

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ, РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
МЕТОДІВ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ФАНДРАЙЗИНГУ ПРОЄКТІВ
RESEARCH, DEVELOPMENT AND SOFTWARE IMPLEMENTATION OF
METHODS FOR FORECASTING THE RESULTS OF PROJECT FUNDRAISING

Виконав: студент(ка) 2 курсу, групи КНТ-219м

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

Суязов С.О.

(прізвище та ініціали)

Керівник Федорончак Т.В.

(прізвище та ініціали)

Рецензент Терещенко Е.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2020 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Суязова Сергія Олександровича
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження, розробка та програмна реалізація методів прогнозування результатів фандрайзингу проєктів
Research, Development and Software Implementation of Methods for Forecasting the Results of Project Fundraising

керівник проєкту (роботи) Федорончак Тетяна Василівна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “3” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 15 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження, 2. Обґрунтування методу прогнозування результатів фандрайзингу проєктів, 3. Проєктування архітектури системи, 4. Основні рішення щодо реалізації компонентів системи, 5. Експлуатація та тестування програми, 6. Економіко-організаційна частина, 7. Охорона праці та безпека в надзвичайних ситуаціях
 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Федорончак Т.В., доцент		
6 Економіко-організаційна частина	Пожуєва Т.О., професор		
7 Охорона праці та безпека в надзвичайних ситуаціях	Коробко О.В., ст. викладач		
Нормоконтроль	Андрєєв М.О., асистент		

7. Дата видачі завдання “ 7 ” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз та дослідження предметної області	2-4 тижні	Розділи 1, 2
3	Розробка архітектури програми	5 тиждень	Розділ 3
4	Розробка програми	6-8 тижні	Розділ 4
5	Тестування та експериментальне дослідження програми	9 тиждень	Розділ 5
6	Розробка економіко-організаційної частини	10 тиждень	Розділ 6
7	Розробка заходів з охорони праці та безпеки в надзвичайних ситуаціях	11 тиждень	Розділ 7
8	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	12 тиждень	Додатки
9	Захист роботи	13 тиждень	

Студент(ка)

(підпис)

Суязов С.О.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Федорончак Т.В.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
117 с., 36 рис., 6 табл., 3 дод., 36 джерел.

ФАНДРАЙЗИНГ ПРОЄКТІВ, ПРОГНОЗУВАННЯ, МАШИННЕ
НАВЧАННЯ, ДОВГА КОРОТКОСТРОКОВА ПАМ'ЯТЬ.

Об'єкт дослідження – процес прогнозування результатів фандрайзингу проєктів. Предмет дослідження – методи прогнозування результатів фандрайзингу проєктів.

Мета роботи – розроблення програмного забезпечення прогнозування результатів фандрайзингу проєктів для визначення політики рекламування проєктів на платформі зі своєчасним привертанням уваги до них.

Матеріали, методи та технічні засоби: мова програмування Python, вебфреймворк Django, мова розмітки HTML, машинне навчання.

Результати. Розроблено веб-застосунок, який дозволяє виконувати роботу над проєктами, вносити вклади на фінансування та аналізувати відповідні дані.

Наукова новизна роботи полягає в тому, що в ній запропоновано метод прогнозування результатів фандрайзингу проєктів, який ґрунтується на використанні нейронних мереж з довгою короткостроковою пам'яттю, використовуючи в якості вхідних ознак суми зібраних коштів на проєкт за перші дні збору, що дозволяє визначити підсумкову зібрану на проєкт суму коштів, визначаючи тим самим політику для роботи над фандрайзинговою компанією на ранніх етапах.

Висновки. Розроблено програмне забезпечення для підтримки процесу фандрайзингу проєктів.

Галузь використання – платформи зі збору коштів на виконання проєктів або проєкти, які беруть участь в спільному зборі коштів на одній з існуючих платформ.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 117 pages, 36 figures, 6 tables, 3 appendixes, 36 sources.

PROJECT FUNDRAISING, FORECASTING, MACHINE LEARNING, LONG SHORT-TERM MEMORY.

The object of research is the process of forecasting the results of fundraising projects. The subject of research is methods of forecasting the results of fundraising projects.

The purpose of the work is to develop software for forecasting the results of project fundraising to determine the policy of advertising projects on the platform with giving timely attention to them.

Materials, methods and tools: Python programming language, Django web framework, HTML markup language, machine learning.

Results. A web application has been developed that allows you to work on projects, make contributions to funding and analyze relevant data.

The scientific novelty of the work is that it offers a method of forecasting the results of fundraising projects, which is based on the use of neural networks with long short-term memory, using as inputs the amount of funds raised for the project for the first days of collection. the amount of funds for the project, thus determining the policy for working on a fundraising company in the early stages.

Conclusions. Software has been developed to support the project fundraising process.

The field of use lies in fundraising platforms for projects or projects which participate in joint fundraising on one of the existing platforms.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	11
1.1 Поняття фандрайзингу	11
1.2 Програмні засоби фандрайзингу проєктів	17
1.3 Висновки за розділом 1	21
2 Обґрунтування методу прогнозування результатів фандрайзингу проєктів.....	22
2.1 Аналіз методів прогнозування результатів фандрайзингу проєктів	22
2.2 Мережі з довгою короткостроковою пам'яттю.....	23
2.3 Метод прогнозування результатів фандрайзингу проєктів.....	26
2.4 Висновки за розділом 2	28
3 Проєктування архітектури системи	29
3.1 Вибір середовища проєктування	29
3.2 Визначення функціональних вимог до програмного забезпечення	31
3.3 Проєктування бази даних.....	35
3.4 Висновки за розділом 3	37
4 Основні рішення щодо реалізації компонентів системи.....	39
4.1 Структура розробленої програмної системи	39
4.2 Основні рішення щодо реалізації вебзастосунку	41
4.3 Основні рішення щодо реалізації підсистеми прогнозування результатів фандрайзингу проєктів	46
4.4 Висновки за розділом 4	48
5 Експлуатація та тестування програми	49
5.1 Призначення програми.....	49
5.2 Умови виконання програми.....	49
5.3 Виконання програми	50
6 Економіко-організаційна частина	64

