Opis US:	Rejestracja nowego użytkownika oraz logowanie
Jako:	Nowy użytkownik aplikacji, który chce założyć konto oraz zalogować się do systemu.
Chciałbym:	Mieć możliwość rejestracji, tworząc nowe konto przy użyciu adresu e-mail i hasła, a następnie logować się do aplikacji za pomocą tych danych.
Aby:	Móc korzystać z funkcji aplikacji, które wymagają zalogowania.
Dodatkowe informacje:	1. Użytkownik podaje unikalny adres e-mail oraz silne hasło 2. Na ekranie logowania użytkownik może skorzystać z opcji „Don't have an account? Register". 3. W przypadku nieudanej próby logowania wyświetlany jest komunikat „Invalid user or password". 4. Udane logowanie wyświetla komunikat „Registration succesful".
Makieta:	Username (pole) Password (pole) Log in (button) Don't have an account? Register (button) > w przypadku rejestracji nowa strona z Username (pole) Password (pole) Register (button)
Wycena:	Analiza: 2h 2h 2h Front: 2h 3h 4h Backend: 3h 4h 5h Testy: 3h 4h 5h PM: 2h 2h 2h

Zadania dla każdego zespołu:

---

1. Analiza

- **Zadanie 1:** Przeanalizować wymagania dotyczące rejestracji i logowania (2h).
- **Zadanie 2:** Przygotować schemat przepływu użytkownika oraz jego interakcji z ekranami rejestracji i logowania (2h).
- **Zadanie 3:** Sprawdzić zależności oraz wymagania dotyczące bezpieczeństwa (np. walidacja e-mail, złożoność hasła) (2h).

---

2. Frontend

- **Zadanie 1:** Stworzyć widok ekranu logowania z polami "Username", "Password", "Log in" oraz przyciskiem "Don’t have an account? Register" (2h).
- **Zadanie 2:** Zaimplementować komunikaty dla użytkownika:
    - Po udanym logowaniu: "Registration successful" (1h).
    - Po nieudanej próbie logowania: "Invalid user or password" (1h).
- **Zadanie 3:** Dodanie walidacji hasła oraz sprawdzenie unikalności adresu e-mail na froncie (3h).

---

3. Backend

- **Zadanie 1:** Implementacja endpointu rejestracji użytkownika (POST /register) z walidacją unikalności adresu e-mail oraz złożoności hasła (3h).
- **Zadanie 2:** Implementacja endpointu logowania (POST /login), weryfikacja danych logowania oraz zwracanie odpowiednich komunikatów (4h).
- **Zadanie 3:** Dodanie obsługi sesji użytkownika lub tokena autoryzacyjnego po zalogowaniu (5h).
- **Zadanie 4:** Zabezpieczenie endpointów przed atakami typu brute force (np. limit prób logowania) (3h).

---

4. Testy

- **Zadanie 1:** Przygotowanie scenariuszy testowych do weryfikacji rejestracji i logowania (1h).
- **Zadanie 2:** Testowanie poprawności rejestracji użytkownika (rejestracja z poprawnymi oraz niepoprawnymi danymi) (4h).
- **Zadanie 3:** Testowanie poprawności logowania użytkownika (logowanie z poprawnymi oraz niepoprawnymi danymi) (4h).
- **Zadanie 4:** Testowanie komunikatów wyświetlanych użytkownikowi w przypadku udanej i nieudanej próby logowania (1h).
- **Zadanie 5:** Testowanie walidacji hasła oraz unikalności adresu e-mail (3h).

---

5. Project Management (PM)

- **Zadanie 1:** Przygotowanie harmonogramu pracy dla zespołu (2h).
- **Zadanie 2:** Koordynacja prac między frontendem a backendem oraz monitorowanie postępów zespołów (2h).
- **Zadanie 3:** Organizacja spotkań codziennych (stand-up) i retrospektywy sprintu (2h).

