**АНОТАЦІЯ**

Метою бакалаврської кваліфікаційної роботи була розроблення програмної системи (веб-платформи) для фінансової підтримки контентмейкерів.

Програмна система розроблена за принципами клієнт-серверної архітектури. Серверна частина була розроблена на платформі ASP.NET Core з використанням таких бібліотек та фреймворків як Entity Framework, ASP.NET Identity та ASP.NET SignalR. Клієнтська частина була розроблена за допомогою наступних бібліотек: Razor Pages, Blazor, Bootstrap, JQuery.

Розроблена програмна система є кросплатформеною веб-платформою. Вона володіє зручним та мінімалістичним інтерфейсом, містить систему ролей, в якій неавторизовані користувачі мають доступ тільки до функціоналу для здійснення пожертв контентмейкеру, та перегляду його (контентмейкера) медійної інформації та посилань на соціальні мережі. Користувачі які авторизовані як контентмейкер мають змогу створювати спеціальні віджети для подальшого використовування на прямих трансляціях, створювати сторінку-візитівку та отримати кошти з пожертв.

До програмної системи створено специфікацію вимог. Проектування та розробка здійснювалась згідно вимог описаних у специфікації та спираючись аналізу предметної області.

Робота містить реалізацію програмної системи та пояснювальну записку. Записка складається з п’яти розділів та шести додатків. Розмір записки без додатків містить 68 сторінок.

**ABSTRACT**

The purpose of the bachelor's thesis was to develop a software system (web platform) for financial support of content makers.

The software system is developed on the principles of client-server architecture. The server part was developed on the ASP.NET Core platform using libraries and frameworks such as Entity Framework, ASP.NET Identity and ASP.NET SignalR. The client part was developed using the following libraries: Razor Pages, Blazor, Bootstrap, JQuery.

The developed software system is a cross-platform web platform. It has a user-friendly and minimalist interface, contains a system of roles in which unauthorized users have access only to the functionality for making donations to the content maker, and view his (content maker) media information and links to social networks. Users who are authorized as content makers have the opportunity to create special widgets for further use on live broadcasts, create a business card page and receive funds from donations.

A specification of requirements has been created for the software system. Design and development was carried out according to the requirements described in the specification and based on the analysis of the subject area.

The work includes the implementation of the software system and an explanatory note. The note consists of five sections and six appendices. The size of the note without attachments contains 68 pages.

**ЗМІСТ**

[**РОЗДІЛ 1 СТВОРЕННЯ НА МОНЕТИЗАЦІЯ ВЕБ КОНТЕНТУ 10**](#_Toc104623020)

[**1.1. Соціальні мережі та платформи для контентмейкерів 10**](#_Toc104623021)

[**1.2. Прямі трансляції 14**](#_Toc104623022)

[**1.3. Монетизація контенту 15**](#_Toc104623023)

[**1.4. Висновки до розділу 1 19**](#_Toc104623024)

[**РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ 20**](#_Toc104623025)

[**2.1. Загальна постановка завдання 20**](#_Toc104623026)

[**2.2. Специфікація вимог до програмного продукту 20**](#_Toc104623027)

[**2.2.1.1. Призначення та мета 20**](#_Toc104623028)

[**2.2.1.2. Продукти-аналоги 21**](#_Toc104623029)

[**2.2.2. Загальний опис 21**](#_Toc104623030)

[**2.2.3. Характеристики системи 22**](#_Toc104623031)

[**2.2.3. Вимоги зовнішніх інтерфейсів 29**](#_Toc104623032)

[**2.2.5. Інші нефункціональні вимоги 30**](#_Toc104623033)

[**2.2.5.3. Атрибути якості програмного продукту 31**](#_Toc104623034)

[**2.3. Вибір засобів для виконання завдання 31**](#_Toc104623035)

[**2.4. Висновок до розділу 2 32**](#_Toc104623036)

[**РОЗДІЛ 3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ/ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ 33**](#_Toc104623037)

[**3.1 . Проектування поведінки розроблюваної системи 33**](#_Toc104623038)

[**3.2. Загальна архітектура системи 34**](#_Toc104623039)

[**3.3. Проектування бази даних 35**](#_Toc104623040)

[**3.4. Проектування серверної частини 37**](#_Toc104623041)

[**3.5. Проектування клієнтської частини 38**](#_Toc104623042)

[**3.6. Висновок до розділу 3 40**](#_Toc104623043)

[**РОЗДІЛ 4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ 41**](#_Toc104623044)

[**4.1. Налаштування та реалізація серверної частини 41**](#_Toc104623045)

[**4.2. Опис роботи системи та програмних рішень на клієнтській частині 42**](#_Toc104623046)

[**4.3. Тестування програмного забезпечення 48**](#_Toc104623047)

[**4.3.1. Вступ 48**](#_Toc104623048)

[**4.3.2. Створення тестових випадків 48**](#_Toc104623049)

[**4.3.3. Модульне тестування 50**](#_Toc104623050)

[**4.3.4. Функціональне тестування 50**](#_Toc104623051)

[**4.3.5. Тестування інтерфейсу 51**](#_Toc104623052)

[**4.3.6. Критерії успішності виконання тестування 51**](#_Toc104623053)

[**4.4. Висновок до розділу 4 51**](#_Toc104623054)

[**РОЗДІЛ 5 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОБОТИ 52**](#_Toc104623056)

[**5.1. Економічна характеристика проектного рішення 52**](#_Toc104623057)

[**5.2. Інформаційне забезпечення та формування гіпотези щодо потреби розроблення проектного рішення 53**](#_Toc104623058)

[**5.3. Оцінювання та аналізування факторів зовнішнього та внутрішнього середовищ 54**](#_Toc104623059)

[**5.4. Формування стратегічних альтернатив 56**](#_Toc104623060)

[**5.5. Бюджетування 59**](#_Toc104623061)

[**5.6. Вибір стратегії 65**](#_Toc104623062)

[**Висновки до розділу 5 66**](#_Toc104623063)

[**ВИСНОВКИ 67**](#_Toc104623064)

[**СПИСОК ЛІТЕРАТУРИ 68**](#_Toc104623065)

[**ДОДАТОК А. Діаграма прецедентів 69**](#_Toc104623066)

[**ДОДАТОК Б. Діаграма бази даних 70**](#_Toc104623067)

[**ДОДАТОК В. Діаграма послідовності 71**](#_Toc104623068)

[**ДОДАТОК Г. Варіанти використання 72**](#_Toc104623069)

[**ДОДАТОК Д. Звіт про модульне тестування 78**](#_Toc104623070)

[**ДОДАТОК Е. Інструкція користувача 79**](#_Toc104623071)

**ВСТУП**

Сучасна людина переживає фантастичні часи. Людство зараз живе під час Інформаційної ери. Інформаційна ера (англійською Digital Age, Computer Age, Information Age) або ще відома як Ера комп'ютерів, інформаційна епоха або електронна епоха – це концепція про сучасний світ та сучасні віки, для яких характерні широкі можливості вільно передавати чи приймати інформацію і мати миттєвий доступ до всесвітніх знань, що в попередні епохи розвитку людства зробити було б важко або неможливо. [1]

Людство розвивається дуже стрімко і технології розвиваються та еволюціонують швидше ніж самі люди. З кожним днем нової інформації стає ще більше і більше, а отримати до неї доступ стає все легше. Щодня через простори інтернету проходить мільйони терабайт інформації і людина потрапляє у інформаційний безлад. Людина має нестримну жагу до нової інформації та новин, вона стає емоційно залежною від неї.

З розвитком інформаційних технологій дуже сильно змінився стиль людського життя. Так само сильно і змінився спосіб проведення дозвілля. Людина дедалі більше надає перевагу перегляду соціальних мереж, телебачення, фільмів, серіалів, інтернет спілкуванню чи просто віртуальному світові над світом реальним та живим спілкуванням. Тому способи споживання інформації дуже стрімко розвиваються та мають шалену конкуренцію. Щороку створюються нові соціальні мережі, стрімко розвиваються старі. За користувача ведеться пекельна війна.

Одним із домінуючих способів отримування інформації є візуальний. Для людини він є найпростішим для сприйняття. Тому зараз шаленої популярності набирають платформи для перегляду фото-відео контенту, накшталт Instagram, YouTube, Tik-Tok, Twitch, Netflix і ще дуже і дуже багато інших. Усі ці платформи мають шалені заробітки від продажі реклами, підписок чи цифрових товарів.

Нова епоха породила і нові професії. Професія «блогер» чи «контентмейкер» стає надзвичайно популярною та навіть престижною. Для цієї професії існує безліч різних інструментів для створення та поширення контенту.

Одним із інструментів сучасних контентмейкерів та лідерів думок є прямі трансляції, або як їх прийнято називати «стріми». Зараз вийти у прямий ефір може кожен бажаючий. Найвідомішими платформами для проведення прямих трансляціє є Facebook, Twitch та YouTube. Вони показують дуже високі статистичні показники. Старший віце-президент Twitch Даміан Бернс, на своїй сторінці у Twitter, поділився статистикою платформи минулого року. За його даними, у 2020-му сервісом щодня користувалися понад 26,5 млн користувачів. Як зазначено у статистиці, продовж 2020 року користувачі спостерігали за прямими трансляціями на Twitch більше трильйона хвилин. Окрім цього, Бернс підмітив, що упродовж цього періоду на сервіс щомісяця приходило понад 6 млн нових контентмйкерів, а середня глядацька аудиторія Twitch в будь-який момент часу склала понад 2 млн людей.

Жодна праця не буває безоплатною, тому віртуальний контент по різному монетизується у мережі. Контентмейкери отримають гроші від платформ, рекламодавців, партнерських програм, спонсорів та напряму від фанатів. Як і фанати та прихильники певних релігій дають пожертвування на свою церкву, так і фанати певного блогера можуть дати пожертвування йому. Цей вид пожертвування має назву – донат.

**РОЗДІЛ 1 СТВОРЕННЯ НА МОНЕТИЗАЦІЯ ВЕБ КОНТЕНТУ**

* 1. **Соціальні мережі та платформи для контентмейкерів**

Сьогодні існують сотні соціальних мереж які щодня заохочують тисячі нових користувачів доєднатися до них. Соціальні мережі поступово доповнюють та частково заміняють реальне життя. Уже існують сервіси спілкування, знайомств та побачень, пошуку роботи, отримання новин, обговорення всіляких питань та тем, перегляду фільмів та серіалів, пошуку усього та усе. Соціальні мережі витісняють та замінюють привичні засоби отримання інформації та розваг, такі як телевізор, газети, книги, радіо, живі вебінари та форуми. Зараз усе переходить у цифровий світ.

Компанія Cloudflare, що створює інтернет-сервіси, спільно з сайтом Alexa.com, та збирає статистику про інтернет-сайтів, випустила рейтинг найпопулярніших сайтів за 2021 рік

Топ-10 — найпопулярніші домени соціальних мереж (кінець) 2021 року за даними Cloudflare:[2]

1. TikTok.com
2. Facebook.com
3. YouTube.com
4. Twitter.com
5. Instagram.com
6. Snapchat.com
7. Reddit.com
8. Pinterest.com
9. LinkedIn.com
10. Quora.com

Щомісяця ці соціальні мережі мають мільярди активних користувачів. За даними Statista.com [3], аудиторія Facebook налічує 2.910 млрд активних користувачів, YouTube 2.562 млрд, Instagram 1.263 млрд.

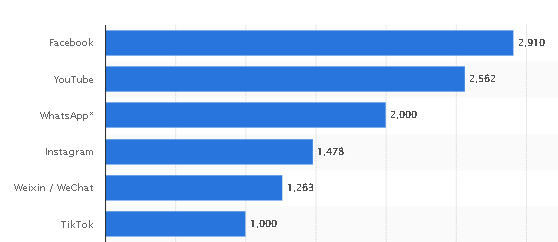


Рис. 1.1. Найпопулярніші соціальні мережі в усьому світі станом на січень 2022 року, ранжовані за кількістю активних користувачів щомісяця

Як можна побачити зі списку, найпопулярнішими є соціальні мережі, які пропонують у переважному відео-контент. TikTok взагалі став феноменом. Ця соціальна мережа дозволяє ділитися людям з усього світу короткими відео (зараз існує можливість створювати відео до 4хв). Також платформа дозволяє редагувати відео, накладати на нього ефекти та музику, спілкуватися з іншими користувачами. Окрім цих інструментів вона також дає можливостей вести прямі трансляції. По даних аналітиків, кількість активних користувачів у 2021р становила більше 1 мільярду.

Інтернет не є тихим місцем, як і люди на Землі (особливо під час пандемії) не тихі чи пасивні, а активні та реактивні. Хоча на вершині рейтингу доменів, здається, не спостерігається різких злетів і падінь протягом року (TikTok і YouTube були винятком), можна побачити, як така подія, як переворот у М’янмі та наступне вірусне відео, могли принести YouTube займає перше місце в нашому рейтингу. Ми також бачили, як вплинула електронна комерція протягом року, як веб-сайти, пов’язані з космосом, мали великий (онлайн) рік із важливими подіями, і як Netflix піднявся під час Різдва.[2]

Темою моєї бакалаврської кваліфікаційної роботи є платформа для підтримки творців які орієнтуються на відео-контент та проведення прямих трансляцій. Тому надалі буде розглядатися саме призначені для цього платформи.

Я виділив дві соціальні мережі, або як їх можна назвати «відео-хостинги», які на мій погляд є найзручнішими та найпопулярнішими для публікації відео матеріалів, це – YouTube та Twitch.

YouTube — веб-платформа для обміну відео в Інтернеті та соціальних мереж, яка функціонує з 2005 року та набула надзвичайної світової популярності. Її творцями є колишні працівники PayPal Стів Чен, Чад Херлі та Джавед Карім. Штаб-квартира знаходиться в Сан-Бруно, Каліфорнія. 9 жовтня 2006 року компанія Google придбала YouTube за 1,65 мільярда доларів. На даний момент компанія вже давно окупилася.

YouTube існує уже більше 15 років, але з часом платформа ставала лише кращою, швидшою та міцнішою. Загалом сервіс отримує 14,3 мільярда відвідувань на місяць: це більше, ніж у Facebook, Вікіпедії, Amazon та Instagram. Якщо говорити про стать користувачів YouTube, то серед них віком від 18 років 46,1% ідентифікують себе як жінки, а 53,9% – як чоловіки. Як повідомляє Statista, деякі користувачі відвідують сервіс навіть кілька разів на день. Тим часом 92% американських користувачів стверджували, що користуються платформою щотижня, а 98% кажуть, що мають доступ до сайту щомісяця.

Маючи щомісяця понад 14 мільярдів відвідувань, YouTube є одним із найбільш відвідуваних сайтів відразу після своєї материнської компанії Google. Пошуковий термін «YouTube» є також третім за популярністю у Google. Щохвилини на сервісі «YouTube» транслюється понад 694 000 годин відео. Це більше, ніж у найвідомішої стрімінгової платформи Netflix, де користувачі транслюють 452 000 годин відео.

За статистичними даними, приблизно 81% користувачів Інтернету бодай раз користувалися YouTube. Зі стрімким розвитком смартфонів, значно збільшилась кількість користувачів, що заходять на сайт через мобільний телефон, їхня частка складає 22% від усіх відвідувачів сервісу. Користувачі операційної системи Android по всьому світу користуються додатком YouTube на своїх телефонах у середньому 23,7 години щомісяця.

Twitch — це популярний американський сервіс для потокового відео в прямому ефірі. Нею керує дочірня компанія Amazon. Сервіс зосереджується на прямій трансляції відеоігор («летсплеї»), включаючи трансляції кіберспортивних змагань, музичні трансляції, трансляції творчого контенту та потоків «в реальному житті». Twitch був представлений у червні 2011 року як додатковий продукт платформи Justin.tv. Вміст на сайті можна переглядати як у прямому ефірі, так і через відео за запитом. Amazon швидко помітив потенціал Twitch і у 2014 році він купив компанію майже за мільярд доларів. Це дозволило отримати їм близько 40 000 000 підписок.

Сервіс Twitch отримав великий наплив нових користувачів саме під час пандемії COVID-19, коли майже уся планета закрилася на карантин. Після активної фази карантину активність сервісу не переставала збільшуватися. Тільки за перші три місяці 2021 року користувачі Twitch переглянули 326 мільярдів хвилин контенту, що є майже половиною від 660 мільярдів за цілий 2019 рік. Тільки в першому кварталі 2021 року це понад 5 мільярдів годин перегляду на Twitch. У 2021 році щомісяця нараховується 9,7 мільйона стрімерів. У 2020 році в середньому за місяць користувачі переглядали Twitch 93 мільярди хвилин. Статистика Twitch за 2020 рік показує, що 140 мільйонів глядачів переглядали контент на платформі щомісяця.

Це звучить дивно для платформи про ігри, але статистика Twitch показує, що це найпопулярніша тема взагалі. «Just Chatting» — раніше тег відео-блогу категорії IRL — призначений для спілкування стрімерів зі своїми глядачами. Хто б міг подумати, що платформа в кінцевому підсумку охопить свої коріння як мережу «життя»? Цей жанр допоміг популяризувати «lifestyle streaming» Багато ігрових стрімерів також використовують цю тему.

Значну частку популярних стрімерів складають саме особи жіночої статі з особливо вираженою фізичною красою. На подібних прямих трансляціях відеогра зазвичай відкладається на задній план, а основний акцент переходить на глибокий виріз декольте, миле личко чи на місце для сидіння на кріслі. Проте досі існують вірні сини та доньки світу геймерів, яким нецікаві ці всі плотські задоволення.

* 1. **Прямі трансляції**

Так як створення нових відеороликів може займати дуже багато часу та потребувати значних зусиль, серед блогерів стало популярним проводити прямі трансляції, або як їх називають у інтернеті стріми, а блогера, який їх проводить – стрімером.

Пряма трансляція — це потокове мультимедіа, яке одночасно записується та транслюється в режимі реального часу через Інтернет. Його часто називають просто «стрім». Пряма трансляція – це ефективний та ефективний спосіб досягнення будь-яких цілей і завдань, які ви хочете досягти. Пряму трансляцію можна використовувати для просування та оголошення ваших нових продуктів і послуг, оголошення новин великої компанії, навчання онлайн-класу тощо. Пряма трансляція дає брендам можливість охопити мільйони людей по всьому світу та взаємодіяти з ними.

«Стрімер»— це особа, яка транслює себе в режимі онлайн за допомогою прямої трансляції або попередньо записаного відео. Жанри включають гру у відеоігри, навчальні посібники або соло-чати. У сучасних реаліях до цієї «касти» доєдналися волонтери та навіть військові (яскравим прикладом є Євген Карась, ще відомий як «Лисий з ЗСУ»).

Одним із типів стрімерів є летсплеєри. Вони набули популярності ще з самого започаткування прямих трансляцій, завдяки цьому вони і до цих пір збирають найбільшу аудиторію та фан-базу в Інтернеті. Летсплей (англ. “Lets-play”) – один з жанрів розважального контенту, який включає проходження комп’ютерних ігор, ігрові марафони, гайди. Для багатьох стрімінг був хобі, але з ростом аудиторії та появою рекламодавців прибутки дозволили заробляти та займатися улюбленою справою у ролі роботи. Щодня з'являється все більше новачків, які пробують зацікавити аудиторію своєю особистістю, а не тільки вмінням грати ігри. Найвідоміші стрімери відеоігор — PewDiePie та Ninja, яким вдається щороку заробляють мільйони доларів лише за допомогою стрімінгу не включаючи пожертви глядачів.

