Barter Bin

BARTER you will. TRADE you must!

Written By: Arnetta Knight

Changes and Updates

The Minimum Viable Product or MVP for the Barter Bin web application is a simple, yet functional platform that will allow its users to post items and services for trade, while expressing their own needs, and matching individuals based on these indicators. This can be done in large part by implementing the core features of user registration and login and the search and match function. Users will be matched based on their wants and offers, and they can search and expand their matches based on their location.

The change that will be implemented is the Ratings and Review table (or section) of Barter Bin will now be moved to a new category called a “stretch feature”. It has been determined that the core functionality to support the MVP can be launched without the reviews feature. Users will still be able to successfully use the web application without the reviews function if time constraints, additional resources, and effort to implement come into play. If there are road blocks this feature can still be added at a later time as a future release which would allow users to rate and review their experiences on the platform.

Designating ratings and reviews as a stretch feature will help to prioritize the core functionality of the application to ensure a successful launch of Barter Bin while still allowing for future improvements and enhancements.

Database Design Overview

To barter means to trade goods and services for other goods and services. Barter Bin will be a unique web application that allows its users to trade their goods and services with each other without the need of cash or other currency. For this project, I will be utilizing a MySQL database. I feel this is a suitable choice for the web application based on the nature of the data that it will be managing.

MySQL is a very popular relational database management system or RDBMS that provides great flexibility, scalability, and performance. It's designed to manage large amounts of data efficiently and effectively and it's commonly used for web applications that require reliable and quick data storage and retrieval. Regarding the Barter Bin web application, MySQL can support the complex transactions and queries needed to handle trades between users, updating trade statuses, and ensuring the consistency of data. MySQL's ability to handle concurrent transactions is also important because multiple users will be accessing the system at the same time.

Brief overview of how the database will interact with the PHP service layer

The database will interact with PHP in many ways to include data insertion and modification, data retrieval, validation and sanitization of data, authentication, and authorization. The PHP service layer will act as the "middle man" between the MySQL database and the front-end user interface. It will return the appropriate response to the user interface to aid in providing an efficient and seamless user experience.

Specifications of Data

The Barter bin MySQL database design and schema will define the fields, tables, and relationships between them. Below is a high-level overview of how this data will be managed:

1. **USERS table:**

```
CREATE TABLE users (
  user_id INT PRIMARY KEY,
  first_name VARCHAR (50) NOT NULL,
  last_name VARCHAR (50) NOT NULL,
  email VARCHAR (255) UNIQUE NOT NULL,
  phone VARCHAR (20),
  address VARCHAR (255),
  city VARCHAR (50),
  state VARCHAR (50),
  zip_code VARCHAR (20),
  password VARCHAR (20) NOT NULL
);
```

The above schema for the users table will be able to manage and store the necessary data for the Barter Bin web app including login credentials and contact information.

2. **ITEMS table:**

```
CREATE TABLE items (
  item_id INT PRIMARY KEY,
  user_id INT,
  title VARCHAR (255) NOT NULL,
  description TEXT,
  image_url VARCHAR (255),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

For the items table schema, I will associate each item listed within the Barter Bin web application with a specific user in the users table.

3. **TRADES table:**

```
CREATE TABLE trades (
  trade_id INT PRIMARY KEY,
  user_id_1 INT,
  item_id_1 INT,
  user_id_2 INT,
  item_id_2 INT,
  status VARCHAR (50),
  initiated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id_1) REFERENCES users(user_id),
  FOREIGN KEY (user_id_2) REFERENCES users(user_id),
  FOREIGN KEY (item_id_1) REFERENCES items(item_id),
  FOREIGN KEY (item_id_2) REFERENCES items(item_id));
```

For the trades table schema, user_id_1 is designated for the user that will be initiating the trade and stored as the foreign key referencing the user id field in the users table. Item_id_1 is the id of the item for the trade initiator, and it will reference the items table. Conversely, user_id_2 will be the one receiving the trade offer and the item id associated with this user will the item being offered in exchange. The status data field will identify the status of the trade (pending, accepted, or declined). The "initiated_at" data field will be the timestamp of when the trade was initiated.

4. **[Stretch Feature] REVIEWS table:**

