Part 2: Technical Knowledge Evaluation

1. Write an SQL query to find all the members of a specific household.

   SELECT *
   FROM Household_users
   WHERE household_id = '17356bf0-21a2-11ef-afb2-99ff98b0e148'

2. Write an SQL query to show the name and role within a household.

   SELECT Households.name, Household_Users.role
   FROM Households
   JOIN Household_Users
   ON Household_Users.household_id = Households.Id;

Part 3: Practical Questions

Case Scenario: Add Members

TC_ADD_USERS_001
Description: Verify that the household can add a new valid member.
Steps:
- Go to https://dashboard-test.zoefin.com/login
- Login as Household with:
  - Email: zoefin.advisor+qaroleadv01@gmail.com
  - Password: <password>
- In the *Pipeline* tab select a Household
- Go to *Household* tab
- Click on *Add Member*
- Add valid data
- Click on *Add Member* button

Expected Behavior: A confirmation message appears saying: 'Member Added!'' and the user appears in the list.

Actual Behavior: The confirmation message appears and the member is successfully added

State: Approved

TC_ADD_USERS_002

Description: Try to add an existing user

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Household with:
    - Email: zoefin.advisor+qaroleadv01@gmail.com
    - Password: <password>
- In the *Pipeline* tab select a Household
- Click on *Add Member*
- Try to add the data of an existing user
- Click on *Add Member* button

Expected Behavior: An error message appears indicating that the email already exists.

Actual Behavior: An error message appears indicating that the email already exists.

State: Approved

TC_ADD_USERS_003

Description: Add a member with missing data

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Household with:
    - Email: zoefin.advisor+qaroleadv01@gmail.com
    - Password: <password>
- In the *Pipeline* tab select a Household
- Click on *Add Member* button
- Leave all the fields empty and try to create the user.

Expected Behavior: The *Add Member* button is not enabled to create the user.

Actual Behavior: The *Add Member* button is not enabled to create the user.

State: Approved

Bug found

Title: Attempt to add user without email

Description: With a new form, and without adding and deleting data, when trying to add a member with all the fields filled out except the email, the *Add Member* button is enabled
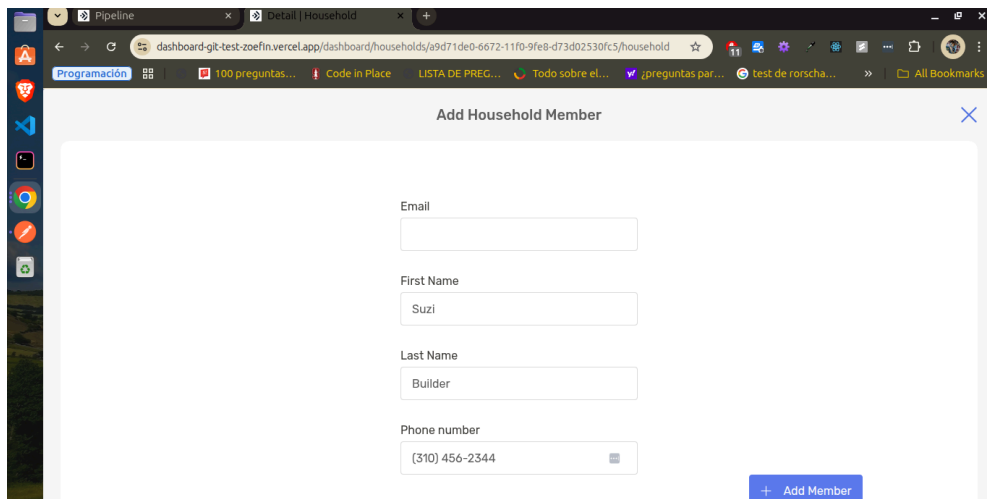
To reproduce:
- Open the form to create a new member
- Fill in only the first name, last name and phone number fields.
- Click on the *Add Member* button
- The Add member button remains loaded and does not display any error message.
-

Expected Behavior: The *Add Member* button is not enabled without all fields being completed

Screenshots:

Case scenario: Edit members

TC_EDIT_USERS_001

Description: Correctly edit user fields

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Household with:
    - Email: zoefin.advisor+qaroleadv01@gmail.com
    - Password: <password>
- In the Pipeline, select a Household
- Go to *Household* tab
- Click on the *Edit* button next to the member.
- Enter the new data
- Save the changes

Expected Behavior: The confirmation message is displayed and the changes are reflected.

Actual Behavior: The confirmation message is displayed and the changes are reflected.