Доступність сервісу породило багато різного контенту у тому числі і негативного. Тому у лютому 2018 року Twitch оновив свою політику щодо прийнятного вмісту, щоб наказати, щоб будь-який вміст, який він вважає ненависним, призупинявся на своїй платформі. Користувачам Twitch тепер заборонено транслювати будь-які ігри, які мають рейтинг «Тільки для дорослих» (AO) у Сполучених Штатах Радою з оцінки розважального програмного забезпечення (ESRB), незалежно від її рейтингу в будь-якому іншому географічному регіоні, а також будь-яку гру, яка містить «вміст відверто сексуального характеру» або «безпідставне насильство», або вміст, який порушує умови використання сторонніх служб.

Для запису відео та проведення прямих трансляцій також існує багато додаткових інструментів. Проте найбільшої набрала OBS (Open Broadcaster Software) - безкоштовне програмне забезпечення з відкритим кодом для запису та створення трансляцій. [4] Програма написана на C/C++ і створена за допомогою Qt, OBS Studio забезпечує захоплення в реальному часі, композицію сцени, запис, кодування та трансляцію через протокол обміну повідомленнями в реальному часі (RTMP). Він може транслювати відео на будь-яке місце призначення з підтримкою RTMP, включаючи YouTube, Twitch, Instagram та Facebook. Хтось або платить вам за ваш контент безпосередньо, або вони платять за доступ до вашої аудиторії.

* 1. **Монетизація контенту**

Усі сучасні платформи та соціальні мережі дають можливість користувачам заробляти на власному контенті. Це зветься монетизацією. Монетизація контенту – це спосіб використання вмісту, щоб, коли користувачі споживали його, ви заробляли гроші. Вам можуть заплатити як самі користувачі, так і третя сторона, яка продає продукти через ваш вміст. Почати експериментувати з монетизацією вмісту легко. Наприклад, такі платформи, як YouTube і Facebook, уже мають вбудовані сервіси, які дають вам змогу користуватися рекламою, коли ваша зацікавленість досягає певного порогу.

Коли контентмейкер безпосередньо продаєте свій контент, він заробляєте гроші безпосередньо від користувачів. Наприклад, коли контент знаходиться за платним доступом або структурований в онлайн-курс, його можна вважати як власний продукт. Хоча насправді ваша сторінка у будь якій соціальній мережі не належить блогеру, а є власністю самої мережі.

Коли контентмейкер продає свою аудиторію, то він працює з третьою стороною, яка часто зветься спонсором, щоб задовольнити їхні маркетингові потреби. Блогери часто продають рекламу, як це відбувається у інших засобах масової інформації.

Платформи монетизації вмісту, такі як Facebook і Youtube, діють як третя сторона між вами та рекламодавцями. Блогер публікує свій вміст на платформі, а платформа платить вам за показ реклами. (Платформа, в свою чергу, оплачується рекламодавцями.) Якщо контентмейкер переважно публікуєте свій контент у соціальній мережі, це найпростіший і прямий спосіб почати його монетизувати.

Ще одним із способу заробітку є продакт-плейсмент. У певному сенсі він є старомодною формою маркетингу. Що стосується контенту, він працює так само, як і з фільмами: блогер показуєте продукт спонсора у своїй роботі, і вони платять йому фіксовану ставку, що є хорошим плюсом. Продакт-плейсмент – це досить простий спосіб почати отримувати прибуток за контент і заробляти гроші.

Існує також афілійований (партнерський) маркетинг. У партнерському маркетингу контентмейкер заохочує своїх користувачів купувати продукти чи послуги. Однак, на відміну від продакт-плейсменту, кожна покупка відстежується, і ви стягуєте комісію на основі того, яка частина йог аудиторії бере участь. Є кілька способів зробити це. Одним з них є використання реферального коду чи промокоду.

Для монетизації бази шанувальників, блогери створюють нові продукти, щоб продавати людям, які люблять їхню роботу. Одяг, кружки, декоративні подушки та інший мерчендайз — все це досить прості варіанти. Популярним способом заробітку від аудиторії є пожертвування або донати.

Простіше кажучи, донат – це різновид подарунка, який робить юридична чи фізична особа (особи), як показує практика, на благодійні цілі та потреби або на користь певної справи. Донат набуває різної форми, включаючи фінансову підтримку, послуги, певні товари (одяг, іграшки, техніка, електронні пристрої, продукти харчування, транспортні засоби тощо). Предмети гуманітарної допомоги, допомоги розвитку, екстреної допомоги, для медичних потреб (кров або органи для трансплантації) також можуть бути складовою пожертвування. Чудовим прикладом є пожертвування, які здійснює український народ та увесь світ для боротьби з гидким та бридким злом яке успішно дохне на кордонах з сусідньою державою болотом, за допомогою чудових подарунків наших західних партнерів. [5]

Величезні можливості відкриваються для некомерційних організацій, які можуть організувати збір коштів у соціальних мережах. З огляду на величезну кількість користувачів інтернету та соціальним мережами, їх варто використовувати, щоб збільшити кількість своєї аудиторії та у результаті і базу донатерів.

Користувачі, яким подобається контент їхнього улюбленого блогера підтримують його фінансово, жертвуючи йому кошти. Окрім прояву любові чи поваги до блогера, фанати привертають до себе увагу улюбленого творця.

Для заохочення глядачів прямих трансляцій жертвувати кошти стрімеру було придумано багато заохочувальних схем. Одна з ним – це можливість засвітити свій псевдонім серед великої аудиторії інтернет користувачів та надіслати якусь звістку улюбленому блогеру. Після того, як фанат надсилає певну грошову суму, у прямому ефірі з’являється повідомлення, яке можу супроводжуватися звуком та анімаційною картинкою, яке містить ім’я користувача та його повідомлення.

Донати стали невід’ємним джерелом заробітку для контентмейкерів. Надзвичайної популярності набирає система Patreon, яка за допомогою спеціальної підписки дозволяє підтримати улюбленого контентмейкера та отримати доступ до ексклюзивного контенту. Patreon— це американська платформа для членства, яка надає бізнес-інструменти для розробників контенту для запуску служби підписки. Він допомагає творцям і художникам отримувати щомісячний дохід, надаючи винагороди та пільги для своєї аудиторії. Patreon стягує комісію в розмірі від 9 до 12 відсотків місячного доходу творців на додаток до плати за обробку платежів.[6]

Окрім Patreon також існує ще одна схожа платформа – OnlyFans. Користувачі OnlyFans за спеціальною підпискою також отримають доступ до ексклюзивного контенту дещо іншого характеру, тому платформа породила у мережі багато мемів та приколів. Здебільшого авторами цього контенту є особи жіночої статі, різного віку, які наділені красивими очима.

OnlyFans — це служба підписки на інтернет-контент, що базується в Лондоні, Великобританія. Творці контенту можуть заробляти гроші від користувачів, які підписалися на їхній контент — «шанувальників». Це дозволяє творцям контенту щомісячно отримувати фінансування безпосередньо від своїх шанувальників, а також одноразові поради та функцію оплати за перегляд (PPV). Веб-сайт має два мільйони творців контенту та 130 мільйонів користувачів.

Заради нових донатів стрімери часто готові піти навіть на нерозумні та небезпечні поступки. Великої популярності набирають прямі трансляції, на яких за пожертвування певної суми можна заставити блогера зробити якусь неприємну річ, як наприклад випити гидку речовину, велику кількість алкоголю, отримати стусанів від інших учасників стріму, отримати розряд током, показати якусь частину тіла, зробити якусь фізичну вправу, потрясти задницею чи грудьми. Коротко кажучи, контентмейкер перетворюється на іграшку чи клоуна для глядачів. На платформах з можливістю публікувати «дорослий» контент, за пожертвування блогер може здійснювати певні вульгарні дії, які у суспільстві можуть набувати аморального характеру. Цікавою можливістю є можливість донатом керувати спеціальним девайсом стрімера. Усі ці дії, це тільки вершина айсбергу фантазії блогера, часто вони роблять жахливіші речі задля грошей.

* 1. **Висновки до розділу 1**

В цьому розділі описано феномен соціальних мереж, їхню тепер уже невід’ємну роль у житті сучасної людини. Середньостатистичний житель планети Земля вже не може уявити своє життя без інтернету. Багато людей використовують соціальні мережі для розваг та проведення вільного часу. У мережі для цього з’явилося безліч сервісів та платформ. Це породило нову професію – блогер або контентмейер.

У багатьох блогерів їхнє хобі, створювати контент перетворилося у справжню та повноцінну роботу. Найпопулярнішим напрямком у блогерстві є створення фото чи відео-контенту. Найпопулярнішими платформами є TikTok, Facebook, YouTube, Twitter та Instagram. Існує багато способів заробітку грошей зі свого контенту, тобто монетизації. Найвідомішими є: продає реклами, спонсорство, пожертвування, продаж мерчендайзу.

**РОЗДІЛ 2** **ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ**

* 1. **Загальна постановка завдання**

Завдання полягає в створенні системи, яка дозволить монетизувати прямий ефір шляхом приймання пожертв від аудиторії. Для звичайного користувача система дасть змогу взаємодіяти з контентмейкером та фінансово його підтримати. У свою чергу контентмейкерам система має дати змогу створювати інструменти для відображення пожертв, візуалізації статистики та певних показників (список з останніх донатерів, список з найкращих за певний час донатерів тощо) на прямій трансляції. Для цього було вирішено розробити веб-застосунок зі зручним та інтуїтивно зрозумілим інтерфейсом та можливістю створювати спеціальні віджети, які є веб-сторінкою, що відображають потрібну інформацію про пожертвування у режимі реального часу.

Вхідними даними для виконання даного завдання є специфікація вимог та список обраних інструментів та рішень.

Вихідними даними є розроблене програмне забезпечення яке реалізує поставлену мету.

* 1. **Специфікація вимог до програмного продукту**
     1. **Вступ**
        1. **Призначення та мета**

Основними призначеннями даної системи є монетизація прямих ефірів шляхом пожертвувань від аудиторії. За допомогою такої системи неавторизовані користувачі зможуть пожертвувати кошти автору прямої трансляції. В даній системі буде реалізована авторизація, яка дозволить створювати користувачів-творців (стрімерів) , які зможуть створювати візуальні елементи для прямої трансляції, які демонструють взаємодію з аудиторію та їхню, фінансову підтримку. Отримувачам пожертв система може дати можливість виводити сповіщення про надходження пожертв, статистики пожертв та прогресу збору коштів у прямому ефірі, даючи веб-посилання на спеціальні «віджети», які можна додати до програми для проведення прямих трансляцій.

* + - 1. **Продукти-аналоги**

Враховуючи те, що прямі трансляції стали дуже популярними та приносять колосальні прибутки, систем, які дозволяють монетизувати контент саме для українських творців, на мій погляд, мало і конкуренції між ними немає. Тому, багато українських блогерів користуються іноземними сервісами. Одним із колись найпопулярніших був рашистський сервіс Donation Alerts (<https://www.donationalerts.com/>), але зараз із зрозумілих причин цей сервіс заблокований на території України та його використання є недоцільним. Популярним сервісом є каліфорнійський streamlabs (https://streamlabs.com/) , що окрім можливості монетизації прямої трансляції, дає спеціальну модифікацію сервісу OBS для проведення прямих трансляцій. Проте сервіс має велику комісію та погано взаємодіє з українською валютою.

Для використання в Україні існують декілька вітчизняних сервісів. Найвідомішим є Дяка (Diaka) ( <https://diaka.ua/amortisation>) – сервіс є дуже простим та має стандартний набір можливостей подібних сервісів. Комісія зі зборів варіюється в залежності від суми (чим більша сума, тим менша комісія). Ще одним українським сервісом є Donatello (<https://donatello.to/>) – він є майже аналогічним, але дає дещо більше можливостей. За даними самого сайту на квітень 2022 року кількість користувачів становить 6176 зареєстрованих. Проте сервіс використовує підсистему Foundy, тому комісія є досить великою, яка включає комісію самого сервісу та Foundy (3,75% + 5%).

* + 1. **Загальний опис**

**2.2.2.1. Характеристики продукту**

Основними можливостями системи є:

* Реєстрація та авторизація
* Надсилання пожертвувань (донатів) творцю
* Створення віджетів для відображення повідомлень (донатів)
* Створення віджетів для відображення статистики
* Створення віджетів для відображення «збору коштів»
* Створення власних віджетів
* Створення сторінки-візитівки автора
* Вивід коштів з системи
* Відображення віджетів у прямому ефірі
* Відображення повідомлень (донатів) у прямому ефірі
* Відображення статистики у прямому ефірі
* Відображення «збору коштів» у прямому ефірі

**2.2.2.2. Класи користувачів**

Система матиме кілька два класи користувачів, що будуть володіти різними привілеями і для яких буде доступний різний функціонал.

* Неавторизований користувач – йому не потрібно проходити процедуру авторизації. З цими правами з системою будуть працювати представники цільового сегмента користувачів – люди, які хочуть фінансово підтримати контентмейкера або переглянути його профіль щоб отримати покликання на соціальні мережі блогера.
* Контентмейкер (блогер, автор) – для входу в систему повинен пройти авторизацію. Має можливість заповнити інформацію про себе, додати опис, додати покликання на соціальні мережі. Має можливість створювати спеціальні віджети, та використовувати їх для прямих трансляцій чи інших власних потреб. Також він має можливість вивести накопичені у системі кошти.

**2.2.2.3. Середовище функціонування**

Для функціонування серверної частини системи потрібні:

● Платформа з встановленим ASP.NET Core та всіма необхідними пакетами, які описані в файлі project.assets.json.

● Сервер бази даних – SQL Server, на якому створені та заповнені необхідні колекції даних.

Для коректної роботи клієнтської частини, потрібен будь-який із сучасних веб-браузерів.

* + 1. **Характеристики системи**
       1. **Створення віджета «Сповіщення»**

2.2.3.1.1. Опис і пріоритет

Користувач з правами контентмейкера має можливість створити та налаштувати віджет «Сповіщення» («Notification»), який у режимі реального часу відображає повідомлення-пожертвування від інших користувачів (фанатів).

Пріоритет: Високий.

2.2.3.1.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Сповіщення» («Notification»), та натиснути «Створити новий». Після цього відкриється форма для створення віджета, яка містить конфігураційні поля та має можливість обрати файли (анімацію та звук), які супроводитимуть сповіщення про донат.

2.2.3.1.3. Функціональні вимоги

1. Відображення доступних змінних у формуванні шаблону тексту-заголовку.
2. Можливість попереднього перегляду та прослуховування доданих файлів.
3. Валідація усіх полів .
4. Можливість додати декілька файлів.
5. Можливість зміни та видалення доданих файлів.
6. Генерація посилання на веб-сторінку відображення віджету.

**2.2.3.2. Створення віджета «Статистика»**

2.2.3.2.1. Опис і пріоритет

Користувач з правами контентмейкера має можливість створити та налаштувати віджет «Статистика» («Statistic»), який у режимі реального часу відображає обрані користувачем статистичні дані у обраному ним вигляді та форматі.

2.2.3.2.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Статистика» («Statistic»), та натиснути «Створити новий». Після цього відкриється форма для створення віджета, яка містить конфігураційні поля та має можливість налаштувати стиль відображення віджету.

2.2.3.2.3. Функціональні вимоги

1. Відображення доступних змінних у формуванні шаблону тексту-заголовку.
2. Можливість вибору типу віджету («Топ» (рейтинг жертводавців), «Останній донатер», «Зібрана сума»).
3. Валідація усіх полів .
4. Можливість вибору способу відображення віджету («Список», «Бігучий рядок», «Слайдер»)..
5. Можливість вибору напрямку руху відображення віджету.
6. Можливість вибору часових рамок (часового періоду). за які будуть обиратися статистичні дані для віджету.
7. Можливість установити швидкість анімації (прокручування).
8. Генерація посилання на веб-сторінку відображення віджету.

**2.2.3.3. Створення віджета «Збір коштів»**

2.2.3.3.1. Опис і пріоритет

Користувач з правами контентмейкера має можливість створити та налаштувати віджет «Збір коштів» («Foundraising»), який у режимі реального часу відображає обрані користувачем статистичні дані у обраному ним вигляді та форматі.

Пріоритет: Високий.

2.2.3.3.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Збір коштів» («Foundraising»), та натиснути «Створити новий». Після цього відкриється форма для створення віджета, яка містить конфігураційні поля та має можливість налаштувати стиль відображення віджету.

2.2.3.3.3. Функціональні вимоги

1. Відображення доступних змінних у формуванні шаблону тексту-заголовку.
2. Можливість увімкнути та вимкнути віджет.
3. Валідація усіх полів .
4. Можливість змінити початкову, кінцеву та зібрану суми.
5. Можливість задати розмір віджету, ширину рамок та заокруглення.
6. Можливість зміни кольори відображення елементів віджету.
7. Генерація посилання на веб-сторінку відображення віджету.

**2.2.3.4. Відображення та робота віджета «Повідомлення»**

2.2.3.4.1. Опис і пріоритет

Користувач який має доступ до посилання на віджет, має можливість відвідати та використовувати веб-сторінку яка відображає роботу віджета у власних потребах. Веб сторінка віджета у режимі реального часу реагує на надходження пожертв контентмейкеру та відображає їх. Веб-сторінка показує заголовок та текст повідомлення, супроводжуючи його появу налаштованим звуком та картинкою. В залежності від налаштувань віджету, повідомлення може бути зачитане синтезом мови.

Пріоритет: Високий.

2.2.3.4.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Повідомлення» («Notification»), обрати у виведеному списку усіх створених віджетів бажаний та натиснути на кнопку «Відобразити». Після цього відкриється веб сторінка з відображенням роботи віджету .

2.2.3.4.3. Функціональні вимоги

1. Відображення заголовку повідомлення відповідно до заданого шаблону.
2. Відображення картинки.
3. Відтворення звуку повідомлення.
4. Відтворення звуків повідомлення та відображення картинок з заданого набору у вибраній послідовності (по черзі чи випадково).
5. Синтез мовлення тексту повідомлення.
6. Відображення повідомлень у режимі реального часу.
7. Можливість установити швидкість анімації (прокручування).

**2.2.3.5. Відображення та робота віджета «Статистика»**

2.2.3.5.1. Опис і пріоритет

Користувач який має доступ до посилання на віджет, має можливість відвідати та використовувати веб-сторінку яка відображає роботу віджета у власних потребах. Веб сторінка віджета оновлюється декілька раз за хвилину, щоб показувати актуальні дані. Веб-сторінка показує статистичні дані у заданому виді та форматі.

Пріоритет: Високий.

2.2.3.5.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Статистика» («Statistic»), обрати у виведеному списку усіх створених віджетів бажаний та натиснути на кнопку «Відобразити». Після цього відкриється веб сторінка з відображенням роботи віджету .

