

Online Office Furniture Store Project

Deliverable 3

CS 631 – Data Management System Design

Section – 005

Group Number – 05

Group Members:

Jyothsna Kaamala (jk734)

Het Patel (hp544)

Bhavya Shah (bs635)

Goals of the Phase:

- The goal of this phase was to develop the database and application programs for the furniture store.
- This involved designing the database schema and instance, implementing the database and application programs using node js and react js, and integrating the database with the application programs using the Postgres database and appropriate APIs.
- The goal was to create a functional and user-friendly system that allows the furniture store to manage its customers, products, and sales transactions, and provides the administrators with the tools to analyze the sales and customers.
- By completing this phase, the furniture store now has a working database and application that can be used to manage its operations and improve its performance.

Implementation of the Project:

- To develop the database for the furniture store, we created the tables based on the entities and their attributes identified in the previous phase.
- This was done using SQL commands to create tables, specify the data types and constraints for each attribute, and establish relationships between tables.
- Once the tables are created, we add the data to the tables using SQL commands such as INSERT, UPDATE, and DELETE.
- We created views to present the data in a specific way and make it easier to access and manage the data.
- Next, we developed the interface for the furniture store using node js and react js.
- This will involve creating the front-end components, such as forms for entering data, tables for displaying data, and buttons for interacting with the data.
- We added features such as user Jwt authentication and validation of user input.
- We integrated it with the database using the Postgres database and appropriate APIs.
- This will allow the interface to communicate with the database and perform the necessary operations, such as adding, updating, and deleting data, as well as running queries to retrieve data for display on the front-end.
- Sales statistics were created by using SQL queries to retrieve the relevant data from the database.
- The results can be displayed and can be filtered by time period (begin date and end date) to focus on specific periods of time.
- This helps the administrators to make more informed decisions and improve the performance of the store.
- Finally, we tested the database and interface to ensure they are working as expected, and make any necessary adjustments.
- Once the testing is complete, the database and interface can be deployed for use by the furniture store.

Problems Faced:

- Complex data relationships: Furniture stores typically have a large number of entities and relationships between them. For example, a customer can have multiple credit cards, shipping addresses, and products in their shopping basket. These complex relationships can make it challenging to design and implement the database and application programs.
- Integration with existing systems: The database and application programs for the furniture store had to be integrated with one another. This introduced the complexity of creating APIs.

Functionalities:

- Login: The login option allows registered users to access their account by providing their email and password. Once logged in, the user will be able to view their account details, manage their credit cards and shipping addresses, and place orders for products. The login stores the jwt token in the localstorage, which is sent
- Register: The register option allows new users to create an account by providing their email, password, and basic personal information. After creating an account, the user will be able to login and access their account.
- Dashboard: The dashboard option provides a user-friendly interface for browsing and searching for products, viewing product details, and adding products to the cart. The dashboard also displays the user's account details, including their status and any special offers that are available.
- Cart: The cart option allows users to view and manage the items in their shopping cart, including changing the quantities of items, applying special offers, and removing items. The cart also displays the total cost of the items, including any applicable discounts or fees.
- Admin : The sales statistics can be accessed through the admin panel of the furniture store, and is available only to users with administrative privileges. The sales statistics can be used to analyze its sales and customers. It allows the administrators to compute various statistics and metrics, such as the most frequently sold products, the products with the highest number of distinct customers, the top customers in terms of money spent, and the average selling product price per product type.

User guide:

1. To use the furniture store, a user must first create an account by clicking on the register option and providing their email, password, and personal information.
2. Once the account is created, the user can login by clicking on the login option and providing their email and password.
3. After logging in, the user can access their account details, manage their credit cards and shipping addresses, and view any special offers that are available.
4. To browse and search for products, the user can click on the dashboard option and use the search bar or filters to find the desired products.
5. To view more details about a product, the user can click on the product image or name to open a detailed product page.
6. To add a product to the cart, the user can click on the "add to cart" button on the product page. The user can view and manage the items in the cart by clicking on the cart option.
7. To place an order, the user can click on the "checkout" button in the cart, and select a shipping address and payment method. The user can then review the order details and confirm the order.
8. If the user has administrative privileges, they can access the sales statistics panel. This helps the administrators to make more informed decisions and improve the performance of the store.
9. To log out of the account, the user can click on the "logout" button in the top right corner of the screen. This will end the user's session and return them to the login screen.