```
CREATE TABLE reviews (
  review_id INT PRIMARY KEY AUTO_INCREMENT,
  user_id INT,
  rating INT,
  comment TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(user_id));
```

In the schema for the reviews table the user id is the id of the user writing the review, stored as a foreign key that references the user's table. A rating out of 5 stars will be given by the user that is writing the review and stored as text. The created_at data field is a timestamp of when the review was submitted. Overall, this table will be able to associate each review with a relevant user. Note that the reviews section is designated as a stretch feature and may be added as future release.

Purpose, Implementation, and User Interactions

- **The Users Table:**
  - **Purpose** – The users table will store data about registered users. The data fields included in this table will be the first name, last name, email, phone, address, city, state, zip code, and password. This data will be used to authenticate and authorize users, display profiles and the information associated with it, and match users with other users and the items they are searching for.
  - **Implementation** – The users table will be implemented as a relational table with fields for each data attribute of a user. The email column will be set as a unique index to ensure that each user has a unique email address. The password column will be encrypted and hashed. When someone signs up on Barter Bin, their user data will be inserted into this table as a new row.

    - **User Interactions** – Users will interact with the "users table" when they sign up to use the Barter Bin application, update their profile information, or search for matches based on location and item criteria.

- **The Items Table:**
    - **Purpose –** The items table will store information about the items that users have listed for trade to include the title, description, image link, "created at" timestamp, and the ID of the user who posted the items. The data is used to display the list of items that a user has to offer for trade and its used to match users with the items that they need.
    - **Implementation –** The items table will be implemented as a relational table with fields for each attribute of an item. The data includes the item id primary key, the user id that references the user id from the users table, item title, description, image link, and the created at column. The created at data attribute will automatically insert the current date timestamp for when a new item was added. When a user lists an item for trade, a new row will be inserted into the table with all relevant data.
    - **User Interactions** – User interaction of this table will occur when searching for items that are needed, to list items for trade, and to initiate a trade with another user.

- **The Trades Table:**
    - **Purpose** – The trades table will store data about individual trades between users to include the id of the items and users involved, and the status of the trade (pending, accepted, or declined). The trades data will be used to track the progress of all trades and ensure that both trading partners are fulfilling their part of the agreement.