2.2.3.5.3. Функціональні вимоги

1. Відображення заголовку рядка-елемента відповідно до заданого шаблону.
2. Оновлення (актуалізація) даних.
3. Відображення статистичних даних у вигляді списку.
4. Відображення статистичних даних у вигляді бігучого рядку.
5. Відображення статистичних даних у вигляді слайдеру.
6. Відтворення анімації (прокручування) за заданою швидкістю.
7. Відображення вибраних статистичних даних («Топ» (рейтинг жертводавців), «Останній донатер», «Зібрана сума»).
8. Відображення руху віджету у вибраному напрямку.
9. Відображення даних віджету відповідно до обраних часових рамок (часового періоду).

**2.2.3.6. Відображення та робота віджета «Збір коштів»**

2.2.3.6.1. Опис і пріоритет

Користувач який має доступ до посилання на віджет, має можливість відвідати та використовувати веб-сторінку яка відображає роботу віджета у власних потребах. Веб сторінка віджета оновлюється декілька раз за хвилину, щоб показувати актуальні дані. Веб-сторінка показує статистичні дані у заданому виді та форматі.

Пріоритет: Високий.

2.2.3.6.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Збір коштів» («Fundraising»), обрати у виведеному списку усіх створених віджетів бажаний та натиснути на кнопку «Відобразити». Після цього відкриється веб сторінка з відображенням роботи віджету .

2.2.3.6.3. Функціональні вимоги

1. Відображення заголовку відповідно до заданого шаблону.
2. Оновлення (актуалізація) даних.
3. Відображення віджету відповідно до заданих параметрів стилю (розміри та кольори елементів).
4. Відображення зібраної суми у відсотках.
5. Можливість вимкнення роботи віджету (нові пожертвування перестають враховуватися).

**2.2.3.7. Керування профілем**

2.2.3.7.1. Опис і пріоритет

Авторизований користувач (контентмейкер) має можливість керувати власним профілем. Він має можливість змінити пароль, змінити адрес електронної пошти, завантажити власні персональні дані та видалити їх з системи.

Пріоритет: Високий.

2.2.3.7.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Керування профілем» («Manage Accaunt»). Далі на лівій боковій панелі з’явиться список чотирьох доступних розділів: «Профіль», «Електронна пошта», «Пароль», «Персональні дані».

2.2.3.7.3. Функціональні вимоги

1. Можливість змінити адресу електронної пошти.
2. Можливість змінити пароль.
3. Можливість змінити персональні дані.
4. Можливість завантажити усі персональні дані.
5. Можливість видалити усі персональні дані

**2.2.3.8. Надсилання пожертв**

2.2.3.8.1. Опис і пріоритет

Неавторизований користувач має можливість надіслати контентмейкеру пожертву (донат).

Пріоритет: Високий.

2.2.3.8.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією користувач повинен мати посилання на профіль блогера, або знати його псевдонім. Далі він переходить за цим посиланням, заповнює потрібні поля (ім’я донатера, сума, повідомлення). Далі він тисне кнопку «Надіслати» і створює платіж.

2.2.3.8.3. Функціональні вимоги

1. Можливість обрати запропоновану суму.
2. Можливість вводу власної суми.
3. Можливість вводу повідомлення.
4. Можливість перейти на профіль (сторінку-візитівку) контентмейкера.

**2.2.3.9. Сторінка-візитівка**

2.2.3.9.1. Опис і пріоритет

Авторизований користувач (контентмейкер) має можливість налатувати свою власну сторінку-візитівку, яка міститиме коротку інформацію про нього та посилання на соціальні мережі. Будь який користувач може переглянути сторінку-візитівку контентмейкера та має змогу перейти на сторінку для пожертв.

Пріоритет: Середній.

2.2.3.9.2. Послідовність дія/відгук

Для того, щоб скористатися даною функцією потрібно у особистому кабінеті користувача перейти до розділу «Панель Користувача» («User panel»). Далі на лівій боковій панелі потрібно натиснути на розділ «Візитівка». Після цього відкриється форма для заповнення потрібних даних та посилань, вибору картинок для профілю.

2.2.3.9.3. Функціональні вимоги

1. Можливість змінити тло профілю.
2. Можливість додати посилання на соціальні мережі.
3. Можливість додати коротку автобіографію.
4. Можливість змінити фото профілю
   * 1. **Вимоги зовнішніх інтерфейсів**
        1. **Користувацькій інтерфейси**

Інтерфейс системи (веб версія) складається з:

* «Домашня» сторінка, яка містить опис сервісу та його можливостей;
* Розділ «Панель користувача» який містить форми для створення/редагування/видалення віджетів, списки створених віджетів та отриманих повідомлень ;
* Розділ «Керування акаунтом» який містить форму для зміни особистих даних користувача;
* Бічне меню в якому містяться список підрозділів та кнопки переходу між розділами;

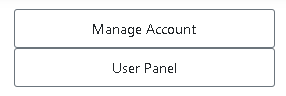


Рис. 2.1. Кнопки переходу між розділами

* На правому-верхньому куті карти розташовані дві кнопки, вигляд яких залежить від того, чи користувач авторизований (неавторизований): перехід до панелі користувача (вхід у систему); вихід з системи (реєстрація).



Рис. 2.2. Кнопки “Hello {username}” – перехід у панель користувача(зліва) та “Logout” – вихід з системи (справа)



Рис. 2.2. Кнопки “Login” – вхід в систему(зліва) та “Register” – реєстрація (справа)

**2.2.5.** **Інші нефункціональні вимоги**

* + - 1. **Вимоги безпеки**

В проекті наявна система авторизації, яка дозволяє авторизованим користувачам додавати або змінювати віджети, налаштовувати персональну сторінку та спосіб отримання коштів. В базі даних, у записах про користувачів системи паролі зберігаються в зашифрованому вигляді, тому, якщо у разі скомпрометації бази даних, зловмисники не зможуть здійснити вхід в систему, так як паролі до облікових засобів, які вони отримають, будуть зашифровані.

* + - 1. **Вимоги продуктивності**

Кожен запит серверної частини має виконуватися асинхронно та максимально оптимізовано для найшвидшого відклику до сторони клієнта. Сповіщення про надходження пожертвування повинне відображатися у режимі реального часу з мінімальною затримкою.

* + - 1. **Атрибути якості програмного продукту**

Веб-аплікація повинна бути повністю адаптивною, тобто користувачі як з мобільних пристроїв повинні також мати змогу зручно і швидко користуватися системою. Повинна бути забезпечена валідація введених користувачем даних в будь-які наявні в системі поля, а також передбачені всі виняткові ситуації. У разі хибності введених даних система повинна вивести повідомлення про помилку. Для того, щоб забезпечити супровід даного продукту, та перенесення ресурсів системи на іншу платформу, повинні бути розроблені супутні інструкції.

* 1. **Вибір засобів для виконання завдання**

Для розробки даної системи було використано клієнт-серверну архітектуру, тому вона відповідно складається з клієнтської та серверної частини. Для розробки серверної частини буде використано платформу Microsoft .NET Core.

Ключовим фреймворком на сервері є ASP.NET Core – це кросплатформенний веб-фреймворк для проектів .NET Core, який надає широкий набір функцій для веб-аплікацій. Фреймворк об’єднує у єдину модель раніше окремі технології ASP.NET MVC та ASP.NET Web API. ASP.NET Core включає у себе такі корисні компоненти як Entity Framework Core, Identity Core, MVC Core та Razor Core. Для побудови клієнтської частини було використано вище згадану компоненту Razor Core, яка дозволяє будувати веб сторінки з використанням як html+css+javascript так і C# коду. Вона взаємодіє з архітектурною компонентою MVC Core яка забезпечує зв’язок між клієнтською та серверною частинами. Для збереження даних буде використовуватись не реляційну базу даних – SQL Server. Для взаємодією з нею, на серверній частині буде використовуватися Entity Framework Core - структура об'єктно-реляційного відображення (ORM) яка використовується для взаємодії із базою даних.

Для створення під-системи авторизації та реєстрації та логіки користувачів була використана Identity Core, що дає готовий, широкий, зручний та гнучкий для змін функціонал. Для стилізації компонент інтерфейсу буде використовуватись фреймворк - Bootstrap. Для реалізації відображення сповіщень у реальному часі було використано SignalR – open source бібліотека для Microsoft ASP.NET, яка дозволяє серверній частині надсилати асинхронні сповіщення до веб-програм на стороні клієнта. Бібліотека також включає в себе компоненти JavaScript на стороні сервера і клієнта.

* 1. **Висновок до розділу 2**

За допомогою результатів аналізу було створено специфікацію вимог до програмного забезпечення, в якій було описано основні сервіси системи та функціональні вимоги до них. Було визначено призначення та мету програмного продукту. Визначено та проаналізовано класи користувачів даної системи. Також були обрані програмні засоби для розробки даної веб-аплікації.

**РОЗДІЛ 3** **АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ/ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ**

* 1. **. Проектування поведінки розроблюваної системи**

В попередньому розділі було визначено два класи користувачів, які будуть наявні в системі, та функціональні вимоги які повинні бути реалізовані в проекті для задоволення потреб всіх цих класів користувачів. В даному розділі описано шляхи їхньої взаємодії з системою та доступний їм функціонал.

*Неавторизований користувач* увійшовши в систему попаде на головну (домашню) сторінку платформі, де він зможу ознайомитися з її можливостями та правилами використовування. Крім цього, він зможе авторизуватися у системі або зареєструватися. Також він має доступ по сторінки-профілю контентмейкера. Для цього йому потрібно мати на неї посилання. На цій сторінці він має змогу ознайомитися з короткою біографією блогера, перейти на соціальні мережі блогера а також здійснити пожертвування. Для пожертвування він має заповнити поле сума і по бажанню вказати свій псевдонім та вміст повідомлення. Також неавторизований користувач має доступ до перегляду роботи віджетів, але тільки у випадку, якщо він має на нього покликання. Проте це покликання не є у публічному доступі, а доступне тільки блогеру (власнику віджета), і блогеру варто зберігати його у таємниці, так як к логікою програми передбачений анонімний доступ до сторінки віджета, для його відображення у програмах для проведення прямих трансляцій.

*Авторизованому користувачу або контентмейкеру (блогеру)* після авторизації до системи доступний весь функціонал, що й неавторизованому користувачу, а також доступ до панелі користувача, яка містить наступні можливості:

* керування персональними даними (зміна паролю, електронної пошти, номеру картки);
* перегляд статистики;
* керування віджетами «Сповіщення» («Notification»), «Збір коштів» («Foundraising») та «Статистика» («Statistic») – створення, редагування, видалення, отримання покликання на сторінку роботи віджета;
* налаштування сторінки профілю (сторінки візитівки) – додавання фото профілю, тла профілю, додавання посилання на соціальні мережі;
* перегляд отриманих сповіщень;

В Додатку A зображена діаграма прецедентів, на якій розміщені актори, які є класами користувачів, а також функціонал для кожного з наявних в системі користувачів

**3.2. Загальна архітектура системи**

Під час розроблення будь-якої програмної системи необхідно недостатньо тільки чітко визначити її вимоги і функціональність, важливо є розроблення архітектури проекту. Нехтування даним етапом, або здійснення значущої помилки в проектуванні, може спричинити на етапі розроблення проблеми які будуть нести за собою фінансові та часові втрати.

Найкращим варіантом для розроблення даної системи, буде побудувати архітектуру, базовану на клієнт-серверній взаємодії та використанням принципів MVC. Уся ця архітектурна буде реалізована за допомогою MVC Core. На стороні клієнта працюватиме веб-застосунок з використанням бібліотеки Microsoft .NET Core, за допомогою якого ми будемо представляти дані клієнту і опрацьовуватимемо його команди. Для авторизації, реєстрації нових користувачів, надсилання пожертв, роботи з даними, та інших операцій, які потребують виконання запитів до бази даних буде створена за допомогою компоненти MVC Core – контролери, які повертатимуть як результат відображення (сторінку) або REST відповідь, що буде серверною частиною системи.

Для реалізації логіки серверних ролей, акаунтів, авторизавції та реєстрації була обрана вбудована компонента Identity Core, яка автоматично створює потрібні модулі та всю логіку.

Для роботи з базою даних буде використано бібліотеку Entity Framework, яка дозволить проводити операції з базою даних за допомогою протоколу SSL. Базою даних для системи служитиме SQL Server.

Логічне представлення вище описаної архітектури зручно зобразити за допомогою діаграми розгортання, яка зображена на рис. 3.1.

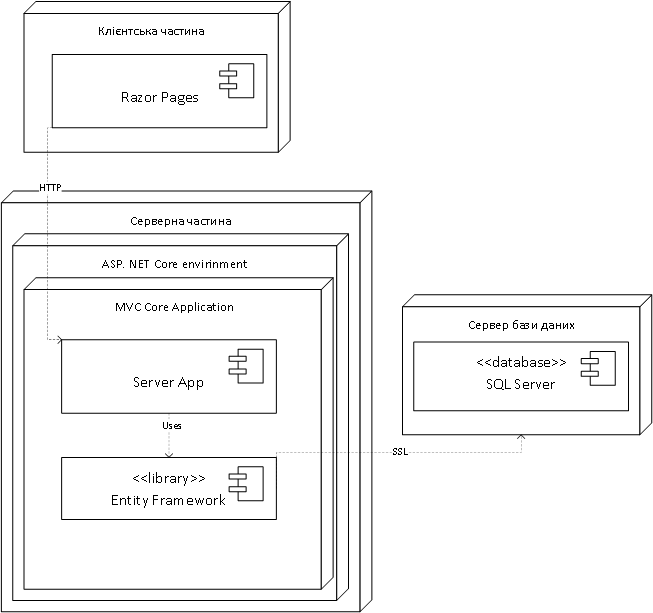


Рис. 3.1. Діаграма розгортання

**3.3. Проектування бази даних**

Для формування нашої бази даних було обрано SQL Server – одну з найпопулярніших баз даних. Дана СУБД підтримує транзакції та дозволяє створити структури даних, які б дозволяла б нам зберігати в певні поля тільки дані певного типу. Дана СУБД добре взаємодіє з ASP.NET та є чудовим вибором для даної технології.

Для створення структури бази даних було обрано так званий підхід “Code First”. Так як структура програми не є складно і не потребує складних рішень, то цей підхід є оптимальним. Цей підхід є одним із можливостей Entity Framework та Visual Studio та дозволяє створювати структуру БД за допомогою структури програми. Тобто, спершу створюються класи програми та налаштовується їх зв’язок між собою. Потім, за допомогою scaffold процедури, середовище Visual Studio створює потрібні таблиці та зв’язки. Для зміни у структуру БД використовуються міграції.

У базі даних присутні таблиці які мають префікс “AspNet”. Ці таблиці створені автоматично за допомогою ASP Identity.

В Додатку Б зображена діаграма бази даних.

У базі існують основні таблиці, та таблиці, які з’явилися у наслідок нормалізації (це таблиці типів/категорій). Дані авторизованого користувача зберігаються у таблиці “AspNetUsers”, яка має зв’язок один до одного з таблицею “UserProfile” – таблиця є собою сторінку профіля користувача. “UserProfile” має два файли для фотографій, тому вона має два зв’язки один до багатьох з таблицею “UploadData” – таблиця містить дані про завантажений до системи файл. Донат-сповіщення зберігаються у таблиці “DonateMsg” яка має зв’язок один до багатьох з таблицею “AspNetUsers”.

Для кожного віджета існує окрема таблиця. Всього їх є три: “FundraisingWidget”, “StatisticWidget” та “NotificationWidget”. Кожен віджет має стиль тексту, тому кожна з цих трьох таблиць має зв’язок один до багатьох з таблицею “TextStyle”, яка містить такі поля як: колір тексту, розмір шрифту, сім’ю шрифту, розмір простору між буквами та словами, прапорці «жирний», «косий», «підкреслений». Кожна з таблиць віджетів має зв’язок один до багатьох з “AspNetUsers”, так як користувач є власником і керівником віджетів та може мати багато їх у власності . Кожна таблиця віджетів має наступні спільні поля: назва, URL-посилання на сторінку відображення віджету, ідентифікатор контентмейкера. “FundraisingWidget” – це таблиця, яка зберігає дані віджета «Збір коштів», зберігає дані про початкову, кінцеву та зібрану суму грошей та кольори стилю відображення віджета. “StatisticWidget” – це таблиця, яка зберігає дані віджета «Статистика». Цей віджет має чотири категоріальних поля, тому таблиця має чотири зв’язки один до багатьох з наступними таблицями: “StatWidgetDirectionType”, “StatWidget”, “StatWidgetDisplayModeType”, “StatWidgetTimeIntervalType”. “NotificationWidget” – це таблиця, яка зберігає дані віджета «Сповіщення». Ця таблиця має зв’язок один до багатьох з таблицею “MsgWidgetContent”, яка зберігає набір зображення + звук для віджету. Так як ці зображення та звук є файлами, то вони зберігаються у вище згаданій таблиці “UploadData”, до якої “MsgWidgetContent” має зв’язок один до багатьох.

**3.4. Проектування серверної частини**

В загальному описі архітектури системи було визначено, що нашому програмному застосунку знадобиться серверна частина, яка надаватиме нам REST API та сторінки-відображення для запитів зі сторони клієнта.

Як було згадано вище, система буде реалізована за допомогою ASP.NET MVC Core, тому серверна частина буде складатися з ASP контролерів. Основу контролерів можна згенерувати за допомогою scaffold процедури і вносити зміни у згенерований код. В основі усіх контролерів лежать декілька основних функцій: створити (Create), редагувати (Edit), видалити (Delete), деталі (Details), список (Index). Проте кожна із цих функцій, окрім Index та Details мають два перевизначених формати для GET та POST запитів. GET запит повертає сторінку-відображення (форму) дії, а POST вже опрацьовує цю дію та вносить зміни до бази даних.

Серверна також частина містить SignalR «Хаб» (Hub), за допомогою якого сповіщення про донат надсилаютсья у режимі реального часу.

Маршрутизація реалізована також за допомогою MVC Core. Левова частка усіх кінцевих точок мають формат “{ControlerName}/{ActionName}”, за рідкісним виключенням для деяких функцій: “Donate/{nickname}” – містить кінцеву точку сторінки профіля контентмейкера з можливістю зробити пожертву, та маршрутів підсистеми Identity.

Підсумовуючи, на серверній частині слід описати декілька кінцевих точок (API endpoints), які будуть обробляти клієнтські запити. Список основних шляхів:

* /Identity/ – містить кінцеві точки для підсистеми акаунів, яка включає авторизацію, реєстрацію, керування акаунтом;
* /Panel – містить кінцеві точки панелі користувача;
* /MsgWidgets – містить групу кінцевих точок для роботи з віджетами «Сповіщення», зокрема для читання, створення, редагування, відображення та видалення;
* /FundraisingWidgets – містить групу кінцевих точок для роботи з віджетами «Збір коштів», зокрема для читання, створення, редагування, відображення та видалення;
* /StatisticWidgets – містить групу кінцевих точок для роботи з віджетами «Статистика», зокрема для читання, створення, редагування, відображення та видалення;
* /UserProfiles/ містить кінцеві точки для роботи з сторінкою-профілем контентмейкера;
* /{ControlerName}/{ActionName} – кінцеві точки які містять функції контролерів;
* /donates – група кінцевих точок для хабу SignalR.

