# Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра “Системи автоматизованого проектування”

**ЗВІТ**

про проходження проектно-технологічної практики

студента ІV курсу групи КН-409

Погуляєв Владислав

**База практики** \_\_\_\_\_\_\_\_НУ «ЛП» м. Львів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(назва та адреса підприємства)

**Термін практики** з «01» травня 2023 р. по з «13» травня 2023 р.

**Керівники практики**

від кафедри \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_к.т.н., доцент каф САП Климкович Т.А.

(підпис) (наук.ст, вч.звання, прізвище, ім’я, по батькові)

від бази практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (посада, прізвище, ім’я, по батькові)

**ОЦІНКА** \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Дата** \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема БКР \_\_\_\_ Платформа для благодійних пожертвувань та волонтерської діяльності \_

Керівник БКР \_\_\_\_\_\_\_\_\_\_\_\_ Станкевич О.М.\_\_\_

(підпис) (Посада, Прізвище, ініціали)

Львів – 2023

ЗМІСТ

[ВСТУП 9](#_Toc484461264)

[РОЗДІЛ 1. Платформа для благодійних пожертвувань та волонтерської діяльності 10](#_Toc484461265)

[1.1. Платформа для благодійних пожертвувань та волонтерської діяльності. Поняття та види 10](#_Toc484461266)

[1.2. Приклади існуючих платформ для благодійних пожертвувань та волонтерської діяльності. 10](#_Toc484461267)

[1.2.1. GoFundMe 10](#_Toc484461268)

[1.2.2 DonorsChoose 12](#_Toc484461269)

[1.2.3 VolonteerMatch 13](#_Toc484461270)

[1.2.4 GlobalGiving 15](#_Toc484461271)

[1.2.5 Patreon 16](#_Toc484461272)

[1.3 Збереження даних 18](#_Toc484461274)

[1.3.1 JSON 18](#_Toc484461275)

[1.3.2 База даних 19](#_Toc484461278)

[Висновок 20](#_Toc484461279)

[РОЗДІЛ 2. ВИКОРИСТАНІ ЗАСОБИ РОЗРОБКИ 21](#_Toc484461280)

[2.1. ASP.NET 21](#_Toc484461281)

[2.2. Entity Framework 24](#_Toc484461282)

[2.3. ReactJS 26](#_Toc484461283)

[2.4. ReduxJS 28](#_Toc484461284)

[2.5. Visual Studio 30](#_Toc484461284)

[2.6. Git 30](#_Toc484461284)

[2.7. SQL мова 33](#_Toc484461284)

# ВСТУП

Платформа для благодійних пожертвувань та волонтерської діяльності є актуальним проектом в сучасному світі. За останні кілька років, зацікавленість людей до благодійності та волонтерства зросла. За допомогою такої платформи, люди можуть з легкістю знайти організації, яким вони можуть допомогти, а також знайти волонтерські можливості для своєї участі у різних проектах.

Метою розробки цієї платформи є полегшення процесу благодійного пожертвування та волонтерської діяльності для користувачів. Крім того, платформа надає можливість швидко знайти необхідну інформацію про благодійні організації та проекти, що можуть зацікавити користувачів.

Науково-технічною новизною цього проекту є те, що він дозволяє пожертвувати кошти на різні проекти з усього світу, а також створювати власні оголошення про благодійну допомогу. Основним розв'язком є створення платформи, яка працює з локальними та віддаленими базами даних.

Однією з головних проблем, що вирішує цей проект, є ускладнення пошуку інформації про благодійні організації та волонтерські можливості. Завдяки розробленій платформі, користувачі можуть швидко знайти необхідну інформацію, зареєструватися як волонтери, а також приєднатися до організацій для допомоги в їхніх проектах.

**РОЗДІЛ 1. Платформа для благодійних пожертвувань та волонтерської діяльності**

# 1.1. Платформа для благодійних пожертвувань та волонтерської діяльності. Поняття та види

Платформа для благодійних пожертвувань та волонтерської діяльності – це програмне забезпечення, яке дає змогу створювати волонтерські оголошення та переглядати їх , використовуючи базу даних (БД).

База даних визначають як середовище зберігання даних, які організовані за визначеними правилами. Існують різні критерії класифікації баз даних. За моделлю організації даних можна виділити такі типи:

* Ієрархічна база даних, яка складається з об’єктів різних рівнів та має зв’язки “предок-нащадок”.
* Мережна база даних, яка має схожу структуру з ієрархічною, але об’єкт може мати декілька предків.
* Реляційна база даних, що зберігає дані у вигляді таблиць та є найпоширенішою моделлю серед систем керування базами даних (СКБД).
* Об’єктно-орієнтована база даних, де дані відображають у вигляді об’єктів.

За розміщенням даних, бази даних поділяють на локальні (централізовані) та розподілені, які можуть бути виконані на різних платформах, таких як веб, стаціонарні системи, мобільні додатки та системи для влаштованих пристроїв.

# 1.2. Приклади існуючих платформ для благодійних пожертвувань та волонтерської діяльності.

# 1.2.1. GoFundMe

GoFundMe – це онлайн-платформа збору коштів для різних цілей, від медичних потреб до благодійності та технологій. Заснована в 2010 році, платформа допомагає людям зібрати кошти на проекти та цілі, що мають значення для них та їхніх близьких.



Рис. 1.1. Логотип GoFundMe.

За допомогою GoFundMe користувачі можуть створювати особисті сторінки та розповідати про свої історії, мету та потребу в коштах. Люди можуть додавати фотографії та відео, щоб краще розповісти про своє запитання на допомогу. Користувачі також можуть ділитися своїми сторінками з друзями та родичами, які можуть пожертвувати на їхню підтримку.

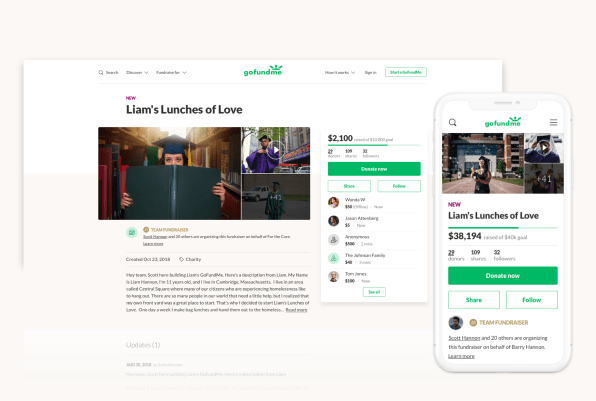


Рис. 1.2. Інтерфейс GoFundMe.

