

# Group 20: CloakedCommerceDB

Owen McDaniel  
University of Idaho  
USA

mcdm0107@vandals.uidaho.edu

Braydyn Proctor  
University of Idaho  
USA

proc3080@vandals.uidaho.edu

## ABSTRACT

This document details a project called CloakedCommerceDB, a database-driven website designed to organize anonymous exchanges between two pairs of users. This project is part of the curriculum for the CS360 class at the University of Idaho.

A key focus for this project will be maintaining anonymity for all users of the system, which will be implemented with hashing. The final result of this paper will be a fully functional website using MySQL to trade between other users.

## CCS CONCEPTS

• **Information systems** → Service discovery and interfaces; RESTful web services; Query representation; *Query intent*; **Query reformulation**; **Query languages for non-relational engines**; Semi-structured data; **Middleware for databases**; **Database query processing**; **Query languages**; • **Human-centered computing** → *Human computer interaction (HCI)*.

## KEYWORDS

Schema Graph, Biological Databases, Data Integration, Ad hoc Querying, Schema Abstraction, Query Reformulation.

### ACM Reference Format:

Owen McDaniel and Braydyn Proctor. 2025. Group 20: CloakedCommerceDB. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23)*, March 27-March 31, 2023, Tallinn, Estonia. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3555776.3577652>

## 1 INTRODUCTION

CloakedCommerceDB is an anonymous bartering platform designed to facilitate the exchange of goods and services without revealing users' identities. In our current time, we face a new issue: the issue of privacy and security. Many websites that offer bartering platforms do not have safeguards for protecting their users' identities. CloakedCommerceDB addresses this issue by utilizing a 16-digit hash key authentication system to ensure complete anonymity during these transactions. Unlike other marketplaces, this system allows users to post, browse, and trade items while safeguarding their personal information.

The platform will be accessible in any web browser through our carefully constructed website, designed using vanilla HTML, CSS,

Bootstrap, and JavaScript to provide a consistent user experience across all devices. Our back end is powered by Node.js with Express, enabling us to create a secure and efficient data processing system. This, paired with our encrypted MySQL databases, provides a reliable and secure platform.

Overall, CloakedCommerceDB will provide a secure platform for trading that redefines the standards followed by similar platforms.

## 2 E.R DIAGRAM

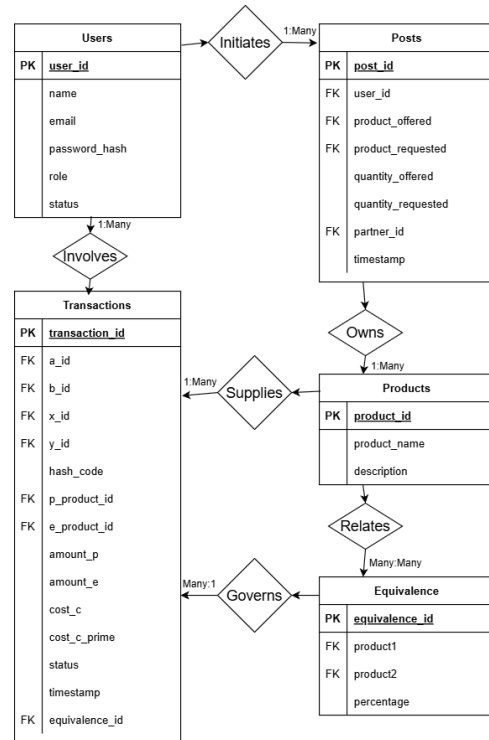


Figure 1: ER Diagram

Our ER diagram outlines the logical structure of the CloakedCommerceDB system, illustrating how data is managed and communicated for anonymous bartering. It includes five key entities: Users, Transactions, Products, Posts, and Equivalence. The Users entity contains primary key user\_id, username, and role, defining each user's role in the barter process such as a trader or admin. The Transactions entity captures details of each exchange with foreign keys a\_id, b\_id, x\_id, and y\_id referencing Users, p\_product\_id and e\_product\_id linking to Products, plus attributes hash\_code, amount\_p, amount\_e, cost\_c\_prime, cost\_c\_double\_prime, and equivalence\_id.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SAC '23, March 27-March 31, 2023, Tallinn, Estonia

© 2025 ACM.