State: Approved

TC_EDIT_USERS_002

Description: Trying to save an invalid phone number

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Household with:
    - Email: zoefin.advisor+qaroleadv01@gmail.com
    - Password: <password>
- In the Pipeline, select a Household
- Go to *Household* tab
- Click on the *Edit* button next to the member.
- Enter in invalid phone number
- Try to save the changes

Expected Behavior: An error message appears saying: 'Please enter a valid phone number' and the *Save Changes* button is not enabled.

Actual Behavior: The error message is displayed and the *Save button* is not enable

State: Approved

TC_EDIT_USERS_003

Description: Cancel editing without saving changes

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Household with:
    - Email: zoefin.advisor+qaroleadv01@gmail.com
    - Password: <password>
- In the Pipeline, select a Household
- Go to *Household* tab
- Click on the *Edit* button next to the member.
- Enter the new data
- Close editing changes without saving them

Expected Behavior: The data is not updated and the original data is preserved.

Actual Behavior: The data is not updated and the original data is preserved.

State: Approved

Case scenario: Delete members

TC_DELETE_USERS_001

Description: Correctly removing a member

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Household with:
    - Email: zoefin.advisor+qaroleadv01@gmail.com
    - Password: <password>
- In the Pipeline, select a Household
- Go to *Household* tab
- Click on *Delete* button next to the member
- In the confirmation modal click on *Remove* button

Expected Behavior: The member no longer appears in the member list.

Actual Behavior: The member no longer appears in the member list but no confirmation message appears.

State: Approved. But it would be nice to see a confirmation message validating that the member has been removed.

TC_DELETE_USERS_002

Description: Cancel the deletion of a member without confirming it

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Household with:
    - Email: zoefin.advisor+qaroleadv01@gmail.com
    - Password: <password>
- In the Pipeline, select a Household
- Go to *Household* tab
- Click on *Delete* button next to the member
- In the confirmation modal click on *Cancel* button

Expected Behavior: The member is not deleted, it remains visible in the members list.

Actual Behavior: The member is not deleted, it remains visible in the members list.

State: Approved

Case scenario: Delete members

TC_ROLE_PERMISSIONS_001

Description: User with 'Client' role cannot edit or delete members

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Client with:
  - Email: zoefin.client+qaroleclient01@gmail.com
  - Password: <password>
- In the Pipeline, select a Household
- Go to *Household* tab
- Check the list of members

Expected Behavior: The client can see the list of members but the *Delete* or *Edit* buttons do not appear.

Actual Behavior: The client can see the list of members but the *Delete* or *Edit* buttons do not appear.

State: Approved

TC_ROLE_PERMISSIONS_002

Description: A user with the "Advisor" role cannot edit or delete a member with a completed investment account.

Steps:

- Go to https://dashboard-test.zoefin.com/login
- Login as Client with:
  - Email: zoefin.advisor+qaroleadv01@gmail.com
  - Password: <password>
- In the Pipeline, select a Household *'New Name'*
- Go to *Household* tab
- Click on the member *'Ken Customer'*
- Verify that the *Edit* and *Delete* buttons are not present.

Expected Behavior: Edit and delete options are disabled.

Actual Behavior: The *Edit* button is no longer visible but the *Delete* button is still displayed.

State: Failed

Part 4: Case Studies

Case Study 1

-   First, I reproduce the problem and try to verify in which environment and with which specific configurations (if any) the bug is being generated. For example if it is in a specific browser or if it is also happening in production, etc. Then, I make sure that it is not a role misconfiguration, or an unexpected restriction in permissions. If the problem can be reproduced continuously, I would document the bug in a ticket with the necessary information and evidence and escalate it to the development team so that it can be solved as quickly as possible.

Case Study 2

When encountering intermittent problems like this I would do the following:

1. The first thing I would do is try to reproduce it under controlled conditions. I would try with different roles (PM, Advisor, Owner), use different browsers, different members, etc, all in order to identify some kind of pattern: if it happens after an update or if it happens only with some data.

2. Then, I would inspect the frontend to validate if the visibility of the edit and delete buttons depend only on the frontend logic or also depend on the backend. I would use tools like DevTools to verify if the status of the investment accounts are loading correctly.

3. Also, I would check the backend responses to make sure that members with completed investment accounts are being identified and that the necessary restrictions are being applied.

4. If I have access to the database, I would check if the account status is being saved correctly.

In order to prevent these problems in the future, I would do the following:

- Ensure validations on both the frontend and backend.

- Implement automated tests that cover the scenarios where editing and deleting should not be allowed.

- Work with test environments that include members with different investment statuses.