SQL commands that populate the tables:

```
CREATE DATABASE desksrus;

CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

CREATE TABLE CUSTOMER (
    CID int PRIMARY KEY DEFAULT SERIAL,
    FName varchar(255) NOT NULL,
    LName varchar(255) NOT NULL,
    CustStatus varchar(255) NOT NULL DEFAULT 'Bronze',
    Email varchar(255) NOT NULL,
    Password VARCHAR(255) NOT NULL,
    CustAddress varchar(255),
    Phone text,
    UserType varchar(255) NOT NULL DEFAULT 'User'
);

CREATE TABLE SILVER_AND_ABOVE (
    CID int NOT NULL,
    CreditLine int,
    PRIMARY KEY (CID),
    CONSTRAINT FK_Silver FOREIGN KEY (CID) REFERENCES Customer
    (CID)
);

CREATE TABLE Credit_Card (
    CardNumber varchar(255) NOT NULL,
    SecNumber int NOT NULL,
    CardOwnerName varchar(255) NOT NULL,
    CardType varchar(255) NOT NULL,
    BillingAddress varchar(255) NOT NULL,
    ExpDate date NOT NULL,
    PRIMARY KEY (CardNumber)
);
```

```

CREATE TABLE Stored_Card (
    CardNumber varchar(255) NOT NULL,
    CID int NOT NULL,
    PRIMARY KEY (CardNumber, CID),
    CONSTRAINT FK_Stored FOREIGN KEY (CID) REFERENCES Customer
    (CID)
);

ALTER TABLE stored_card
    ADD FOREIGN KEY (CardNumber) REFERENCES Credit_Card
    (CardNumber);

CREATE TABLE SHIP_ADDRESS (
    CID int NOT NULL,
    SAName varchar(255) NOT NULL,
    SNumber int,
    Country varchar(255) NOT NULL,
    State varchar(255) NOT NULL,
    City varchar(255) NOT NULL,
    Street varchar(255) NOT NULL,
    ZIP int NOT NULL,
    RecipientName varchar(255),
    PRIMARY KEY (CID, SAName),
    FOREIGN KEY (CID) REFERENCES Customer (CID) ON DELETE CASCADE
);

CREATE TABLE TRANSACTION (
    TransactionID int NOT NULL,
    TStatus varchar(255) NOT NULL,
    TDate date NOT NULL,
    TotalAmount int NOT NULL,
    CardNumber varchar(255) NOT NULL,
    CID int NOT NULL,
    PRIMARY KEY (TransactionID),
    FOREIGN KEY (CardNumber) REFERENCES credit_card (CardNumber)
    ON DELETE CASCADE,

```

```
    FOREIGN KEY (CID) REFERENCES Customer (CID) ON DELETE CASCADE
)
CREATE TABLE CART (
    CartID int NOT NULL,
    TransactionID int NOT NULL,
    Status varchar(255) NOT NULL,
    Date date NOT NULL,
    TotalAmount int NOT NULL,
    ShipAddName varchar(255) NOT NULL,
    CID int NOT NULL,
    PRIMARY KEY (CartID),
    FOREIGN KEY (CID, ShipAddName) REFERENCES ship_address (CID,
    SAName) ON DELETE CASCADE,
    FOREIGN KEY (CID) REFERENCES Customer (CID) ON DELETE CASCADE,
    FOREIGN KEY (TransactionID) REFERENCES TRANSACTION
    (TransactionID) ON DELETE CASCADE
);
```

```
CREATE TABLE Product (
    ProductID int NOT NULL,
    Name varchar(255) NOT NULL,
    Price int NOT NULL,
    Type varchar(255) NOT NULL,
    Quantity int NOT NULL,
    Description varchar(255),
    ImageSrc varchar(255),
    PRIMARY KEY (ProductID)
);
```

```
CREATE TABLE Appears_in (
    CartID int NOT NULL,
    ProductID int NOT NULL,
    Quantity int NOT NULL,
    PriceSold int NOT NULL,
    PRIMARY KEY (CartID, ProductID),
    FOREIGN KEY (CartID) REFERENCES cart (CartID) ON DELETE
    CASCADE,
```

```
    FOREIGN KEY (ProductID) REFERENCES product (ProductID) ON
DELETE CASCADE
);

CREATE TABLE Offer_Product (
    ProductID int NOT NULL,
    OfferPrice int NOT NULL,
    PRIMARY KEY (ProductID),
    FOREIGN KEY (ProductID) REFERENCES product (ProductID) ON
DELETE CASCADE
);

CREATE TABLE Desks (
    ProductID int NOT NULL,
    Material varchar(100),
    Drawers int,
    PRIMARY KEY (ProductID),
    FOREIGN KEY (ProductID) REFERENCES product (ProductID) ON
DELETE CASCADE
);

CREATE TABLE Chairs (
    ProductID int NOT NULL,
    Fabric varchar(100),
    Type varchar(100),
    PRIMARY KEY (ProductID),
    FOREIGN KEY (ProductID) REFERENCES product (ProductID) ON
DELETE CASCADE
);

CREATE TABLE Book_Cases (
    ProductID int NOT NULL,
    Material varchar(100),
    Shelves int,
    PRIMARY KEY (ProductID),
    FOREIGN KEY (ProductID) REFERENCES product (ProductID) ON
DELETE CASCADE);
```

SQL commands that populate the tables:

