

## **Обрана архітектура проектування системи**

Для створення мого застосунку використовується трирівнева архітектура.

Трирівнева архітектура - це архітектура клієнт-сервер, в якій функціональна логіка процесу, доступ до даних, зберігання даних у комп'ютері та користувальницький інтерфейс розробляються та підтримуються як незалежні модулі на окремих платформах.

Три яруси в трирівневій архітектурі:

### **1) Рівень представлення:**

На рівні представлення забезпечується взаємодія з користувачем програми - це інтерфейс користувача і рівень обміну даними. Його основне призначення полягає у відображенні інформації та отриманні її від користувача. Цей рівень може працювати у веб-браузері або як графічний інтерфейс користувача комп'ютерної або мобільної програми.

Рівень представлення веб-застосунку для моєї курсової роботи буде розроблено за допомогою *react та typescript*.

### **2) Рівень програми**

Також відомий як логічний або проміжний рівень, є центральною ланкою програми. На цьому рівні обробляється інформація, зібрана з попереднього рівня за допомогою бізнес-логіки, яка є набір бізнес-правил. Крім того, рівень програми може додавати, змінювати та видаляти дані, розташовані на рівні даних.

Рівень програми веб-застосунку (API) для моєї курсової роботи буде розроблено за допомогою *C# та технологій EntityFramework та ASP.NET Core Identity*.

### **3) Рівень даних**

Рівень даних є рівнем доступу до даних або базовим рівнем, призначеним для зберігання та управління інформації, обробленою програмою. Його роль може виконувати реляційна система управління базами даних.

Рівень даних для моєї курсової роботи буде розроблено за допомогою *MSSQL*.

## **Взаємодія додатку з сервером**

Фронтенд частина буде відправляти дані на сервер за допомогою бібліотеки *axios*. Axios це широко відома JavaScript-бібліотека. Вона є

HTTP-клієнтом, заснованим на промісах і призначена для браузерів і Node.js.

### Взаємодія додатку з локальною базою даних

Сервер, у свою чергу, відправляє дані до бази даних за допомогою *EntityFramework* та *ASP.NET Core Identity*.

Entity Framework – це рішення для роботи з базами даних, яке використовується у програмуванні мовами сімейства .NET. Воно дозволяє взаємодіяти з СУБД з допомогою сутностей (entity), а не таблиць.

ASP.NET Identity представляє вбудовану в ASP.NET систему автентифікації та авторизації. Ця система дозволяє користувачам створювати облікові записи, автентифікуватися, керувати обліковими записами або використовувати для входу на сайт облікові записи зовнішніх провайдерів, таких як Facebook, Google, Microsoft, Twitter та інші.

### Діаграма прецедентів

