

CS-GY 6083 – B Principles of Database Systems

Project – Part II

- Need a login page for both customers and admin users.
 - o Username for customers will be account number a_uid from acd_safe_acnt.
 - o Password for customers (let's set it as their username for now)
 - o Let's create 3-4 admin users with their own usernames and passwords.
- To store all customers and admin usernames and passwords, let's create a new table for it. We set usernames as primary key, password, is_admin (true/false) as columns.

```
-- Create the user_auth table
CREATE TABLE user_auth (
    username BIGINT(20) PRIMARY KEY,
    password VARCHAR(255) NOT NULL,
    is_admin BOOLEAN NOT NULL DEFAULT FALSE
);
```

We can use the below command to update this user_auth table whenever there's new users created in acd_safe_acnt to update records accordingly.

```
-- Add existing customer usernames from acd_safe_acnt
INSERT IGNORE INTO user_auth (username, password, is_admin)
SELECT DISTINCT a_uid, a_uid, FALSE
FROM acd_safe_acnt;
```

- Customer Features
 - o Can change their address. (Update from CRUD activities)
 - o View all their accounts, loans, all details basically. (Read)
 - o Can change their password. (Update)
 - o Logout button to get back to login page.
- Admin Features
 - o Can view customer details based on account number search. (Read)
 - o Can update any details of that customer. (Update)
 - o Can delete the entire customer details from the bank. (Delete)
 - o Can create new customers into the bank. (Create)
 - o Can change their password. (Update)
 - o logout button to get back to login page.
- Other Features
 - o Prevent SQL injection (if Django supports it), if not check and sanitize inputs before concatenating into query strings.
 - o Multiple users should be able to use site at same time.
 - o Deadlock prevention.