Insert Queries for adding the data

```
INSERT INTO customer (CID, FName, LName, CustStatus, Email,
Password, CustAddress, Phone) VALUES
('1','Bhavya', 'Shah', 'Gold', 'bhavyashah1@gmail.com',
'bhavya', '305 - 14th Ave. S. Suite 3B', '2015514879'),
('2','Jyothsna', 'Kamala', 'Silver',
'Jk21@gmail.com','jyothsna','Keskuskatu 45', '551201456'),
('3','Het', 'Patel', 'Platinum',
'hetpatel23@gmail.com','het','ul. Filtrowa 68', '2015124567'),
('4','Mike', 'Ross', 'Bronze', 'mikeross@yahoo.com','mike',
'Skagen 21', NULL),
('5','Harvey', 'Specter', 'Gold', 'hs1234@gmail.com', 'hervey',
'123 Broad St', NULL),
('6','James', 'Butt', 'Gold', 'jbutt@gmail.com', 'james', '6649
N Blue Gum St', '5046218927'),
('7','Josephine', 'Darakjy', 'Silver',
'josephine_darakjy@darakjy.org', 'josephine', '4 B Blue Ridge
Blvd', '8102929388'),
('8','Art','Venere', 'Platinum', 'art@venere.org', 'art', '8 W
Cerritos Ave #54', '8566368749'),
('9','Lenna', 'Paprocki', 'Bronze', 'lpaprocki@hotmail.com',
'lenna', '639 Main St', '9073854412'),
('10','Donette', 'Foller', 'Gold', 'donette.foller@cox.net',
'donnette', '34 Center St', '5135701893'),
('11','Simona', 'Morasca', 'Silver', 'simona@morasca.com',
'simona', '3 Mcauley Dr', '4195032484'),
('12','Mitsue', 'Tollner', 'Platinum',
'mitsue_tollner@yahoo.com', 'mitsue', '7 Eads St',
'7735736914'),
('13','Leota', 'Dilliard', 'Bronze', 'leota@hotmail.com',
'leota', '7 W Jackson Blvd', '4087523500'),
('14','Sage', 'Wieser', 'Platinum', 'sage_wieser@cox.net',
'sage', '5 Boston Ave #88', '6054142147'),
```

```
('15','Kris', 'Marrier', 'Bronze', 'kris@gmail.com', 'kris',
'228 Runamuck Pl #2808', '4106558723');
('16','Erica', 'Fernendes', 'Platinum', 'erica23@gmail.com',
'erica', '117 Liberty Ave', '4106558723', 'admin')
```

```
INSERT INTO silver_and_above (CID, CreditLine) VALUES
('1', '50'),
('2', '20'),
('3', '70'),
('5', '20'),
('6', '20'),
('7', '10'),
('8', '100'),
('10', '20'),
('11', '30'),
('12', '50'),
('14', '90');
```

```
INSERT INTO credit_card (CardNumber, SecNumber, CardOwnerName,
CardType, BillingAddress, ExpDate) VALUES
('123123123123', '123', 'Bhavya Shah', 'MasterCard', '305 - 14th
Ave. S. Suite 3B', '2024-12-02'),
('456456456456', '456', 'Jyothsna Kamala', 'MasterCard',
'Keskuskatu 45', '2026-01-16'),
('789789789789', '789', 'Het Patel', 'Visa', 'ul. Filtrowa 68',
'2024-05-10'),
('147147147147', '147', 'Mike Ross', 'Visa', 'Skagen 21',
'2026-11-24'),
('258258258258', '258', 'Harvey Specter', 'MasterCard', '123
Broad St', '2024-08-16'),
('369369369369', '369', 'Het Patel', 'MasterCard', 'ul. Filtrowa
68', '2022-12-21'),
('987987987987', '987', 'James Butt', 'MasterCard', '6649 N Blue
Gum St', '2027-11-21'),
('654654654654', '654', 'Josephine Darakjy', 'MasterCard', '4 B
Blue Ridge Blvd', '2028-01-24'),
```

```
('321321321321', '321', 'Art Venere', 'Visa', '8 W Cerritos Ave  
#54', '2025-5-21'),  
('963963963963', '963', 'Lenna Paprocki', 'Visa', '639 Main St',  
'2024-07-02'),  
('852852852852', '852', 'Donette Foller', 'Visa', '34 Center  
St', '2024-08-02'),  
('741741741741', '741', 'Simona Morasca', 'MasterCard', '3  
Mcauley Dr', '2023-01-02'),  
('159159159159', '159', 'Leota Dilliard', 'MasterCard', '7 W  
Jackson Blvd', '2025-01-22'),  
('357357357357', '357', 'Sage Weiser', 'MasterCard', '5 Boston  
Ave #88', '2026-05-17'),  
('148148148148', '148', 'Lenna Paprocki', 'MasterCard', '639  
Main St', '2024-11-02'),  
('368368368368', '368', 'Josephine Darakjy', 'Visa', '4 B Blue  
Ridge Blvd', '2027-05-02');
```

```
INSERT INTO stored_card (CardNumber, CID) VALUES  
('123123123123', '1'),  
('147147147147', '4'),  
('148148148148', '9'),  
('159159159159', '13'),  
('258258258258', '5'),  
('321321321321', '8'),  
('357357357357', '14'),  
('369369369369', '3'),  
('456456456456', '2'),  
('741741741741', '11'),  
('789789789789', '3'),  
('963963963963', '9');
```

```
INSERT INTO ship_address (CID, SName, SNumber, Country, State,  
City, Street, ZIP, RecipientName) VALUES  
('1', 'Home', '1', 'USA', 'Washington', 'Seattle', '14th Ave',  
'98128', NULL),  
('2', 'Home', '1', 'Finland', 'Uusimaa', 'Helsinki', 'Keskuskatu  
45', '21240', NULL),
```

```
('3', 'Work', '1', 'USA', 'New Jersey', 'Newark', '123 Broad  
St', '03230', 'Het'),  
('4', 'Home', '1', 'Norway', 'Rogaland', 'Stavanger', 'Skagen  
21', '4006', 'Rachael'),  
('5', 'Work', '2', 'USA', 'New Jersey', 'Jersey City', 'Van  
Winkle', '07078', 'Jyothsna'),  
('6', 'Work', '1', 'USA', 'New York', 'NYC', '424 BroadWay',  
'25412', 'Donna'),  
('7', 'Home', '1', 'USA', 'Illinois', 'Chicago', '128 Lake St',  
'57812', 'James'),  
('8', 'Home', '1', 'USA', 'Texas', 'Arlington', '404 st',  
'25584', 'Joseph'),  
('9', 'Home', '1', 'USA', 'California', 'Los Angeles', '179  
Columbia St', '10113', 'Art'),  
('10', 'Home', '1', 'USA', 'New Jersey', 'Jersey City', '278  
Columbus St', '078078', 'Lenna'),  
('11', 'Home', '1', 'USA', 'New York', 'Long Island', '689  
Bleeker St', '56574', 'Don'),  
('12', 'Work', '1', 'USA', 'New Jersey', 'Newark', '328 Martin  
St', '14758', 'Lenna');
```

```
INSERT INTO product (ProductID, Name, Price, Type, Quantity,  
Description, ImageSrc) VALUES  
('1', 'Harvard Study Desk', '150', 'Desk', '17', 'Fine wooden  
desk',  
'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSm131Svl-  
qDVCsEKzztPFTElE26D5k_FNGBQ&usqp=CAU'),  
('2', 'Adan Desk Chair', '260', 'Chair', '10', 'Fabric -  
Upholstery',  
'https://assets.pbimgs.com/pbimgs/ab/images/dp/wcm/202225/0025/l  
ayton-upholstered-swivel-desk-chair-1-c.jpg'),  
('3', 'Bella Metal Bookcase', '100', 'Bookcase', '15', 'Metal  
Bookcase', 'https://m.media-amazon.com/images/I/51iFIygR19L._AC_.  
jpg'),  
('4', 'Glass Dining Desk', '250', 'Desk', '30',  
NULL, 'https://images.furnituredealer.net/img/products%2Fsteve_si  
lver%2Fcolor%2Folson%20ss_os480db%2Bgt-b1.jpg'),
```