Одна з найбільш важливих функцій GoFundMe полягає в тому, що користувачі можуть збирати гроші швидко та ефективно, не залежно від того, де вони живуть. Таким чином, люди можуть отримати допомогу та підтримку з усього світу.

GoFundMe також є корисною для благодійних організацій та некомерційних організацій, які можуть створювати свої власні сторінки та збирати кошти на благодійні проекти та інші потреби. Це може допомогти підвищити увагу до їхніх проблем та підтримати їхню місію.

# 1.2.2 DonorsChoose

# DonorsChoose – це онлайн-платформа для збору коштів на проекти, які пов'язані з освітою. Заснована в 2000 році, платформа допомагає вчителям з США створювати проекти та збирати кошти на придбання матеріалів та обладнання для своїх класів.

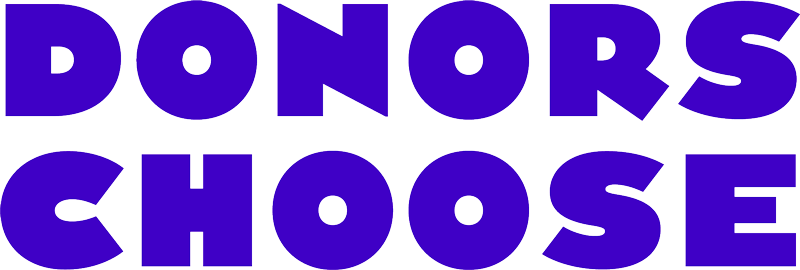


Рис. 1.3. Логотип DonorsChoose.

За допомогою DonorsChoose вчителі можуть створювати особисті сторінки та описувати свої проекти та потреби. Люди можуть знайти проекти, які їм цікаві та пожертвувати на їхню підтримку. Користувачі також можуть ділитися своїми сторінками з друзями та родичами, щоб залучити ще більше підтримки.

Одна з головних переваг DonorsChoose полягає в тому, що це дає змогу вчителям отримати необхідні матеріали та обладнання для своїх класів, які можуть допомогти покращити якість освіти та навчання учнів. Крім того, це допомагає підвищити зацікавленість учнів та збільшити їхню мотивацію до навчання.

DonorsChoose також є корисною для тих, хто хоче пожертвувати на освітні проекти та допомогти вчителям та учням отримати необхідні ресурси. Це може допомогти зробити позитивний внесок у майбутнє дітей та підтримати освітню місію вчителів.

Узагалі, DonorsChoose є потужним інструментом для збору коштів та підтримки освітніх проектів. Це дає можливість вчителям отримати необхідні матеріали та обладнання, а учням - якісну освіту та навчання.

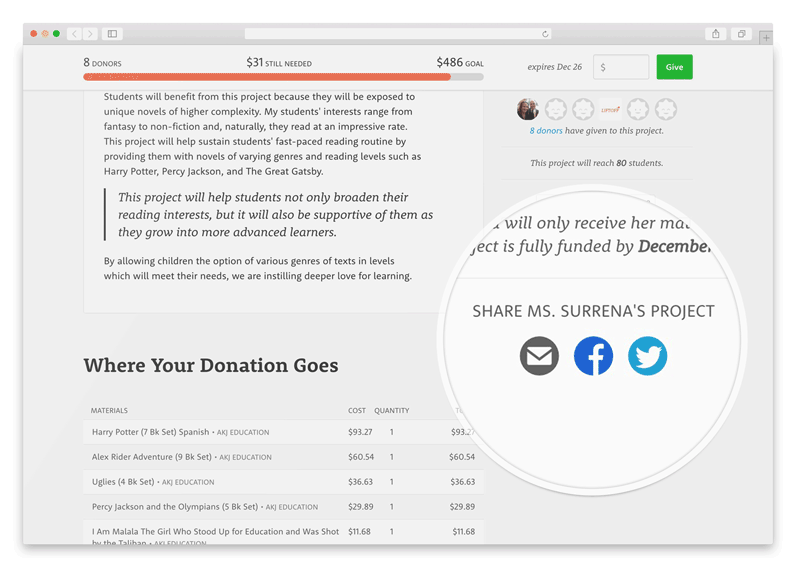


Рис. 1.4. Інтерфейс DonorsChoose.

# 1.2.3 VolunteerMatch

VolunteerMatch – це онлайн-платформа, яка об’єднує волонтерів та неприбуткові організації. Заснована в 1998 році, платформа допомагає знаходити волонтерські можливості у багатьох країнах світу, допомагаючи організаціям та волонтерам знаходити один одного.



Рис. 1.5. Логотип VolunteerMatch.

За допомогою VolunteerMatch неприбуткові організації можуть розміщувати вакансії для волонтерів та знаходити людей, які готові надати свою безоплатну допомогу. Волонтери також можуть шукати відповідні вакансії на платформі та зв'язуватися з організаціями.