ACM ISBN 978-1-4503-9517-5/23/03

<https://doi.org/10.1145/3555776.3577652>

The Products entity includes primary key `product_id`, `product_name`, very basic MySQL implementation to show understanding of database design and querying. The report for the CloakedCommerceDB will be four pages long with more details of implementation.

and description, representing items available for barter, with transfer costs `cost_c_prime` and `cost_c_double_prime` tracked in Transactions. The Posts entity with primary key `post_id` links to Users via `user_id` and `partner_id`, and to Products via `product_offered` and `product_requested`, capturing barter offers and requests. The Equivalence entity with primary key `equivalence_id`, `product1` and `product2` as foreign keys to Products, and percentage forms the equivalence ratio for product value comparisons. Relationship sets include Users-Transactions as one-to-many since one user can have many transactions, Products-Transactions as one-to-many because one product can be in many transactions, Users-Posts as one-to-many where one user can create many posts, Products-Posts as one-to-many since one product can appear in many posts, and Equivalence-Products as many-to-many because multiple products can have various equivalences.

### 3 TOOLS

Our main development tool for both the front-end and back-end will be JavaScript. HTML & JS will be used for the user interface and frontend. TO help with the back end of the website, we will be using Node.js which will allow us to use JavaScript for connecting to the database aspect. The barter system is designed without real-world payment processing to maintain complete anonymity. CloakedCommerceDB will be using MySQL for the database management system in order to securely store and manage data from users.

For working on the project we will be using Visual Studio Code. VSCode has built-in integration for Git/Github as well as numerous other extensions to ease the programming process, which makes it a fantastic choice for this project. VSCode also includes automatic formatting support for HTML, Php, MySQL, and JavaScript which significantly impacts the time it takes to implement.

For version control, we will be using Git and GitHub so both partners can work on the CloakedCommerceDB in a standardized environment.

### 4 PLAN / TIMELINE

Development of CloakedCommerceDB will be done in phases with updated documentation and presentations to demonstrate progress made in implementation and understanding. This project will continue through the course of the University of Idaho Spring 2025 semester and will end on May 1st with a completed project.

#### 4.1 Phase I - February 25th

By February 25, we will have completed two pages of documentation which will include an ER diagram with an explanation of database design. The paper will also include an implementation methodology with details of the tools that will be used for individual parts of the development.

#### 4.2 Phase II - March 21st

By March 21, we will have a working prototype for the CloakedCommerceDB website, with the front-end mostly implemented to demonstrate the implemented functionality. This will also include

### 4.3 Phase II - Results

For our Phase 2 Demo we were able to implement everything we wanted. The front end is heavily developed. We have implemented most of the required pages with minor issues. We changed from react to a basic HTML CSS and bootstrap front end for ease of use. Currently for our back-end development, we are using Node.js /w Express and modules for different libraries that will be discussed later.

We have created a backend SQL database with "dummy data" that is for testing of products. The database takes into account any information that is necessary to save and follows the configuration of the ER diagram. We have a fully implemented Register & Login system, all information is stored within the database's "Users" table. The password is encrypted via Bcrypt which generates a secure hashed password, and it does not showcase the password in plain-text anywhere.

There is a product posting page where users can enter their partners' hash code (a unique user ID that is only known by your partner(s) ). This is the space where you can request trades and input what you are offering.

We have an admin panel for users that are marked as an admin in their role, this is done via code & verification through the database. The backend uses authorization routing to change HTML pages in order to have different features. In this deliverable due date the largest struggle was learning how to do back-end routing for sessions, and making the HTML of the normal site change when a user is logged in, versus when they are not logged in.

#### 4.4 Phase III - April 18th

By April 18, a fully functioning prototype will be available with all intended functional features. The prototype will include all the trading interfaces and feature a fully implemented front end. The phase III report will be six pages long further detailing the CloakedCommerceDB implementation.

#### 4.5 Phase IV - May 1st

By May 1st, the fully working final product will be released and the final touches for the product will be implemented. The system should be fully functional with no large errors or bugs for all systems, including administrator tools, account creation, and database queries. There will be an eight-page document specifying implementation of our trading system with encountered challenges being detailed.