('5', 'Comfort Sofa Chair', '350', 'Sofa', '20', 'Yellow Color','https://target.scene7.com/is/image/Target/GUEST_3b4179f0-dfa5-4f25-9d51-375f6bc17391?wid=488&hei=488&fmt=pjpeg'),
('6', 'Wooden Office Bookcase', '80', 'Bookcase', '50', 'Strong','<https://secure.img1-fg.wfcdn.com/im/03337342/compr-r85/8975/89755064/glastonbury-84-h-x-45-w-solid-wood-standard-bookcase.jpg>'),
('7', 'Amsterdam Visitor Chair', '300', 'Chair', '25', 'Comfort at its best in USA','https://www.humanscale.com/userFiles/images/seating/freedomheadrest/17_humanscale_freedom_headrest_chair_prod1.jpg'),
('8', 'Zillow Glass Bookcase', '150', 'Bookcase', '10', 'Handle with Care', 'https://images.thdstatic.com/productImages/dad2f276-6be1-43f8-9d0e-8f0308541d16/svn/white-home-decorators-collection-bookcases-bookshelves-js-3424-a-31_600.jpg'),
('9', 'Tough Metal Desk', '210', 'Desk', '14', NULL,'[https://img.uline.com/is/image/uline/H-5685BL?Mobile_Zoom\\$](https://img.uline.com/is/image/uline/H-5685BL?Mobile_Zoom$)');