6.1 Планування розробки програмного продукту	64
6.2 Розрахунок основної заробітної платні	66
6.3 Розрахунок додаткової заробітної платні	67
6.4 Відрахування на соціальні заходи	67
6.5 Визначення витрат на матеріали	68
6.6 Витрати на спеціальне обладнання	69
6.7 Розрахунок накладних витрат.....	71
6.8 Економічна ефективність.....	72
7 Охорона праці та безпека в надзвичайних ситуаціях.....	73
7.1 Аналіз потенційних небезпек	73
7.2 Заходи по забезпеченню безпеки.....	73
7.3 Заходи по забезпеченню промислової санітарії та гігієни праці	74
7.4 Заходи по забезпеченню пожежної безпеки	80
7.5 Заходи по забезпеченню безпеки у надзвичайних ситуаціях	81
Висновки.....	82
Перелік джерел посилання	83
Додаток А Технічне завдання.....	87
Додаток Б Текст програми.....	92
Додаток В Слайди презентації	110

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- LSTM – Long Short Term Memory;
ВДТ – візуальний дисплейний термінал;
ЕОМ – електронно-обчислювальна машина;
ПК – персональний комп'ютер;
РНМ – рекурентна нейронна мережа.

ВСТУП

Термін «фандрайзинг» був запозичений в США, де він використовується багато десятиліть переважно в сфері залучення фінансування в так званий третій сектор, що розвивається з кінця XIX – початку XX сторіччя і функціонує поряд з приватним і державним секторами, об'єднує некомерційні недержавні організації, покликані реалізовувати завдання, ігноровані приватним і державним секторами: активний розвиток фандрайзингу в другій половині XX сторіччя пов'язаний зі зростанням ролі третього сектору в США, питома вага загальної чисельності зайнятих в некомерційному секторі зростає в геометричній прогресії [1].

Україна має високий рівень інтелектуального потенціалу, однак є однією з не багатьох держав, де існує велика різниця між наявним інтелектуальним потенціалом та можливістю його повноцінно реалізовувати в готовий продукт, що спричинено складною економічною ситуацією в нашій країні, що є наслідком трансформаційних перетворень останніх десятиліть, які спровокували скорочення видатків держави на науково-дослідні роботи, фінансування науки і освіти тощо, в реалізації творчих здібностей і знань науковців через забезпечення їх необхідними ресурсами для довгострокових і середньострокових проєктів і програм можуть надати підтримку та допомогу вітчизняні і закордонні організації, фонди і установи: актуальними проблемами сьогодення є невміння знайти необхідні фонди, пробудити інтерес до свого проєкту, презентувати його, чітко сформулювати заявку, розробити програму заходів для залучення додаткових ресурсів на втілення програм, проєктів та дослідних робіт, вирішенню даних питань, а також забезпеченню розвитку і ефективної діяльності некомерційних і громадських організацій сприяє впровадження в практику роботи неприбуткових підприємств і організацій фандрайзингу [2].

В результаті в якості мети роботи було визначено розроблення програмного забезпечення прогнозування результатів фандрайзингу проєктів для визначення політики рекламування проєктів на платформі зі своєчасним привертанням уваги до них.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Поняття фандрайзингу

Поняття фандрайзингу можна визначити наступними характеристиками:

- багатостадійний або разовий процес;
- взаємодія, з передісторією або без неї, як правило, двох сторін (фізичних або юридичних осіб) – прохача і спонсора;
- процес, що має на меті збір коштів та залучення ресурсів на реалізацію соціально значимої діяльності однією стороною і прийняття рішення про їх надання іншою стороною;
- процес, що проходить у присутності консультантів, посередників і суперників;
- процес, в рамках узгодженої або запропонованої однієї зі сторін процедури;
- процес у формі, що є в даний час легітимною для середовища, де він здійснюється [3].

Фандрайзинг може бути внутрішнім і зовнішнім:

- у першому випадку мова йде про розробку і реалізацію стратегії пошуку джерел забезпечення діяльності силами менеджменту самої організації;
- у другому випадку фандрайзинг – це предмет консалтингової діяльності, коли пошук фінансування здійснюється за допомогою залучення професійних консультантів з фандрайзингу: існують терміни: «професійний фандрайзер» – той, хто надає послуги із залучення ресурсів самостійно або за контрактом, фандрайзер – співробітник організації, яка шукає шляхи залучення ресурсів, для залучення ресурсів в фандрайзингу можуть залучатися волонтери – групи населення, які готові пожертвувати своїм вільним часом і (або) коштами для пошуку необхідних ресурсів [4].

Джерелом отримання необхідних ресурсів можуть бути:

- державні органи (федеральні, державні та інші);
- комерційні організації (фірми, підприємства, корпорації тощо);
- приватні особи (бізнесмени, меценати та інші);
- партнери з числа некомерційних організацій (загальноосвітні або спортивні школи, заклади вищої освіти тощо);
- фонди, союзи, асоціації та інші некомерційні об'єднання [5].

До методів онлайн фандрайзингу належать зокрема наступні:

а) “peer-to-peer”, який можливо перекласти українською як “рівноправний фандрайзинг”, даний спосіб передбачає залучення волонтерів для проведення фандрайзингової кампанії, дозволяє проводити фандрайзинг не “зверху” (виключно персоналом організації), а долучати до цього процесу величезну кількість людей, які фактично стають агентами змін, для цього потрібно:

1) підшукати особистостей – волонтерів, які буквально можуть репрезентувати організацію: вони мають бути відкритими, мати вихід на широку аудиторію людей;

2) створити платформу для збору коштів – це може бути певна платіжна система або спеціальний вебсайт;

3) мотивувати волонтерів: зарядити волонтерів своєю енергією, розповісти про важливість фандрайзингу для активностей;

4) волонтери поширюють посилання на сторінки зі збору коштів серед своїх знайомих, розповідають про історію організації та свою особисту історію у зв'язку із організацією;

б) фандрайзингові листи: такий інструмент використовується переважно за допомогою email-розсилки та орієнтований саме на зацікавлену аудиторію, а не широку громадськість, для використання даного інструменту потрібно мати свою базу розсилки, такий інструмент надає можливість:

