**Pytania dotyczące technologii:**

1. Dlaczego zdecydowałeś się na wykorzystanie technologii WPF? Jakie są jej zalety w porównaniu do WinForms lub innych frameworków?

WPF oferuje nowoczesne możliwości projektowania interfejsów użytkownika, takie jak dynamiczne style, binding danych oraz obsługę XAML, co ułatwia oddzielenie logiki aplikacji od jej wizualnej części. W porównaniu do WinForms, WPF zapewnia bardziej zaawansowaną obsługę grafiki, możliwość tworzenia responsywnych aplikacji oraz wsparcie dla nowoczesnych standardów UI. Jest także bardziej elastyczny w dostosowywaniu wyglądu aplikacji.

1. Co sprawiło, że wybrałeś język C# do realizacji projektu? Jakie jego cechy uznałeś za kluczowe?

C# to nowoczesny, obiektowy język programowania, który świetnie integruje się z WPF i .NET Framework. Jego silne typowanie, łatwa obsługa zdarzeń oraz wsparcie dla wzorców projektowych (np. MVVM) były kluczowe. Dodatkowo, szerokie wsparcie społeczności oraz zajęcia na uczelni uczyniły go naturalnym wyborem.

1. Dlaczego zdecydowałeś się na użycie bazy SQLite? Jakie były powody wyboru tej bazy danych zamiast np. MySQL, PostgreSQL lub SQL Server?

SQLite jest lekką, wydajną i łatwą w implementacji bazą danych, która nie wymaga instalowania serwera, co idealnie pasuje do lokalnych aplikacji. Jest także przenośna, co oznacza, że cała baza danych mieści się w jednym pliku. W przypadku bardziej złożonych systemów, takich jak MySQL czy PostgreSQL, potrzebna byłaby dodatkowa konfiguracja, co mogłoby być nadmiarowe dla mojego projektu.

1. Czy WPF i XAML są odpowiednie do skalowalnych aplikacji? Jakie wyzwania wiążą się z ich użyciem?

Tak, WPF i XAML są odpowiednie do skalowalnych aplikacji, szczególnie dzięki mechanizmowi bindingu danych, wsparciu dla MVVM i obsłudze stylów. Jednak mogą pojawić się wyzwania, takie jak zarządzanie wydajnością w aplikacjach o dużej liczbie elementów wizualnych, czy bardziej złożona konfiguracja w porównaniu do prostszych frameworków. Krótko ujmując – rozbudowując aplikację jesteśmy elastyczni ale aby zachować wydajność musielibyśmy wprost proporcjonalnie do wykonanej naszej pracy optymalizować procesy, ładowania i zrzucania danych z pamięci na każdym etapie działania aplikacji.

1. Jakie alternatywy technologiczne rozważałeś podczas projektowania aplikacji? Dlaczego ostatecznie z nich zrezygnowałeś?

Rozważałem użycie WinForms jako prostszej alternatywy dla WPF, jednak brak nowoczesnych funkcji interfejsu użytkownika oraz brak grafiki wektorowej zadecydował o jego odrzuceniu. Rozważałem także użycie MySQL jako bazy danych, jednak wymagałoby to instalacji serwera, co zwiększyłoby złożoność wdrożenia i zmieniłoby nasze założenia odnośnie lokalności i niezależności aplikacji.

**Pytania dotyczące projektu:**

1. Jakie problemy rozwiązuje Twoja aplikacja w porównaniu do istniejących systemów takich jak Museeo czy ArtSaas?

Moja aplikacja działa lokalnie, co eliminuje zależność od internetu i opłat abonamentowych. Zapewnia pełną kontrolę nad danymi, umożliwia personalizację funkcji i integrację specyficznych rozwiązań, które odpowiadają konkretnym potrzebom muzeów.

1. Czy Twoja aplikacja jest skalowalna i czy możliwe jest jej przeniesienie do środowiska chmurowego?

Tak, aplikacja jest skalowalna. Przeniesienie do środowiska chmurowego wymagałoby zmiany bazy danych na np. MySQL lub PostgreSQL oraz zaimplementowania API do obsługi komunikacji. Można też podejść do problemu inaczej i zintegrować usługi Google aby baza zapisywała się do chmury za pomocą API.

1. Jakie trudności napotkałeś podczas implementacji projektu? Jak je rozwiązałeś?

**Pytania dotyczące funkcjonalności:**

1. Jak działa proces dodawania i edytowania eksponatów? Jak zapewniłeś spójność danych w bazie? prezentacja
2. W jaki sposób realizowana jest wysyłka e-maili? Jakie mechanizmy do tego wykorzystałeś? prezentacja
3. Jakie dane zawierają generowane raporty? W jaki sposób użytkownicy mogą z nich korzystać? prezentacja
4. Jak zaimplementowałeś funkcję historii zmian eksponatów i logowań? Czy działa to w czasie rzeczywistym? prezentacja
5. Czy aplikacja wspiera wielojęzyczność? Jeśli nie, w jaki sposób można to zaimplementować?

**Pytania dotyczące przyszłego rozwoju: Takiego typu mogą być?**

1. Jakie funkcjonalności można by dodać w przyszłości, aby zwiększyć użyteczność aplikacji?
2. Czy rozważałeś implementację funkcji opartych na sztucznej inteligencji, np. do analizy zbiorów muzealnych?
3. Jak zmodyfikowałbyś aplikację, aby działała w środowisku sieciowym, a nie tylko lokalnie?
4. Czy możliwe jest stworzenie aplikacji mobilnej jako rozszerzenia Twojego projektu? Jakie technologie byś wybrał?