INSERT INTO `transaction` (`TransactionID`, `TStatus`, `TDate`, `TotalAmount`, `CardNumber`, `CID`) VALUES
('1', 'Completed', '2022-12-06', '230', '123123123123', '1'),
('2', 'Completed', '2022-11-23', '350', '147147147147', '4'),
('3', 'Completed', '2022-09-07', '100', '148148148148', '9'),
('4', 'Completed', '2022-02-09', '180', '321321321321', '8'),
('5', 'Completed', '2022-06-04', '410', '123123123123', '1'),
('6', 'Completed', '2022-04-18', '360', '123123123123', '1'),
('7', 'Completed', '2022-02-11', '80', '369369369369', '3'),
('8', 'Completed', '2021-11-03', '500', '456456456456', '2'),
('9', 'Completed', '2021-11-08', '350', '368368368368', '7'),
('10', 'Completed', '2021-08-17', '210', '987987987987', '6'),
('11', 'Completed', '2022-08-09', '80', '852852852852', '10'),
('12', 'Completed', '2022-09-14', '250', '741741741741', '11'),
('13', 'Completed', '2022-03-16', '150', '654654654654', '7');

```
INSERT INTO `cart` (`CartID`, `TransactionID`, `Status`, `Date`,  
`TotalAmount`, `ShipAddName`, `CID`) VALUES  
('1', '1', 'Order Placed', '2022-12-06', '230', 'Home', '1'),  
('2', '2', 'Delivered', '2021-11-23', '350', 'Home', '4'),  
('3', '3', 'Delivered', '2022-07-29', '100', 'Work', '3'),  
('4', '4', 'Delivered', '2022-02-09', '180', 'Home', '8'),  
('5', '5', 'Delivered', '2022-06-04', '410', 'Home', '1'),  
('6', '6', 'Delivered', '2022-04-18', '360', 'Home', '1'),  
('7', '7', 'Delivered', '2022-02-11', '80', 'Work', '3'),  
('8', '8', 'Shipped', '2022-11-03', '500', 'Home', '2'),  
('9', '9', 'Delivered', '2021-11-08', '350', 'Home', '7'),  
('10', '10', 'Delivered', '2021-08-17', '210', 'Work', '6'),  
('11', '11', 'Delivered', '2022-08-09', '80', 'Home', '10'),  
('12', '12', 'Out for Delivery', '2022-10-14', '250', 'Home',  
'4'),  
('13', '13', 'Delivered', '2022-03-16', '150', 'Home', '7');
```

```
INSERT INTO `appears_in` (`CartID`, `ProductID`, `Quantity`,  
`PriceSold`) VALUES  
('1', '1', '1', '150'),  
('1', '6', '1', '80'),  
('2', '5', '1', '350'),  
('3', '3', '1', '100'),  
('4', '7', '1', '100'),  
('4', '6', '1', '80'),  
('5', '1', '1', '150'),  
('5', '2', '1', '260'),  
('6', '9', '1', '210'),  
('6', '1', '1', '150'),  
('7', '6', '1', '80'),  
('9', '5', '1', '350'),  
('8', '1', '1', '150'),  
('8', '5', '1', '350'),  
('10', '9', '1', '210'),  
('11', '6', '1', '80'),  
('12', '4', '1', '250'),  
('13', '7', '1', '150');
```

```
INSERT INTO `offer_product` (`ProductID`, `OfferPrice`) VALUES  
( '5', '300' ),  
( '1', '100' );
```

```
INSERT INTO `book_cases` (`ProductID`, `Material`, `Shelves`) VALUES  
( '3', 'Metal', '4' ),  
( '6', 'Wood', '6' ),  
( '8', 'Glass', '3' );
```

```
INSERT INTO `desks` (`ProductID`, `Material`, `Drawers`) VALUES  
( '1', 'Wood', '4' ),  
( '4', 'Glass', '0' ),  
( '9', 'Metal', '3' );
```

```
INSERT INTO `chairs` (`ProductID`, `Fabric`, `Type`) VALUES  
( '2', 'Upholestry', 'Desk' ),  
( '7', 'Linen', 'Visitor' ),  
( '5', 'Velvet', 'Sofa' );
```

SQL commands for sales statistics:

1. For a given time period (begin date and end date) compute the most frequently sold products.

```
SELECT P.PName, SUM(CT.Quantity) as total
From product P, Appears_in CT, cart C
WHERE P.ProductID = CT.ProductID AND CT.CartID = C.CartID AND
C.Date > '2022-12-01' AND C.Date < '2023-12-01'
GROUP BY P.ProductID
ORDER BY total DESC
```

2. For a given time period (begin date and end date) compute the products which are sold to the highest number of distinct customers.

```
SELECT P.ProductID, COUNT(DISTINCT C.cid) AS customer_count
FROM product P, Appears_In CT, Cart C
WHERE P.ProductID = CT.ProductID AND CT.CartID = C.CartID AND
C.Date > '2021-12-01' AND C.Date < '2023-12-01'
GROUP BY P.ProductID
ORDER BY customer_count DESC
LIMIT 10
```

3. For a given time period (begin date and end date) compute the 10 best customers (in terms of money spent) in descending order.

```
SELECT C.Fname, SUM(CT.TotalAmount) as total
FROM customer C, cart CT, Transaction T
WHERE C.CID = CT.CID AND CT.Date > '2022-12-01' AND CT.Date <
'2023-12-01' AND T.TStatus = "Completed"
GROUP BY C.CID
ORDER BY total DESC
LIMIT 10
```

4. For a given time period (begin date and end date) compute the 5 best zip codes (in terms of shipments made).

```
SELECT SA.ZIP, COUNT(SA.ZIP) as total
FROM transaction T
JOIN cart C
    ON T.TransactionID = C.TransactionID
JOIN ship_address SA
    ON C.CID = SA.CID AND C.ShipAddName = SA.SAName
WHERE T.TDate > '2022-12-01' AND T.TDate < '2023-12-01' AND
T.TStatus = 'Completed'
GROUP BY SA.ZIP
ORDER BY total DESC
LIMIT 5
```

5. For a given time period (begin date and end date) compute the average selling product price per product type for desks, chairs and bookcases.