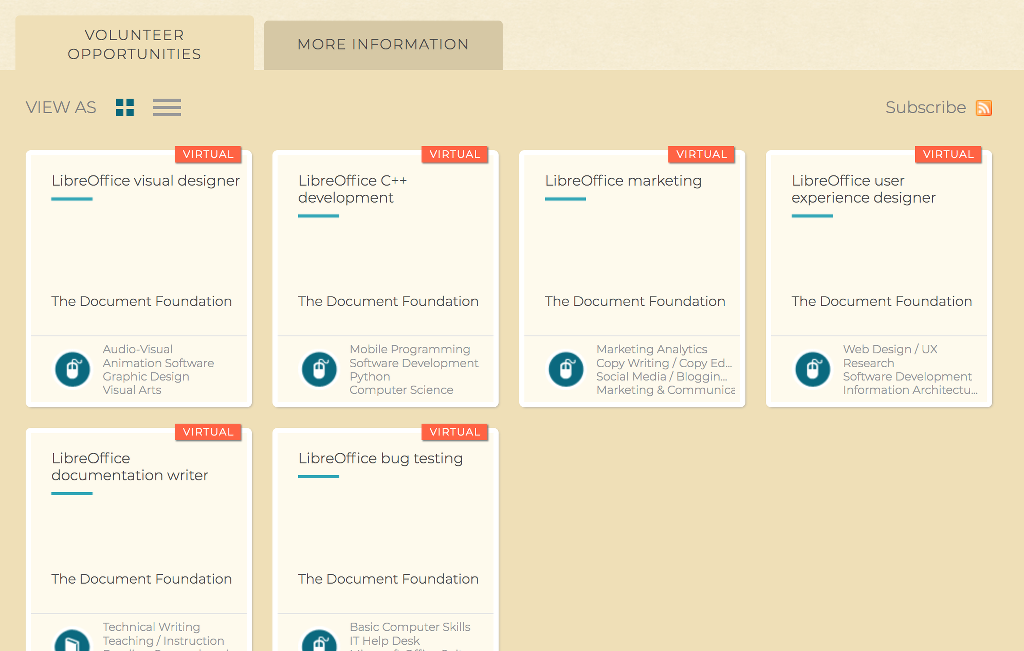


Рис. 1.6. Інтерфейс VolunteerMatch.

Одна з головних переваг VolunteerMatch полягає в тому, що волонтери можуть знайти відповідну волонтерську роботу, яка відповідає їхнім інтересам та здібностям. Також це допомагає неприбутковим організаціям знаходити відповідних волонтерів, які можуть допомогти їм виконувати їхні місії.

VolunteerMatch також є корисною для тих, хто хоче зайнятися волонтерською діяльністю та зробити позитивний внесок у свій спільноту та світ. Це може допомогти відчути себе корисним, знайти нових друзів та розвиватися в процесі надання допомоги.

Загалом, VolunteerMatch є потужним інструментом для пошуку волонтерських можливостей та збільшення відчуття корисності та зацікавленості в наданні допомоги. Це дає змогу організаціям та волонтерам знаходити один одного та виконувати важливу місію для спільної користі.

# 1.2.4 GlobalGiving

GlobalGiving - це онлайн-платформа для збору коштів, яка забезпечує зв'язок між неприбутковими організаціями та людьми, які бажають надати допомогу. Заснована в 2002 році, GlobalGiving дозволяє організаціям розміщувати свої проекти на платформі та збирати кошти від людей, які хочуть пожертвувати на підтримку цих проектів.



Рис. 1.7. Логотип GlobalGiving.

Організації можуть зареєструватися на платформі та створювати сторінки для своїх проектів, де вони можуть ділитися інформацією про свою місію, цілі та плани. Люди з усього світу можуть переглядати проекти та пожертвувати на ті, які вони хочуть підтримати.

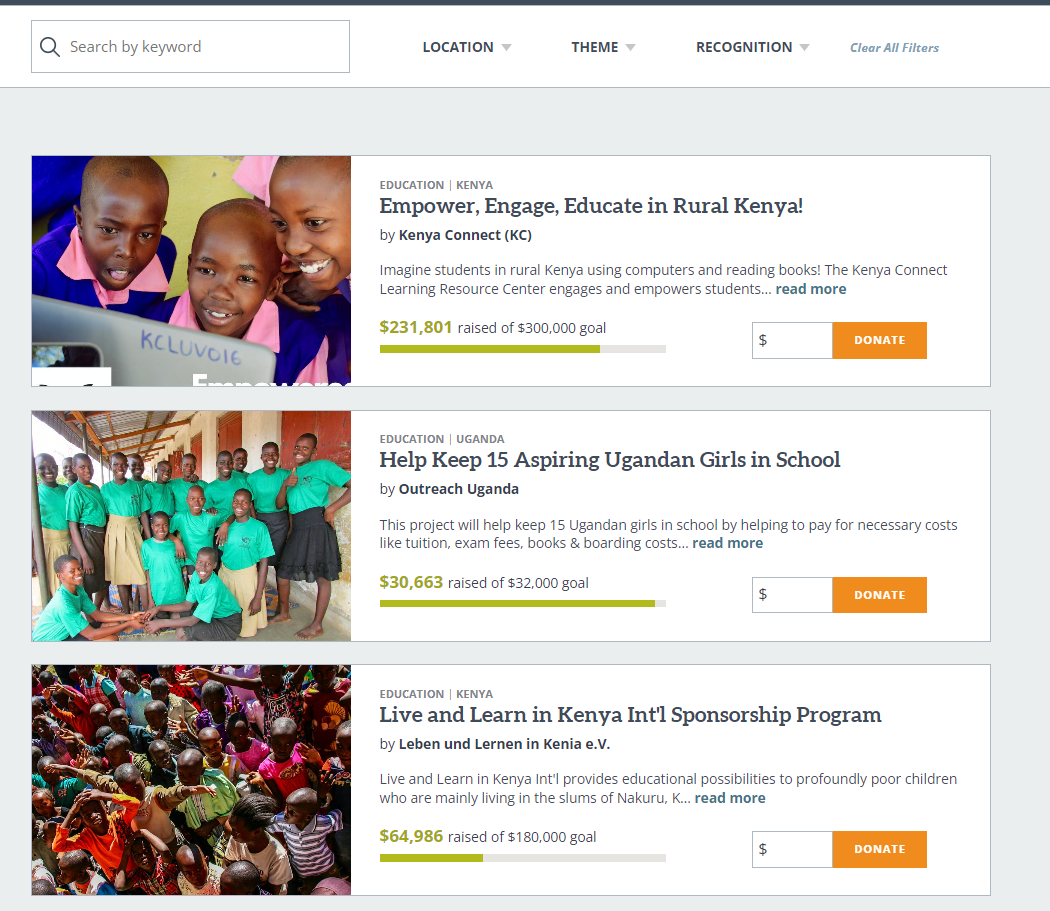


Рис. 1.8. Інтерфейс GlobalGiving.

Одна з переваг GlobalGiving полягає в тому, що платформа пропонує можливості збору коштів для проектів у більш ніж 170 країнах світу. Крім того, організації можуть отримувати фінансову підтримку не тільки від індивідуальних спонсорів, але й від компаній та інших організацій, які бажають підтримати благодійні проекти.