Діаграма послідовності (дивись Додаток В). дає змогу вдало продемонструвати роботу серверної частини системи, візуалізацію запитів, та їх подальше опрацювання.

**3.5. Проектування клієнтської частини**

Хоча робота програми MVC керується головним чином контролерами, але безпосередньо користувачеві програма доступна у вигляді сторінка-вигляд, яке формує зовнішній вигляд програми. В ASP.NET MVC сторінка-вигляд - це файли з розширенням cshtml, які містять код інтерфейсу користувача в основному на мові html. Тому для реалізації клієнтської частини буде використовуватися компонента Razor Core, яка є частиною MVC Core. Razor Core дозволяє використовувати код C# разом з html, що дозволяє швидко та зручно будувати потрібні веб-сторінки, які використовуватимуть модель даних, яка передаватиметься з серверної частини. Технологія Razor дозволяє створювати partial view (часткові сторінки), що дозволяє використовувати їх у декількох інших сторінках, це своєрідна модульність.

При виклик методу View контролер не виробляє рендеринг уявлення і не генерує розмітку html. Контролер лише готує дані та вибирає, яке уявлення треба повернути як об'єкт ViewResult. Потім вже об'єкт ViewResult звертається до двигуна представлення для рендерингу представлення у вихідний результат.

Клієнтська частина буде складатися з чотирьох типів веб сторінок: Razor Page (.cshtml+.cshtml.cs), Razor View(.cshtml), Razor Layout View (.cshtml) та Razor Component (.razor). Клієнтська частина є складовою єдиного проектного рішення (Visual Studio Project). Клієнтські сторінки розділені на дві папки: \Views\ (містить основні сторінки системи, тобто контролерів) та \Area\Identity\Pages \(містить сторінки для під-системи ідентифікації).

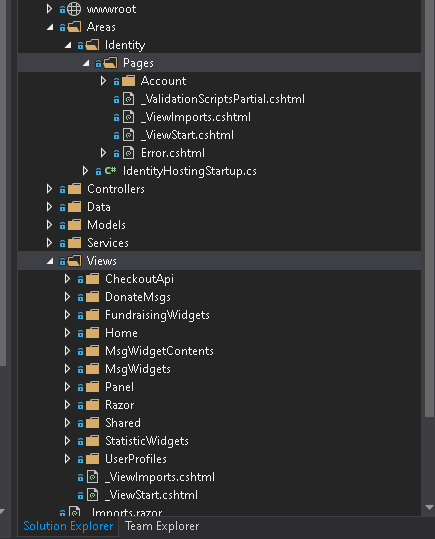


Рис.3.5.1. Клієнтська частина проекту

У папці \Views\ знаходяться під-папки, які мають назви контролерів і які містять сторінки контролерів. Також, у папці \Views\ знаходиться папка “Shared”, яка містить у собі макети (Razor Layout), які є основою інших сторінок.

У папці \Area\Identity\Pages міститься під-система ідентифікації. Вона включає у себе як і клієнтське представлення (.cshtml) так і серверну бізнес логіку (.cshtml.cs), яка під час компіляції розділяється. Дані сторінки реалізовані за допомогою Razor Page, яка використовує Code Behind.

Для зручної та швидкої роботи було створено декілька часткових сторінок та компонент:

* \_FundraisingWidgetForm.cshtml – форма для віджету «Збір коштів»;
* \_MsgWidgetGrid.cshtml – форма для віджету «Сповіщення»;
* \_StatWidgetForm.cshtml – форма для віджету «Статистика»;
* \_PanelNav.cshtml – навігаціна панель;
* DisplayName.razor – компонента для власного відображення назви поля;

Для надання сторінка стилю я використав бібліотеку Bootstrap, яка дозволяє швидко та просто заставити сайт виглядати сучасно, мінімалістично та красиво.

**3.6. Висновок до розділу 3**

В даному розділі було описано та визначено загальну архітектуру системи у вигляді клієнта та сервера з використанням дизайн шаблону MVC. Для візуального відображення системи було створено діаграму розгортання. Для демонстрації доступних сценаріїв взаємодії та використання системи для двох наявних класів користувачів та наглядного відображення було створено діаграму прецедентів. Також було визначено структуру та зв’язки таблиць бази даних. Для відображення деталей таблиць (полів, їхніх типів даних, ключів, зв’язків) було сформовано діаграму бази даних. Було описано роботу серверної частини та її зв’язок з клієнтською частиною. Так як використано підхід MVC, то обидві частини між собою тісно пов’язані. Було описано наявні кінцеві точки, на які клієнт буде відправляти запити, і сценарії для опрацювання кожного з цих запитів. Для демонстрації роботи кінцевих точок та їх сценаріїв опрацювання було створено діаграму послідовності.

**РОЗДІЛ 4** **РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПЛАТФОРМИ ДЛЯ ФІНАНСОВОЇ ПІДТРИМКИ КОНТЕНТМЕЙКЕРІВ**

**4.1. Налаштування та реалізація серверної частини**

Основним інструментом для створення серверної частини програмної системи є ASP.Net MVC Core. З його допомогою було створено контролери та функції, що слугують кінцевими точками, до яких буде звертатися клієнтська сторона.

Для налаштування серверної сторони використовується файл Startup.cs, який містить потрібні функції.

Частина коду з файлу Startup.cs яка відповідає за налаштування серверної частини:

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

app.UseDatabaseErrorPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

endpoints.MapRazorPages();

endpoints.MapBlazorHub();

endpoints.MapHub<DonateHub>("/donates");

});

var cultureInfo = new CultureInfo("en-US");

CultureInfo.DefaultThreadCurrentCulture = cultureInfo;

CultureInfo.DefaultThreadCurrentUICulture = cultureInfo;

}

Для безпечного та гнучкого під’єдняння до бази даних, налаштування хостингу та деяких інших налаштувань створено файл appsettings.json який містить певні параметри та їх значення (ConnectionStrings, DefaultConnection, AllowedHosts тощо).

Для взаємодії з базою даних використовується Entity Framework Core. Основним його елементом є контекст бази даних (ApplicationDbContext) який містить інформацію про структуру БД.

**4.2. Опис роботи системи та програмних рішень на клієнтській частині**

Платформа для фінансової підтримки контентмейкерів, повністю адаптивна, та може використовуватися через будь-який сучасний браузер, за наявності інтернету на пристрої користувача. Інтерфейс користувача даної системи було розроблено за принципами евристики Якоба Нільсена, а отже він простий та зручний у використанні.

На рис. 4.1. зображена веб-версія інтерфейсу користувача системи, яку побачить користувач під час першого відкриття веб-застосунку . Як видно на рисунку, користувача вітає початкова сторінка додатку, на якій коротко описано можливості платформи.

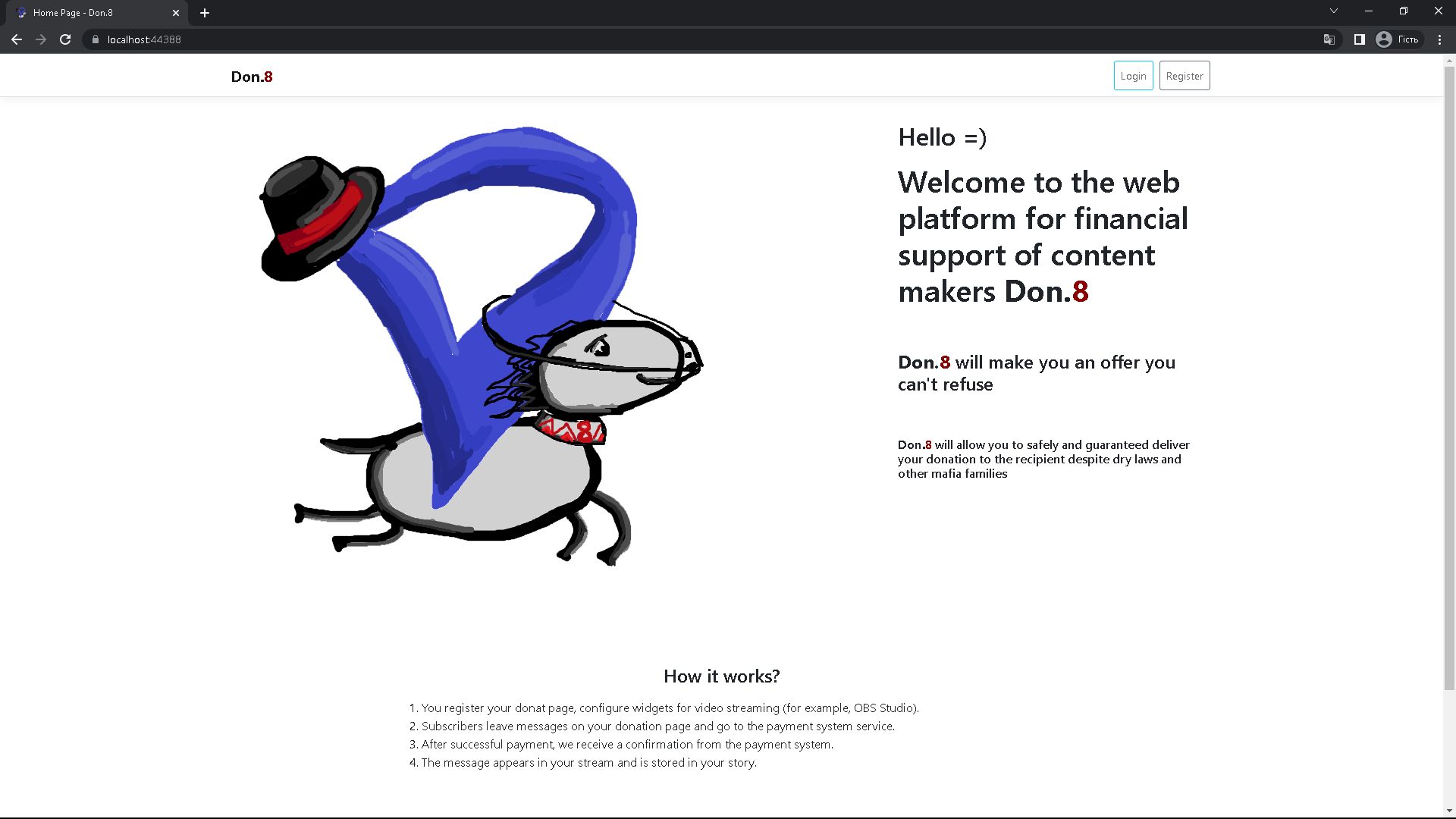


Рис. 4.1. Користувацький інтерфейс неавторизованого користувача для веб-версії платформи

Зверху даної сторінки розміщена верхня навігаційна панель, на якій доступні дві кнопки “Login”(Увійти) та “Register” (Зареєструватися). Після натиснення користувачем на одну з цих кнопок, від переходить до форми авторизації або реєстрації відповідно (див. рис.4.2. ).

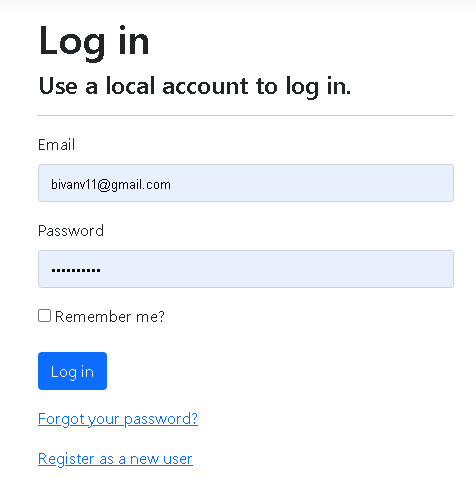
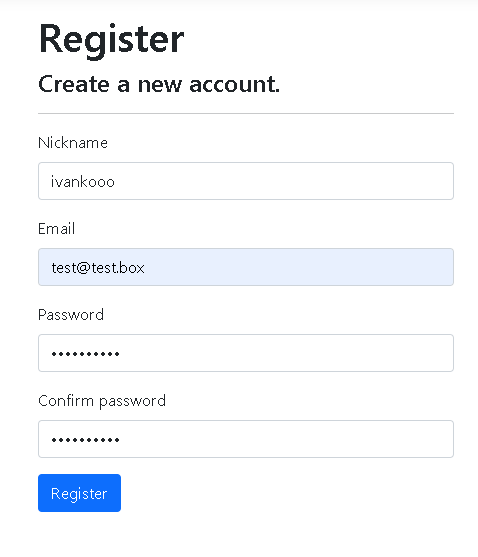
 

Рис. 4.2. Форми авторизації та реєстрації

Після завершення процесу авторизації, кнопки на навігаційній панелі замінюються на дві наступні відповідно: вхід до кабінету користувача (відображається як “Hello, {nickname}”, де замість {nickname} відображається псевдонім користувача ), вихід з системи (відображається як “Logout”).

При виході з системи, користувач повертається на початкову сторінку.

При натиску “Hello, {nickname}”користувач переходить до кабінет користувача, у розділ “Dashboard”, де він може побачити статистичні дані за певний період часу щодо пожертв: гістограма сума/день, список з 10 найкращих донорів, мінімальні, максимальні та середні суми пожертв. На рисунку 4.3. зображена панель користувача та розділ “Dashboard”.

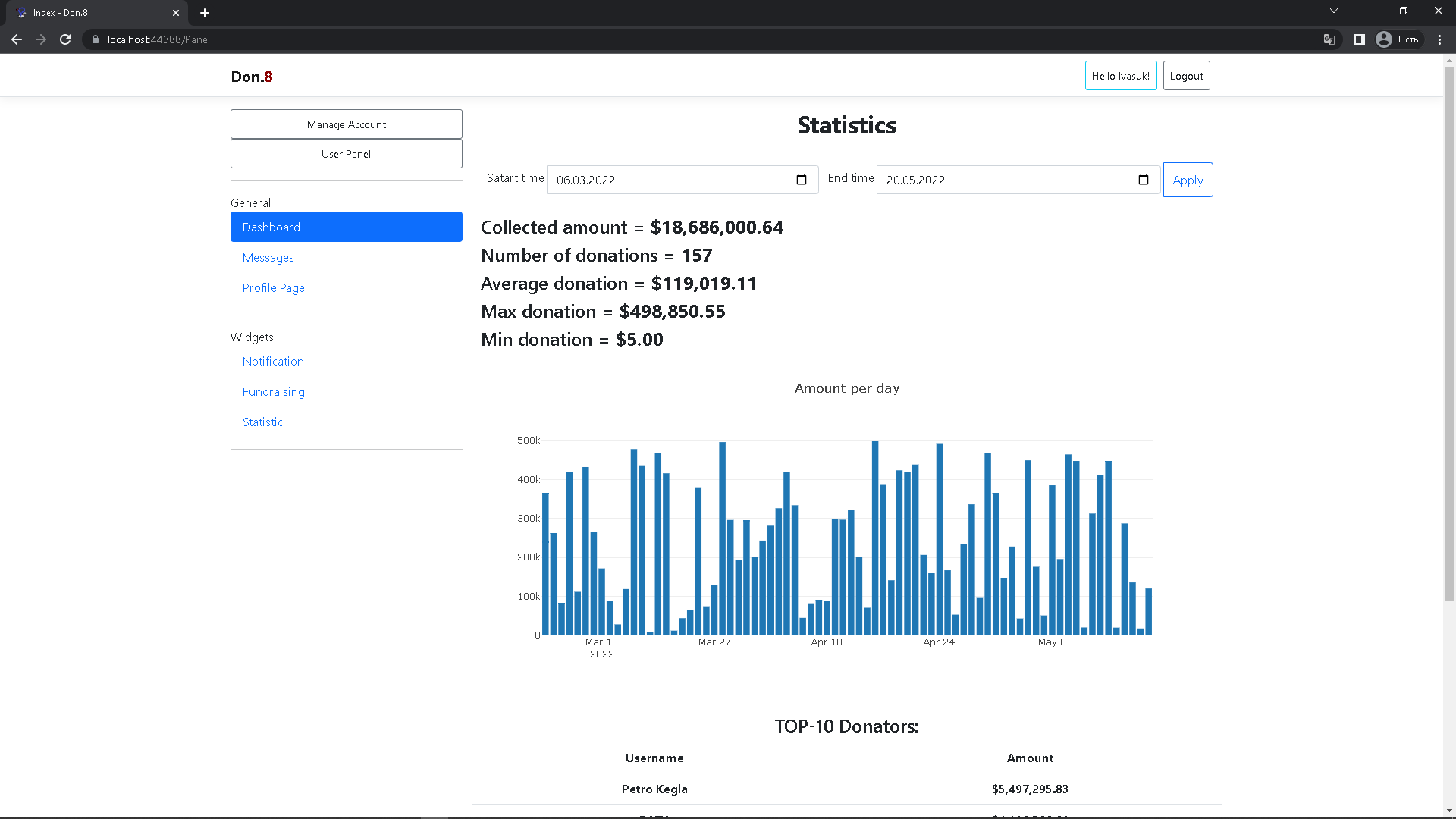


Рис. 4.3. Панель користувача, розділ “Dashboard”

У панелі користувача, зліва знаходиться бічне навігаційне меню, яке умовно поділене на три частини: перехід між розділами кабінету користувача «Панель користувача» та «Керування акаунтом», група підрозділів «Загальний», група підрозділів «Віджети».

У групі «Загальний» є три підрозділи: “Dashboard” (описаний вище), “Messages”, “Profile Page”. “Messages” – містить список усіх вхідних повідомлень (пожертв) за вибраний проміжок часу. “Profile Page” – містить форму редагування сторінки профілю контентмейкера (сторінки-візитівки).

У групі «Віджети» містяться підрозділи для роботи з віджетами: “Notification”, “Fundraising”, “Statistic”. При переході на будь який з підрозділів система виводить список віджетів того типу. У списку можна перейти на редагування, перегляд, деталі або видалення віджету. На рисунку 4.4. зображено підрозділ віджету “Statistic”. Для двох інших підрозділів Notification”, “Fundraising” список виглядає подібним чином, відрізняючись колонками.