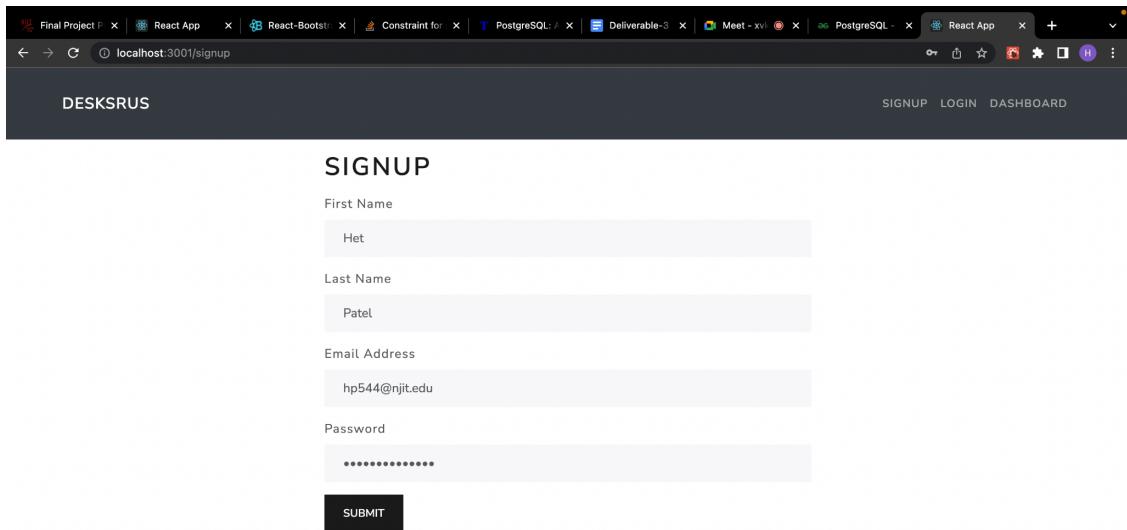
```
SELECT P.type, AVG(CT.PriceSold) as total
FROM product P, Appears_in CT, cart C
WHERE P.ProductID = CT.ProductID AND CT.CartID = C.CartID AND
C.Date > '2022-12-01' AND C.Date < '2023-12-01'
GROUP BY P.type
```

Source code:

<https://github.com/phet2309/Projects>

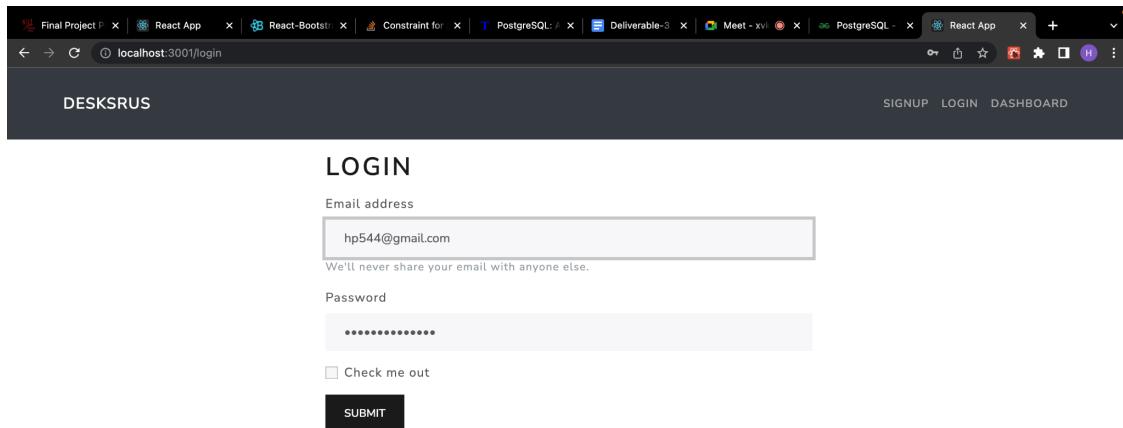
Application screenshots:

Sign-up page



The screenshot shows a web browser window with multiple tabs open at the top. The active tab is labeled "localhost:3001/signup". The page itself has a dark header with the text "DESKSRUS" on the left and "SIGNUP" "LOGIN" "DASHBOARD" on the right. Below the header, the word "SIGNUP" is centered in a large, bold font. There are four input fields: "First Name" containing "Het", "Last Name" containing "Patel", "Email Address" containing "hp544@njit.edu", and "Password" containing a series of asterisks. A black "SUBMIT" button is located below the password field.

Login Page



The screenshot shows a browser window with the URL `localhost:3001/login`. The page has a dark header with the text "DESKSRUS" on the left and "SIGNUP LOGIN DASHBOARD" on the right. Below the header is a "LOGIN" section. It contains a form with fields for "Email address" (containing "hp544@gmail.com") and "Password" (represented by a series of dots). There is also a checkbox labeled "Check me out" and a "SUBMIT" button.

Transactions Page



The screenshot shows a browser window with the URL `localhost:3000/transactions`. The page has a dark header with the text "DESKSRUS" on the left and "DASHBOARD STORE ORDERS TRANSACTIONS LOGOUT" on the right. Below the header is a table showing three transaction entries:

\$230	Success	\$410	Success
\$2022-12-06T05:00:00.000Z		\$2022-06-04T04:00:00.000Z	
View More	View More	View More	View More

The Store Page

The screenshot shows the homepage of a furniture store named "DESKSRUS". The top navigation bar includes links for DASHBOARD, STORE, ORDERS, TRANSACTIONS, and LOGOUT. A "GO TO CART" button is located in the top-left corner of the main content area. Below it, there is a grid of five furniture items:

- HARVARD STUDY DESK**: \$150. View Details. ADD TO CART.
- ADAN DESK CHAIR**: \$260. View Details. ADD TO CART.
- BELLA METAL BOOKCASE**: \$100. View Details. ADD TO CART.
- GLASS DINING DESK**: \$250. View Details. ADD TO CART.
- COMFORT SOFA CHAIR**: \$350. View Details. ADD TO CART.