Ще одна перевага GlobalGiving полягає в тому, що платформа дозволяє неприбутковим організаціям отримувати доступ до різних ресурсів та інструментів для розвитку та підвищення ефективності їхньої діяльності. Наприклад, організації можуть отримувати курси з фінансового менеджменту, маркетингу та фондової роботи.

# 1.2.5 Patreon

Patreon – це онлайн-платформа, яка допомагає творчим людям отримувати гроші від своїх шанувальників, щоб вони могли продовжувати свою роботу. На платформі є багато творців від музикантів, художників та блогерів до письменників та журналістів.



Рис. 1.9. Логотип Patreon.

Платформа дозволяє творцям створювати різні рівні підписки, на які можуть підписатися їхні шанувальники. Кожен рівень підписки має свої вигоди та винагороди, які творець може пропонувати своїм підписникам, наприклад, ексклюзивний контент, ранній доступ до нових робіт або особистий доступ до творця.

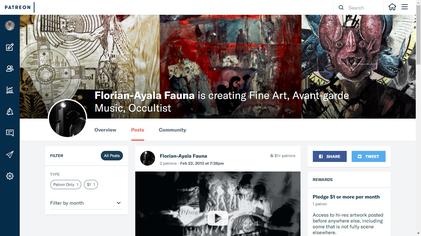


Рис. 1.10. Інтерфейс Patreon.

Творці можуть встановлювати місячну плату за кожен рівень підписки або встановлювати мінімальний поріг для платежів від своїх шанувальників. Також є можливість зробити одноразовий внесок на розвиток творчості певного автора.

Patreon забезпечує творцям можливість заробляти на своїй роботі, зокрема на зйомках відео, виготовленні музики, написанні книг чи роботі над іншими проектами. Платформа дає можливість творцям зосередитися на творенні контенту, не витрачаючи час на пошук спонсорів та збір коштів.

Підтримка платформи Patreon надає творцям можливість мати стабільний прибуток та продовжувати свою творчість, що допомагає забезпечити стабільність та розвиток культурної та творчої сфери.

# 1.3 Збереження даних

Під час роботи з великими обсягами даних потрібно забезпечити їх структуризоване зберігання. Зберігання може бути локальним або віддаленим, на сервері, і для цього використовують різні системи, такі як Block storage, File storage, Object storage та Key-value storage. Щоб вибрати оптимальну систему зберігання, потрібно провести тести та перевірити певні параметри в різних умовах.

Основні характеристики, які потрібно врахувати під час вибору системи зберігання:

• Надійність у роботі та відмовостійкість

• Швидкість відновлення даних за виникнення проблем

• Продуктивність, яка відповідає вашим запитам та потребам

• Консистентність даних для забезпечення їх коректності та цілісності.

**1.3.1 JSON**

JSON є текстовим форматом обміну даними між комп'ютерами, який може бути зрозумілим людиною і дозволяє описувати об'єкти та інші структури даних. Основне призначення JSON полягає в передачі структурованої інформації через мережу, завдяки процесу серіалізації. Формат був розроблений і популяризований Дугласом Крокфордом і головним чином використовується для написання веб-програм, зокрема технології AJAX. У порівнянні з XML, JSON займає менше місця, дозволяє складні структури в атрибутах та прямо інтерпретується за допомогою JavaScript в об'єкти. Це робить його більш придатним для серіалізації складних структур. JSON також підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші. У контексті веб-застосунків JSON доречний для обміну даними між браузером і сервером (AJAX) або між серверами (програмні HTTP-інтерфейси).

# 1.3.2 База даних

БД – це набір даних, які організовані відповідно до певної концепції, що описує їх характеристики та зв’язки між ними, і містить щонайменше одну область застосування (відповідно до стандарту ISO/IEC 2382:2015). Загалом, база даних включає схеми, таблиці, представлення, збережені процедури та інші об’єкти, і її дані організовані згідно з моделлю організації даних. Сучасні бази даних також можуть містити засоби для обробки даних та їх опис.

БД – це структурована набір даних, що відображає стан об'єктів певної предметної області та зв'язки між ними. СКБД – це програма, яка забезпечує можливість створення БД та виконання операцій з даними, які в ній зберігаються. СКБД не є окремим видом програмного забезпечення, а лише документи, з якими оперують спеціальні прикладні програми.

Загалом будь-який впорядкований набір даних можна вважати базою даних, наприклад, паперову картотеку про працівників підприємства в відділі кадрів. Застосунки для роботи з базами даних є дуже поширеними прикладними програмами, і сучасні інформаційні системи використовують СКБД для забезпечення роботи з базами даних. СКБД – це система, яка ґрунтується на програмних та технічних засобах, забезпечує визначення, створення, маніпулювання, контроль, керування та збереженням баз даних.

# Висновок до розділу 1

Для розробки Платформи для благодійних пожертвувань та волонтерської діяльності насамперед необхідно провести аналіз існуючих додатків на ринку. У розділі розглянуто найпопулярніші системи, які можна знайти, використовуючи пошуковик Google. Усі системи мають свої переваги та неділоки, деякі платформи мають доволі застарілий інтерфейс, що робить не дуже зручним їх використання, але на томість платформа Patreon має дуже хороший пророблений інтерфейс та хорошу вітальну сторінку. Усі перелічені платформи створені в форматі веб-додатків, що є логічно, оскільки це досить зручно як для людини, яка вирішить зробити пожертвування, так і для самих авторів оголошень.

Для реалізації проекту прийнято рішення використовувати СКБД MS SQL. Для початку роботи з базою даних її необхідно створити з новими таблицями та налаштувати зв’язки між ними.

# РОЗДІЛ 2. ВИКОРИСТАНІ ЗАСОБИ РОЗРОБКИ

Комп’ютери є універсальними інструментами, які здатні здійснювати складні обчислення, полегшувати повсякденні завдання, автоматизувати нудні процеси та дають змогу легко обмінюватись інформацією, незалежно від місцязнаходження. Проте, комп’ютери не мають “інтелекту”, як люди, що створює мовний бар’єр між користувачем та машинами. Для того, щоб комп’ютер виконав певну дію, необхідно подати набір інструкцій, створених людиною, оскільки сам комп’ютер не може нічого зробити самостійно.