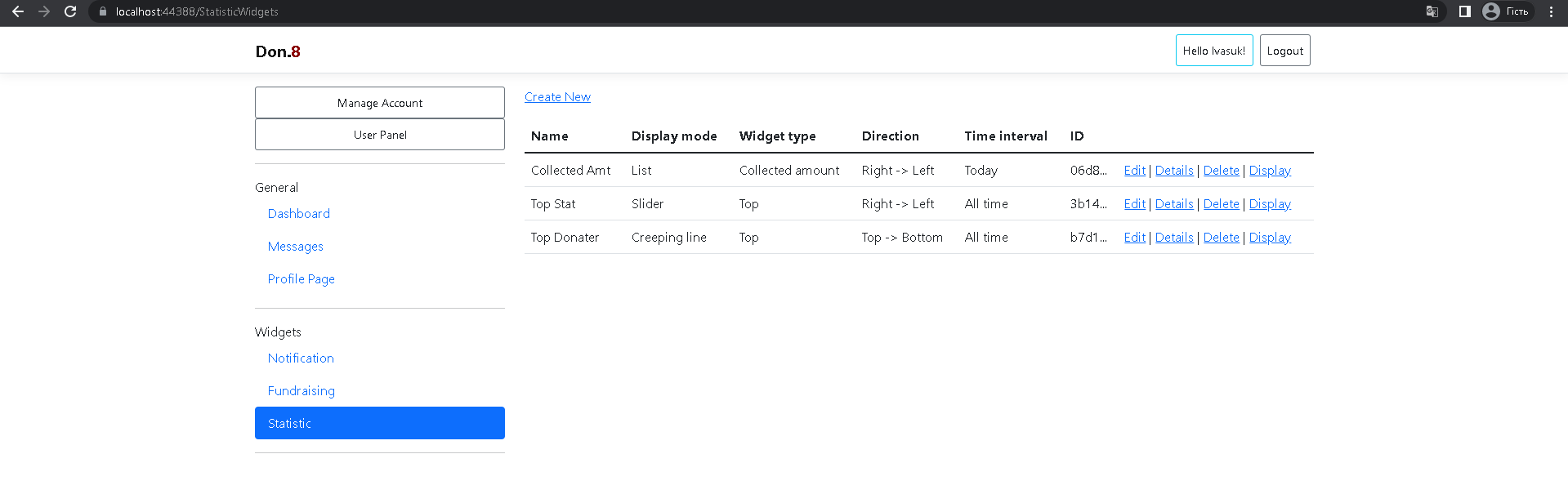


Рис. 4.4.Підрозділ “Statistic” для роботи з віджетом «Статистика»

Кожен з віджетів має деякі спільні поля, проте є і відмінні. Форми створення, редагування та видалення є схожими, проте для віджету “Сповіщення”, редагування та створення має додатковий візуальний елемент таблицю, яка слугує для вибору багатьох файлів (див. рис.4.5.). Після загальних полів є таблиця, яка має дві колонки: анімація, звук. Обидві колонки містять контролер для вибору файлу. Після вибору анімації, вище з’являється попередній перегляд анімованого зображення. Після вибору звуку, у музичному плеєрі, який розміщений вище контролера вибору файлу, появляється можливість прослухати обраний музичний файл.

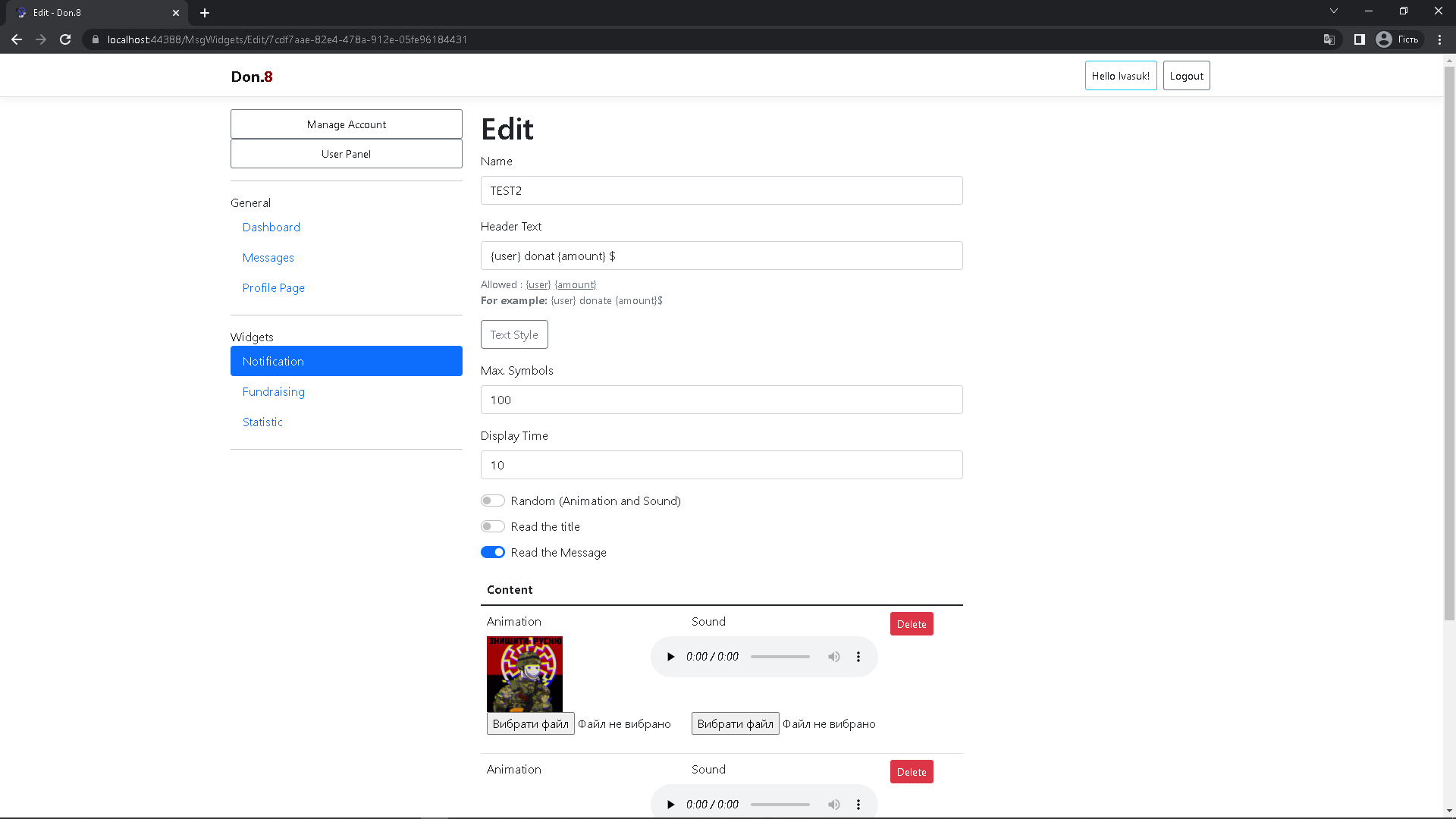


Рис. 4.5. Форма для створення/редагування віджету «Сповіщення»

Кожен із віджетів має свою сторінку його роботи, посилання на яку, контентмейкер і зможе використати для додавання віджета до програми для проведення прямих трансляцій. Для кожного типу віджета ця сторінка виглядає по різному та залежить від налаштувань кожного віджету. Для кожного створеного віджету створюється унікальна сторінка, посилання на яке міститься у деталях кожного віджету. На рисунку 4.6. зображено роботу усіх трьох віджетів. Для віджета «Сповіщення» – відображається одна з вибраних картинок, порядок якої залежить від параметру ‘Random”, ім’я донатера та сума донату у ораному форматі та текст повідомлення. Ім’я донатера та повідомлення можуть бути пустими. Для віджета «Збір коштів» – відображається процес збору коштів з назвою збору, початковою, кінцевою сумами та зібраним відсотком. Для віджета «Статистика» - відображається список з п’яти кращих донатерів, який відображається обраною анімацією.



Рис. 4.6. Відображення роботи усіх віджетів

Розділ кабінету користувача «Керування акаунтом» містить чотири підрозділа для роботи з особистими даними користувача: “Profile”, “Email”, “Password”, “Personal data”. Кожен з підрозділів містить звичайну форму для редагування полів певних даних. На рисунку 4.7. зображена форма редагування особистих даних у підрозділі “Profile”, яка містить поля: ім’я користувача, псевдонім, номер телефону, номер банківської картки.

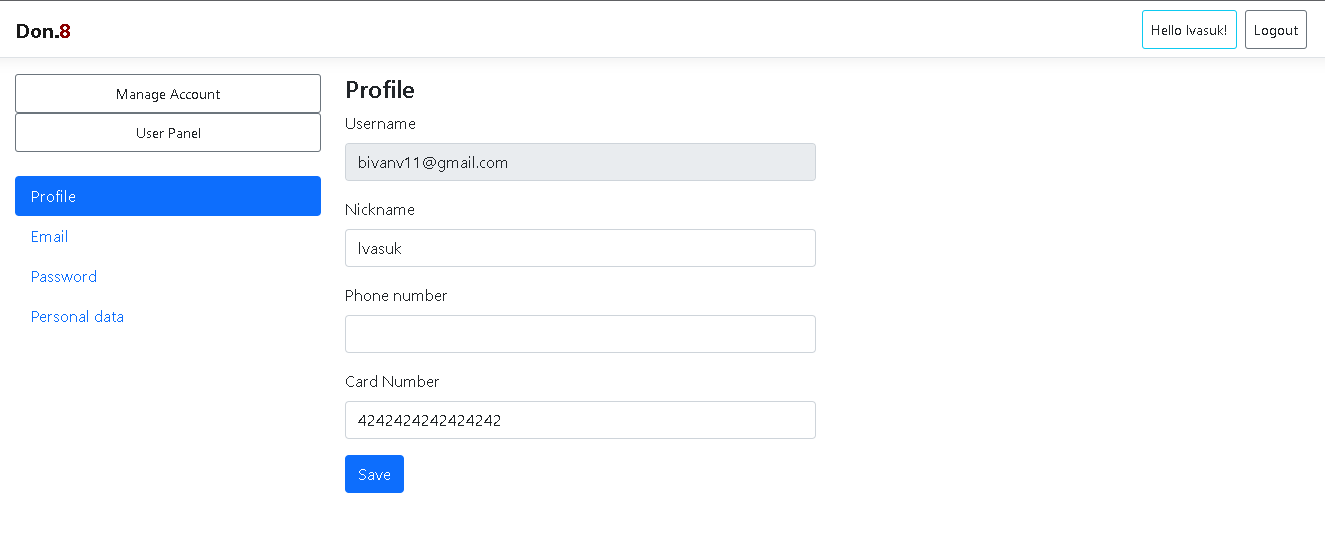


Рис. 4.7. Форма для редагування особистих даних користувача

На рисунку 4.8. зображена сторінка профілю контентмейкера (сторінка-візитівка) на якій знаходиться зображення профілю, іконки-посилання на соцмережі, коротка біографія та форма надсилання пожертви. Після заповнення потрібних полів, користувач натискає на кнопку “Donate” (пожертвувати), та його перекидує на сторінку оплати, де він оплачує суму пожертву.

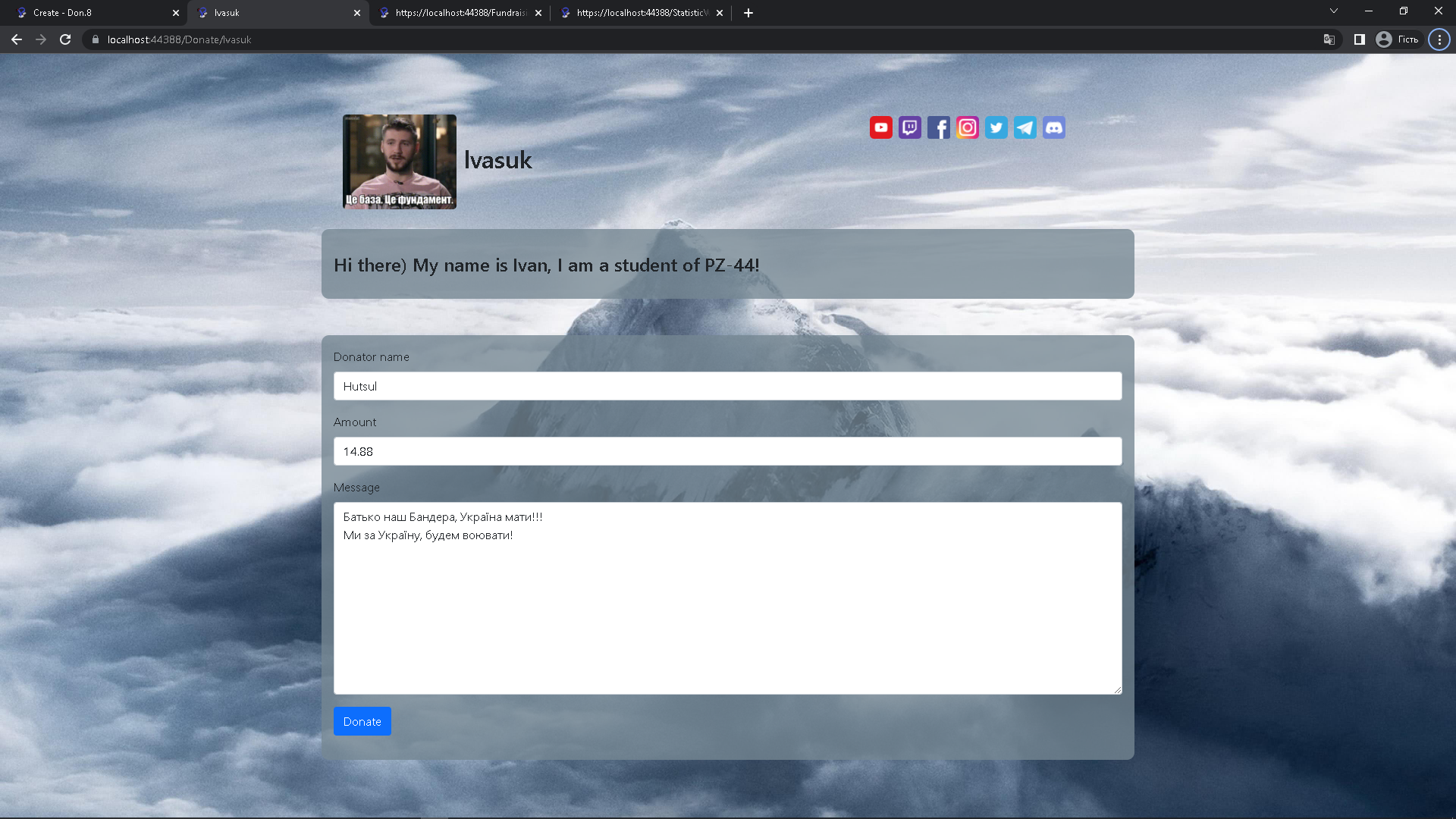


Рис. 4.8. Сторінка профілю контентмейкера та форма для надсилання донату

Після успішної оплати пожертви, система надсилає сповіщення усім віджетам отримувача, а також змінює дані у базі даних, що потім відображається на інших віджетах. Обробка вхідних сповіщень у реальному часі відбувається за допомогою SIgnalR. Нище наведений код, у якому відбувається отримання сповіщень.

var hubConnection = new signalR.HubConnectionBuilder()

.withUrl("/donates?userid="+userID)

.withAutomaticReconnect()

.build();

// Get message from hub

hubConnection.on('ReceiveMessage', function (donate) {

console.log("Mesg Received")

displayDonate(donate);

});

hubConnection.start();

Для реалізації синтезу мовлення (озвучення повідомлення з донату) було використано два підходи: вбудований window.speechSynthesis та за допомогою стороннього сервісу responsivevoice. Це було зроблено через те, що у вбудованому вузький набір мов та немає найкращої та найчарівнішої на нашій планеті – української мови. Сторонній сервіс дає значно ширший набір мов, але він не виділяється надійністю та має обмежений доступ, тому, у разі відмови першого, використовуватиметься вбудований. Так як одночасно може відтворюватися звук однією мовою, клієнтська частина сама визначає мову сповіщення. Нище наведений код цієї можливості.

function voiceMessage(message) {

let utter = new SpeechSynthesisUtterance();

if (/^[a-zA-Z]+$/.test(message))

{

utter.lang = 'en-US';

}

else

{

utter.lang = 'ru-RU';

}

utter.text = message;

utter.volume = 1;

window.speechSynthesis.speak(utter);

}

**4.3. Тестування програмного забезпечення**

**4.3.1. Вступ**

Процес тестування системи є вкрай важливим етапом життєвого циклу ПЗ. В даному підрозділі описані результати тестування платформи для фінансової підтримки контентмейкерів. Проводилось функціональне, модульне (unit test) та тестування інтерфейсу. Ці види тестування дають змогу упевнитися у коректній реалізації функціонування програмної системи на рівні коду та інтерфейсу користувача.

**4.3.2. Створення тестових випадків**

Так як у системі присутні два класи користувачів, то тестові випадки відповідно поділяються на дві групи – для неавторизованого користувача та контентмейкера. Проте, можливості контентмейкера повністю включають можливості неавторизованого користувача.

**4.3.2.1. Модульне тестування**

Так як усі основні можливості реалізована на серверній частині, а на клієнтській є тільки декілька простих та допоміжних функцій, які легко перевіряються вручну, то модульне тестування доцільне для перевірки коду серверної частини. Це тестування проводилося для перевірки роботи окремих модулів та методів які містяться в системі. Основною складовою серверної частини є контролер, тому основна увага зосереджувалася на їхньому тестуванні.

Для тестування було використано фреймворк NUnit, яка дозволяє запускати створені тести для перевірки функціональності. Для тестування потрібна база даних, тому, для симуляціїї її роботи я використав бібліотеку Microsoft.EntityFrameworkCore.InMemory. Створені юніт тести містяться у окремому проекті але є частиною проектного рішення, та мають доступ до даних основної системи.

**4.3.2.2. Функціональне тестування**

Для виконання функціонального тестування було створено тридцять п’ять тестових випадків, які включають проведення позитивного (коректні вхідні дані), та негативного тестування (помилковими вхідні дані). Для створення тестових випадків використовувалась специфікація вимог. Дане тестування проводилося у ручному режимі. Під час проведення функціонального тестування паралельно виконувалося тестування інтерфейсу користувача, так деякі варіанти використання пересікаються та мають однаковий алгоритм перевірки. Це дозволяє скороти час на тестування. Дане тестування також частково відбувалось власне під час реалізації проекту.

Таблиця 4.1

**Варіанти використання для функціонального тестування**

|  |  |  |
| --- | --- | --- |
| **Варіанти використання** | **Тестові випадки** | **Тестові дані** |
| Створення віджета «Сповіщення» | 4 | 10 |
| Створення віджета «Збір коштів» | 2 | 6 |
| Створення віджета «Статистика» | 2 | 8 |
| Редагування віджетів | 10 | 24 |
| Редагування сторінки профілю | 5 | 8 |
| Робота дашборду (статистики) | 4 | 13 |
| Авторизація користувача | 2 | 8 |
| Робота віджета «Сповіщення» | 1 | 10 |
| Робота віджета «Збір коштів» | 1 | 6 |
| Робота віджета «Статистика» | 1 | 12 |
| Надсилання пожертви | 3 | 10 |
| **Разом** | **35** | **115** |

**4.3.2.3. Тестування інтерфейсу**

Цей тип тестування проводиться для перевірки навігації між сторінками, коректного відображення роботи кожного з віджетів, дотримання стилістики та оформлення, коректної роботи форм створення, редагування та видалення. Сформовано набір тестових випадків. (табл. 4.2.)

Таблиця 4.2

**Варіанти використання для тестування інтерфейсу**

|  |  |
| --- | --- |
| Варіант використання | Тестові дані |
| Перевірка роботи віджета «Сповіщення» | 10 |
| Перевірка роботи віджета «Збір коштів» | 6 |
| Перевірка роботи віджета «Статистика» | 12 |
| Перевірка форми створення/редагування віджета «Сповіщення» | 8 |
| Перевірка форми створення/редагування віджета «Збір коштів» | 7 |
| Перевірка форми створення/редагування віджета «Статистика» | 10 |
| Перевірка вікна для реєстрації | 6 |
| Перевірка дашборду | 10 |
| Перевірка розділу «Керування акаунтом» | 25 |
| Загалом | 94 |

**4.3.3. Модульне тестування**

Після проведення тестування, усі тести були виконані успішно. В такому разі модульне тестування можна вважати успішним.