1) розповісти про онлайн фандрайзингову кампанію: можливо вставити посилання на ресурс, за допомогою якого збираються кошти компанією;

2) залучити волонтерів для “рівноправного” фандрайзингу: можна персонально запропонувати аудиторії стати волонтерами та підкреслити їхню важливість;

3) запросити на фандрайзингові заходи: усім подобаються персональні запрошення на заходи, а також можна запропонувати невеликий презент відвідувачам, які покажуть такого листа;

в) фандрайзинг через інфокампанію: дана ідея потребує більше ресурсів, аніж попередні способи, але вона дозволяє ілюструвати діяльність, запуск інформаційної кампанії – це окреме мистецтво, проте якщо йдеться саме про фандрайзингові цілі, то послідовність кроків повинна виглядати приблизно таким чином:

1) обрати головні повідомлення для кампанії: це має бути заклик або допомогти безпосередньо групам, з якими відбувається взаємодія, або допомогти вирішенню певних проблем;

2) розробити візуальний супровід обраних повідомлень: фандрайзингові кампанії рідко виглядають “нейтрально”, вони містять умовно позитивне або негативне зображення, наприклад, риби, які помирають від пластику, або дитина, яка щасливо грається із іграшками, перший приклад містить негативну ілюстрацію і змушує людей злякатися: якою буде доля цих риб, якщо я не пожертвую кошти на очищення води, другий приклад містить позитивну ілюстрацію і мотивує людей наблизити цю картинку, щоб дитина одужала і змогла щасливо гратися;

3) прив’язати публікацію із інфокампанією до фандрайзингових ресурсів – платіжних систем або вебсайтів: показати аудиторії чіткий зв’язок між картинкою, яку вони бачать, та внеском, який вони можуть зробити [6].

В рамках фандрайзингу приватні особи, організації, фонди і всі інші джерела отримання ресурсів для діяльності некомерційних організацій

прийнято називати донорами, всі мотиви донорів діляться, як правило, на дві групи:

- безкорисливі мотиви: керуючись такими мотивами, донори підтримують діяльність установ культури, виходячи з безоплатних інтересів (наприклад, бажання допомогти установі культури, керуючись власними переконаннями або відповідно до культурних, етичних, соціальних, релігійних та інших вимог);

- комерційні мотиви – дані мотиви характерні для донорів, які прагнуть здійснити підтримку діяльності установ на підставі отримання певного прибутку, а саме символічного прибутку: як символічний прибуток можуть виступати будь-які значення і цінності, які виникають по відношенню до донорів: позитивна репутація, реклама, поліпшення іміджу, налагодження комунікацій, підвищення соціального і культурного статусу в суспільстві та інші [5].

Основними поняттями, без яких не може існувати фандрайзинг, є такі поняття як:

- грант – кошти, безоплатно передані дарувальником (фондом, корпорацією, урядовою установою) некомерційній організації або приватній особі для виконання конкретної роботи: на відміну від позики грант не потрібно повертати;

- заявка – письмове звернення з проханням про виділення гранту: на відміну від бізнес-плану, заявка пишеться, щоб отримати кошти на некомерційний проєкт, тобто на справу, яка свідомо не орієнтована на отримання прибутку;

- донор – юридична або фізична особа, яка надає матеріальну, фінансову, організаційну та іншу благодійну допомогу неприбутковим організаціям на добровільній безкорисливій основі;

- спонсор – юридична або фізична особа, яка надає на добровільній і неприбутковій основі матеріальну підтримку благодійній діяльності з метою популяризації виключно свого імені (назви), торгової марки;

- грантодавці – приватні особи або організації, які розглядають заявки на грант і які надають гранти;
- заявник – приватна особа або організація, що подає заявки на грант;
- проєкт – логічна сукупність заходів, що здійснюються для досягнення поставлених цілей в рамках встановленого бюджету, плану-графіка реалізації та прогнозування результатів діяльності;
- спонсорство – взаємовигідний обмін грошових коштів, товарів, послуг, робіт на можливості розміщення реклами, PR-заходів, що стимулюють збут;
- благодійність – добровільна, безкорислива, різна за формами підтримка фінансових і юридичних осіб в загальнокорисних цілях;
- пожертвування – дарування речі або права обмеженому колу благоотримувачів в загальнокорисних цілях;
- патронаж – надання довгострокової матеріальної та організаційної підтримки некомерційної діяльності у відповідь на отримання прав керування нею;
- членство – регулярне відрахування в некомерційну організацію фіксованих коштів з метою отримання стандартного пакета некомерційних послуг;
- меценатство – організація або підтримка суспільно-значущих заходів в просвітницьких цілях (переважно в сфері культури і мистецтв);
- державні фонди – держустанови, що фінансуються з бюджету своєї держави;
- посередницькі фонди – громадські організації, що фінансуються державними чи приватними фондами і розподіляють фінанси по заявникам, що задовольняє вимогам фонду або програми;
- приватні фонди – це недержавні некомерційні організації, які отримують кошти від громадян (у вигляді пожертвувань) або комерційних організацій, а також приватні особи [1].

Успішність фандрайзингу оцінюється розміром залучених ресурсів, приклади фандрайзингу в усьому світі демонструють, що в залежності від тієї мети, яку переслідує організація, його можна розділити на два основних види:

а) оперативний фандрайзинг – такий вид фандрайзингу передбачає організацію фандрайзингової кампанії з метою забезпечення діяльності тієї чи іншої організації: завданням фандрайзера може стати отримання будь-якого виду ресурсів (від фінансової допомоги до покупки устаткування), які необхідні для роботи установи, особливістю даного виду фандрайзингу є відсутність конкретного призначення залучених ресурсів і, відповідно, їх різноманітність, негативним моментом в даному випадку слід вважати «розмитість» завдань фандрайзингу, відсутність конкретних результатів від використання ресурсів, що може знижувати ефективність практик фандрайзингу, позитивною ж є можливість використовувати отримані ресурси на розсуд організації, в тому числі, при плануванні майбутніх перспектив, приклади оперативного фандрайзингу в музеї – залучення ресурсів для підтримки діяльності установи культури у вигляді:

