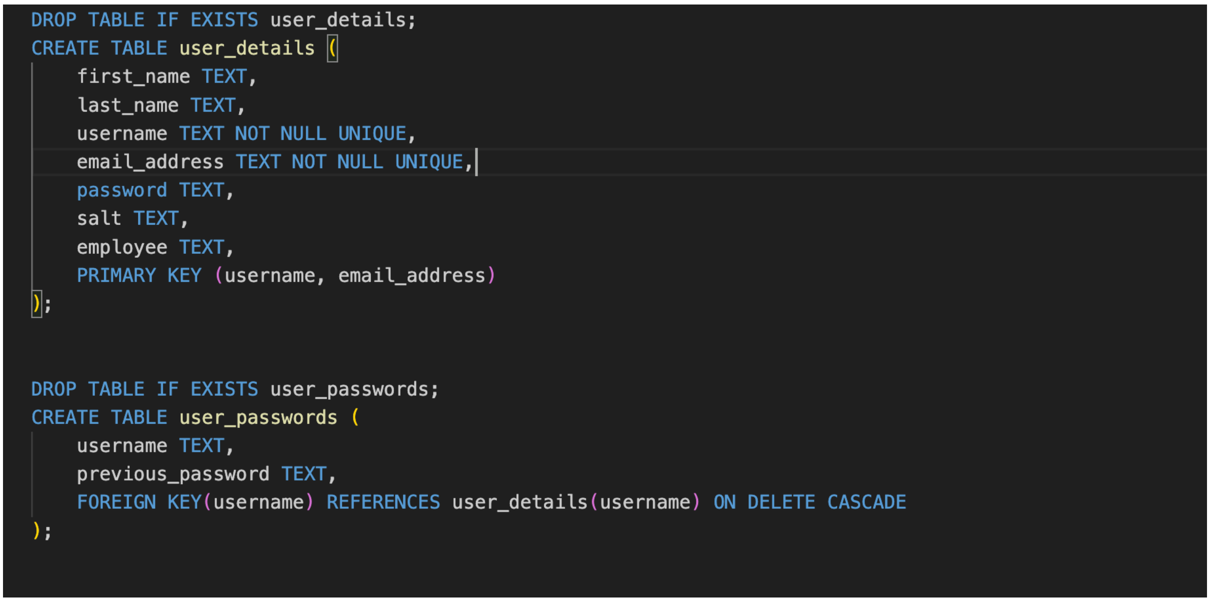
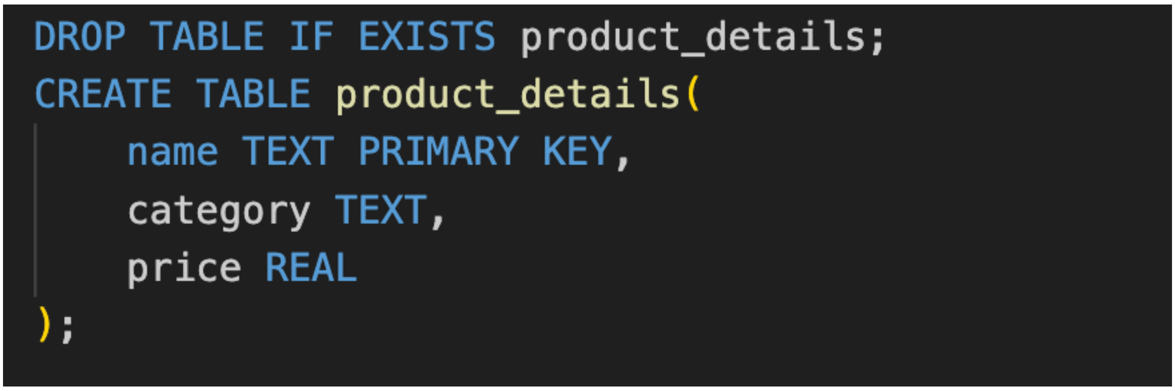
DataBase Schema
Project – 03

