PRACTICAL TRAINING REPORT

ON

Banking Automation

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

COMPLETION OF TRAINING

BACHELOR OF COMPUTER APPLICATION

Summary And Recommendation

1. Database Initialization

- The script connects to an SQLite database (`bank.sqlite`) and attempts to create a table named `acn` to store user information (Account Number, Name, Password, Email, Mobile, Balance, Open Date, Gender).
- If the table already exists, the script will print "something went wrong, might be table already exists".

2. Main Window (Login Screen)

- The main screen (`main_screen` function) has fields for entering the account number (ACN) and password.
- It provides the option to log in, clear fields, or proceed to account creation or password retrieval if the user is a new user or has forgotten their password.
- The login button verifies the entered account number and password by querying the database.

3. Forgot Password Screen

- The `forgotpass_screen` function allows a user to retrieve their password by providing their account number (ACN), email, and mobile number.
- It checks these details against the database and returns the password if the details match.

4. New User Registration

- The `newuser_screen` function enables new users to create an account by entering their name, password, email, mobile number, and gender.

- It validates the mobile number and email format using regular expressions before storing the data in the SQLite database.
- After successful registration, the user is shown the newly created account number (ACN).

5. Welcome Screen (Post-Login)

- Once logged in, the `welcome_screen` function is displayed.
- This screen shows the user's account information (account details, balance, etc.) and offers options like:
    - **Details:** View personal account details.
    - **Update:** Update account details (name, password, email, mobile).
    - **Deposit:** Deposit money into the account.
    - **Withdraw:** Withdraw money from the account.
    - **Transfer:** Transfer money to another account.

6. Banking Operations

- **Deposit:** The `deposit_db` function allows the user to deposit a specified amount into their account.
- **Withdraw:** The `withdraw_db` function lets the user withdraw an amount, provided they have sufficient balance.
- **Transfer:** The `transfer_db` function facilitates transferring money to another account. It ensures that the source and destination account numbers are different, checks if the destination account exists, and verifies sufficient balance for the transfer.

7. SQLite Database Operations

- Each function that interacts with the database (`login`, `forgotpass_db`, `newuser_db`, `details`, `update_db`, etc.) performs SQL queries to read or write data to the `acn` table.

Recommendations for Improvement or Enhancement:

1. **Security**:
    - Passwords are currently stored in plain text. It’s advisable to hash passwords before storing them using libraries like `bcrypt` or `hashlib` for better security.
2. **Error Handling**:
    - The code lacks detailed error handling. For example, during database operations, exceptions like `sqlite3.DatabaseError` can be caught and handled appropriately.
3. **User Interface**:
    - While the UI uses basic `Tkinter` widgets, it can be improved by adding more visual elements like icons, better layout management (using `pack`, `grid`, or `place` more effectively), and ensuring the UI is more responsive.
4. **Database Optimization**:

    - The database operations like fetching account data or checking login credentials can be optimized with better indexing or query structure to ensure the app scales well with a larger number of users.
5. **User Session Management**:
    - After login, the session can be managed in a better way (e.g., storing the account number in a global or session variable to avoid frequent queries for user data).
6. **Log out**:
    - You might want to add additional functionality for the "logout" button to clean up any session-related data or navigate back to the login screen properly.