- 1) фінансових коштів;
- 2) покупки спеціального експозиційного обладнання;
- 3) волонтерів, які здійснюють допомогу в організації наукової конференції;

б) проєктний фандрайзинг – даний вид фандрайзингу передбачає організацію фандрайзингової кампанії з метою залучення конкретних ресурсів для реалізації проєктів: проєктний фандрайзинг володіє явною перевагою в порівнянні з оперативним, перш за все, його позитивним моментом слід назвати «оформленість» і ясність пропозицій про співпрацю для організацій-донорів та конкретизації необхідних ресурсів, фандрайзинг в даному випадку передбачає акцентування уваги на планованому результаті реалізації проєкту (відкриття музею, проведення фестивалю), але слід зауважити, що нерідко професійні фахівці даної області діяльності

рекомендують для проведення фандрайзингу представляти будь-яку діяльність в якості того чи іншого проєкту, такий проєктом може бути будь-яка дія, яке має свої терміни, ресурси, результат, тоді список ресурсів буде прозорий, зрозумілий, а завдання донора і результат культурної діяльності очевидними, приклад проєктного фандрайзингу в театрі – залучення ресурсів для проведення театрального фестивалю у вигляді:

- 1) грошових кошти на оплату виготовлення необхідних декорацій і покупку костюмів;
- 2) інформаційна підтримка засобів масової інформації;
- 3) придбання сувенірів та призів;
- 4) волонтерів, здатних надати допомогу в організації фестивалю [5].

1.2 Програмні засоби фандрайзингу проєктів

У світі на даний момент існує багато різних платформ, які використовуються для фандрайзингу проєктів.

Airfunding [7] (рис. 1.1) – це платформа для фандрайзингу, що базується в Японії, де можна збирати кошти з метою позитивного впливу на життя людей чи громади: проєкти мають свої цілі, ліміти фінансування та суми, зібрані в межах лімітів фінансування, передаються власникам проєктів незалежно від того, досягнуто цілі чи ні, окремої плати за платформу не стягується, а 6% буде стягнуто із зібраної суми з урахуванням комісії за отримання банком власника проєкту, платформа на даний момент має понад 10 мільйонів прихильників, підтримуються різні проєкти, такі як медичні, сімейні, проєкти громад, подолання катастроф, імміграції, освітні, мистецькі, бізнес, навчання за кордоном та подорожі [8].

Вебзастосунок Airfunding дозволяє виконувати зокрема наступні функції:

- створювати власні проєкти;

- виконувати пошук проєктів;
- фінансувати проєкти;
- переглядати детальну інформацію про проєкт.

