

## 1. Desktopна версія з використанням WPF

### 1.1 Огляд:

Windows Presentation Foundation (WPF) - це фреймворк інтерфейсу користувача для створення додатків для Windows. СКБД, побудована на WPF, встановлюється на локальних машинах користувачів, має графічний інтерфейс і взаємодіє з локальними базами даних.

### 1.2 Переваги:

- **Багатий досвід роботи з інтерфейсом користувача:** WPF включає в себе інструменти для проектування інтерактивних інтерфейсів користувача.
- **Продуктивність:** Додатки виконуються локально, що може бути перевагою для завдань з інтенсивною обробкою даних.
- **Відсутність залежності від браузера:** Додатки WPF працюють безпосередньо в Windows, що дозволяє уникнути проблем, пов'язаних з кроссбраузерною сумісністю.
- **Офлайн-функціональність:** Користувачі можуть працювати з додатком без підключення до Інтернету, що підходить для автономних середовищ.

### 1.3 Недоліки:

- **Обмежена крос-платформна підтримка:** WPF в першу чергу призначений для Windows, що робить його несумісним з macOS або Linux.
- **Розгортання та обслуговування:** Desktopні додатки вимагають ручного встановлення та оновлення, що може ускладнити розгортання порівняно з веб-додатками.
- **Обмежена масштабованість:** Кожен користувач повинен встановити додаток на свій пристрій.
- **Проблеми з безпекою:** Локальні бази даних можуть стикатися з ризиками безпеки, а підтримка безпеки на декількох машинах може бути складнішою, ніж захист централізованих систем.

### 1.4 Відповідні випадки використання:

- Середовища з користувачами на системах Windows.
- Програми, які вимагають автономного використання.
- Системи з невеликою кількістю користувачів, наприклад, корпоративні рішення.

## 2. Монолітна веб-архітектура з використанням ASP.NET Core MVC

### 2.1 Огляд:

ASP.NET Core MVC - це серверний веб-фреймворк для створення динамічних веб-додатків.

### 2.2 Переваги:

- **Централізоване управління:** Весь додаток упакований в єдиний проект, що може спростити розгортання та контроль версій.

- **Безпека:** Централізовані додатки дозволяють реалізувати аутентифікацію, авторизацію та контроль доступу до баз даних.
- **Незалежність від платформи:** Веб-додатки можуть працювати на різних платформах (Windows, macOS, Linux) через веб-браузери.
- **Масштабування:** Монолітні додатки можна масштабувати, розгортаючи їх на більш потужних серверах або використовуючи балансувальники навантаження.
- **Модель розробки:** Менша кількість компонентів спрощує розробку та підтримку, особливо для невеликих команд.

### 2.3 Недоліки:

- **Тісно пов'язана архітектура:** Зміни в одній частині системи можуть призвести до необхідності перебудови та перерозгортання всієї програми.
- **Обмежена гнучкість:** У міру зростання додатку, підтримка кодової бази може стати більш складною задачею.
- **Вузькі місця в продуктивності:** Якщо бізнес-логіка управляється в рамках одного процесу, це може призвести до проблем з продуктивністю.
- **Менша масштабованість для великих систем:** Великими монолітними додатками може бути складно керувати, розробляти, тестувати та розгортати як єдине ціле.

### 2.4 Відповідні варіанти використання:

- Малі та середні рішення для СУБД з тісно інтегрованою бізнес-логікою та користувацькими інтерфейсами.
- Додатки, де перевага надається уніфікованій моделі розробки.
- Проекти з помірними потребами в масштабованості і меншим акцентом на швидкі цикли розгортання.

## 3. Веб-орієнтована версія з використанням REST

### 3.1 Огляд:

СУБД на основі REST слідує принципам передачі представницького стану (Representational State Transfer, REST), організовуючи систему в окремі сервіси, які взаємодіють через HTTP. Зазвичай це передбачає розділення інтерфейсу та бек-енду, причому бек-енд надає API для взаємодії з клієнтом.

### 3.2 Переваги:

- **Розділення фронтенду та бекенду:** Дозволяє незалежно розробляти користувацький інтерфейс та серверну логіку.
- **Масштабованість:** REST-архітектури підтримують горизонтальне масштабування окремих сервісів, наприклад, сервісу бази даних.
- **Кроссплатформенність:** REST API не залежать від платформи і можуть бути доступними для різних клієнтів, включаючи веб-браузери та мобільні додатки.
- **Гнучкість інтерфейсу користувача:** розробники можуть використовувати різні технології інтерфейсу або декілька інтерфейсів, адаптованих для різних груп користувачів.

- **Сумісність з мікросервісами:** REST підтримує архітектуру мікросервісів, що дозволяє СКБД складатися з окремих сервісів, які взаємодіють через API.
- **Оновлення та підтримка:** Кожен компонент може бути незалежно оновлений або замінений.

### 3.3 Недоліки:

- **Складність:** REST-системи можуть ускладнювати дизайн, включаючи версіювання API та міжсервісну взаємодію.
- **Мережева затримка:** HTTP-зв'язок може збільшувати затримку, особливо коли задіяно кілька сервісів.
- **Узгодженість даних:** Підтримка узгодженості даних між розподіленими сервісами може бути більш складним завданням, ніж в монолітних системах.
- **Проблеми з безпекою:** Роз'єднана природа REST-систем може виявити додаткові вразливості безпеки; кожен сервіс вимагає належної автентифікації та авторизації.

### 3.4 Відповідні випадки використання:

- Системи, що вимагають високої масштабованості та гнучкості, наприклад, хмарні рішення на основі СУБД.
- Додатки, розроблені для різних типів клієнтів (веб-, мобільні, десктопні).
- Проекти, які передбачають майбутнє зростання і можуть використовувати архітектуру мікросервісів.
- Команди розробників, які мають досвід роботи як з front-end, так і з back-end технологіями.

Критерій	WPF Desktop	Монолітна архітектура	REST
UI/UX	Обширний десктопний UI	Традиційний веб UI	Гнучкі веб-технології
Кросплатформеність	Windows	Незалежність	Незалежність
Масштабованість	Обмежена на машину	В міру (вертикальна)	Висока (горизонтальна)
Продуктивність	Залежить від OEM	Залежить від розгортання	Залежить від мережі
Підтримка	Ручне завантаження і оновлення	Централізоване оновлення	Комплексний deployment та оновлення
Робота офлайн	Так	Ні	Ні
Розгортання	Ручне	Централізоване	Централізоване
Внесення оновлень	Регульоване	Просте	Врахування компонентів
Безпека	Локальні ризики	Централізована	Розподілена