# 2.1. ASP.NET

ASP.NET – це один з найпопулярніших фреймворків для створення веб-додатків. Він розробляється Microsoft і використовується для створення веб-сайтів, веб-додатків та веб-сервісів. ASP.NET базується на платформі .NET і використовує мови програмування C# або Visual Basic .NET для розробки веб-додатків.



Рис. 2.1. Логотип ASP.NET.

ASP.NET має багато переваг, серед яких варто виокремити:

1) Швидкість та продуктивність – ASP.NET забезпечує високу продуктивність завдяки використанню принципу компіляції на етапі встановлення та підтримки кешування на сервері.

2) Безпека – ASP.NET забезпечує безпеку веб-додатків завдяки вбудованим механізмам автентифікації, авторизації та оброблення помилок. Крім того, він має вбудований механізм захисту від атак Cross-Site Scripting (XSS) та Cross-Site Request Forgery (CSRF).

3) Масштабованість – ASP.NET забезпечує масштабованість веб-додатків завдяки використанню розподіленої архітектури та можливості використання кластерів серверів.

4) Легкість використання – ASP.NET має широкий вибір інструментів для розролення веб-додатків, включаючи інтегровану розробку середовища (IDE) Visual Studio, що дає змогу швидко створювати веб-додатки.

5) Підтримка – Microsoft надає широку підтримку ASP.NET, включаючи документацію, навчальні матеріали та форуми спільноти, що дозволяє розробникам швидко вирішувати проблеми.

ASP.NET дає моєливість розробникам створювати веб-додатки різних типів, включаючи веб-сайти, веб-додатки та веб-сервіси. Для цього ASP.NET пропонує різні типи проектів, такі як ASP.NET Web Forms, ASP.NET MVC та ASP.NET Web API. Кожен із цих типів проектів має свої особливості та спрощує розробку веб-додатків певного типу.

ASP.NET Web Forms – це тип проекту, який базується на концепції сторінок, тобто кожна сторінка має своє представлення та код, що відповідає за її функціональність. ASP.NET Web Forms дає змогу швидко створювати веб-сайти та веб-додатки з використанням компонентів, таких як текстові поля, кнопки та інші.

ASP.NET MVC – це тип проекту, який базується на концепції розділення сторінки на моделі, представлення та контролери. Кожен із цих компонентів має свою функцію під час оброблення запитів та відображення відповіді. ASP.NET MVC дає змогу швидко створювати веб-додатки з більш гнучким управлінням, представленням та логікою додатку.

ASP.NET Web API - це тип проекту, який дозволяє створювати веб-сервіси з використанням технологій RESTful. Це дає можливість розробникам створювати веб-сервіси з використанням HTTP-протоколу та стандартних форматів даних, таких як JSON та XML.

Для розробки веб-додатків з використанням ASP.NET розробникам потрібно мати знання мови програмування C# або Visual Basic .NET та веб-технологій, таких як HTML, CSS та JavaScript. Крім того, розробникам слід мати знання про архітектуру веб-додатків та підходи до розроблення з використанням ASP.NET.

ASP.NET – це потужний та популярний фреймворк для розробки веб-додатків, який забезпечує швидкість, безпечні та масштабовані веб-додатки. ASP.NET має вбудовану підтримку для багатьох інших технологій, таких як Entity Framework, SignalR, Razor та інших. Це дає можливість розробникам швидко створювати веб-додатки з використанням сучасних технологій та підходів.

Одна з головних переваг ASP.NET полягає у тому, що він забезпечує високу продуктивність та ефективність роботи з базами даних. ASP.NET має вбудовану підтримку для баз даних, таких як Microsoft SQL Server, та може використовувати інші бази даних, такі як MySQL та Oracle. Крім того, ASP.NET забезпечує високу безпеку веб-додатків, шляхом використання різних методів, таких як автентифікація та авторизація, валідація введення користувача та захист від атак типу Cross-Site Scripting (XSS) та SQL Injection.

Загалом, ASP.NET – це потужний та високопродуктивний фреймворк для розробки веб-додатків, який забезпечує розробникам широкі можливості для створення різноманітних веб-додатків та веб-сервісів. Будучи частиною платформи .NET, ASP.NET дає змогу розробникам використовувати різноманітні засоби, бібліотеки та сервіси для створення високоякісних веб-додатків.

# 2.2. Entity Framework

Entity Framework – це технологія об’єктно-орієнтованого доступу до даних (ORM), яка дозволяє розробникам працювати з базами даних, як з об’єктами у програмі. Це забезпечує високу продуктивність та ефективність роботи з базами даних та зменшує кількість коду, необхідного для створення та роботи з базами даних.



Рис. 2.2. Логотип EF Core.

Entity Framework має вбудовану підтримку різних баз даних, включаючи Microsoft SQL Server, MySQL та PostgreSQL. Він дає можливість розробникам працювати з даними за допомогою LINQ (Language Integrated Query) – розширення мови C#, яке дає змогу виконувати запити до бази даних з використанням звичайних мовних конструкцій.

Entity Framework має кілька різновидів:

1) EF Core – це платформонезалежний, легкий та швидкий виконавчий код, який працює на Windows, Linux та Mac.

2) EF6 – це більш старіший варіант EF, який працює тільки на платформі .NET Framework та має менше можливостей, ніж EF Core.

Entity Framework забезпечує можливість розробникам створювати різні типи запитів до баз даних, такі як вибірка, вставка, оновлення та видалення, використовувати транзакції для збереження даних у БД та забезпечує підтримку роботи з даними багатьох користувачів.

Entity Framework дає змогу використовувати підхід “Code First”, коли розробники можуть створювати моделі даних на основі коду класів, а потім за допомогою міграцій змінювати структуру бази даних. Крім того, EF дає можливість використовувати підхід “Database First”, коли розробники можуть створювати моделі даних на основі структури бази даних.

Підсумовуючи, Entity Framework – є потужним та універсальним інструментом для роботи з базами даних, який дає можливість розробникам швидко та ефективно працювати з даними у своїх програмах. Він дає змогу створювати складні запити до баз даних та використовувати міграції для зміни структури баз даних, що робить роботу з базами даних зручнішою та простішою.

Entity Framework також підтримує підключення до баз даних з допомогою різних провайдерів, що дозволяє розробникам використовувати той провайдер, який найбільш підходить для їх потреб.

