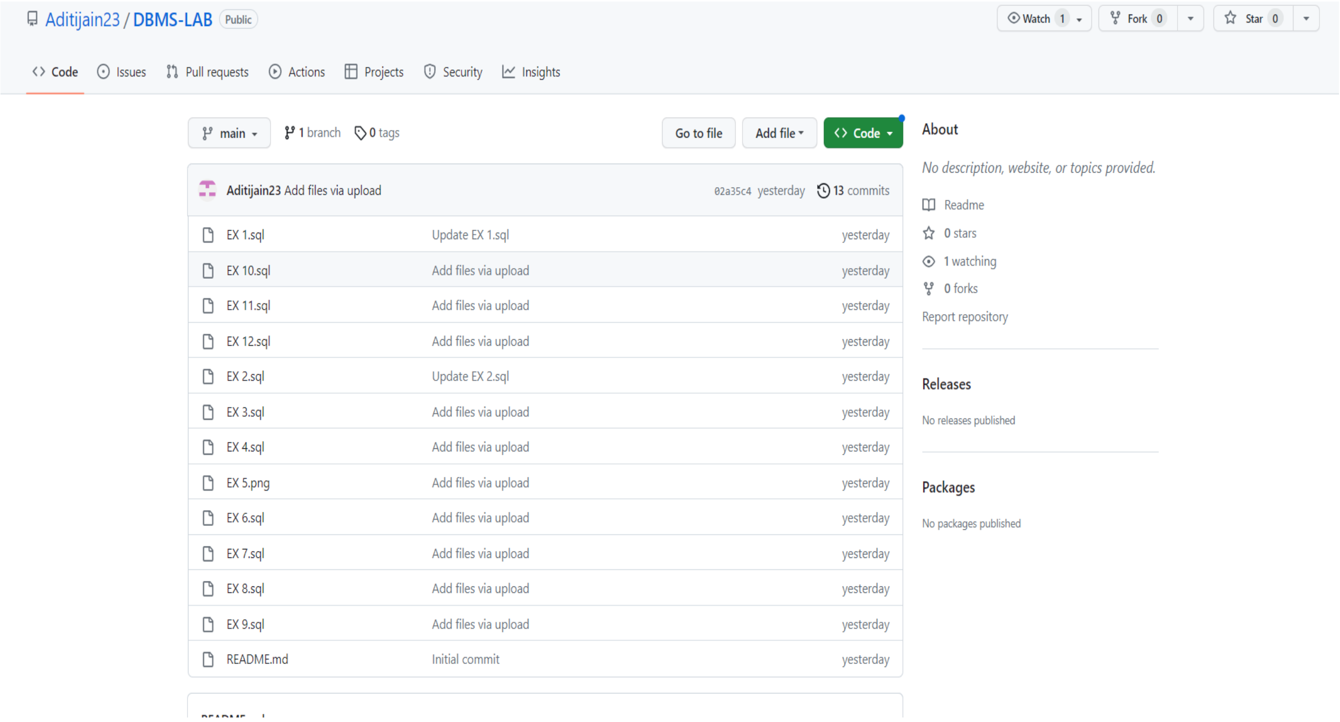
No releases published

Create a new release

Packages

No packages published

Publish your first package



Aditijain23 Add files via upload 02a35c4 yesterday 13 commits

EX 1.sql	Update EX 1.sql	yesterday
EX 10.sql	Add files via upload	yesterday
EX 11.sql	Add files via upload	yesterday
EX 12.sql	Add files via upload	yesterday
EX 2.sql	Update EX 2.sql	yesterday
EX 3.sql	Add files via upload	yesterday
EX 4.sql	Add files via upload	yesterday
EX 5.png	Add files via upload	yesterday
EX 6.sql	Add files via upload	yesterday
EX 7.sql	Add files via upload	yesterday
EX 8.sql	Add files via upload	yesterday
EX 9.sql	Add files via upload	yesterday
README.md	Initial commit	yesterday