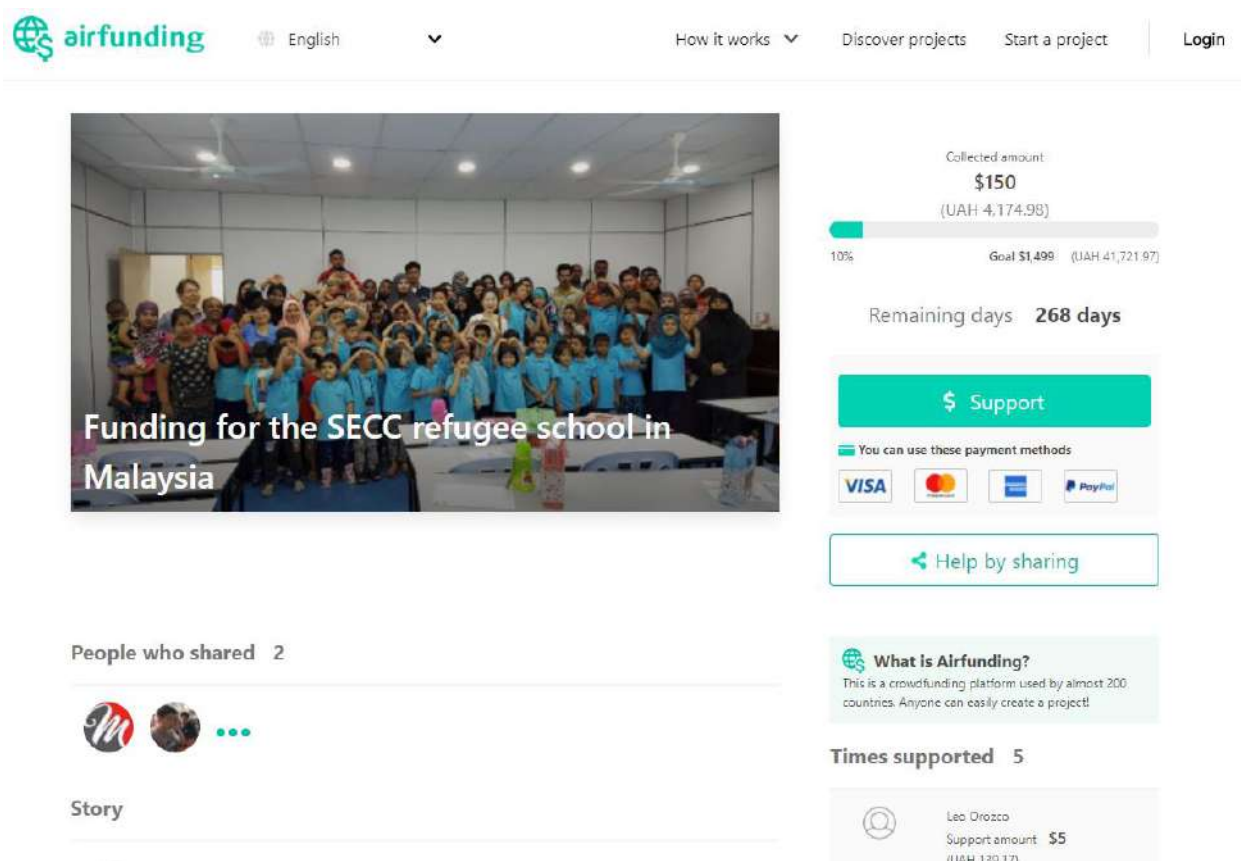


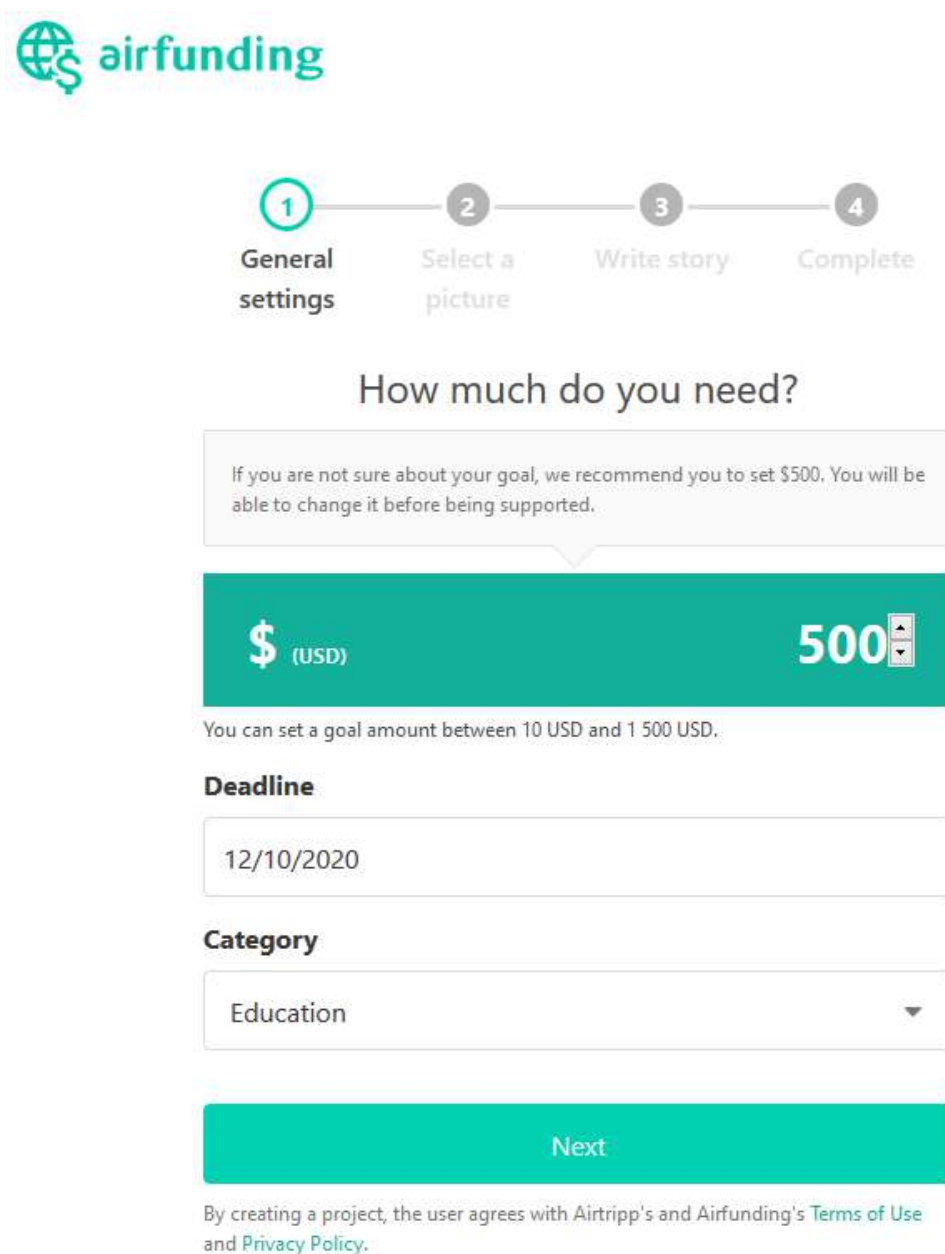
Рисунок 1.1 – Інтерфейс вебзастосунку Airfunding [7]

За кожним проєктом демонструється поточна зібрана на нього сума, мета, процент завершення збору коштів та кількість днів, що лишилась для збору. Будь-яких прогнозів щодо успішності завершення компанії зі збору коштів не надається.

Під час створення власного проєкту (рис. 1.2) користувач повинен вказати наступну інформацію:

- суму коштів, яку необхідно зібрати;
- строк для завершення збору коштів (дата);
- категорію проєктів;
- зображення;

- назву проєкту;
- історію проєкту (опис).



airfunding

1 General settings 2 Select a picture 3 Write story 4 Complete

How much do you need?

If you are not sure about your goal, we recommend you to set \$500. You will be able to change it before being supported.

\$ (USD) 500

You can set a goal amount between 10 USD and 1 500 USD.

Deadline

12/10/2020

Category

Education ▼

Next

By creating a project, the user agrees with Airtripp's and Airfunding's [Terms of Use](#) and [Privacy Policy](#).

Рисунок 1.2 – Внесення даних проєкту для вебзастосунку Airfunding [7]

GoFundMe [9] (рис. 1.3) – це платформа для фандрайзингу, яка була створена в 2010 році і є найбільш надійною платформою безкоштовного соціального збору коштів для приватних осіб, груп та організацій, це реалізація соціальної платформи Indiegogo, спеціально розроблена для допомоги як малим, так і великим групам, які мають справу або проєкт для

збору коштів: за допомогою платформи на даний момент зібрано понад 4 мільярди доларів для людей, які потребують коштів, забезпечується платформа, на якій окремі люди та групи співпрацюють для збору коштів шляхом надання інструментів соціального обміну разом із безкоштовною платформою, не має плати за платформу, не має термінів і не вимагає обмежень в строках [8].