### 5 CHALLENGES

#### 5.1 Anticipated Challenges

While Developing the CloakedCommerceDB system we anticipate several challenges, primarily in ensuring anonymity, cross-platform compatibility, data security and accurate cost simulations. To maintain anonymity we will use a 16-digit hash key for authentication anonymously, this will ensure user identities are protected during

(simulated) transactions. For Cross-Platform compatibility to work properly we will use vanilla HTML paired with Bootstrap in order to keep a consistent user interface and experience across any device.

Ensuring data security while maintaining integrity for these transactions is another challenge, we will be using MySQL with encryption methods in order to securely manage user information. Finally accurately simulating barter exchanges and cost adjustments will require using an equivalence table (T) for value comparisons and the Node.JS backed logic in order to perform constant real-time calculations to simulate real prices changing for bartering. Overall these solutions will help maintain anonymity, security, and fair value simulations.

## 5.2 Encountered Challenges

We have encountered various challenges while developing this portion. The first major challenge involved understanding the issue we are trying to solve. It took some time to dissect what we need for the project and what we shouldn't exactly have in it. The second Challenge came when having the foreign key constraints whilst creating barter listings. Originally the system required users to input their partners userID directly. However this would defeat the purpose of remaining anonymous as someone could easily tell who they were trading with. So we created a partner hash code for a unique identifier that only the partner knows.

We had first tried using React.js for the frontend, but it seemed like too unnecessary for what we were trying to accomplish. It would've worked, but it was adding complexity that wasn't really needed. We were already a lot more used to working with regular HTML, CSS, Bootstrap, and JavaScript, so it made more sense to just stick with that. It ended up saving us a lot of time and made things much easier to test and change as we went. Allowing us to get more done and streamline the whole process.

Dealing with these issues along the way helped the project come together more. It pushed us to work harder on making the platform secure, keeping user info anonymous, and making sure everything worked the way it should.

## 6 USER INTERFACE

### 6.1 Homepage & Navbar

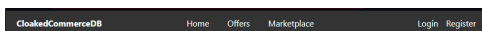


Figure 2: Navigation Bar

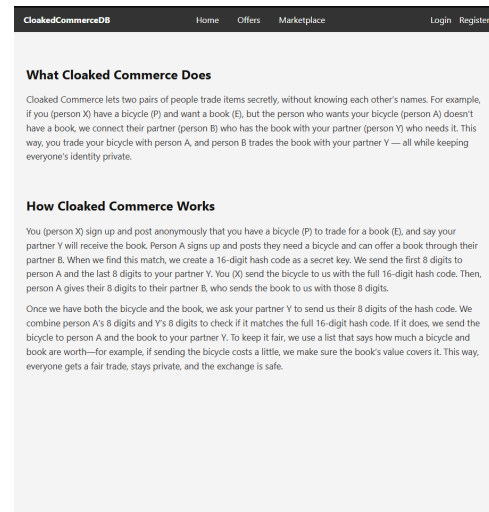


Figure 3: Home Page

The homepage links to the backend through a session check. If the user is logged in, the navigation bar will dynamically show their name and a profile page via a dropdown. If not logged in the homepage will prompt the login and registration options. This check is performed using a fetch call to `/auth/session`

### 6.2 Login / Registration

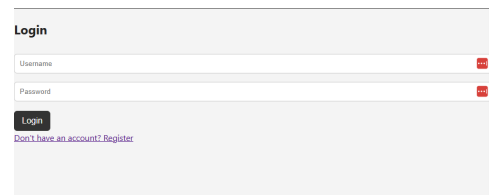


Figure 4: Login Page

The login and registration pages submit form data to the backend through POST requests to `/auth/login` and `/auth/register`. Registration hashes the user's password with bcrypt and then generates a 16-character anonymous hash identification key for users to share with their known partner. After a successful login, the session is initialized, and the user is redirected to the homepage. The logout button terminates the session as well.