Результати модульного тестування зображені в додатку Д.

**4.3.4. Функціональне тестування**

У ручному режимі було пройдено усі тестові сценарії. Функціональне тестування вважається успішним, тому що усі тестові випадки пройдено успішно. В додатку Г зображені варіанти використання.

**4.3.5. Тестування інтерфейсу**

На цьому етапі були виявлені проблеми з навігацією по панелі користувача, зокрема під час переходу між розділами, на бічні панелі підсвічувався активним завжди перший. Даний дефект вдалося виправити. Також на сторінці відображення віджету «Збір коштів», після збору кінцевої суми, шкала прогресу показувала хибні дані. Під час тестування роботи віджета «Сповіщення» була виявлена проблема з невирівняним текстом сповіщення. Футер у нижній частині екрану мав абсолютну позицію, та у деяких випадках накладався поверх іншої інформації

**4.3.6. Критерії успішності виконання тестування**

Під час проведення тестування програмного продукту потрібно було забезпечити виконання наступних умови:

* Усі функції, алгоритми, сервіси, моделі та програми працюють правильно.
* Усі вимоги зазначені у специфікації вимог, реалізовані у системі та покриті тестами.
* Проведено тестування інтерфейсу користувача та перевірено його відповідність усім вимогам.

За результатами тестування, описаними у попередніх пунктах, можна зробити висновок, що тестування було успішно.

**4.4. Висновок до розділу 4**

В цьому розділі було детально описано реалізацію клієнтської та серверної частин системи. Було описано особливість роботи та взаємодію з продуктом. Були продемонстровані результати реалізації наявних в системі ключових модулів. Для впевненості в коректності роботи системи, було проведено три типи тестування: модульне, функціональне та тестування інтерфейсу. Усі тестові випадки було пройдено успішно. В додатку Е наведено створену інструкцію користувача.

**РОЗДІЛ 5** **ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОБОТИ**

**5.1. Економічна характеристика проектного рішення**

Програмна система для фінансової підтримки контентмейкерів, яка є частиною моєї бакалаврської кваліфікаційної роботи, повинна дозволити будь-яким бажаючим користувачам, які хочуть фінансово підтримати певну особу, організацію чи спілку своїми власними коштами та надіслати власне повідомлення. Натомість отримувачам пожертв має дати змогу у прямому ефірі виводити сповіщення про надходження пожертв, статистики пожертв та прогресу збору коштів. Відображення пожертви виглядає як поява анімованої картинки, імені жертводавця, суми пожертви та повідомлення, яке можу бути озвучене синтезом мови. Під час надходження повідомлення також відтворюється звук сповіщення.

Основний функціонал програмного продукту – надання користувачам можливості фінансово підтримати фізичну чи юридичну особу та надіслати їй певне повідомлення. Цей функціонал буде доступний усім користувачам без авторизації, проте потребуватиме мінімального розміру пожертви. Розмір мінімальної пожертви встановлюється контентмейкером.

Візуалізація вхідних пожертв дозволяє засвітити своє ім’я на широку аудиторію, та донести контентмейкеру своє повідомлення, що у свою чергу дає додаткове заохочення для користувачів надсилати нові донати.

Візуалізація прогресу збору коштів у режимі реального часу демонструє вклад аудиторії, а візуалізація статистики, наприклад «ТОП донатерів», може надовго продемонструвати псевдонім найбільш щедрого користувача на велику аудиторію. Інколи декілька таких користувачів вступають у гонку за місце на вершині списку

За надання своїх можливостей, сервіс може стягувати певний відсоток, тобто комісію, з кожної пожертви. Для гнучкості, цей відсоток варіюється від загальної суми пожертв у місяць, тобто, чим більша сума пожертв – тим менша комісія.

**5.2. Інформаційне забезпечення та формування гіпотези щодо потреби розроблення проектного рішення (програмного продукту)**

На ринку існує багато іноземних сервісів, проте, так як сервіси мають фінансову складову, то вони все ще не дуже коректно працюють у Україні або зовсім недоступні. Вони не мають підтримки користувачів з України погано взаємодіять з банківською системою України. Існує тільки декілька вітчизняних сервісів, проте вони мають дуже вузький функціонал та дуже високу комісію, що не робить їх привабливими для користувачів.

Кожен схожий сервіс має певний базові характеристики:

* можливість відображення повідомлень;
* можливість відображення статистики пожертв;
* можливість відображення прогресу збору коштів.

Для аналізу аналогічного продукту можна розглянути два сервіси: Diaka, Donatello. Diaka – дуже простий сервіс та має стандартний набір можливостей подібних сервісів. Комісія зі зборів залежить від суми (чим більша сума, тим менша комісія). Donatello – є майже аналогічним, але дозволяє гнучкіше конфігурувати віджети. За даними самого сервісу Donatello, за квітень 2022 року кількість зареєстрованих користувачів становить 6176. Сервіс використовує підсистему Foundy, тому комісія є досить великою, яка включає комісію самого сервісу та Foundy (3,75% + 5%).

Основними користувачами (споживачами) даних сервісів є блогери чи контентмейкери які проводять прямі трансляції та їхня аудиторія. Зараз, під час активної фази війни, денацифікації та демілітеризації рашистських військ до споживачів даних сервісів почали активно долучатися волонтерські організації та публічні люди. На своїх прямих трансляціях вони збирають кошти на благодійні організації, на потреби ЗСУ, гуманітарну допомогу тощо.

Так як подібні сервіси використовують сторонню систему оплати P2P, то вони керуються законодавством цього фінансового сектору. Звісно, так як сервіс бере комісію з донатів, то він має бути унормований законами про прибутки та для запуску повинен пройти консультацію у юристів.

Зробивши аналіз вимог на ринку, та функціональні можливості програмного продукту, можна дійти висновку, що дана система буде активно використовуватись медійними людьми на території України які проводять прямі трансляції.

Окрім сервісів для надсилання прямих пожертв, які спеціалізуються саме на авторів прямих трансляцій, існують ще сервіси, які дозволяють підтримати творців шляхом щомісячного членського внеску. За спеціальну плату, користувачі отримають ексклюзивний доступ до певного контенту. Плата може бути різною, та встановлюється автором індивідуально. Привілеї користувачів часто залежать від виду їхньої підписки, зазвичай чим дорожча підписка – тим більше привілеїв.

Яскравим представником такого сервісу є платформа Patreon. Вона набула шаленої популярності, так як допомагає творцям з невеликою аудиторією отримувати прибуток. Вона працює за принципом платної підписки, та надає ексклюзивні матеріали своїм підписникам.

**5.3. Оцінювання та аналізування факторів зовнішнього та внутрішнього середовищ**

В наступному розділі було проаналізовано та оцінено фактори внутрішнього і зовнішнього середовищ групою експертів. Для оцінки зовнішніх факторів було використано шкалу [-5; 5], де -5 — це негативний вплив на організацію, 5 — це позитивний вплив, а 0 — нейтральний вплив. Фактори внутрішнього середовища оцінюються за шкалою [0;5], де 0 демонструє низький розвиток, відсутність чи катастрофічний стан фактору внутрішнього середовища, оцінка 5 демонструє високий рівень розвитку даного фактору. Рівень вагомості для кожного фактору визначається за допомогою коефіцієнтів, сума всіх коефіцієнтів дорівнює одиниці. Зважений рівень впливу факторів дорівнює добутку впливу фактору у балах та рівня вагомості. Результати експертних оцінок впливу факторів зовнішнього середовища наведено у табл. 5.1.

Таблиця 5.1

**Результати експертного оцінювання впливу факторів зовнішнього та внутрішнього середовищ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Фактори** | **Середня експертна оцінка, бали** | **Середня вагомість факторів** | | **Зважений рівень впливу, бали** | |
| *Фактори зовнішнього середовища* | | | | | |
| Споживачі | 2 | 0,11 | | 0,22 | |
| Постачальники | 3 | 0,1 | | 0,3 | |
| Конкуренти | 1 | | 0,1 | | 0,1 |
| Державні органи влади | 0 | | 0,05 | | 0 |
| Інфраструктура | 2 | | 0,06 | | 0,12 |
| Законодавчі акти | -1 | | 0,1 | | -0,1 |
| Профспілки, партії та інші громадські організації | 2 | | 0,05 | | 0,1 |
| Система економічних відносин в державі | 1 | | 0,06 | | 0,06 |
| Організації-сусіди | 2 | | 0,01 | | 0,02 |
| Міжнародні події | -2 | | 0,01 | | -0,02 |
| Міжнародне оточення | 4 | | 0,03 | | 0,12 |
| Науково-технічний прогрес | 5 | | 0,07 | | 0,35 |
| Політичні обставини | 3 | | 0,06 | | 0,18 |
| Соціально-культурні обставини | 4 | | 0,05 | | 0,20 |
| Рівень техніки та технологій | 5 | | 0,04 | | 0,20 |
| Особливості міжнародних економічних відносин | 0 | | 0,02 | | 0 |
| Стан економіки | -3 | | 0,08 | | -0,24 |
| Загальна сума |  | | 1 | | 1,61 |

Продовження таблиці 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| *Фактори внутрішнього середовища* | | | |
| Цілі | 5 | 0,11 | 0,55 |
| Структура | 3 | 0,16 | 0,48 |
| Завдання | 3 | 0,07 | 0,21 |
| Технологія | 4 | 0,2 | 0,8 |
| Працівники | 3 | 0,21 | 0,63 |
| Ресурси | 2 | 0,25 | 0,50 |
| Загальна сума |  | 1 | 3,17 |

Хоч під час військового стану розглядати вплив зовнішніх факторів дуже тяжко, бо більшість з них зараз приносять негативний вплив, проте, зважений рівень впливу зовнішнього середовища є позитивним, його значення становить 1,61. Найбільш позитивно на продукт впливають наступні фактори; науково-технічний прогрес, постачальники та рівень техніки та технологій. Найбільшим негативним фактором є економічна система під час війни зазнає шалених ударів та збитків. Політична ситуація у сьогоденні реалії також дає негативний ефект, проте, це значно збільшило к-кість волонтерів. У внутрішньому середовищі сумарний зважений рівень впливу становить 3,17.

Вплив зовнішнього та внутрішнього середовища є позитивним, і тому вихід на ринок є можливим, проте слід зважити на зовнішні події, особливо війну на території нашої країни, проте, очевидно, що у цій війні переможе Україна, бо кожен із нас вірить та допомагає ЗСУ, окупантам ніколи не вдасться знищити наш бойовий дух і їх залишається тільки «удобряти» наш чорнозем, тому після нашої перемоги ризики знизяться.

**5.4. Формування стратегічних альтернатив**

Основою стратегічного планування є вибір стратегії, а будування стратегічних альтернатив відіграє важливу роль при становленні цілей організації, а також допомагає обрати оптимальний варіант, враховуючи її можливості, переваги та недоліки. Зазвичай є декілька можливих варіантів стратегій, серед яких організаціям доводиться обирати. Перша група стратегічних альтернатив зображена на рисунку 5.1., критеріями поділу якої є існуючий та новий продукт (програмний), а також відповідні послуги.

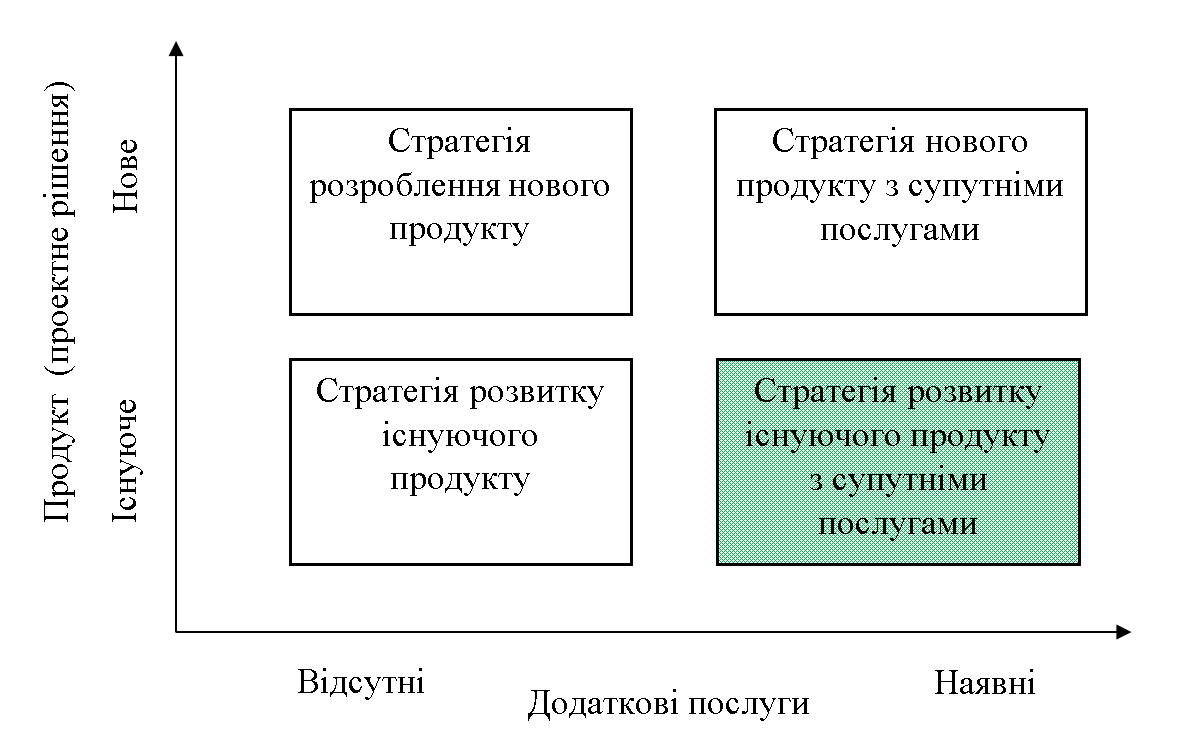


Рис. 5.1 Стратегічні альтернативи першої групи

**Стратегія розроблення нового продукту (програмної системи)** – це створенням нового програмного забезпечення, яке дає змогу вирішити новоутворені потреби людини, економіки, певного колективу або суспільства загалом тощо.

**Стратегія розвитку існуючого продукту (програмного засобу)** розуміє собою внесення змін до вже існуйчого програмного забезпечення а також його характеристик.

**Стратегія розвитку існуючого продукту (програмного засобу) з супутніми послугами** означає представлення на ринок модифікованого програмного забезпечення із додатковими послугами (встановлення, коригування, супроводження, адаптування до специфіки конкретного підприємства).

**Стратегія нового продукту (програмного засобу) з супутніми послугами** означає розроблення нового програмного забезпечення та пропонування додаткових послуг при його експлуатації.

Друга група стратегічних альтернатив зображена на рисунку 5.2., критеріями поділу якої є існуючий ринок та продукт (програмний засіб), новий ринок та продукт (програмний засіб).

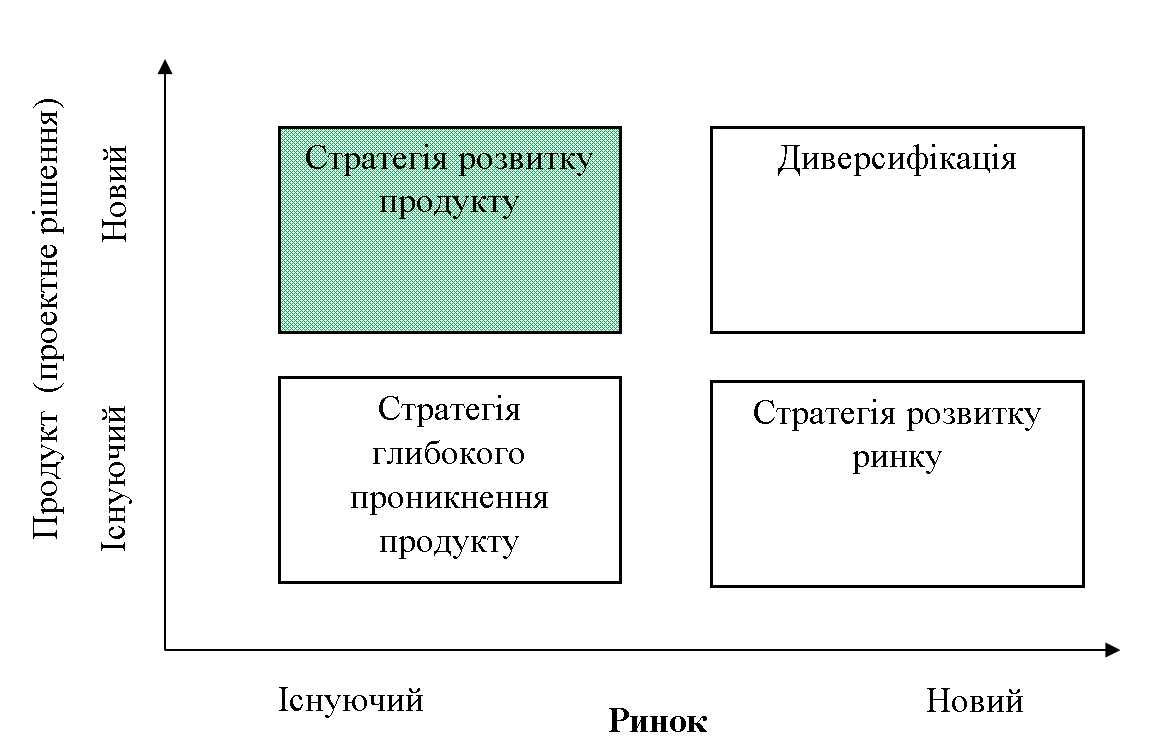


Рис. 5.2 Стратегічні альтернативи другої групи

**Глибоке проникнення на ринок** означає використання існуючого продукту для збільшення частки на існуючому ринку. За умови достатніх запасів ресурсів та великої потужності ця стратегія є найменш ризикованою, проте, при активному зростанні на ринку збільшується конкуренція. Дана стратегія може бути успішною за умови, якщо конкурентні продукти будуть уступати у своїх потужностях та ресурсах, або якщо розвиток ринку буде дуже стрімкий.

**Стратегія розвитку ринку** означає використання існуючого продукту або його змінених версій для виходу на новий ринковий сегмент. Ризик цієї стратегії є дуже високий та пов’язаний з неперевіреністю успішності даного типу продукту на новому ринку.

**Стратегія розвитку продукту** означає створення нового продукту для існуючого ринкового сегменту. Ризик знову ж таки є доволі високим, бо ця стратегія вимагає створення нового продукту для вже сформованого сегменту користувачів. Однак, цю стратегію варто використовувати при зменшенні обсягів існуючого продукту на етапі зрілості.

**Стратегія диверсифікації** означає вихід на нові сфери бізнесу.

Зробивши аналіз першої групи стратегічних альтернатив, я обрав стратегію розвитку існуючого продукту з супутніми послугами, адже схожих систем для фінансової підтримки контентмейкерів є досить багато на світовому ринку, проте на ринку України їх все ж недостатньо і вони мають досить низький розвиток.