Однією з великих переваг Entity Framework є можливість використовувати його у складі ASP.NET, що дає змогу розробникам створювати високопродуктивні веб-додатки зі зручним доступом до баз даних. Крім того, Entity Framework має декілька інструментів для покращення продуктивності, таких як кешування та пакетні операції. Вони дають можливість збільшити швидкість роботи з базами даних та зменшити кількість запитів до них, що позитивно впливає на продуктивність та ефективність додатка в цілому.

Загалом, Entity Framework є потужним та корисним інструментом для роботи з базами даних, який дає можливість розробникам швидко та легко створювати високопродуктивні та надійні програми. Він має багато можливостей та підтримує різні провайдери баз даних, що дає змогу розробникам використовувати його в різних проектах та у різних середовищах.

# 2.3. React.JS

React.js – це бібліотека для створення інтерфейсів користувача, що дозволяє розробникам створювати динамічні веб-сторінки та веб-додатки. Вона розроблена Facebook в 2011 році і відтоді стала дуже популярною у розробці фронтенду.

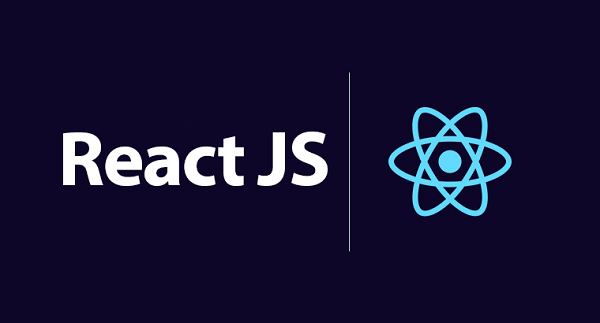


Рис. 2.3. Логотип React.JS.

React.js працює з компонентами, які є самодостатніми частинами веб-сторінки. Кожен компонент містить HTML-код, CSS-стилі та JavaScript-логіку. За допомогою компонентів, розробники можуть створювати складні інтерфейси, розділяючи їх на простіші блоки.

Один з головних принципів React.js – це “одне джерело правди” (single source of truth). Це означає, що стан компонента повинен бути збережений в одному місці, а не розподілений по всій програмі. Це забезпечує простіше управління станом та запобігає появі непередбачуваної поведінки.

Ще одним важливим аспектом React.js є використання віртуального DOM (Document Object Model). Він є копією реального DOM, але не містить реальних елементів веб-сторінки. Віртуальний DOM зберігає стан всіх компонентів та забезпечує ефективнішуу роботу з реальним DOM. Коли стан компонента змінюється, React.js змінює тільки необхідні елементи реального DOM, а не всю сторінку.

Ще одним важливим аспектом React.js є можливість використовувати JSX-синтаксис, що дає змогу вбудовувати JavaScript-код в HTML-код. Це забезпечує простішу та зрозумілішу розробку інтерфейсу.

React.js також підтримує широкий діапазон інструментів для розробки, включаючи багато корисних бібліотек, таких як Redux, React Router та Material-UI. Redux дає змогу зберігати стан додатка в одному місці, що спрощує його управління. React Router – це бібліотека для роутингу веб-додатків, що дає можливість змінювати вміст сторінки без перезавантаження сторінки, а Material-UI – це бібліотека для розроблення стильних та сучасних інтерфейсів.

React.js також працює дуже добре з іншими технологіями, такими як Node.js та Express.js. Node.js дає змогу розробникам створювати серверну частину веб-додатків, а Express.js – це фреймворк для розроблення серверних додатків на Node.js. За допомогою React.js та Node.js можна створювати повноцінні веб-додатки з різними можливостями, такими як авторизація користувачів, відправка електронної пошти, робота з базами даних та багато іншого.

React.js має також велику спільноту розробників, яка допомагає розвивати та підтримувати цю бібліотеку. У спільноті розробників React.js є багато корисних матеріалів для вивчення та розробки, таких як документація, курси, відеоуроки, блоги та інші ресурси.

У підсумку, React.js – це потужна технологія для розробки інтерфейсів користувача, яка дає моєливість створювати динамічні та привабливі веб-додатки. Вона підтримує широкий діапазон інструментів та має велику спільноту розробників, що допомагає вивчати та розвивати цю технологію. Якщо ви шукаєте потужну технологію для розробки фронтенду, React.js може бути ідеальним вибором для вас.

# 2.4. Redux.JS

Redux.js - це бібліотека для керування станом додатка у React-додатках. Вона була розроблена, щоб спростити управління станом додатка та забезпечити однорідність в управлінні станом для різних компонентів React-додатків.



Рис. 2.4. Логотип Redux.JS.

Redux заснований на патерні Flux, який розроблений у Facebook. Патерн Flux включає односторонню потік даних, де дії (actions) виконуються з компонентів React та пересилаються в диспетчер (dispatcher), який відповідає за розподіл цих дій до магазину (store). Магазин містить стан додатка та логіку, яка відповідає за зміну стану за допомогою редукторів (reducers), які виконуються на кожній дії та змінюють стан додатка відповідно до логіки, яка описана в них. Компоненти React можуть підписуватись на зміни стану магазину та оновлювати свій стан відповідно до змін.

Redux дає змогу зберігати стан додатка в одному місці, що спрощує його управління та підтримку. Також він дає можливість зберігати історію змін стану, що дає змогу повернутись до будь-якої попередньої версії стану додатка. Це можна використати для налагодження та відлагодження додатка.

Redux також дає можливість розбити стан додатка на окремі частини та зберігати їх в окремих редукторах. Це сприяє спрощенню логіки додатка та підтримки коду. Крім того, Redux дає змогу розробникам використовувати бібліотеки middleware для додавання додаткової логіки до процесу зміни стану додатка.

Redux має велику спільноту розробників, що допомага є вирішувати проблеми та надавати підтримку для розробників, що робить його ще більш привабливим для використання.

Redux можна використати для будь-якого типу додатка, від простих до складних, забезпечуючи однорідний підхід до управління станом додатка. Зокрема, Redux дає змогу писати тестований та декларативний код, що забезпечує більшу надійність та стабільність додатку.