About

No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Report repository

Releases

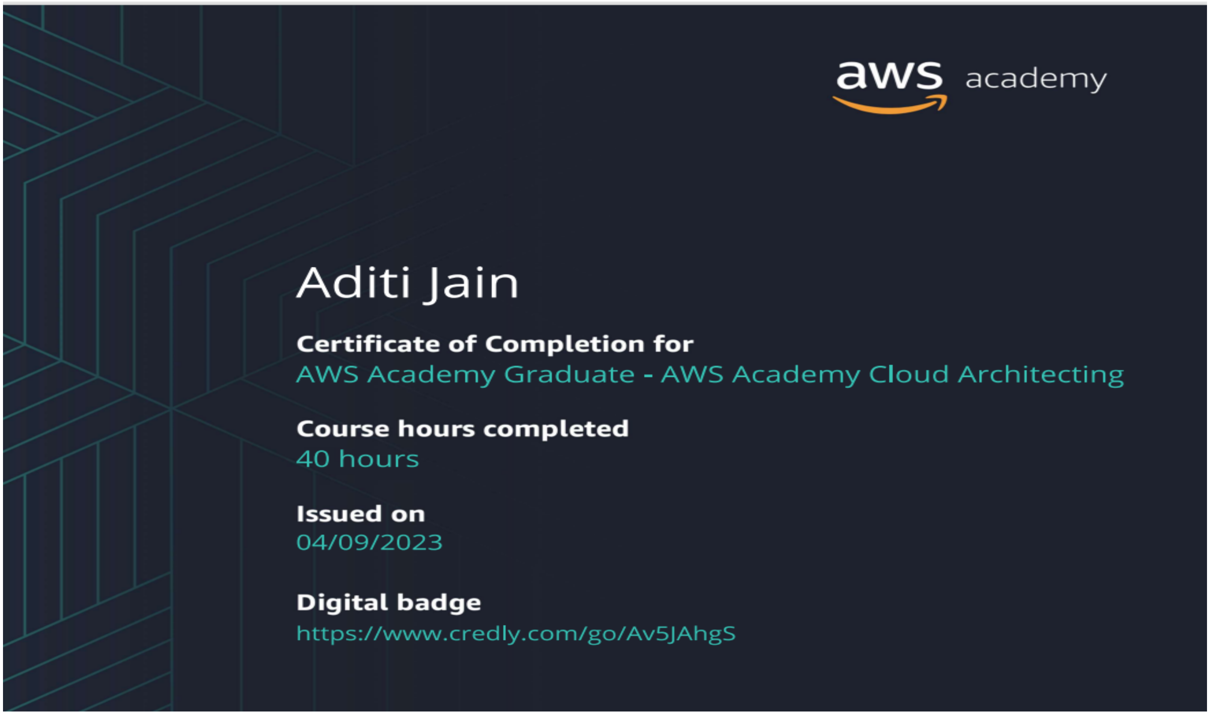
No releases published

Packages

No packages published

APPENDIX-II

AWS CERTIFICATE



aws academy

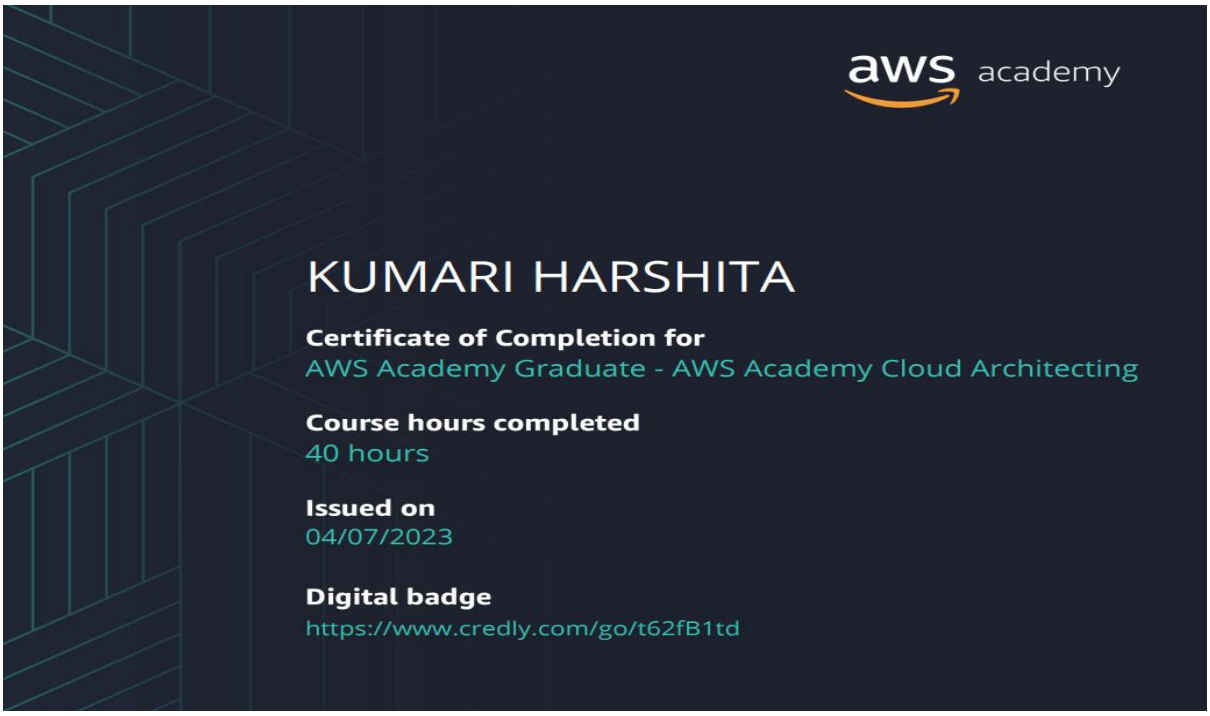
Aditi Jain

Certificate of Completion for
AWS Academy Graduate - AWS Academy Cloud Architecting

Course hours completed
40 hours

Issued on
04/09/2023

Digital badge
https://www.credly.com/go/Av5JAhgS



aws academy

KUMARI HARSHITA

Certificate of Completion for
AWS Academy Graduate - AWS Academy Cloud Architecting

Course hours completed
40 hours

Issued on
04/07/2023

Digital badge
https://www.credly.com/go/t62fB1td