    - **Implementation** – The trades table will be implemented relationally like the other data tables mentioned previously. It will contain the trade id which also serves as the primary key (an auto-integer). Each row within the table will represent a single trade between two users and contain columns for user id 1, item id 1, user id 2, item id 2, status, and the "initiated at" timestamp which will automatically insert a current date timestamp for when the trade process begins.
    - **User Interactions** – An individual will interact with the trades table when they start or accept a trade, update the status of a trade, or view their trade history.

- **[Stretch Feature] The Reviews Table:**
    - **Purpose** – The reviews table will store data and information about each review left by a registered user to include the user id, rating (1-5), comment, and the auto-generated timestamp of when the review was submitted. The data is used to display user reviews and ratings on Barter Bin and to calculate the average rating of the website.

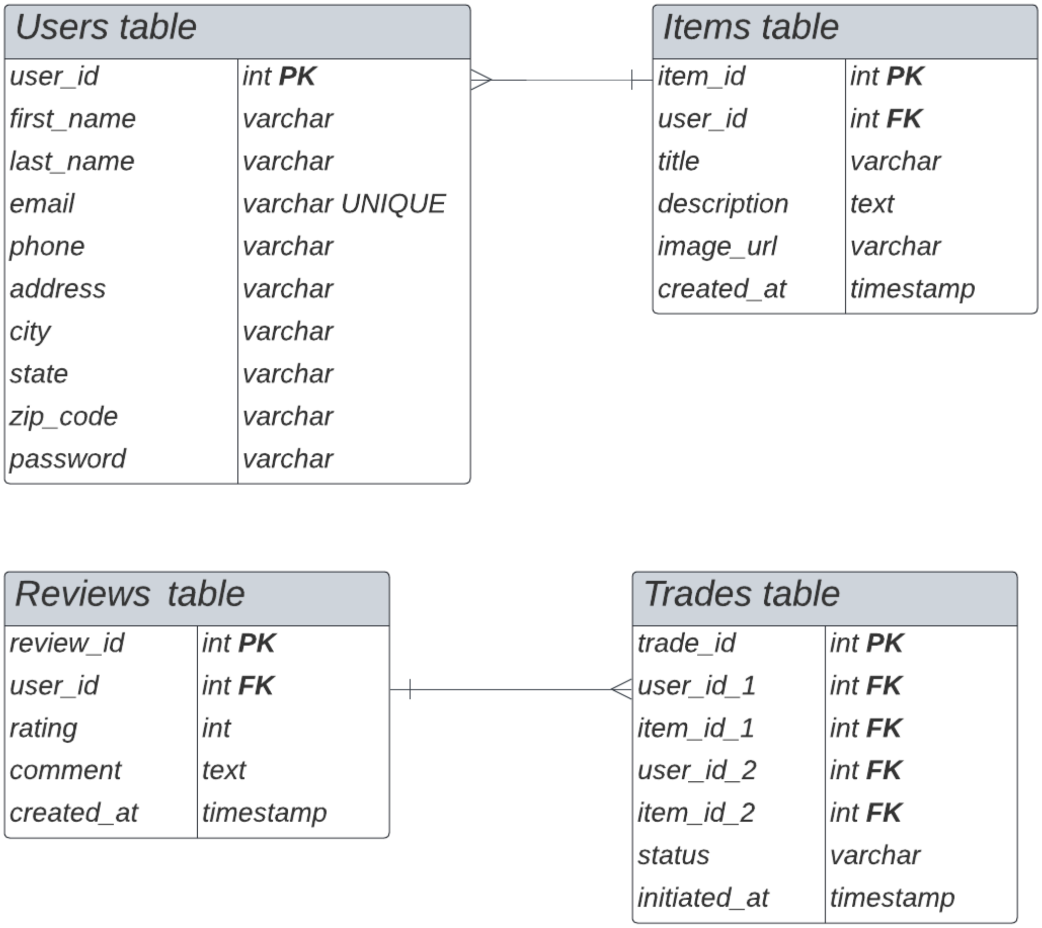
- **Implementation** – The reviews table will be implemented relationally with a review id primary key. Each row within the table will represent a single review left by a registered user and will contain columns for the user id (which references the id in the users table), rating, comment, and the "created at" auto-generated timestamp. The ratings column will be an integer between 1 and 5, representing the number of "stars" given by a user. A timestamp will be automatically inserted into the "created at" column once a review is submitted.
- **User Interactions** – An individual will interact with the reviews table when they write or leave a review, view their own reviews, and view the reviews left by other registered users.

Entity-Relationship (ER) Diagram

In the ER diagram below, the relationships between the tables are indicated by lines connecting the tables. This diagram provides a more detailed view of the relationships between the various tables in the Barter Bin database. It also helps to illustrate how the data is stored and related to each other. The following relationships are represented:

- Each item in the items table is associated with a single user within the users table. This is represented by the user id foreign key in the items table.
- Each trade in the trades table will involve two users and two (or possibly more) items represented by the user_id_1, item_id_1, user_id_2, and item_id_2. It's associated with a trade status and an initiation stamp.
- Each user review in the reviews table is associated with a single user from the users table which is represented by the user_id foreign key in the reviews table. Note that the reviews section is designated as a stretch feature and may be added as future release.

Barter Bin ER Diagram



Users table	
user_id	int **PK**
first_name	varchar
last_name	varchar
email	varchar UNIQUE
phone	varchar
address	varchar
city	varchar
state	varchar
zip_code	varchar
password	varchar

Items table	
item_id	int **PK**
user_id	int **FK**
title	varchar
description	text
image_url	varchar
created_at	timestamp

Reviews table	
review_id	int **PK**
user_id	int **FK**
rating	int
comment	text
created_at	timestamp

Trades table	
trade_id	int **PK**
user_id_1	int **FK**
item_id_1	int **FK**
user_id_2	int **FK**
item_id_2	int **FK**
status	varchar
initiated_at	timestamp

[Stretch Feature]