Іншою корисною можливістю Redux є можливість розширення за допомогою плагінів та middleware. Middleware є функціями, які дають можливість додавати додаткову логіку до процесу зміни стану додатка, наприклад, логування або зберігання історії змін. Це дає змогу розробникам зберігати чистоту та простоту коду додатка, знаючи, що додаткова логіка може бути додана у будь-який час.

Redux також інтегрується добре з React-додатками, що дає змогу розробникам зберігати стан додатка та логіку в Redux, а компоненти React можуть підписуватись на зміни стану та оновлювати свій стан відповідно до змін. Це дає моживість розробникам використовувати привабливі функції React, такі як контекст та власники, та використовувати Redux для управління станом.

Redux є потужною бібліотекою для управління станом додатка в React-додатках. Він забезпечує простий та однорідний підхід до управління станом, дозволяє зберігати стан додатка в одному місці та підтримує розширення за допомогою плагінів та middleware. Redux інтегрується добре з React-додатках та дає можливість розробникам використовувати функціональність React, щоб створювати привабливі та потужні додатки. Крім того, Redux забезпечує можливість писати тестований та декларативний код, що допомагає зберегти надійність та стабільність додатку.

Однак, на вибір Redux впливає складність додатка та його потреби в управлінні станом. Якщо додаток є простим та має небагато стану, можливо, не потрібно використовувати Redux. Однак, якщо додаток має складні взаємодії між компонентами та багато стану, то Redux може бути потужним інструментом для управління станом.

У будь-якому випадку, якщо ви використовуєте Redux, важливо мати на увазі, що він не є срібною кулею. Іноді розробники можуть піти занадто далеко з використанням Redux, створивши занадто багато додаткової логіки та middleware. Тому важливо зберігати баланс між простотою та потужністю, використовуючи Redux у доречних випадках та зберігаючи код додатку простим та легким для розуміння.

Загалом, Redux є потужним інструментом для управління станом додатка, який дозволяє зберігати стан в одному місці та забезпечує декларативний та тестований код. Він інтегрується добре з React-додатками та дозволяє розробникам створювати привабливі та потужні додатки. Однак, використовуйте Redux розумно, зберігаючи баланс між простотою та потужністю, щоб забезпечити стабільність та надійність додатку.

# 2.5. Visual Studio

Набір інструментів Microsoft Visual Studio призначений для створення програмного забезпечення, включаючи планування, розробку інтерфейсу користувача, написання коду, тестування, налагодження, аналіз якості коду та продуктивності, розгортання у клієнтських середовищах та збір телеметрії щодо використання. Всі ці інструменти можна знайти в одному інтегрованому середовищі розробки (IDE) – Visual Studio. Вони були створені з метою максимально ефективної спільної роботи.



Рис. 2.5. Логотип Visual Studio.

Visual Studio є універсальним інструментарієм для розробдення різних типів додатків, від простих мобільних додатків і ігор до складних підприємств та центрів обробки даних. Наприклад, ви можете створювати додатки та ігри для Windows, Android та iOS, веб-сайти та веб-служби на основі платформ ASP.NET, JQuery та AngularJS, додатки для різних платформ та пристроїв, таких як Office, Sharepoint, Hololens, Kinect та “Інтернет речей”, а також ігри та графічні додатки для Windows пристроїв, включаючи Xbox, з підтримкою DirectX.

Visual Studio за замовчуванням підтримує C#, C, C++, JavaScript, F# та Visual Basic, і легко інтегрується зі сторонніми додатками, такими як Unity та Apache Cordova, за допомогою розширень, таких як Набір засобів Visual Studio для Unity та Інструменти Visual Studio для Apache Cordova відповідно. Ви можете самостійно розширити Visual Studio, створивши власні інструменти для виконання спеціалізованих завдань.

**2.6. Git**

Git – це система контролю версій, яка дає змогу розробникам зберігати та керувати змінами в програмному коді та інших файлах. Вона стала незамінною частиною сучасного програмного розроблення та дає можливість командам розробників ефективно співпрацювати, зберігати історію змін, відслідковувати помилки та швидко відновлювати стан додатків.



Рис. 2.6. Логотип Git.

Git дає змогу зберігати код у віддаленому репозиторії, який можна синхронізувати з локальними копіями. Кожен розробник може створювати свою власну гілку (branch), що дозволяє працювати зі своїми власними змінами, не впливаючи на головний код. Після завершення роботи, зміни можна об’єднати (merge) з головною гілкою, забезпечуючи збереження коду в одному місці та уникнення конфліктів.

Однією з найважливіших функцій Git є можливість повернення до попередніх версій коду. Якщо під час розробки виникають проблеми, Git дозволяє відновити попередню версію коду та продовжити роботу з цієї точки. Крім того, Git дає змогу співпрацювати з іншими розробниками, які можуть вносити зміни до коду та вирішувати проблеми шляхом взаємодії та обговорення.

Git також має багато інших корисних функцій, таких як відслідковування змін та дозвіл на доступ до репозиторію. Він інтегрується з багатьма редакторами коду та іншими інструментами, що дає можливість розробникам легко і ефективно працювати з Git.

Хоча Git може здатися складним на початку, він має дуже добре документовану та розгалужену систему допомоги, що дозволяє вирішувати будь-які проблеми, які можуть виникнути. Багато розробників вважають Git незамінною частиною своїх робочих процесів та рекомендують його для будь-яких проектів.

Однією з найбільших переваг Git є те, що він є відкритою системою, що означає, що ви можете безкоштовно використовувати його та вносити свої власні зміни в систему. Багато відомих компаній, таких як Google, Facebook, Microsoft та інші, використовують Git для своїх проектів та активно співпрацюють зі спільнотою розробників, що дозволяє їм розвивати цю систему контролю версій та покращувати її функціональні можливості.

Загалом, Git є дуже корисним інструментом для будь-якої команди розробників, яка працює над програмними проектами. Він дозволяє розробникам зберігати та керувати змінами в коді, співпрацювати з іншими розробниками, відслідковувати помилки та швидко відновлювати стан додатків. Хоча Git може здатися складним на початку, він має дуже потужну та детально документовану систему, яка дозволяє вирішувати будь-які проблеми, які можуть виникнути. Тому використання Git є необхідним для будь-якого проекту програмного розроблення та допомагає зберігати якість коду та робочий процес команди на високому рівні.

