* User Story: Sign up

As a user, I want to sign up in order to create a quotation template. Below are the steps to complete the needed action:

* Click on the ‘Register’ button
* Display and fill the Form that contains:
  + First name (Mandatory)
  + Last name (Mandatory)
  + Email (Mandatory)
  + Phone number (Mandatory)
  + Password (Mandatory)
  + Confirm Password (Mandatory)

The system ensures mandatory fields are filled and data formats are correct. If not, users are prompted to complete fields.

It checks if users exist in the database; existing users proceed to sign-in, new users undergo a sign-up process with two-factor authentication via SMS or authenticator app.

Task 1: [Quotation - FrontEnd] – Sign up

1. Design a user-friendly registration form with input fields for first name, last name, email, phone number, password and confirm password.
2. Implement validation mechanisms to ensure all mandatory fields are filled and data formats are correct before submission.
3. Display appropriate error messages to prompt users to complete any missing or incorrectly formatted fields.
4. Provide a button to initiate the registration process.
5. Integrate functionality for two-factor authentication if the user is a new registrant.

Task 2: [Quotation -BackEnd] - Signup

1. Create an API endpoint to handle user registration requests.
2. Implement logic to validate mandatory fields (first name, last name, email, phone number, password and confirm password) and ensure correct data formats.
3. Check the database to determine if the user already exists.
4. If the user does not exist, initiate the sign-up process, including two-factor authentication via SMS or authenticator app or email.
5. Save the user data to the database upon successful registration.

* User Story: Sign in

Title:Walk\_in – Sign In

As a registered user, I want to sign in to access my account and manage my quotation templates.

Below are the steps to complete the necessary action:

* Enter your email and password.
* Verify that the entered email and password are in the correct formats; prompt to refill if incorrect.
* Check the database to verify the existence of the user.
* Existing users proceed to authenticate (2-factor authentication) before being redirected to the dashboard.
* New users are directed to the sign-up process.

Related Tasks:

Task 1: [Quotation - FrontEnd] - Sign in

1. Design a sign-in form with fields for email and password.
2. Implement client-side validation for email and password formats.
3. Send a login request to the backend API upon form submission.
4. Handle responses from the backend, displaying appropriate error messages if credentials are incorrect or if the user needs to complete 2-factor authentication.
5. Upon successful authentication, redirect the user to the dashboard page.

Task 2: [Quotation - BackEnd]- Sign in

1. Implement an API endpoint to handle user login requests.
2. Validate the format of the email and password inputs.
3. Check the database for the existence of the user based on the provided email.
4. If the user exists, verify the password against the stored hash in the database.
5. Implement 2-factor authentication logic if enabled for the user's account.
6. If authentication is successful, generate and return an authentication token.