**Pytania teoretyczne i ogólne:**

1. Wyjaśnij, jak działa WPF i w jaki sposób XAML współpracuje z logiką aplikacji w C#.

WPF to framework umożliwiający tworzenie interfejsów użytkownika. XAML opisuje warstwę wizualną w postaci XML-a, co pozwala na oddzielenie UI od logiki aplikacji. Logika biznesowa jest zaimplementowana w C#, a elementy UI są dynamicznie wiązane z danymi przez binding.

1. Jakie wzorce projektowe wykorzystałeś w aplikacji? Dlaczego je wybrałeś?

Użyłem wzorca MVVM (Model-View-ViewModel), ponieważ umożliwia on separację logiki biznesowej od warstwy wizualnej, co ułatwia testowanie i rozwój aplikacji.

1. Czym różni się SQLite od innych systemów baz danych? Jakie są jego ograniczenia?

SQLite nie wymaga serwera, jest bardzo szybki i lekki. Jednak nie obsługuje równoczesnych transakcji na dużą skalę, co czyni go mniej odpowiednim dla aplikacji wielodostępnych.

**Czym jest nasza aplikacja:**

Nasza aplikacja została napisana z wykorzystaniem WPF i C#. Jest to uwarunkowane u podstaw tym, że nasze ostatnie lata na uczelni były związane właśnie z tym językiem i chcieliśmy nie tylko wykorzystać tę widzę i przetestować ale także poszerzyć tworząc coś własno co przysłuży się innym. Ale nie tylko to było powodem – postawiliśmy sobie za cel zrobić aplikację bazodanową opartą w pełni o usługi lokalne dla instytucji muzealnych, które potrzebują aplikacji o niskim progu wejścia pod kątem szkoleń z obsługi, złożoności systemu i itp.

System jest uniwersalny tzn. nie koniecznie musi znaleźć się w muzeum (muzeum jest przykładem z racji chęci wypełnienia luki systemem) ponieważ bez problemu może posłużyć np. bibliotece (książki zbiory), wypożyczalni (przedmiot i wypożyczający) i itp. Tutaj przechodzimy do drugiej dużej zalety – jest to system zaprojektowany w MVVM a to oznacza możliwość modyfikowania naszego projektu poprzez: bindowanie w widoku i wykorzystanie już napisanych przez nas metod w modelu. Oczywiście nic nie stoi na przeszkodzie aby zmodyfikować nieco projekt pod klienta preferencje jednak fundament pozostaje taki sam.

Warto zwrócić uwagę na zasadność użycia SQLite do naszej pracy – bo oprócz oczywistych zalet jak lekkość, szybkość ważnym elementem jest też dla nas możliwość pliku .db. Co to oznacza? Otóż w małych aplikacjach baza jest wystarczająca jednak gdyby zaszła potrzeba przeniesienia bazy do innego serwisu czy po prostu archiwizacja w chmurze to jest to bardzo proste do przeprowadzenia z racji ustandaryzowania tego formatu u jego podstaw. Zatem nie ograniczamy administratora tego systemu a wręcz ułatwiamy mu modyfikację danych i ich moderowanie. Oczywiście hasła w bazie są przed nim ukryte funkcją hasz SHA256 a sama baza gdyby utworzyć kopię zabezpieczona kluczem symetrycznym AES.

Wyjaśnienie tabel specjalnych: current\_user to tabela celowo przechowująca sam login obecnego usera. Jest tak ze względów bezpieczeństwa aby dane użytkownika ale też ich obróbka była prosta w opanowaniu – bowiem do sytuacji takich jak wysyłanie maili czy rejestr zdarzeń w tabeli historia – jest potrzebny tylko login. Czyści się po zamknięciu.

Tabela historia jest oparta wyłącznie na triggerach czyli wyzwalaczach, które kiedy coś zadzieje się w bazie to wykonają (w tym przypadku) rejestr o zmianie w bazie – służy głównie administratorowi jako element jego uprawnień.

Optymalizacja aplikacji – polegała głównie na zwolnieniu zasobów (połączeniu DbConnect i Window) za pomocą funkcji Dispose() i Close() z wykorzystaniem instrukcji using. Pozwoliło to nam na większą kontrolę zasobów systemu. Natomiast samo wykorzystanie WPF, który wykorzystuje grafikę wektorową zamiast rastrowej pozwoliło nam na zmniejszenie wymagań aplikacji w tym przede wszystkim wykorzystanie ram-u na poziomie ~180 MB. Zrzucenie rejestrowania na bazę a przy tym przyspieszanie procesu rejestracji zdarzeń.

### ****Funkcja haszująca****

Funkcja haszująca to algorytm, który przekształca dane wejściowe w unikalny ciąg znaków o stałej długości (hash). Jest jednokierunkowa, co oznacza, że nie można łatwo odzyskać oryginalnych danych. Służy np. do przechowywania haseł (SHA-256, bcrypt).

### ****Szyfrowanie AES****

AES (Advanced Encryption Standard) to algorytm szyfrowania symetrycznego, który używa tego samego klucza do szyfrowania i deszyfrowania. Jest bezpieczny i stosowany np. w komunikacji, bankowości i plikach ZIP.