З другої групи стратегічних альтернатив, була вибрана стратегія розвитку продукту з огляду на те, що присутні на ринку продукти аналоги мають досить вузький спектр можливостей та є досить схожими один на одного, тому розвиток самого продукту, шляхом додавання до нього нового функціоналу, дасть явні переваги над і так не дуже розвинутими конкурентами.

**5.5. Бюджетування**

Бюджетування являє собою комплексно-обґрунтовану систему розрахунку витрат, пов’язаних з реалізацією продукту, тобто програмної системи, та допомагає створити заходи для підвищення рентабельності виробництва. Під час цього етапу потрібно визначити собівартість продукту, забезпечити виробничу та комерційну діяльність підприємства.

Бюджет матеріальних витрат зображено в таблиці 5.2.

Таблиця 5.2

**Бюджет витрат матеріалів та комплектуючих виробів**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва матеріалів та комплектуючих | Марка, тип, модель | Фактична кількість, шт. | Ціна за одиницю, грн. | Разом, грн. |
| 1 | 2 | 3 | 4 | 5 |
| SSD диск | Kingston NV1 250GB | 1 | 1200,0 | 1200,0 |
| Програмне забезпечення | Visual Studio 2022 | 1 | 1850,0 | 1850,0 |
| Разом: | | | | 3050,0 |

**Бюджет витрат на купівлю основних засобів**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Витрати на куповані засоби** | **Марка, тип, модель** | **Фактична кількість, шт.** | **Ціна за одиницю, грн.** | **Разом, грн** | **Термін експлуатації, років** | **Місячна сума амортизації, грн.** |
| Ноутбук | Dell 5700 | 1 | 15000 | 15000 | 5 | 250 |

Систему будуть розробляти троє осіб. Бюджет витрат на оплату праці наведений у таблиці 5.3*.*

Таблиця 5.3

**Бюджет витрат на оплату праці виробничих працівників**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Посада,  спеціальність | Кількість працівників, осіб | Час роботи, дні | Денна заробітна плата працівників, грн. | Сума витрат на оплату праці, грн. |
| *Основна заробітна плата* | | | | |
| Front End Developer (Розробник клієнтської частини) | 1 | 28 | 450,0 | 12600,0 |
| Back End Developer (Розробник серверної частини) | 1 | 42 | 520,0 | 21840,0 |
| Дизайнер та мануальний тестувальник | 1 | 14 | 380,0 | 5320,0 |
| Разом: | | | | 39760,0 |

Бюджет обов’язкових відрахувань та податків наведений в таблиці 5.4.

Визначаємо суму військового збору для кожної категорії працівників:

= 12600,0\* 0,015 = 189,0 (грн)

= 21840,0 \* 0,015 = 327,6 (грн)

= 5320,0 \* 0,015 = 79,8 (грн)

Визначаємо суму податку з доходів фізичних осіб:

= 12600,0 \* 0,18 = 2 268,0 (грн)

= 21840,0 \* 0,18 = 3 931,2 (грн)

= 5320,0 \* 0,18 = 957,6 (грн)

Таблиця 5.4

**Бюджет обов’язкових відрахувань та податків**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Посада,  спеціальність | Сума  основної заробітної плати | | Сума додаткової заробітної плати | Разом витрат на оплату праці | | | Сума податку з доходів фізичних осіб, грн. | Сума військового збору, грн.  (1,5%) | |
| Розробник клієнтської частини | 12600,0 | - | | | 12600,0 | 2 268,0 | | | 189,0 |
| Розробник серверної частини | 21840,0 | - | | | 21840,0 | 3 931,2 | | | 327,6 |
| Дизайнер та мануальний Тестувальник | 5320,0 | - | | | 3600,0 | 957,0 | | | 79,8 |
| Разом: | 39 760,0 | - | | | 39 760,0 | 7 156,2 | | | 596,4 |

Бюджет загальновиробничих витрат зображено в таблиці 5.5.

Таблиця 5.5

**Бюджет загальновиробничих витрат**

|  |  |
| --- | --- |
| Статті витрат | Сума, грн. |
| *Змінні загальновиробничі витрати, у т.ч.:* | |
| - заробітна плата допоміжного персоналу; | 4200,0 |
| - нарахування на заробітну плату (ЄСВ 22%) | 924,0 |
| - витрати на МШП; | 300,0 |
| - витрати на електроенергію; | 700,0 |
| - витрати на ремонт; | 1000,0 |
| - інші змінні витрати; | 600,0 |
| Разом змінних витрат: | 7724,0 |
| *Постійні загальновиробничі витрати, у т.ч.:* | |
| - комунальні послуги; | 1000,0 |
| - витрати на оренду; | 4000,0 |
| - інші постійні витрати; | 1800,0 |
| Разом постійних витрат: | 6800,0 |
| *Разом загальновиробничих витрат:* | 14 524 |

Бюджет адміністративних витрат та витрат на збут наведено в таблиці 5.6.

Таблиця 5.6

**Бюджет адміністративних витрат та витрат на збут**

|  |  |
| --- | --- |
| Статті витрат | Сума, грн. |
| *Адміністративні витрати, у т.ч.:* | |
| - заробітна плата адміністративного персоналу; | 3500,0 |
| - нарахування на заробітну плату (ЄСВ 22%) | 770,0 |
| - витрати на МШП; | 300,0 |
| - витрати на відрядження; | 0,0 |
| - облаштування адміністративного обладнання; | 250,0 |

Продовження табл. 5.6

|  |  |
| --- | --- |
| - інші адміністративні витрати; | 300,0 |
| Разом адміністративних витрат: | 5 120 |
| *Витрати на збут, у т.ч.:* | |
| - заробітна плата менеджерів зі збуту; | 4000,0 |
| - нарахування на заробітну плату (ЄСВ 22%) | 880,0 |
| - витрати на відрядження; | 0,0 |
| - витрати на гарантійне обслуговування; | 300,0 |
| - витрати на налагодження і експлуатацію; | 200,0 |
| - витрати на рекламу; | 1000,0 |
| - інші витрати на збут; | 320,0 |
| Разом витрат на збут: | 6700,0 |

Виробнича собівартість формується на основі витрат на виробництво продукції. Цей показник є одним з ключових економічних показників господарської діяльності підприємства, у якому дістають відображення зростання продуктивності праці, економія ресурсів, технічний прогрес. Зведений кошторис витрат на розробку проектного рішення наведено в таблиці 5.7.

Таблиця 5.7

**Зведений кошторис витрат на розробку проектного рішення (продукту)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Статті витрат | Одиниці виміру | Фактична кількість, шт. | Ціна одиниці, грн. | Разом, грн. |
| Сировина і матеріали | шт. | - | - | - |
| Купівельні напівфабрикати та комплектуючі вироби | грн. | - | 3050,0 | 3050,0 |
| Зворотні відходи | грн. | - | - | - |
| Паливо та електроенергія на технологічні цілі | грн | - | 200 | 200 |

Продовження табл. 5.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Амортизаційні відрахування основних фондів | грн. | - | - | 250,00 |
| Основна заробітна плата | грн. | - | - | 39760,0 |
| Додаткова заробітна плата | грн. | - | - | 0,0 |
| Відрахування на соціальне страхування (ЄСВ 22%) | грн. | - | - | 8747,2 |
| Витрати на утримання й експлуатацію устаткування | грн. | - | 300,0 | 300,0 |
| Загальновиробничі витрати, у т.ч.: | | | | |
| - змінні; | грн. | - | - | 7724,0 |
| - постійні; | грн. | - | - | 6800,0 |
| ***Разом виробничих витрат:*** | **грн.** | **-** | **-** | **66831,2** |
| Адміністративні витрати | грн. | - | - | 5 120 |
| Витрати на збут | грн. | - | - | 6700,0 |
| Інші операційні витрати | грн. | - | - | - |
| ***Разом виробничих і операційних витрат:*** | **грн.** | **-** | **-** | **78 651,2** |

Ціну продукту визначаємо за формулою:

Ц = СБ + СБ \* Р, де Ц – ціна програмного продукту, грн. СБ – собівартість програмного продукту, грн.Р – рентабельність, 33 %.

Ц = **66831,2**\* 0,33 + **66831,2**‬‬ = **88885,5 грн.**

Ц з ПДВ = 88885,5 \* 1,2 = **106662,6 грн**

Дохід від реалізації продукції = Ц \* к-сть реалізованої продукції = 106662,6\* 1 = **106662,6грн.** ПДВ = 106662,6/6 = **17777,1 грн.**

Чистий дохід від реалізації продукції = 101 076,6 – 17777,1 = 88885,5 грн.

Валовий прибуток = Чистий дохід від реалізації продукції – Собівартість реалізованої продукції = 88885,5 – 66 831,2‬‬ = 22054,3 грн.

Фінансовий результат від операційної діяльності = Валовий прибуток - адміністративні витрати - витрати на збут - інші операційні витрати = 22054,3 – 5120,0 – 6700,0 = 10234,3 грн.

Податок на прибуток = Фінансовий результат від операційної діяльності \* 18% = 10234,3 \* 0,18 = 1842,17грн.

Чистий прибуток (збиток) = Фінансовий результат від операційної діяльності - Податок на прибуток = 10234,3 – 1842,17= 8392,12 грн.

Бюджет фінансових результатів наведено в таблиці 5.8.

Таблиця 5.8

**Бюджет фінансових результатів**

|  |  |
| --- | --- |
| Показники | Сума, грн. |
| Дохід від реалізації продукції | 106662,6 |
| Податок на додану вартість (20 %) | 17777,1 |
| Чистий дохід від реалізації продукції | 88885,5 |
| Собівартість реалізованої продукції | 66831,2 |
| Валовий прибуток | 22054,3 |
| Операційні витрати: | |
| - адміністративні витрати | 5 120 |
| - витрати на збут | 6700,0 |
| - інші операційні витрати | 0,0 |
| Фінансовий результат від операційної діяльності | 10234,3 |
| Податок на прибуток (18 %) | 1842,17 |
| Чистий прибуток (збиток) | 8392,12 |

**5.6. Вибір стратегії**

Отримані результати аналізу дозволяють зробити висновок, що рівень впливу зовнішнього середовища є позитивним, він становить 1,61. Найбільш позитивно на продукт впливають такі фактори як науково-технічний прогрес, постачальники та рівень техніки та технологій. Через надзвичайно складну ситуацію в Україні негативних факторів значно побільшало і найгіршими є стан економіки та міжнародні події. Найбільш розвинутими серед внутрішніх факторів є цілі та технології. У внутрішньому середовищі, сумарний зважений рівень впливу становить 3,17. Проаналізувавши фактори внутрішнього та зовнішнього середовищ, можна дійти висновку, що суттєвих ризиків під час розробки та впровадження програмного продукту немає.

З першої групи було обрано стратегію розвитку існуючого продукту з супутніми послугами, тому що на ринку вже є схожі системи, але вони не відрізняються різноманітністю. Після внесення додаткових змін у існуючу функціональність та після додавання нової функціональності, продукт зможе виділитися на ринку та перетягнути на себе вже існуючих споживачів та захопити нових.

Для даного програмного продукту конкуренція присутня, але доступні в Україні сервіси не покривають необхідний функціонал для системи даного типу. Закордонні конкуренти через їхню банківську систему та їхні закони є не зовсім зручними для використання. Здійснивши аналіз вимоги на ринку та функціональних можливостей програмного продукту, зроблено висновок, що дана система зможе активно використовуватись фізичними чи юридичними особами які проводять прямі трансляції, та мають певну медійну аудиторію.

**Висновки до розділу 5**

В даному розділі «Економічне обґрунтування доцільності роботи» була обґрунтована доцільність розробки існуючого продукту з супутніми послугами, розраховано всі його економічні характеристики. Здійснення розробки програмного продукту та запровадження його в дію може відбутися за період, який складає 50-60 робочих днів. Загальна сума виробничих витрат складає 66831,2 грн. Під час розробки наявні також і адміністративні витрати у розмірі 5 120,0 грн та витрати на збут у розмірі 6700,0 грн. Через виділення великої частки коштів на рекламу, витрати на збут є значними. Прибуток приноситимуть саме користувачі, тому дуже важливим є залучення нових споживачів. Через низький відсоток рентабельності та великі виробничі витрати, чистий прибуток складає 8392,12 грн.

# ВИСНОВКИ

Під час виконання даної бакалаврської кваліфікаційної роботи було проведене дослідження сучасних соціальних мереж та блогерів (контентмейкерів), які є однією з ключових атрибутів кожної з мереж. Було визначено основні способи подачі ними інформації та заробітку грошей за допомогою їхньої творчості, тобто – монетизація контенту. Основним типом контентмейкерів для дослідження було взято саме «стрімерів» - творці, які проводить прямі трансляції залучаючи певну аудиторію. У сучасних реаліях у цю роль цих «доєдналися волонтери та навіть військові. Основним предметом дослідження монетизації було обрано пожертви, або як їх називають «донати». Був здійснений аналіз ринку, на наявність продуктів, які можуть дати змогу фінансово підтримати контентмейкрів, а їм, у свою чергу, дозволити певним чином за них віддячити, продемонструвати пожертви аудиторії та взаємодіяти з жертводавцями. Після аналізу було визначено, що на світовому ринку наявно багато подібних продуктів, проте ринок України ще не сильно розвинутий.

Початковим етапом розробки стала постановка задачі, який включає у себе створення специфікації вимог, та опис майбутнього функціоналу, яким система повинна володіти. Для реалізації поставлених задач було прийнято рішення використовувати за основу платформу ASP.NET Core та додаткові бібліотеки MVC Core, Identity Core, Entity Framework Core, SignalR та Razor Pages Core. Система реалізована за принципами клієнт-серверної архітектури. В якості

Під час проектування та розробки архітектури системи було створено наступні діаграми: прецедентів, використання, розгортання, бази даних. Було описано загальну структуру складових серверної та клієнтської частин. В якості архітектурного шаблону було використано MVC.

Внаслідок етапу розробки, був написаний єдиний вихідний код для клієнтської та серверної частин. Було успішно проведено три типи тестування: модульне, функціональне та тестування інтерфейсу.

**СПИСОК ЛІТЕРАТУРИ**

1. Інформаційна ера Вікіпедія – відкрита енциклопедія [Електронний ресурс]. – Режим доступу : <https://uk.wikipedia.org/wiki/Інформаційна_ера>
2. In 2021, the Internet went for TikTok, space and beyond [Електронний ресурс] – Режим доступу:<https://blog.cloudflare.com/popular-domains-year-in-review-2021/>
3. Most popular social networks worldwide as of January 2022, ranked by number of monthly active users [Електронний ресурс]. – Режим доступу :

<https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>

1. OBS [Електронний ресурс]. – Режим доступу :<https://obsproject.com/uk>
2. Донат – що це таке в соціальних мережах, відеоіграх та на стрімах. [Електронний ресурс]. – Режим доступу : <https://termin.in.ua/donat/>
3. Patreon, Вікіпедія – відкрита енциклопедія [Електронний ресурс]. – Режим доступу : <https://en.wikipedia.org/wiki/Patreon>
4. Документація по ASP.NET [Електронний ресурс]. – Режим доступу :

<https://docs.microsoft.com/uk-ua/aspnet/core/?view=aspnetcore-6.0>

1. Документація по Entity Framework [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/uk-ua/ef/>
2. Entity Framework Tutorial [Електронний ресурс]. – Режим доступу :

<https://www.entityframeworktutorial.net/>

1. Документація то ASP.NET Razor Pages [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/razor-pages/>
2. System Requirements for ASP.NET Core [Електронний ресурс]. – Режим доступу : <https://help.syncfusion.com/aspnet-core/installation-and-upgrade/system-requirements>
3. Збірник наукових матеріалів XC Міжнародна інтернет — конференція «РОЗВИТОК НАУКИ В УМОВАХ ВОЄННОГО СТАНУ» [Електронний ресурс]. – Режим доступу :<https://el-conf.com.ua/архів-конференцій/>