Opis US:	Generowanie przykładowej propozycji silnego hasła
Jako:	Użytkownik, który tworzy nowe konto w systemie lub zmienia hasło

Chciałbym:	Otrzymać automatycznie wygenerowaną propozycję silnego hasła spełniającą wymogi bezpieczeństwa (ilość znaków, wielkie i małe litery, cyfry oraz znaki specjalne), którą mogę zaakceptować lub odrzucić.
Aby:	Z łatwością stworzyć bezpieczne hasło, które będzie trudne do odgadnięcia i chronić moje konto przed nieautoryzowanym dostępem.
Dodatkowe informacje:	1. Użytkownik podaje Serwis do którego chce zgenerować hasło i Email (opcjonalnie) 2. Generujemy losową propozycję hasła o odpowiedniej sile (użytkownik sam podaje minimum znaków oraz zawartość znaków specjalnych). 3. Zwracamy komunikat z wyświetlanym hasłem 4. W przypadku odrzucenia propozycji, generujemy nowe hasło na żądanie użytkownika. 5. Użytkownik może zapisać wygenerowane hasło.
Makieta:	Service (pole) Email (Optional) (pole) Include Special Symbols (tak/nie) Password Length (scroll button) Generate Password (button) Save Password (button)
Wycena:	Analiza: 3h 3h 3h Front: 1h 2h 3h Backend: 2h 4h 6h Testy: 1h 2h 2h PM: 2h 2h 2h

Zadania dla każdego zespołu:

---

1. Analiza

- **Zadanie 1:** Przeanalizowanie wymagań dla funkcjonalności generowania i personalizacji silnych haseł (3h).
- **Zadanie 2:** Opracowanie algorytmu generującego hasło, uwzględniającego minimalną długość, wielkie i małe litery, cyfry oraz znaki specjalne (3h).
- **Zadanie 3:** Określenie sposobu zapisu i przechowywania wygenerowanych haseł z uwzględnieniem opcji „Save Password” (3h).

---

2. Frontend

- **Zadanie 1:** Stworzenie widoku formularza generowania hasła z polami:
    - „Service” (pole tekstowe),

    - „Email” (pole tekstowe, opcjonalne),
    - „Include Special Symbols” (przełącznik Tak/Nie),
    - „Password Length” (suwak wyboru długości) (2h).
- **Zadanie 2:** Implementacja przycisków „Generate Password” i „Save Password” (1h).
- **Zadanie 3:** Wyświetlanie wygenerowanego hasła z możliwością jego zaakceptowania lub odrzucenia przez użytkownika (2h).
- **Zadanie 4:** Dodanie funkcji ponownego generowania hasła na żądanie użytkownika, jeśli pierwsza propozycja zostanie odrzucona (3h).

---

3. Backend

- **Zadanie 1:** Implementacja endpointu generowania hasła (POST /generate-password), który przyjmuje parametry takie jak długość hasła, znaki specjalne i wymagane elementy (2h).
- **Zadanie 2:** Opracowanie logiki generowania hasła, która uwzględnia wymagane parametry (np. długość, użycie wielkich i małych liter, cyfr, znaków specjalnych) (4h).
- **Zadanie 3:** Implementacja endpointu zapisu wygenerowanego hasła, który zapisuje wygenerowane hasło powiązane z serwisem i e-mailem (6h).
- **Zadanie 4:** Zabezpieczenie zapisanych haseł, w tym opcja szyfrowania przed zapisem w bazie danych (3h).

---

4. Testy

- **Zadanie 1:** Opracowanie scenariuszy testowych dla funkcji generowania hasła, w tym walidacja długości, użycia znaków specjalnych, wielkich/małych liter i cyfr (1h).
- **Zadanie 2:** Testowanie opcji „Generate Password” przy różnych kombinacjach parametrów (np. z i bez znaków specjalnych, różne długości hasła) (2h).
- **Zadanie 3:** Testowanie funkcji „Save Password”, w tym sprawdzenie, czy hasło jest prawidłowo zapisane i zaszyfrowane w bazie danych (2h).
- **Zadanie 4:** Weryfikacja działania funkcji odrzucenia hasła i generowania nowego na żądanie (1h).

---

5. Project Management (PM)

- **Zadanie 1:** Przygotowanie harmonogramu sprintu oraz koordynacja pracy zespołów frontend, backend i testów (2h).
- **Zadanie 2:** Monitorowanie postępów prac zespołów oraz rozwiązywanie ewentualnych problemów (2h).
- **Zadanie 3:** Organizacja spotkań codziennych (stand-up), podsumowań sprintu oraz retrospektywy sprintu (2h).

