1. Общее описание

Модули регистрации и авторизации обеспечивают:

* Создание новых учетных записей сотрудников.
* Аутентификацию пользователей в системе.
* Хранение профилей пользователей в базе данных (Supabase).

2. Модуль авторизации (/login)

Функционал:

* Вход пользователя по email и паролю.
* Валидация полей формы.
* Обработка ошибок (неверные данные, проблемы с сервером).
* Перенаправление на главную страницу (/main-now) при успешной аутентификации.

Компоненты и зависимости

1. Blazor-страница:/login
2. Сервисы:
   * Supabase.Client– для аутентификации через Supabase.
   * NavigationManager– для перенаправления.
3. Модель:LoginModel(email, password).

Алгоритм работы

1. Пользователь вводит email и пароль.
2. При нажатии кнопки"Войти"вызывается методHandleAuthorization().
3. Система отправляет запрос к Supabase (SignInWithPassword).
4. Если аутентификация успешна:
   * Пользователь перенаправляется на/main-now.
5. Если ошибка:
   * Выводится сообщение:*"Неверный email или пароль"*(или другая ошибка).

Обработка ошибок

|  |  |
| --- | --- |
| Ошибка | Сообщение для пользователя |
| Неверный email/пароль | "Неверный email или пароль." |
| Проблемы с сервером | "Произошла ошибка авторизации. Попробуйте позже." |

3. Модуль регистрации (/register)

Функционал:

1. Создание нового аккаунта (email, пароль, подтверждение пароля).
2. Сохранение профиля пользователя (имя, фамилия, возраст).
3. Валидация данных:
   * Минимальная длина пароля (6 символов).
   * Возраст > 0.
   * Совпадение паролей.
4. Обработка ошибок (дубликат email, серверные ошибки).

Компоненты и зависимости

1. Blazor-страница:/register
2. Сервисы:
   * Supabase.Client– для работы с базой данных.
   * AuthService– для регистрации в Supabase Auth.
   * NavigationManager– для перенаправления.
3. Модель: RegisterModel(email, password, confirmPassword, firstName, lastName, age).

Алгоритм работы

1. Пользователь заполняет форму.
2. При нажатии "Зарегистрироваться" вызывается HandleRegistration().
3. Проверяется валидность данных:
   * Пароль ≥ 6 символов.
   * Возраст > 0.
4. Система регистрирует пользователя через AuthService.RegisterAsync().
5. Если успешно:
   * Профиль сохраняется в таблицеUserProfile.
   * Перенаправление на/login.
6. Если ошибка:
   * Выводится сообщение (например,*"Пользователь с таким email уже существует"*).

Обработка ошибок:

|  |  |
| --- | --- |
| Ошибка | Сообщение для пользователя |
| Пароль < 6 символов | "Пароль должен содержать минимум 6 символов." |
| Возраст ≤ 0 | "Возраст должен быть больше 0." |
| Email уже занят | "Пользователь с таким email уже зарегистрирован." |
| Ошибка сервера | "Ошибка при регистрации. Попробуйте позже." |

4. Интеграция с Supabase

* Auth:
  + Аутентификация через supabase.auth.signInWithPassword().
  + Регистрация через supabase.auth.signUp().
* База данных:
  + Профили пользователей хранятся в таблицеUserProfile.