

Aplikacja WPF z wykorzystaniem architektury MVVM

Celem tego projektu jest zaprojektowanie i implementacja zaawansowanej aplikacji z wykorzystaniem technologii Windows Presentation Foundation (WPF) i wzorca Model-View-ViewModel (MVVM).

Temat: Aplikacja do zarządzania biblioteką książek

Podstawowe Wymagania:

1. **Interfejs użytkownika:** Użytkownik powinien być w stanie przeglądać listę książek, dodawać nowe książki, edytować istniejące książki i usuwać książki z biblioteki.
2. **Detale książki:** Każda książka powinna zawierać takie informacje jak tytuł, autor, rok wydania, wydawnictwo, gatunek i krótki opis.
3. **Wyszukiwanie książek:** Użytkownik powinien mieć możliwość wyszukiwania książek według różnych kryteriów, takich jak tytuł, autor, gatunek itp.
4. **Sortowanie i filtracja:** Użytkownik powinien mieć możliwość sortowania i filtrowania listy książek według różnych kryteriów.

Dodatkowe Wymagania:

1. **Zaawansowane funkcje wyszukiwania:** Umożliwić użytkownikowi wykonywanie zaawansowanych zapytań wyszukiwania, takich jak wyszukiwanie książek wydanych w określonym roku, książek określonego autora itp.
2. **Wypożyczanie książek:** Użytkownik powinien być w stanie wypożyczać i zwracać książki. System powinien śledzić, które książki są wypożyczone i kto je wypożyczył.
3. **Powiadomienia:** Użytkownik powinien otrzymywać powiadomienia o zbliżającym się terminie zwrotu wypożyczonej książki.

Oczekiwane Umiejętności:

1. **Zrozumienie i zastosowanie wzorca MVVM:** Widoki powinny być oddzielone od logiki biznesowej za pomocą ViewModeli.
2. **Używanie wiązań danych:** Wszystkie dane wyświetlane w interfejsie użytkownika powinny być wiązane z danymi w ViewModelach.
3. **Komendy:** Akcje użytkownika, takie jak kliknięcie przycisku, powinny być obsługiwane za pomocą komend, a nie bezpośrednio w kodzie widoku.
4. **Używanie usług:** Logika biznesowa, taka jak ładowanie i zapisywanie danych, powinna być obsługiwana przez serwisy.

Stworzenie RESTful API z połączeniem do bazy danych

Celem tego projektu jest zaprojektowanie i implementacja RESTful API, które łączy się z bazą danych.

Temat: API do zarządzania systemem rezerwacji hoteli

Podstawowe Wymagania:

1. **Modele danych:** Powinieneś utworzyć modele danych reprezentujące hotele, pokoje i rezerwacje. Każdy hotel powinien mieć nazwę, adres, ocenę i listę pokoi. Każdy pokój powinien mieć numer, typ (np. jednoosobowy, dwuosobowy) i cenę. Każda rezerwacja powinna mieć datę początkową, datę końcową, informacje o pokoju i dane klienta.
2. **Operacje CRUD:** Twoje API powinno obsługiwać podstawowe operacje CRUD (Utwórz, Odczytaj, Zaktualizuj, Usuń) dla każdego z modeli danych.
3. **Autentykacja i autoryzacja:** Powinieneś zaimplementować prosty system autentykacji i autoryzacji. Tylko zautoryzowani użytkownicy powinni mieć dostęp do operacji CRUD.

Dodatkowe Wymagania:

1. **Wyszukiwanie:** Użytkownik powinien być w stanie wyszukiwać hotele według nazwy, adresu lub oceny. Powinien też być w stanie wyszukiwać dostępne pokoje w danym hotelu według typu i ceny.
2. **Rezerwacje:** Użytkownik powinien być w stanie dokonywać rezerwacji. System powinien sprawdzać, czy dany pokój jest dostępny w wybranym terminie.
3. **Raporty:** Użytkownik powinien być w stanie generować raporty dotyczące rezerwacji, takie jak liczba rezerwacji w danym miesiącu, średnia długość pobytu, najbardziej popularne hotele itp.

Oczekiwane Umiejętności:

1. **Zrozumienie i zastosowanie REST:** Twoje API powinno stosować zasady architektury REST.
2. **Połączenie z bazą danych:** Powinieneś korzystać z jakiegoś ORM (np. Entity Framework dla .NET), aby łączyć się z bazą danych i wykonywać na niej operacje.
3. **Autentykacja i autoryzacja:** Powinieneś zaimplementować mechanizm zabezpieczający dostęp do danych.
4. **Testy jednostkowe i integracyjne:** Powinieneś napisać testy jednostkowe dla swojego API, sprawdzające zarówno scenariusze poprawne, jak i niepoprawne.

Aplikacja webowa z wykorzystaniem ASP.NET MVC

Celem tego projektu jest zaprojektowanie i implementacja aplikacji webowej z wykorzystaniem frameworka ASP.NET MVC.