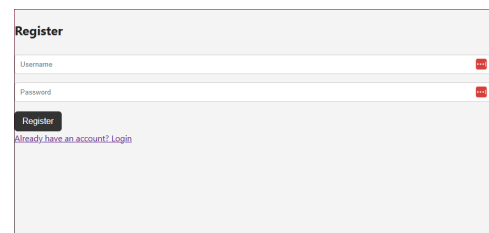


Figure 5: Registration Page

### 6.3 Profile Page

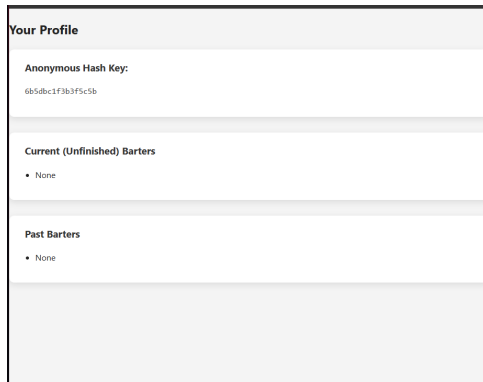


Figure 6: Profile Page

The profile page grabs the personal data using a GET request to /dashboard. This includes the user's anonymous id key, Username and role. This page is protected and only accessible if the session is valid, if the user attempts to access without a valid session it will return them to the login page.

### 6.4 Listings

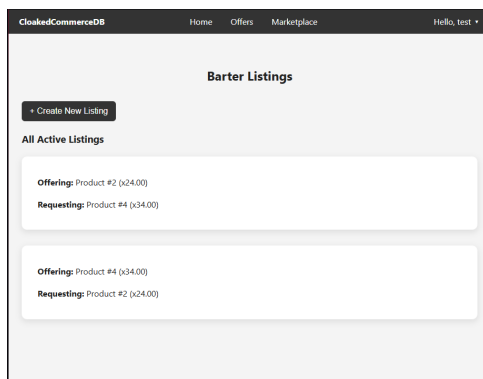


Figure 7: Listings Page

The listing page communicates with the backend to perform several API calls. It fetches the products data from /barter/products, loads the existing unfulfilled listings from /bater/postsm and submits new listings with POST to /barter/posts. Users create listings by submitting a partner's anonymous hash key, which the backend uses to figure out the user ID prior to storing the posts.

### 6.5 Admin Panel

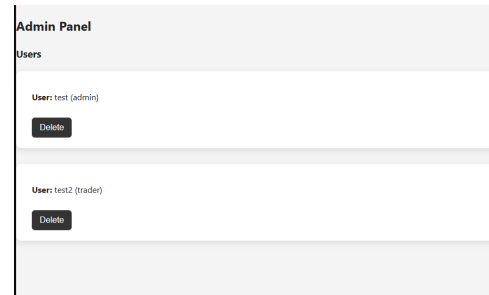


Figure 8: Admin Panel

The admin panel is a protected section only accessible to administrators. This role is designated within the code, and verified in the database. This allows for protection of user identities, codes, trades etc. If a user has the role of admin they will be able to access this via the Admin section on the dropdown. It accesses the backend via set routes. To remove a user it uses /admin/remove in the future you will be able to view peoples trades, and add products via this panel using the admin route /admin/verifyprod. This will allow regular users to add their product and admins to view them and verify worth and update the equivalence table.

## 7 CONCLUSION

Overall, CloakedCommerceDB is a system designed to match users looking to exchange products and services while maintaining their own anonymity. CloakedCommerceDB will be accessible through a user-friendly website which provides a seamless and efficient platform to make transactions between individuals who wish to exchange goods.

The users of the system can easily view their postings or transactions by logging into their verified accounts. From within their account, they can view and manage posts they have made, track any ongoing transactions, and review their account history. Admins can log into the website and view and manage current postings on the website. Admins are also responsible for approving new account-creation requests in order to ensure only real people can create accounts.

Overall, CloakedCommerceDB will offer a secure and anonymous platform for users to exchange goods efficiently and privately. With this focus on user's anonymity, security, and ease of use. It has the potential to change the way bartering is done by providing a trusted and straightforward process.

## REFERENCES

<https://github.com/B-Proctor/ClockedCommerceDB>