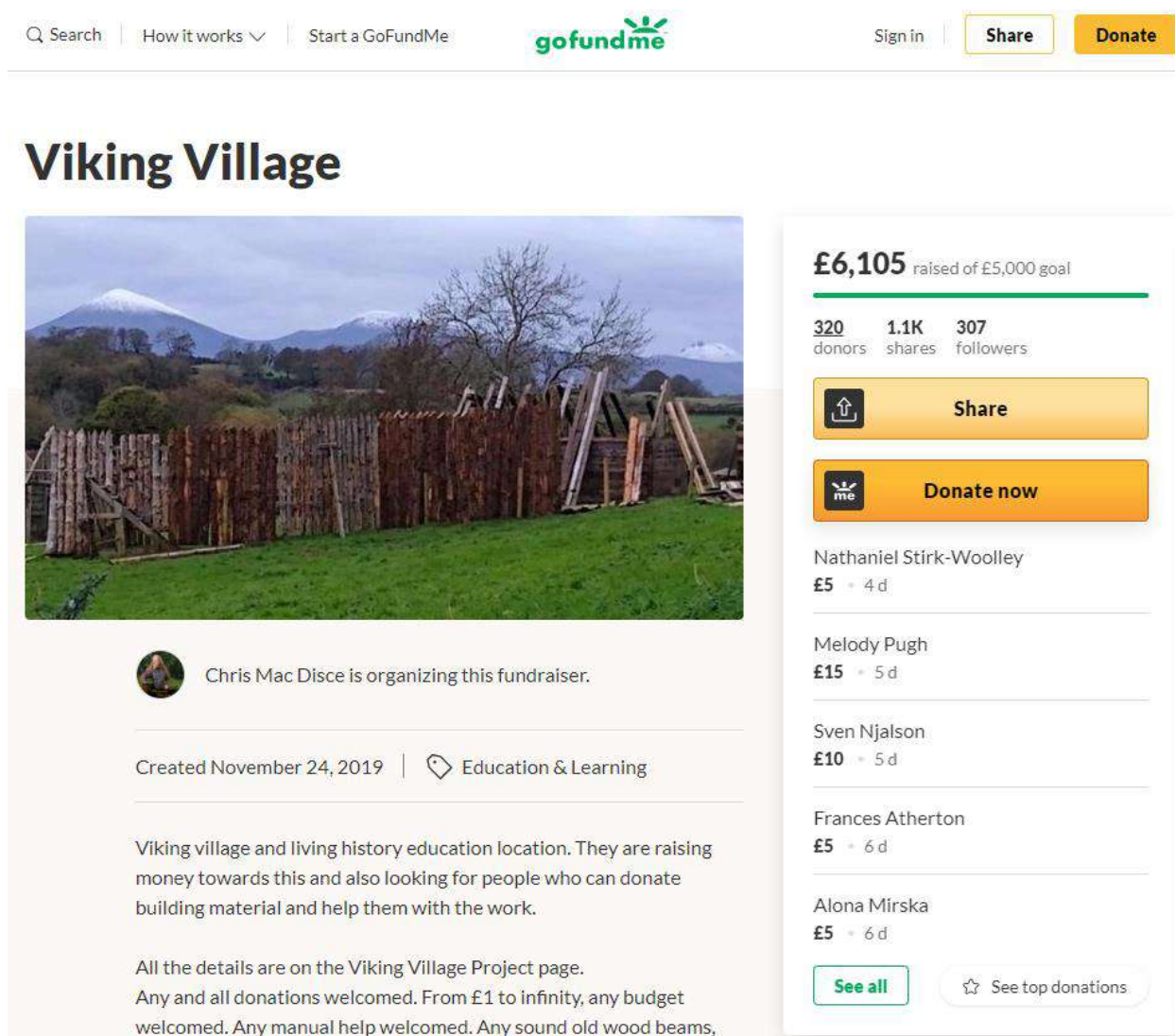


Рисунок 1.3 – Інтерфейс вебзастосунку GoFundMe [9]

Платформа дозволяє переглядати, ким саме і яку саме суму було перелічено на даний проєкт. Окрім того під час внесення пожертвування можна залишити коментар. Це основні функціональні відмінності (окрім ідеологічних) у даній платформі відносно попередньої.

1.3 Висновки за розділом 1

Виконано огляд проблеми фандрайзингу з наведенням основних понять, типів фандрайзингу, описом даного процесу. Виконано огляд програмних засобів, які використовуються для фандрайзингу проєктів. Визначено, що вони не мають засобів прогнозування результатів збору коштів на проєкти у своєму складі, окрім того порівняно інші наявні функціональні можливості.

Завданнями дипломної кваліфікаційної роботи магістра є:

- дослідження методів прогнозування результатів фандрайзингу проєктів;
- проєктування програмного забезпечення;
- розроблення програмного забезпечення;
- тестування та налагодження програми;
- економічне обґрунтування доцільності розробки та забезпечення засобів охорони праці.

2 ОБҐРУНТУВАННЯ МЕТОДУ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ФАНДРАЙЗИНГУ ПРОЄКТІВ

2.1 Аналіз методів прогнозування результатів фандрайзингу проєктів

Фандрайзинг проєктів є відносно новим напрямком діяльності, а тому прогнозування результатів фандрайзингу є ще недостатньо вивченою проблемою. Тому для загального розуміння стану проблеми доцільним буде об'єднати під час огляду в одну групу методи, які було досліджено для виконання прогнозування результатів фандрайзингу проєктів та прогнозування успішності виконання фандрайзингу проєктів (тобто виконання задачі класифікації).

У роботі [10] виконується прогнозування вкладів, які надаються під час фандрайзингу проєктів, на основі використання регресійної моделі. При цьому за даними попередніх періодів (днів) прогнозується зібрана сума коштів на наступний період (день). Недоліком такої моделі є те, що вона здатна тільки узагальнювати дані, які були зібрані за попередніми фандрайзингами проєктів.

У роботі [11] розглядається підхід до прогнозування успішності виконання фандрайзингу проєктів на основі використання нейронних мереж, при цьому в якості такого механізму використовуються ансамблі нейронних мереж.

У роботі [12] досліджується класифікація на основі використання наступних методів:

- дерева рішень;
- метод опорних векторів;
- логістична регресія;
- метод k-найближчих сусідів.

У роботі [13] було проаналізовано дані щодо виконання фандрайзингу проєктів. Лише невелика частина неуспішних проєктів (близько 10 % змогла

її подолати), при цьому успішні проєкти навпаки головним чином цю лінію долають.

