Project Overview and Report of What is Implemented

Must create a fake bank system to trick a scammer into believing they are accessing a victim's true bank account.

Requirements

The project would include the following pages:

- Login
- Account overview
- Transaction history
- Payments/transfers

Functional requirements

- Enable users to create, edit and delete one or more user profiles.
- Add/manage/remove accounts to user profiles.
- Generate a realistic recent transaction history and history that reflects the account balances.
- Admin area that can be to configure the access credentials of fake-users, the numbers and names of the accounts of each user and the amount of money in each account.

Non-functional requirements

- The login process requires only a single password.
- Application supports multiple currencies (GBP, USD and a third currency)
- Admin superusers can add, edit and delete user accounts.
- The system can generate a random recent transaction history for an end-user.
- Some of the peripheral screens, not essential to the refund scam, have realistic content and a realistic look and feel.

Notes

- Must **not** use the name, logo, look and feel (e.g. bespoke fonts and colours) of a genuine online banking service.
- Must **not** use copyright or trademarked materials of a real organisation.

Deployment

The deployed version should be preconfigured with consumer and administrator accounts. The consumer should have multiple bank accounts with transaction history. Correct access credential to user and admin account should be provided.

Report of What is implemented:

- User Authentication (Log-in and Log-out)
    - User will be able to sign up to the webpage by inserting:
        - Name, surname, email, password, password confirmation, Date of Birth and Phone Number
    - After the user signed up successfully, the user will be able to go to the make “payment page”, check the ‟ transactions page”, asl well as ‟edit account” page.
    - If a user has not signed in yet, he/she will be unable to see those pages mentioned above such as: payments, sign-put and transactions etc.
    - After a user signed in successfully, the user can sign-out button will appear and he/she can sign out without problems.

- Search Bar
    - All the links are implemented on top of the page inside of a search bar, where users can find the relevant pages by clicking the buttons.

- The web page displays well on the desktops, tablets as well as mobile devices.

- About Us and Contact Us
    - About us and contact us pages were created and linked to the other elements of the bank’s user interface. User interaction in the specific pages have also been created so that the user is able to view the different data throughout these pages in various ways. A unified design pattern was implemented to allow for an eye-appealing user interface between all the web pages yet promoting the distinct features on each page.

- Implemented admin interface, you can login with an email and password.
    - Normal users cannot login, only admin users who have been approved can log in.

    - The admin interface can be accessed by putting /admin in front of the url.

    - There is a dashboard which is the default page displayed once the admin has logged in.

    - On the menu bar there are sections for admin users, comments, users and transactions.

    - The admin user section can create, edit or delete admin users.

    - The Users section can create, edit or delete any user. All the users are displayed on the page with their relevant details.

    - On the transactions page, admin can view the transaction history, the admin can create new transactions, delete existing ones or edit them. To make a transaction, the admin needs sender_id which can be found in the users section, currency_id, recipient, sort_code and account number.

- Implemented transactions
    - Created a database tables for different currencies and their corresponding values in pounds; this allows users to make payments in any currency, but to have their account balance decreased in the GBP equivalent
    - Added a sort code and balance attribute to each user, where "050505" is the sort code of Nyala Bank.
    - Added very thorough validation to the transactions controller before any transaction is made to ensure only secure and legitimate transactions are accepted (this is done in the back-end to ensure any manually created POST requests will face the same validation).
    - Added a check to ensure that recipient users exist if the recipient of the transaction is with Nyala Bank.
    - Created a 'Your Statement' page to display a list of all previous transactions made by a given user.
    - Added a balance section to the dashboard to allow the user to check their balance quickly.
    - On each section there are filters which can help an admin to find a particular piece of data or sort it based on the requirements.
    - Created admin user database and active admin comments database.

- Extensively used the website to try and find flaws to be corrected. Noticed some things which should not be tolerated by the website but thus got corrected.

- Implemented tests for the controls of the website to make sure it does what it is supposed to do.