Below this grid, there are three more furniture items visible but not fully part of the main grid:

- A wooden bookshelf.
- A black office chair.
- A white cabinet.

The Shopping cart

The screenshot shows the shopping cart page. On the left, a sidebar displays the items currently in the cart:

- ZILLOW GLASS BOOKCASE**: Price: \$150. Total: \$150.00.

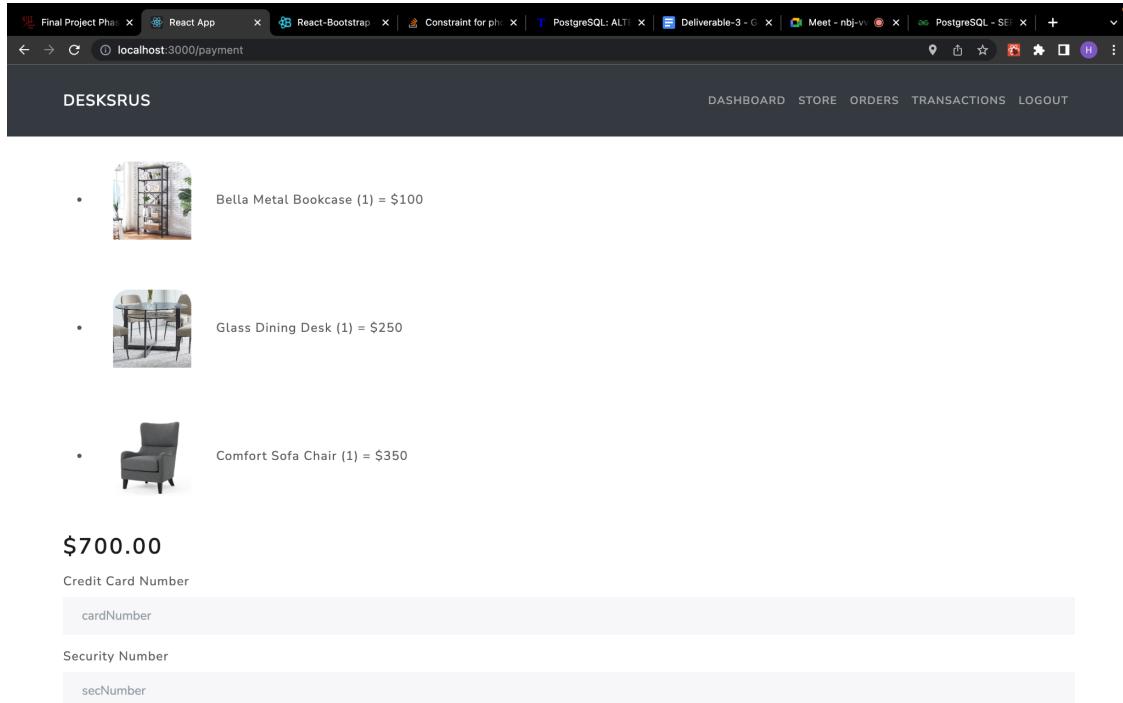
Below this, there is a quantity selector (set to 1) and a large image of the Zillow Glass Bookcase.

On the right, the main content area shows a grid of furniture items:

- DESK CHAIR**: View Details. ADD TO CART.
- BELLA METAL BOOKCASE**: \$100. View Details. ADD TO CART.
- GLASS DINING DESK**: \$250. View Details. ADD TO CART.
- COMFORT SOFA CHAIR**: \$350. View Details. ADD TO CART.

At the bottom left of the sidebar, there is a "CHECKOUT" button.