Opis US:	Sprawdzanie bezpieczeństwa i siły hasła
Jako:	Użytkownik, który tworzy nowe konto w systemie
Chciałbym:	Sprawdzić, czy hasło, które wprowadzam, spełnia wymagania bezpieczeństwa i jest wystarczająco silne. Po wpisaniu hasła system automatycznie analizuje jego moc.
Aby:	Zabezpieczyć swoje konto przed nieautoryzowanym dostępem poprzez użycie silnego hasła.
Dodatkowe informacje:	1. Hasło zbyt słabe i zawarte w bazie słabych haseł: Zwracamy komunikat „Password Strength: Very Weak (Common Password)” 2. Hasło słabe: Zwracamy komunikat „Password Strength: Weak” 3. Hasło średnie: Zwracamy komunikat „Password Strength: Moderate” 4. Hasło silne: Zwracamy komunikat „Password Strength: Strong”
Makieta:	Enter Password to check Progress bar
Wycena:	Analiza: 1h 1h 1h Front: 2h 3h 4h Backend: 1h 2h 2h Testy: 1h 1h 1h PM: 1h 1h 1h

Zadania dla każdego zespołu:

---

1. Analiza

- **Zadanie 1:** Przeanalizowanie wymagań dotyczących oceny siły hasła oraz kryteriów klasyfikacji haseł jako "Very Weak", "Weak", "Moderate" i "Strong" (1h).
- **Zadanie 2:** Określenie algorytmów i baz danych do weryfikacji, czy hasło jest powszechnie używane i zawarte w bazie słabych haseł (1h).
- **Zadanie 3:** Przegląd wymagań dotyczących komponentu frontendowego, w tym paska postępu pokazującego siłę hasła (1h).

---

2. Frontend

- **Zadanie 1:** Stworzenie widoku pola "Enter Password to check" z paskiem postępu (progress bar) do wyświetlania siły hasła w czasie rzeczywistym (2h).
- **Zadanie 2:** Implementacja wizualnych wskaźników siły hasła na pasku postępu:

    - Bardzo słabe: Czerwony kolor, komunikat „Password Strength: Very Weak (Common Password)”
    - Słabe: Pomarańczowy kolor, komunikat „Password Strength: Weak”
    - Średnie: Żółty kolor, komunikat „Password Strength: Moderate”
    - Silne: Zielony kolor, komunikat „Password Strength: Strong” (3h).
- **Zadanie 3:** Dodanie dynamicznej aktualizacji poziomu bezpieczeństwa hasła na podstawie danych z backendu oraz zmiany koloru paska postępu (4h).

---

3. Backend

- **Zadanie 1:** Implementacja logiki analizy siły hasła, która ocenia:
    - Długość hasła,
    - Złożoność (wielkie i małe litery, cyfry, znaki specjalne),
    - Obecność w bazie słabych haseł (1h).
- **Zadanie 2:** Utworzenie endpointu API (POST /check-password-strength), który przyjmuje hasło, analizuje jego moc i zwraca odpowiedni wynik („Very Weak”, „Weak”, „Moderate”, „Strong”) oraz powód, jeśli hasło jest „Very Weak” (2h).
- **Zadanie 3:** Dodanie funkcji do porównania hasła z bazą powszechnie używanych lub słabych haseł (np. porównanie z listą popularnych haseł) (2h).

---

4. Testy

- **Zadanie 1:** Opracowanie scenariuszy testowych dla funkcji analizy siły hasła, uwzględniając różne przypadki, np. krótkie i proste hasła, złożone hasła, hasła powszechnie używane (1h).
- **Zadanie 2:** Testowanie integracji frontend-backend w celu sprawdzenia, czy siła hasła jest poprawnie wyświetlana użytkownikowi oraz że kolory i komunikaty są zgodne z oczekiwaniami (1h).
- **Zadanie 3:** Walidacja działania paska postępu na podstawie siły hasła i poprawności komunikatów (1h).

---

5. Project Management (PM)

- **Zadanie 1:** Przygotowanie harmonogramu sprintu i koordynacja działań między zespołami frontend, backend i testów (1h).
- **Zadanie 2:** Monitorowanie postępów prac oraz rozwiązywanie ewentualnych problemów, które mogą wystąpić podczas implementacji (1h).
- **Zadanie 3:** Organizacja spotkań codziennych (stand-up) oraz retrospektywy sprintu (1h).

Opis US:	Chronienie haseł przez użytkownika w menedżerze haseł
Jako:	Użytkownik, który chce bezpiecznie przechowywać swoje hasła do różnych aplikacji i serwisów
Chciałbym:	Mieć możliwość zapisywania i zabezpieczania moich haseł w menedżerze haseł, który jest chroniony jednym silnym hasłem głównym.
Aby:	Zarządzać wszystkimi moimi hasłami w jednym miejscu, bez ryzyka ich zapomnienia lub utraty, jednocześnie zapewniając ich bezpieczeństwo.
Dodatkowe informacje:	1. Po wprowadzeniu prawidłowego hasła do logowania użytkownik ma dostęp do listy zapisanych haseł. 2. Hasła są przechowywane w formie zaszyfrowanej 3. W przypadku zgubienia hasła głównego użytkownik nie będzie mógł odzyskać przechowywanych haseł. 4. Każdy użytkownik ma dostęp tylko do swojej listy haseł 5. Użytkownik może edytować, usuwać oraz kopiować hasła.
Makieta:	Search Service or Email (dynamic search) Scroll View: ● Service ● Email ● Password Edit screen: ● Service (pole) ● Email (Optional) (pole) ● Password (pole) ● Save (button) ● Cancel (button) ● Delete (icon button) Copy (Icon Button)
Wycena:	Analiza: 3h 3h 3h Front: 4h 5h 6h Backend: 4h 6h 8h Testy: 3h 4h 5h PM: 3h 3h 3h