Users.sql



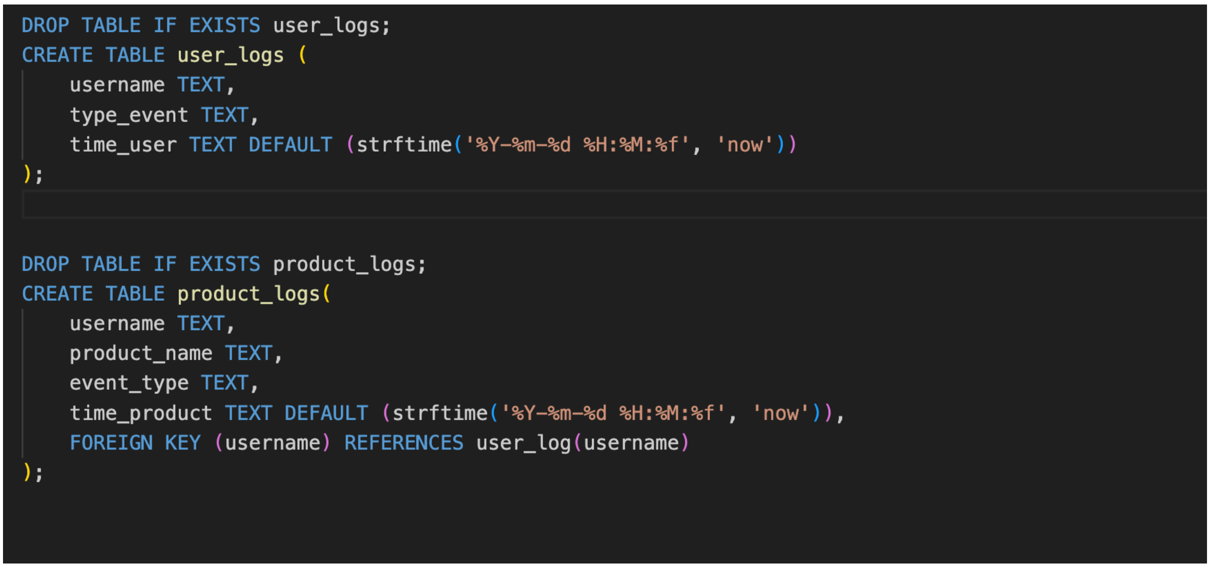
```sql
DROP TABLE IF EXISTS user_details;
CREATE TABLE user_details (
    first_name TEXT,
    last_name TEXT,
    username TEXT NOT NULL UNIQUE,
    email_address TEXT NOT NULL UNIQUE,
    password TEXT,
    salt TEXT,
    employee TEXT,
    PRIMARY KEY (username, email_address)
);


DROP TABLE IF EXISTS user_passwords;
CREATE TABLE user_passwords (
    username TEXT,
    previous_password TEXT,
    FOREIGN KEY(username) REFERENCES user_details(username) ON DELETE CASCADE
);
```

Products.sql



```sql
DROP TABLE IF EXISTS product_details;
CREATE TABLE product_details(
    name TEXT PRIMARY KEY,
    category TEXT,
    price REAL
);
```

Logs.sql



```sql
DROP TABLE IF EXISTS user_logs;
CREATE TABLE user_logs (
    username TEXT,
    type_event TEXT,
    time_user TEXT DEFAULT (strftime('%Y-%m-%d %H:%M:%f', 'now'))
);


DROP TABLE IF EXISTS product_logs;
CREATE TABLE product_logs(
    username TEXT,
    product_name TEXT,
    event_type TEXT,
    time_product TEXT DEFAULT (strftime('%Y-%m-%d %H:%M:%f', 'now')),
    FOREIGN KEY (username) REFERENCES user_log(username)
);
```

Table Design

Users Table:
This table stores the core profile for each user. Columns are first_name, last_name, username, email_address, password, salt, and employee. The composite primary key made up of username and email_address guarantees that every user is uniquely identified by both fields, preventing duplicates and preserving data integrity.

User_Passwords Table:
Keeps a history of each user's previous password hashes so that old passwords cannot be reused. It has two columns: username and previous_password. The username column is a foreign key that references user_details and is defined with ON DELETE CASCADE, so all stored hashes are removed automatically if the user is deleted.

Product_details:
Contains the master list of products available in the system. It has name, category, and price columns. The name column is the primary key, ensuring every product is uniquely identified. All catalog lookups, edits, and cost calculations reference this table.

User_logs table:
Records every high-level action a user performs that is not tied to a particular product, such as sign-up or login. It stores username, type_event, and time_user.

The timestamp has a default value generated by strftime so each entry is time-stamped automatically, creating a chronological audit trail.

Product_logs table:
Captures events that involve both a user and a product, such as product creation, edits, searches, and orders. Columns are username, product_name, event_type, and time_product. The username field is a foreign key back to user_logs, letting you join quickly to a user’s general activity while keeping product-specific events in their own table.