Відповідно такі результати аналізу дозволяють говорити про те, що збір коштів у перші дні є одним з важливих показників того, на скільки успішною буде компанія зі збору коштів у кінці.

До архітектур штучних нейронних мереж, які можуть використовуватися для прогнозування часових рядів, належать:

- шар довгої короткострокової пам'яті (Long Short Term Memory (LSTM)) дозволяє повторно задіяти в процесі навчання попередню інформацію і тим самим вирішити проблему загасання градієнта;

- шар керованих зворотних блоків (Gated Recurrent Unit) заснований на тому ж принципі, що і шар LSTM, однак рекурентні блоки представляють собою більш прості структури в порівнянні з LSTM і, відповідно, менш затратні в обчислювальному сенсі;

- згорткові нейронні мережі (convolutional neural network), що застосовуються завдяки їх здатності до згортання параметрів, вилучення ознак з локальних вхідних шаблонів, отримання ефективних і модульних уявлень даних, незважаючи на те, що основна сфера їх застосування – розпізнавання образів на зображеннях: час можна розглядати як просторовий вимір, подібно висоті або ширині двовимірного зображення [14].

2.2 Мережі з довгою короткостроковою пам'яттю

Рекурентна нейронна мережа (РНМ) є мережею прямого поширення сигналу, але нейрони отримують інформацію не тільки від попереднього шару, але і від свого рекурентного зв'язку, це означає, що мережа навчається запам'ятовувати також послідовність даних, що надходять: складність РНМ – це проблема зникаючого градієнта: мережа швидко втрачає інформацію з плином часу, хоча це впливає тільки на ваги, а не на стан нейронів, але інформація накопичується саме в них, рекурентні нейронні мережі

найчастіше використовуються для автоматичного доповнення інформації [15].

Мережа з довгою короткостроковою пам'яттю, або LSTM, вирішує проблему втрати інформації в РНМ, використовуючи фільтри і явно задану клітку пам'яті, кожен нейрон має клітку пам'яті і три фільтри (вхідний, вихідний і забуваючий, мета фільтрів – захистити інформацію):

- вхідний фільтр визначає, скільки інформації з попереднього шару буде зберігатися в клітці;
- вихідний фільтр визначає, скільки інформації отримають наступні шари;
- забуваючий фільтр контролює міру збереження значення в пам'яті: наприклад, якщо мережа вивчає книгу і переходить на нову главу, то якісь символи з попередньої вона забуває, такі мережі LSTM здатні створювати складні структури, але для цього вони вимагають велику кількість ресурсів [15].

На рис. 2.1 приведено архітектуру нейронної мережі LSTM [16].

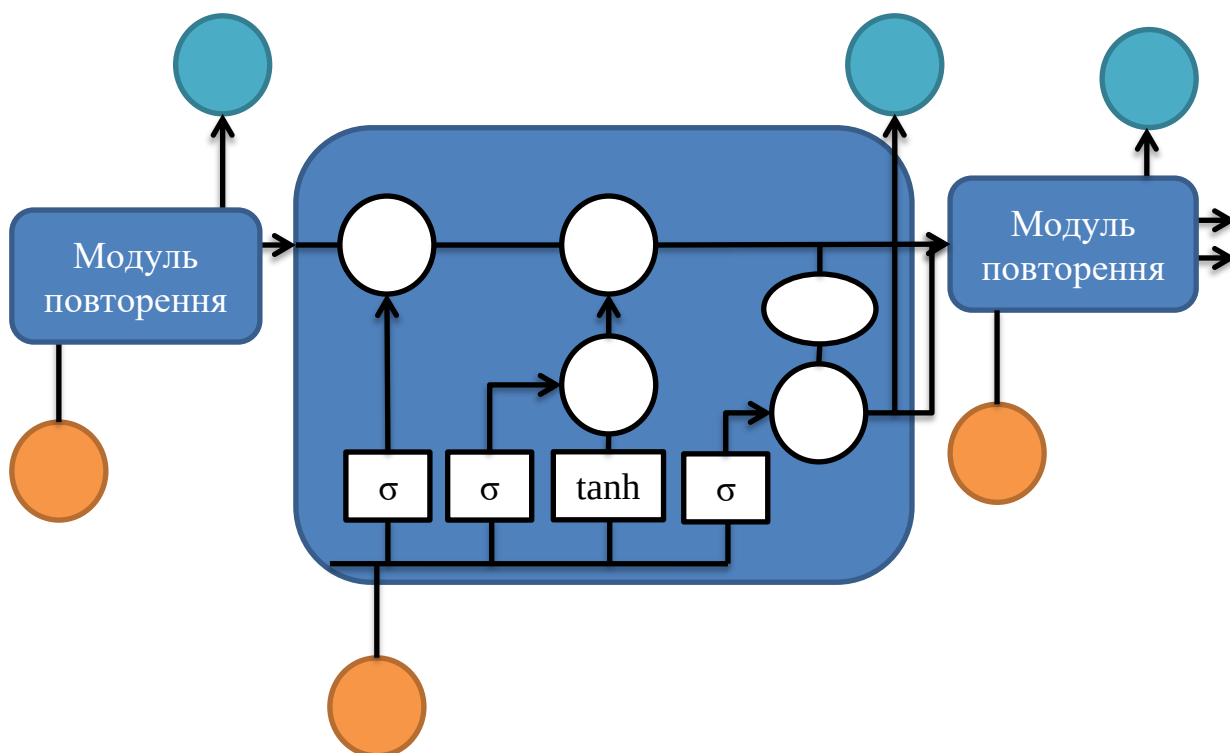


Рисунок 2.1 – Архітектура нейронної мережі LSTM

На рис. 2.1 зображено чотири шари нейронної мережі в білих прямокутниках, точкові оператори в білих колах, введення в помаранчевих колах і стан клітинки в бірюзових колах: модуль LSTM має стан клітинки і три шлюзи, що дає їм можливість вибірково вивчати, відключати або зберігати інформацію від кожного з модулів, стан клітинки в LSTM допомагає інформації проходити через одиниці без зміни, дозволяючи лише кілька лінійних взаємодій [16].