Payment Page: Adding payment details and shipping details

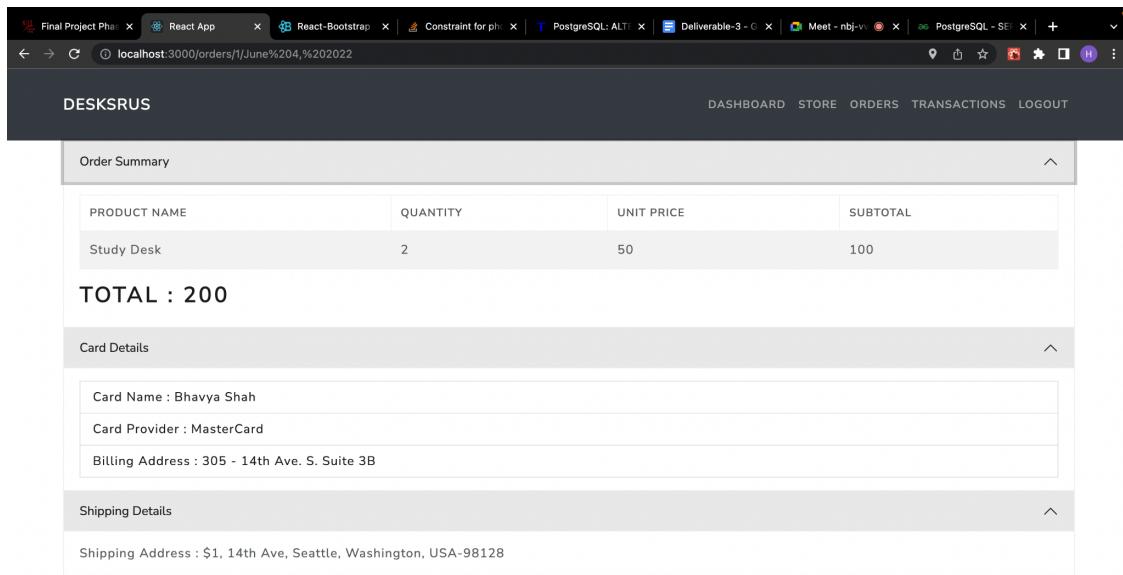


The screenshot shows a payment page for an order. At the top, there's a navigation bar with tabs for DASHBOARD, STORE, ORDERS, TRANSACTIONS, and LOGOUT. Below the navigation, the store name "DESKSRUS" is displayed. The main content area lists three items with their descriptions and prices:

- Bella Metal Bookcase (1) = \$100
- Glass Dining Desk (1) = \$250
- Comfort Sofa Chair (1) = \$350

The total amount is displayed as **\$700.00**. Below the total, there are input fields for Credit Card Number and Security Number.

Order Placed Successfully



The screenshot shows a confirmation page for an order placed successfully. The URL in the browser is `localhost:3000/orders/1/June%204,%202022`. The page has a similar navigation bar as the previous screen. The main content is divided into sections:

- Order Summary:** A table showing the product name, quantity, unit price, and subtotal for a Study Desk (2 units at \$50 each).
- TOTAL : 200**
- Card Details:** Fields for Card Name (Bhavya Shah), Card Provider (MasterCard), and Billing Address (305 - 14th Ave. S. Suite 3B).
- Shipping Details:** Shipping Address (1, 14th Ave, Seattle, Washington, USA-98128).

Sales Statistics

1.

```
deskrus5# 
deskrus5# 
deskrus5# SELECT P.Name, SUM(CT.Quantity) as total
deskrus5# From product P, Appears_In CT, cart C
deskrus5# WHERE P.ProductID = CT.ProductID AND CT.CartID = C.CartID AND C.Date > '2021-12-01' AND C.Date < '2023-12-01'
deskrus5# GROUP BY P.ProductID
deskrus5# ORDER BY total DESC;
      name          total
-----|-----
Wooden Office Bookcase |    4
Harvard Study Desk   |    4
Amsterdam Visitor Chair |  2
Tough Metal Desk     |    1
Global Chair Desk    |    1
Comfort Sofa Chair   |    1
Adam Desk Chair      |    1
Bella Metal Bookcase |    1
(8 rows)
deskrus5# 
```

2.

```
deskrus5# SELECT P.ProductID, COUNT(DISTINCT C.cid) AS customer_count
deskrus5# FROM product P, Appears_In CT, Cart C
deskrus5# WHERE P.ProductID = CT.ProductID AND CT.CartID=C.CartID AND C.Date > '2021-12-01' AND C.Date < '2023-12-01'
deskrus5# GROUP BY P.ProductID
deskrus5# ORDER BY customer_count DESC
deskrus5# LIMIT 10;
productid | customer_count
-----|-----
       6 |        4
       7 |        2
       1 |        2
       4 |        1
       5 |        1
       9 |        1
       2 |        1
       3 |        1
(8 rows)
deskrus5# 
```

3.

```
deskrus5# SELECT C.Fname, SUM(CT.TotalAmount) as total
deskrus5# FROM transaction T, cart CT, transaction C
deskrus5# WHERE C.CID = CT.CID AND CT.Date > '2022-12-01' AND CT.Date < '2023-12-01' AND T.TStatus = 'Completed'
deskrus5# GROUP BY C.CID
deskrus5# ORDER BY total DESC
deskrus5# LIMIT 10;
      fname          total
-----|-----
Bhavya |  2998
(1 row)
deskrus5# 
```

4.

```
deskrus5# SELECT SA.ZIP, COUNT(SA.ZIP) as total
deskrus5# FROM transaction T
deskrus5# JOIN cart CT ON T.TransactionID = CT.TransactionID
deskrus5# JOIN ship_address SA ON C.CID = SA.CID AND C.ShipAddName = SA.SAName
deskrus5# WHERE T.TDate > '2021-12-01' AND T.TDate < '2023-12-01' AND T.TStatus = 'Completed'
deskrus5# GROUP BY SA.ZIP
deskrus5# ORDER BY total DESC
deskrus5# LIMIT 5;
      ZIP          total
-----|-----
98128 |    3
3230  |    2
4066  |    2
25584 |    1
57812 |    1
(5 rows)
deskrus5# 
```

5.

```
deskrsus5=# SELECT P.type, AVG(CT.PriceSold) as total
deskrsus5# FROM product P, Appear_in CT, cart C
deskrsus5# WHERE P.ProductID = CT.ProductID AND CT.CartID = C.CartID AND C.Date > '2021-12-01' AND C.Date < '2023-12-01'
deskrsus5# GROUP BY P.type;
   type |      total
-----+-----
  Desk  | 176.666666666666667
Bookcase | 84.00000000000000000
Chair   | 215.000000000000000
(3 rows)

deskrsus5#
```