Temat: Aplikacja do zarządzania zasobami firmy

Podstawowe Wymagania:

1. **Modele danych:** Powinieneś utworzyć modele danych reprezentujące zasoby, pracowników i działy. Każdy zasób powinien mieć nazwę, opis, kategorię (np. sprzęt IT, biurowy, pojazd) i status

(np. dostępny, w użyciu, w naprawie). Każdy pracownik powinien mieć imię, nazwisko, stanowisko i dział. Każdy dział powinien mieć nazwę i listę pracowników.

2. **Operacje CRUD:** Twoja aplikacja powinna obsługiwać podstawowe operacje CRUD (Utwórz, Odczytaj, Zaktualizuj, Usuń) dla każdego z modeli danych.
3. **Autentykacja i autoryzacja:** Powinieneś zaimplementować prosty system autentykacji i autoryzacji. Tylko zautoryzowani użytkownicy powinni mieć dostęp do operacji CRUD.

Dodatkowe Wymagania:

1. **Wyszukiwanie:** Użytkownik powinien być w stanie wyszukiwać zasoby według nazwy, kategorii lub statusu. Powinien też być w stanie wyszukiwać pracowników według imienia, nazwiska, stanowiska lub działu.
2. **Rezerwacje zasobów:** Użytkownik powinien być w stanie rezerwować zasoby dla pracowników. System powinien śledzić, który zasób jest aktualnie w użyciu i przez którego pracownika.
3. **Raporty:** Użytkownik powinien być w stanie generować raporty dotyczące zasobów, takie jak ilość dostępnych zasobów, zasoby w użyciu, najbardziej używane zasoby itp.

Oczekiwane Umiejętności:

1. **Zrozumienie i zastosowanie ASP.NET MVC:** Powinieneś zrozumieć i zastosować zasady działania architektury MVC (Model-View-Controller).
2. **Połączenie z bazą danych:** Powinieneś korzystać z Entity Framework, aby łączyć się z bazą danych i wykonywać na niej operacje.
3. **Autentykacja i autoryzacja:** Powinieneś zaimplementować mechanizm zabezpieczający dostęp do danych.
4. **Testy jednostkowe:** Powinieneś napisać testy jednostkowe dla swojego kodu, sprawdzające zarówno scenariusze poprawne, jak i niepoprawne.

Aplikacja Webowa z wykorzystaniem Blazor

Celem tego projektu jest zaprojektowanie i implementacja aplikacji webowej z wykorzystaniem technologii Blazor.

Temat: Aplikacja do zarządzania systemem zamówień w restauracji

Podstawowe Wymagania:

1. **Modele danych:** Powinieneś utworzyć modele danych reprezentujące posiłki, zamówienia i klientów. Każdy posiłek powinien mieć nazwę, opis, cenę i kategorię (np. przystawki, dania główne, desery). Każde zamówienie powinien zawierać listę posiłków, datę i czas zamówienia, dane klienta i status (np. nowe, w trakcie przygotowania, gotowe do odbioru). Każdy klient powinien mieć imię, nazwisko i numer telefonu.
2. **Operacje CRUD:** Twoja aplikacja powinna obsługiwać podstawowe operacje CRUD (Utwórz, Odczytaj, Zaktualizuj, Usuń) dla każdego z modeli danych.

3. **Autentykacja i autoryzacja:** Powinienes zaimplementować prosty system autentykacji i autoryzacji. Tylko zautoryzowani użytkownicy powinni mieć dostęp do operacji CRUD.

Dodatkowe Wymagania:

1. **Złożenie zamówienia:** Klienci powinni mieć możliwość składania zamówień za pomocą interaktywnego menu.
2. **Śledzenie statusu zamówienia:** Klienci powinni mieć możliwość sprawdzenia statusu swojego zamówienia.
3. **Powiadomienia:** Klienci powinni otrzymywać powiadomienia o zmianie statusu ich zamówienia.
4. **Raporty:** Użytkownicy powinni mieć możliwość generowania raportów dotyczących zamówień, takich jak liczba zamówień na dzień, najpopularniejsze posiłki, średnia wartość zamówienia itp.

Oczekiwane Umiejętności:

1. **Zrozumienie i zastosowanie Blazor:** Powinienes zrozumieć i zastosować zasady działania technologii Blazor.
2. **Połączenie z bazą danych:** Powinienes korzystać z Entity Framework, aby łączyć się z bazą danych i wykonywać na niej operacje.
3. **Autentykacja i autoryzacja:** Powinienes zaimplementować mechanizm zabezpieczający dostęp do danych.
4. **Testy jednostkowe i funkcjonalne:** Powinienes napisać testy jednostkowe i funkcjonalne dla swojego kodu, sprawdzające zarówno scenariusze poprawne, jak i niepoprawne.

Aplikacja Mobilna z wykorzystaniem .NET MAUI

Celem tego projektu jest zaprojektowanie i implementacja aplikacji mobilnej z wykorzystaniem platformy .NET MAUI (Multi-platform App UI).