# ДОДАТОК А. Діаграма прецедентів

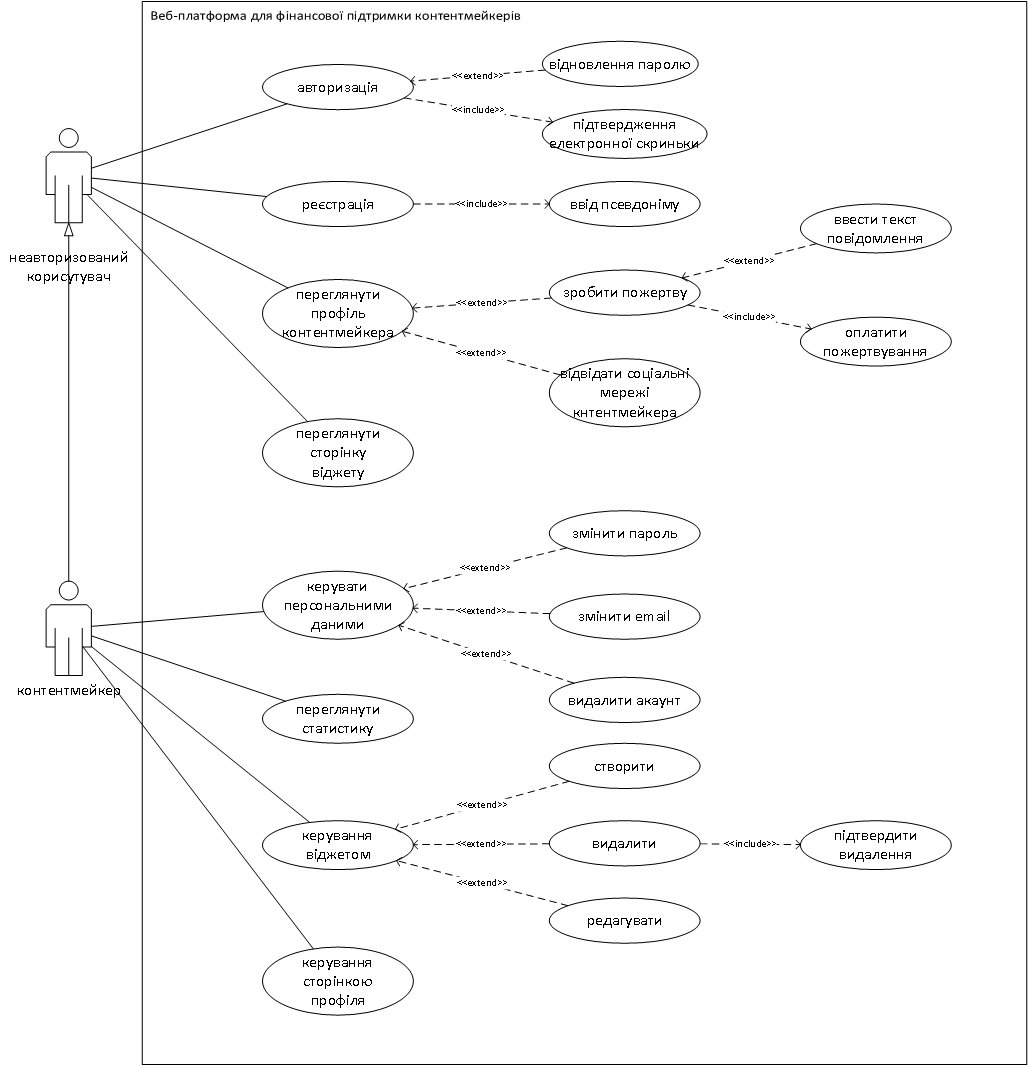


Рис. А.1. Діаграма прецедентів

# ДОДАТОК Б. Діаграма бази даних

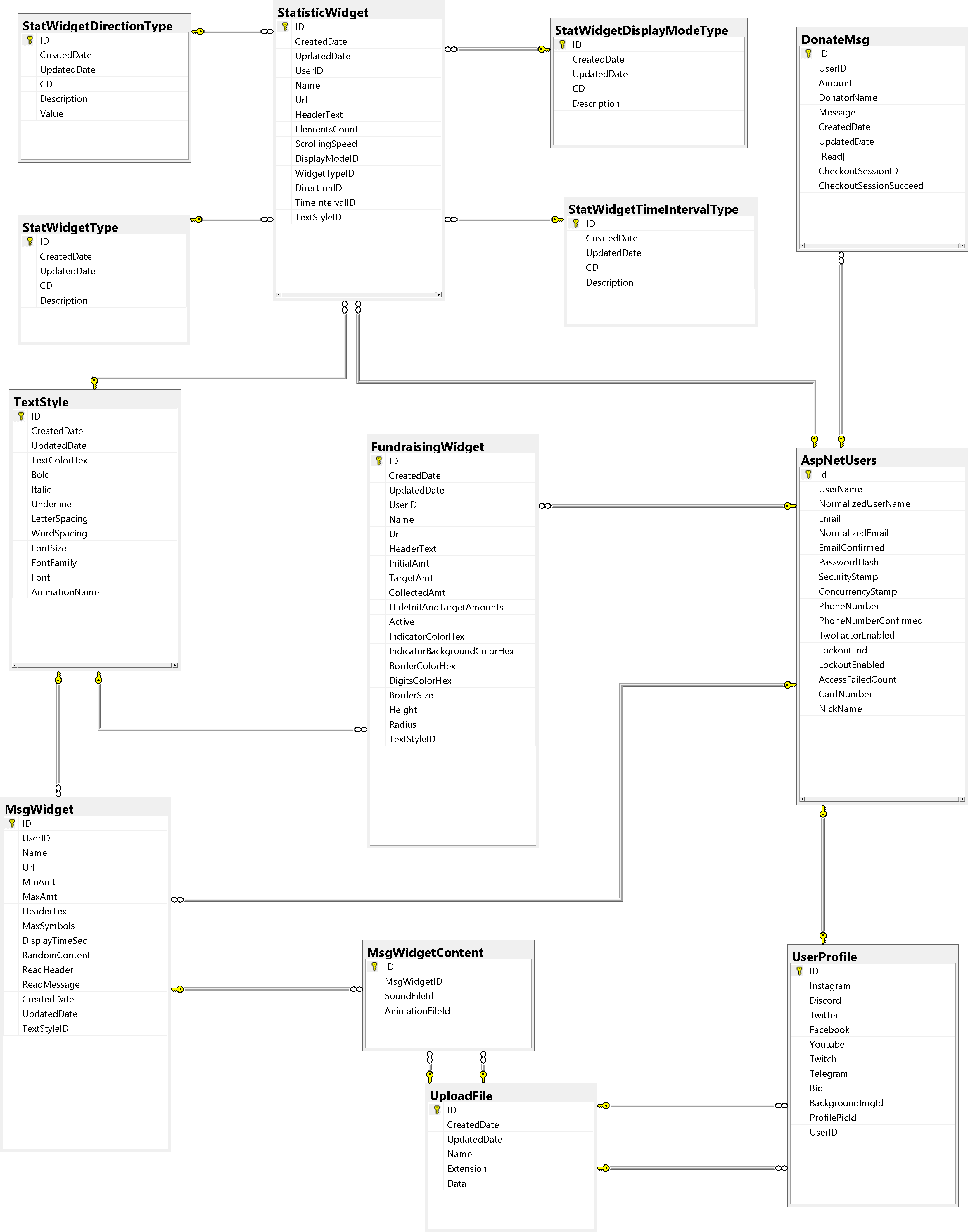


Рис. Б.1. Діаграма бази даних

# ДОДАТОК В. Діаграма послідовності

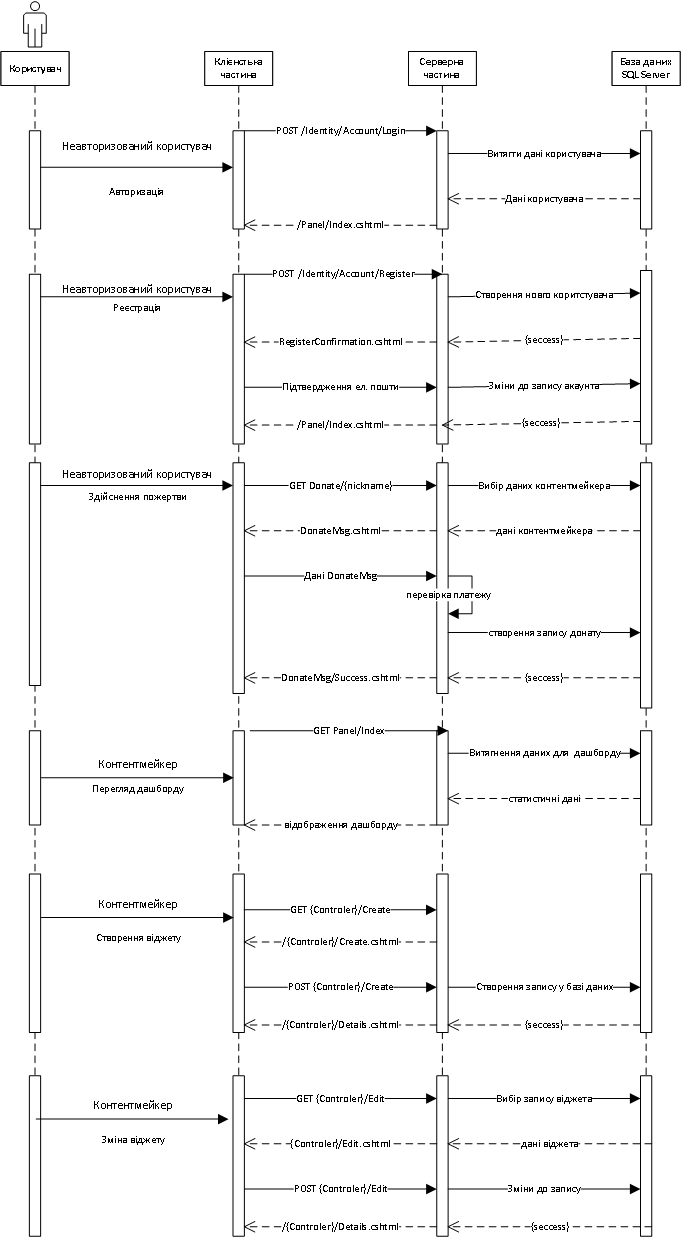


Рис. В.1. Діаграма послідонсті

# ДОДАТОК Г. Варіанти використання

Таблиця Г.1

**FT1. Авторизація. Валідні дані**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Користувач натискає на кнопку для «Login» | Відкривається форма входу. |
|  | Введення правильних логіна та паролю | Успішний вхід у систему. Відкриття домашньої сторінки. |

Таблиця Г.2

**FT2. Авторизація. Неправильні дані**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Неавторизований користувач натискає на кнопку для «Login» | Відкривається форма входу. |
|  | Введення неправильних логіна та паролю | Система виводить повідомлення про неправильно введені дані |

Таблиця Г.3

**FT3. Надсилання пожертви**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Неавторизований користувач переходить на сторінку профіль контентмейкера. | Відкривається сторінка профіль контентмейкера (сторінка-візитівка) |
|  | Заповнення полів: ім’я, сума, повідомлення. Натиснення «Donate» | Перенаправлення на сторінку оплати. |
|  | Внесення даних для опали (номер карти, email, CVV тощо) | Повідомлення про помилки відсутні |
|  | Натиснення кнопки “Pay” та підтвердження оплати | Успішна оплата. Перенаправлення на сторінку успішної оплати. |

Таблиця Г.4

**FT4. Створення віджета «Сповіщення»**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Авторизований користувач переходить у панель користувача | Відкривається панель користувача |
|  | У розділі “Widgets” користувач обирає тисне “Notification” | Система виводить список існуючих віджетів «Сповіщення» |
|  | Користувач тисне на“Create new” | Система виводить форму створення нового віджета |
|  | Користувач заповнює текстові та числові поля, обирає параметри чек-боксів | Повідомлення про помилки відсутні. Система запропоновує значення текстових полів |
|  | Користувач тисне “Text Style” | Система виводить вікно налаштування стилю тексту |
|  | Користувач змінює стиль тексту | Зміни стилю тексту відображаються у прикладі |
|  | Користувач тисне “Ok” | Вікно зміни стилю закривається |
|  | Користувач обирає файл для анімації | Вище контролера вибору з’являється попередній перегляд анімації |
|  | Користувач обирає аудіо файл | Оновляється аудіо-контролер, та з’являється можливість прослухати обраний файл |
|  | Користувач тисне «відтворити» на аудіоконтролері | Система відтворює вибраний аудіо-файл. |
|  | Користувач тисне кнопку «Create» | Система зберігає віджет та перенаправляє на список існуючих віджетів «Сповіщення», де з’являється запис нового віджету |

Таблиця Г. 5

**FT5. Відображення роботи віджета «Сповіщення»**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Авторизований користувач переходить у панель користувача | Відкривається панель користувача |
|  | У розділі “Widgets” користувач обирає тисне “Notification” | Система виводить список існуючих віджетів «Сповіщення» |
|  | Авторизований користувач у розділі віджета тисне “Display” | Система перенаправляє користувача на сторінку роботи віджета, та чекає вхідного повідомлення |
|  | Чекає повідомлення | ЗА умови надходження повідомлення, воно виводиться на екран, у супроводі звуку та агнімації |

Таблиця Г.6

**FT6. Створення віджета «Збір коштів»**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Авторизований користувач переходить у панель користувача | Відкривається панель користувача |
|  | У розділі “Widgets” користувач обирає тисне “Fundraising” | Система виводить список існуючих віджетів «Збір коштів» |
|  | Користувач тисне “Create new” | Вивід форми створення віджету |
|  | Користувач заповнює текстові, числові та булеві поля. | Система запропоновує значення полів |
|  | Користувач тисне “Text Style” | Система виводить вікно налаштування стилю тексту |
|  | Користувач змінює стиль тексту | Відображаються змін тексту |
|  | Користувач тисне “Ok” | Вікно зміни стилю закривається |

Продовження таблиці Г.6

|  |  |  |
| --- | --- | --- |
|  | За допомогою повзунків обирається розміри стилю віджету | Над повзункоми відображається значення розмірів |
|  | Користувач обирає кольори стилю віджета | Контролер вибору кольору показує вибраний колір |
|  | Користувач тисне кнопку «Create» | Система зберігає віджет та перенаправляє на список існуючих віджетів «Збір коштів» де появляється новий |

Таблиця Г.7

**FT7. Відображення роботи віджета «Збір коштів»**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Авторизований користувач переходить у панель користувача | Відкривається панель користувача |
|  | У розділі “Widgets” користувач обирає тисне “Fundraising” | Система виводить список існуючих віджетів «Збір коштів» |
|  | Користувач у розділі віджета тисне “Display” | Система перенаправляє користувача на сторінку роботи віджета та будує шкалу прогресу з початковою, кінцевою , зібраною сумою та відсотками. |

Таблиця Г.8

**FT8. Створення віджета «Статистика»**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Авторизований користувач переходить у панель користувача | Відкривається панель користувача |
|  | У розділі “Widgets” користувач обирає тисне “Statistic” | Система виводить список існуючих віджетів «Збір коштів» |
|  | Користувач тисне “Create new” | Вивід форми створення віджету |

Продовження таблиці Г.8

|  |  |  |
| --- | --- | --- |
|  | Користувач заповнює тестові поля та вибирає кількість елементів | Система запропоновує значення полів |
|  | Користувач тисне “Text Style” | Система виводить вікно налаштування стилю тексту |
|  | Користувач змінює стиль тексту | Відображаються змін тексту |
|  | Користувач тисне “Ok” | Вікно зміни стилю закривається |
|  | Користувач обирає опції з випадаючих списків | Система показує опції випадаючих списків |
|  | Користувач повзунком обирає швидкість прокручування | Над повзунком показується значення |
|  | Користувач тисне кнопку «Create» | Система зберігає віджет та перенаправляє на список існуючих віджетів «Статистика» де появляється запис про щойно створений віджет |

Таблиця Г.9

**FT9. Відображення роботи віджета «Статистика»**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Авторизований користувач переходить у панель користувача | Відкривається панель користувача |
|  | У розділі “Widgets” користувач обирає тисне “Statistic” | Система виводить список існуючих віджетів «Статистика» |
|  | Користувач у розділі віджета тисне “Display” | Система перенаправляє користувача на сторінку роботи віджета та будує шкалу прогресу, яка містить назву збору, суму зібраних коштів, суму кінцевої мети та відсоток зібраних коштів |

Таблиця Г.10

**FT10. Редагування віджетів**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Авторизований користувач переходить у панель користувача | Відкривається панель користувача |
|  | У розділі “Widgets” користувач обирає потрібний тип віджета | Система виводить список існуючих віджетів даного типу |
|  | Користувач у розділі віджета тисне “Edit” | Система відкриває форму редагування віджета |
|  | Користувач вносить зміни у поля | Система відображає зміни значень полів |
|  | Користувач тисне кнопку «Save» | Система зберігає віджет та перенаправляє на список існуючих віджетів |

Таблиця Г.11

**FT11. Редагування сторінки профілю (сторінки візитівки)**

|  |  |  |
| --- | --- | --- |
|  | Дія | Очікуваний результат |
|  | Авторизований користувач переходить у панель користувача | Відкривається панель користувача |
|  | У розділі “General” користувач обирає “Profile Page” | Відкривається форма редагування профілю |
|  | Користувач заповнює біографію | Повідомлень про помилки немає |
|  | Користувач обирає тло профілю та фото профілю | Система виводить вибрані фото |
|  | Користувач вносить зміни у поля-посилання на соціальні мережі | Система відображає зміни значень полів |
|  | Користувач тисне кнопку «Save» | Система зберігає віджет та перенаправляє на список існуючих віджетів |

# ДОДАТОК Д. Звіт про модульне тестування

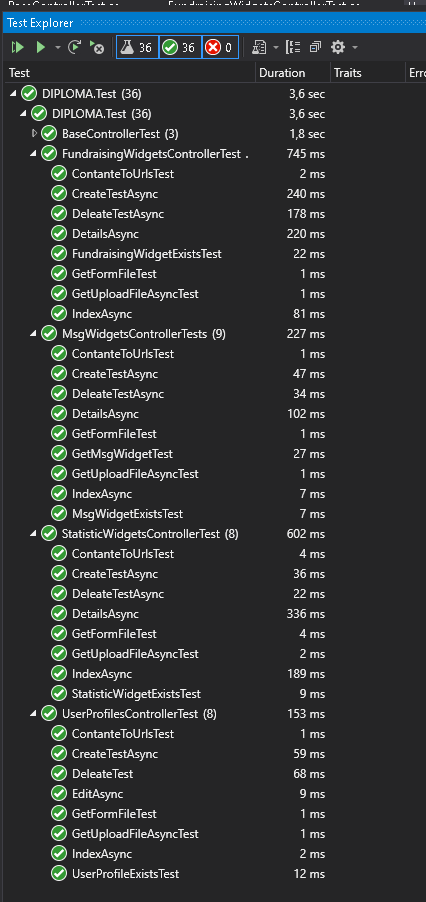


Рис. Д.1. Результати тестування серверної частини

# ДОДАТОК Е. Інструкція користувача

**Д.1. Компоненти ПЗ**

Програмна система складається з двох частин – клієнтської та серверної частин, які реалізовані за допомогою інструментів платформи ASP.NET Core. Серверна частина реалізована за допомогою MVC Core, а клієнтська за допомогою Razor Core. Середовищем в якому розроблялась система є засіб для створення і редагування програм - Visual Studio 2019 Community Edition. Для розгортання продукту Продукт можна розгорнути на машині з наступними ОС: Windows 7 SP1 і новіші; Windows 8.1; Windows 10 Version 1607 і новіші; Windows Server 2008 R2 SP1, 2012 SP1, 2012 R2, 2016 (Full Server, Server Core, or Nano Server); Mac OS X 10.11, 10.12\*; Red Hat Enterprise Linux 7; Ubuntu 14.04, 16.04, 17 . Процесор машини повинен бути x86 або x64. Мінімальний розмір оперативної пам’яті 512 Мб, а рекомендований 1Гб. Обсяг дискового простору може знадобитися до 3 Гб вільного місця. Також на сервері має бути встановлений SQL Server 2008 або новіші версії, SDK NET Core 3.1 та .NET Framework : Minimum .NET 4.5.1 upto .NET 4.7 .

**Д.2. Встановлення ПЗ**

Якщо програмна система вже розгорнута на певній машині чи хмарному сервісі, то вона не потребує встановлення. Для початку роботи з системою достатньо тільки скористатися будь-яким сучасним браузером, та перейти за посиланням на платформу.

У випадку, якщо є необхідність запустити чи розгорнути веб-застосунок на локальній машині, то спершу необхідно завантажити вихідний код та підготувати свою машину до «білду» та запуску програми (див. пункт Д.1). Найпростішим способом розгортання є використання середовища Visual Studio. Потрібно також налаштувати файл appsettings.json, у якому основним налаштуванням є стрічка підключення до бази даних. За допомогою нього, можна просто запустити процес будування програми, після чого запустити програмну систему за допомогою IIS Express. Тоді середовище все зробить саме та розгорне програму на одному з портів localhost, та відкриє посилання у встановленому за замовчуванням браузері. Також, за допомогою Visual Studio можна розгорнути систему на IIS, Azure, Local Folder тощо. Так як у проектуванні використовувався патерн MVC, то клієнтська та серверна частина є складовими одного проекту, тому вони розгортаються разом.

**Д.3. Базові функції**

При володінні сучасним веб-оглядачем, будь який не авторизований користувач отримує доступ до наступного функціоналу:

* Реєстрація;
* Авторизація;
* Відновлення паролю;
* Перегляд сторінки профілю контентмейкера;
* Можливість здійснити пожертву;
* Можливість використання роботи віджета, за умови наявності посилання на нього.

Якщо користувач авторизується до системи, він отримає роль контентмейкера та, окрім можливостей неавторизованого користувача, у нього з’явиться доступ до панелі користувача яка містить наступний функціонал:

* Керування обліковим записом – зміна особистих даних;
* Робота з віджетами – створення, редагування, видалення, тощо;
* Редагування сторінки профілю – додавання короткої біографії, посилання на соціальні мережі, зміни тла профілю та аватарки;
* Робота з дашбордом – перегляд статистики вхідних пожертв за певний період часу.

**Д.4. Аналіз помилок**

Система містить у собі підсистему обробки помилок, тому у разу настання будь-якої помилки, система виводить інформаційні повідомлення. У разі критичної помилки, для клієнта система виводить тільки описове повідомлення, з яким сам клієнт зможе звернутися до підтримки.