Zadania dla każdego zespołu:

---

1. Analiza

- **Zadanie 1:** Określenie wymagań dotyczących przechowywania i zabezpieczania haseł użytkownika w menedżerze (3h).
- **Zadanie 2:** Przegląd wymagań szyfrowania haseł, aby zapewnić bezpieczeństwo przechowywanych danych (3h).
- **Zadanie 3:** Opracowanie przepływu użytkownika do zarządzania zapisanymi hasłami (dodawanie, edycja, usuwanie, kopiowanie) i zapewnienie, że każdy użytkownik ma dostęp tylko do swojej listy haseł (3h).

---

2. Frontend

- **Zadanie 1:** Stworzenie widoku listy haseł z funkcją wyszukiwania (pole "Search Service or Email") oraz funkcją przewijania listy zapisanych haseł (Scroll View) (4h).
- **Zadanie 2:** Implementacja ekranu szczegółów hasła z polami:
    - "Service" (pole tekstowe),
    - "Email" (opcjonalne pole tekstowe),
    - "Password" (pole tekstowe z możliwością kopiowania hasła) (5h).
- **Zadanie 3:** Dodanie przycisków akcji na ekranie szczegółów hasła:
    - "Save" (zapisanie zmian),
    - "Cancel" (anulowanie edycji),
    - "Delete" (usunięcie hasła),
    - "Copy" (skopiowanie hasła do schowka) (6h).

---

3. Backend

- **Zadanie 1:** Implementacja endpointu do przechowywania haseł (POST /store-password), który przechowuje zaszyfrowane hasło powiązane z danym użytkownikiem (4h).
- **Zadanie 2:** Implementacja endpointu do pobierania listy zapisanych haseł dla zalogowanego użytkownika (GET /passwords), z filtrowaniem dostępu wyłącznie do własnych danych (6h).
- **Zadanie 3:** Opracowanie logiki edycji (PUT /edit-password), usuwania (DELETE /delete-password) i kopiowania hasła (GET /copy-password), przy czym wszystkie operacje są zabezpieczone i możliwe tylko dla zalogowanego użytkownika (8h).
- **Zadanie 4:** Szyfrowanie haseł przed ich zapisem w bazie danych, aby uniemożliwić nieautoryzowany dostęp do danych (4h).

---

4. Testy

- **Zadanie 1:** Opracowanie scenariuszy testowych dla operacji CRUD (Create, Read, Update, Delete) na hasłach, w tym sprawdzenie poprawności zapisu, edycji i usuwania danych (3h).

- **Zadanie 2:** Testowanie funkcji wyszukiwania haseł na liście w celu sprawdzenia, czy dynamiczne wyszukiwanie działa poprawnie (1h).
- **Zadanie 3:** Testowanie operacji kopiowania hasła oraz sprawdzenie bezpieczeństwa przy operacji kopiowania (1h).
- **Zadanie 4:** Weryfikacja, czy dostęp do haseł jest ograniczony tylko do właściciela konta i że inne konta nie mogą uzyskać dostępu do tych danych (4h).
- **Zadanie 5:** Testowanie scenariusza, w którym użytkownik zapomina hasło główne i nie ma możliwości odzyskania zapisanych haseł (5h).

---

5. Project Management (PM)

- **Zadanie 1:** Przygotowanie harmonogramu sprintu oraz koordynacja działań między zespołami frontend, backend i testów (3h).
- **Zadanie 2:** Monitorowanie postępów oraz rozwiązywanie potencjalnych problemów technicznych, które mogą wystąpić podczas implementacji (3h).
- **Zadanie 3:** Organizacja spotkań codziennych (stand-up) oraz retrospektywy sprintu w celu oceny wyników sprintu i identyfikacji możliwych usprawnień (3h).

Link do projektu na GitHub: https://github.com/KIvan-byte/IvliPass