# 2.7. SQL мова

SQL (Structured Query Language) – це мова, яка дає змогу створювати та працювати з реляційними базами даних, які є наборами пов'язаної інформації, збереженої у таблицях.

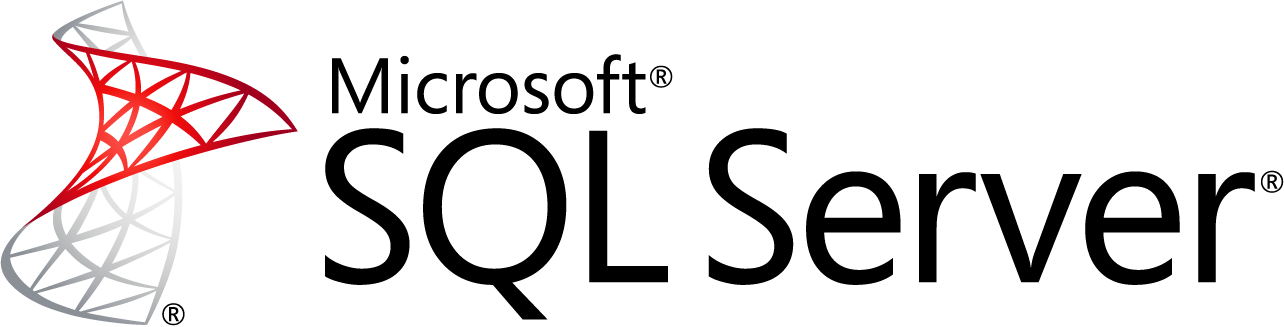


Рис. 2.7. Microsoft SQL Server.

Завдяки постійному зростанню інформаційного простору, з’явилась необхідність у створенні стандартної мови, яка могла б бути використана в різних типах комп’ютерних середовищ. Стандартна мова дає можливість користувачам, які знають один набір команд, використовувати їх для створення, пошуку, зміни та передачі інформації, незалежно від типу комп'ютера.

У нашому все більш і більш взаємопов’язаному комп’ютерному світі, користувач, який володіє цією мовою, має значну перевагу у використанні та узагальненні інформації з різних джерел.

SQL є основною стандартною мовою технології реляційних баз даних завдяки своїй елегантності та незалежності від специфіки комп’ютерних технологій. Стандарт SQL визначається Американським Національним Інститутом Стандартів (ANSI) і Міжнародною Організацією по Стандартизації (ISO).

Більшість комерційних програм баз даних розширюють SQL, додаючи різні особливості в цю мову, які вони вважають корисними. Іноді ці особливості порушують стандарт мови, але деякі з них стають стандартами “ринку” через свою корисність.

Microsoft SQL Server є високопродуктивною платформою для обробки та аналізу даних будь-якого розміру, яка відповідає всім сучасним вимогам роботи з даними будь якої складності. SQL Server має широкий спектр функцій та інструментів, що дозволяють ефективно управляти даними, виконувати запити, забезпечувати безпеку та доступність даних.

За допомогою SQL Server можна створювати реляційні бази даних, виконувати транзакції, встановлювати зв’язки між таблицями, індексувати дані для швидкого пошуку та виконувати різноманітні запити. Також платформа підтримує роботу з геоданими, що дає змогу створювати геопросторові додатки та аналізувати дані за географічними параметрами.

SQL Server надає зручні інструменти для резервного копіювання та відновлення даних, а також має вбудовану систему безпеки, яка дозволяє керувати доступом до даних та забезпечувати їх захист від несанкціонованого доступу. Платформа також має вбудовану підтримку реплікації, що дозволяє забезпечити доступність даних в режимі реального часу.

Окрім того, SQL Server підтримує інтеграцію з різноманітними додатками, включаючи різні сервіси та інші бази даних. SQL Server можна використовувати для розробки та реалізації багатошарових додатків, які працюють з великим обсягом даних та мають складну структуру. Це робить SQL Server потужним інструментом для розробки підприємницьких додатків та систем управління даними будь-якого розміру та складності.

## Висновки до розділу 2

В даному розділі були докладно розглянуті всі основні технології розробки, наведена їх сутність, переваги та призначення. Кожна з цих технологій була обрана з урахуванням масштабів проекту і вимог, які він повинен виконувати. Зазначені технології виявляються дуже ефективними в своїх сферах застосування, крім того, вони взаємодіють добре з іншими інструментами і утворюють єдину гармонійну систему, що забезпечує комфортну, продуктивну та високоякісну розробку дипломного проекту.

## Висновок:

Отже, після витраченого часу на практику в НУ "ЛП" я приходжу до висновку, що набрав достатньо досвіду, щоб мати змогу створити дипломну роботу на тему "Платформа для благодійних пожертвувань та волонтерської діяльності". Мені задоволенням служить пройдена практика, і я переконаний, що цей досвід буде корисним у майбутніх проектах, допомагаючи мені досягти успіху. Я знайомий з різними аспектами платформи, від проектування до волонтерської координації, і вірю, що це буде справжнім природженим завданням для мене. Мій досвід у побудові та розробці додатків, спілкуванні з користувачами та вирішенні проблем допоможуть мені виконати цей проєкт з високою якістю і ефективністю. Я відчуваю готовність взятися за цю роботу, використовуючи свої знання та навички, набуті під час практики, і розширюючи їх у процесі розробки цієї платформи. Моя мета полягає в тому, щоб створити функціональний та інтуїтивно зрозумілий інструмент, який забезпечить зручність для користувачів та сприятиме благодійній діяльності та волонтерству. Я впевнений, що результат буде задовольняючим і відповідатиме всім вимогам дипломної роботи, а мій досвід знайде практичне застосування у реальному світі.

# Використана література

1. [https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0.](https://testmatick.com/ru/testirovanie-metodom-serogo-yashавіа%20авіа.)
2. <https://learn.microsoft.com/en-us/ef/core/>
3. <https://legacy.reactjs.org/docs/getting-started.html>
4. <https://redux.js.org/>
5. <https://learn.microsoft.com/en-us/visualstudio/windows/?view=vs-2022>
6. <https://git-scm.com/doc>
7. https://learn.microsoft.com/en-us/sql/?view=sql-server-ver16