Temat: Aplikacja do zarządzania systemem planowania zadań

Podstawowe Wymagania:

1. **Modele danych:** Powinienes utworzyć modele danych reprezentujące zadania, kategorie i użytkowników. Każde zadanie powinno mieć nazwę, opis, datę i czas, priorytet i przypisaną kategorię. Każda kategoria powinna mieć nazwę i listę zadań. Każdy użytkownik powinien mieć imię, nazwisko, adres e-mail i listę zadań.
2. **Operacje CRUD:** Twoja aplikacja powinna obsługiwać podstawowe operacje CRUD (Utwórz, Odczytaj, Zaktualizuj, Usuń) dla każdego z modeli danych.
3. **Autentykacja i autoryzacja:** Powinienes zaimplementować prosty system autentykacji i autoryzacji. Tylko zautoryzowani użytkownicy powinni mieć dostęp do operacji CRUD.

Dodatkowe Wymagania:

1. **Powiadomienia:** Użytkownicy powinni otrzymywać powiadomienia o zbliżających się zadaniach.

2. **Sortowanie i filtrowanie:** Użytkownicy powinni mieć możliwość sortowania i filtrowania zadań według różnych kryteriów (np. daty, priorytetu, kategorii).
3. **Synchronizacja z chmurą:** Zadania powinny być synchronizowane z chmurą, aby użytkownik mógł mieć do nich dostęp na różnych urządzeniach.

Oczekiwane Umiejętności:

1. **Zrozumienie i zastosowanie .NET MAUI:** Powinieneś zrozumieć i zastosować zasady działania platformy .NET MAUI.
2. **Tworzenie interfejsu użytkownika:** Powinieneś umieć tworzyć atrakcyjny i użyteczny interfejs użytkownika z wykorzystaniem XAML.
3. **Połączenie z bazą danych:** Powinieneś korzystać z Entity Framework, aby łączyć się z lokalną bazą danych i wykonywać na niej operacje.
4. **Autentykacja i autoryzacja:** Powinieneś zaimplementować mechanizm zabezpieczający dostęp do danych.
5. **Testy jednostkowe i interfejsu użytkownika:** Powinieneś napisać testy jednostkowe dla swojego kodu, a także testy interfejsu użytkownika, sprawdzające działanie aplikacji na różnych platformach.

Praca z Azure i GitHub CI/CD

Celem tego projektu jest zaprojektowanie i implementacja systemu zarządzania treścią (CMS) z wykorzystaniem Azure Services, oraz zastosowanie praktyk DevOps z użyciem GitHub CI/CD.

Temat: System zarządzania treścią (CMS)

Podstawowe Wymagania:

1. **Azure SQL Database:** Użyj Azure SQL Database do przechowywania danych aplikacji. Zaprojektuj i zaimplementuj strukturę bazy danych, która będzie przechowywać treści, takie jak strony, posty bloga, użytkowników i role.
2. **Azure Web Apps:** Użyj Azure Web Apps do hostowania aplikacji webowej. Aplikacja powinna oferować funkcjonalność CRUD (Utwórz, Odczytaj, Zaktualizuj, Usuń) dla różnych typów treści.
3. **Azure Virtual Machines:** Użyj Azure Virtual Machines do hostowania środowiska testowego. Testuj swoją aplikację w tym środowisku przed jej wdrożeniem.
4. **GitHub CI/CD:** Użyj narzędzi CI/CD dostępnych na GitHubie do automatyzacji procesu budowy, testowania i wdrażania aplikacji. Ustaw, aby po każdym commitcie do głównej gałęzi kod był automatycznie budowany, testowany i wdrażany na środowisko testowe.

Dodatkowe Wymagania:

1. **Autentykacja i autoryzacja:** Zaimplementuj system autentykacji i autoryzacji z wykorzystaniem Azure Active Directory. Tylko zautoryzowani użytkownicy powinni mieć dostęp do operacji CRUD.

2. **Azure Storage:** Użyj Azure Storage do przechowywania i serwowania mediów, takich jak obrazy i pliki do pobrania.
3. **Monitorowanie i logowanie:** Użyj Azure Monitor i Application Insights do monitorowania wydajności aplikacji i logowania błędów.

Oczekiwane Umiejętności:

1. **Zrozumienie i zastosowanie usług Azure:** Powinieneś zrozumieć i zastosować różne usługi Azure wymienione w wymaganiach.
2. **Praca z bazą danych:** Powinieneś znać podstawy pracy z bazą danych SQL, takie jak tworzenie tabel, pisanie zapytań SQL, projektowania schematu bazy danych.
3. **Zrozumienie i zastosowanie DevOps:** Powinieneś zrozumieć i zastosować praktyki DevOps, takie jak integracja ciągła (CI), dostarczanie ciągłe (CD), testowanie i monitorowanie.
4. **Praca z GitHub:** Powinieneś znać podstawy pracy z GitHubem, takie jak zarządzanie repozytorium, korzystanie z funkcji CI/CD