Обчислення в межах організації LSTM організовуються на основі наступних формул (2.1)-(2.6):

$$f = \sigma(W_f h_{t-1} + U_f x_t), \quad (2.1)$$

$$g = th(W_g h_{t-1} + U_g x_t), \quad (2.2)$$

$$i = \sigma(W_i h_{t-1} + U_i x_t), \quad (2.3)$$

$$o = \sigma(W_o h_{t-1} + U_o x_t), \quad (2.4)$$

$$c_t = (c_{t-1} \otimes f) \oplus (g \otimes i), \quad (2.5)$$

$$h_t = th(c_t) \otimes o, \quad (2.6)$$

де i , f , o , g – вхідний шлюз, шлюз забуття, вихідний шлюз, внутрішній прихований стан відповідно;

W_i , W_f , W_o , W_g – матриці ваг вхідного шлюзу, шлюзу забуття, вихідного шлюзу, внутрішнього прихованого стану відповідно;

U_i , U_f , U_o , U_g – матриці зміщень вхідного шлюзу, шлюзу забуття, вихідного шлюзу, внутрішнього прихованого стану;

c_t – поточний стан клітинки;

c_{t-1} – попередній стан клітинки;

σ – сигмовидна функція [14].

Нейронні мережі LSTM – це штучна нейронна мережа, яка використовується в області глибокого навчання: подібні мережі здатні обробляти не тільки окремі поодинокі дані (наприклад, зображення), а й цілі

послідовності даних (наприклад, аудіозапис мови або відеозапис), тому LSTM мережі здатні вирішувати такі завдання, як розпізнавання рукописного введення, розпізнавання мови, визначення аномалій у великих потоках даних (мережевий трафік, банківські транзакції), LSTM нейронні мережі добре підходять для класифікації, обробки і побудови прогнозів на основі часових рядів, де взаємопов'язані явища можуть відбуватися з невизначеним часовим лагом, цей часовий лаг призводить до ускладнень у використанні класичних нейронних мереж в вирішенні даних проблем через загасання градієнта, в той час як LSTM мережі нечутливі до величини часового лагу [17].

2.3 Метод прогнозування результатів фандрайзингу проєктів

Метод прогнозування результатів фандрайзингу проєктів було побудовано на основі основних рішень, які було виділено за результатами аналізу, а також з застосуванням логіки прогнозування на майбутній період, яку було використано зокрема в роботі [18].

Відповідно проблему, що розв'язується в роботі, можна сформулювати наступним чином на основі формули (2.7):

$$v^c(t) = f(v(t-d), v(t-d-1), \dots, v(1)), \quad (2.7)$$

де $v^c(t)$ – загальна сума, зібрана в процесі фандрайзингу проєкту, тобто сума, зібрана в момент часу t , що визначає тривалість реалізації фандрайзингу;

$v(t-d)$ – сума коштів, яка була зібрана на реалізацію проєкту в момент часу $t-d$, тобто в останній момент часу, який використовується для подальшого прогнозування (для прогнозування поточний момент часу);

$v(t-d-1)$ – сума коштів, яку було зібрано в попередній момент $t-d-1$, тобто в попередній день під час прогнозування;

$v(1)$ – сума коштів, яку було зібрано в перший день фандрайзингу;

f – функція, яка моделює відповідну залежність.

Відповідно метод має дозволяти виконувати моделювання і на основі нових даних про збір коштів у перші дні виконувати прогнозування кумулятивної зібраної суми за результатами останнього дня фандрайзингу.

Для реалізації методу на першому етапі виконується підготовка даних. Під час підготовки даних вся множина наявних проєктів має бути представлена у вигляді записів $v(1), v(2), \dots, v(t - d - 1), v(t - d), v(t)$. Зі всіх таких наявних записів мають бути виділені навчальна та тестова вибірка.

Для кожного запису має бути виконана нормалізація даних. Для цього виконується ділення кожного значення на суму, яку було заплановано для збору під час фандрайзингу. Якщо фандрайзинг завершується при досягненні максимальної суми, то значення будуть розподілені в діапазоні від 0 до 1. Якщо фандрайзинг на цьому не завершується, то можливі значення більше 1.

Під час опробовування методу було використано дані про збір коштів за перші 7 днів фандрайзингу. Можна використовувати також коротші періоди. Можливим є застосування періоду в 3 дні, але при цьому знижується точність прогнозів. Тому такий спосіб можливий тоді, коли потрібно якомога швидше отримати прогноз про долю проєкту.

Після цього на другому етапі реалізується створення нейронної мережі архітектури LSTM.

На наступному етапі виконується навчання моделі на основі навчальної вибірки, створеної на першому етапі.

На останньому етапі реалізується прогнозування на основі навчаної моделі. Для цього за новими проєктами, фандрайзинг яких тільки розпочато (перші 7 днів), виконується прогнозування результату збору коштів.

2.4 Висновки за розділом 2

За результатами проведеного аналізу методів, які використовуються для прогнозування під час фандрайзингу проєктів, було виділено, що на даний момент прогнозування результатів фандрайзингу проєктів, тобто зібраних коштів, виконується за допомогою використання регресійної моделі, в той же час для прогнозування успішності збору використовуються дерева рішень, метод k-найближчих сусідів, ансамблі нейронних мереж. У той же час актуальним рішенням для розв'язання ширшої проблеми, прогнозування часових рядів, є нейронні мережі, тому було запропоновано використовувати даний механізм для виконання прогнозування результатів фандрайзингу проєктів. Проаналізовано архітектуру мереж з довгою короткостроковою пам'яттю.

Запропоновано метод прогнозування результатів фандрайзингу проєктів, який ґрунтується на використанні нейронних мереж з довгою короткостроковою пам'яттю, використовуючи в якості вхідних ознак суми зібраних коштів на проєкт за перші дні збору, що дозволяє визначити підсумкову зібрану на проєкт суму коштів, визначаючи тим самим політику для роботи над фандрайзинговою компанією на ранніх етапах.

3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

3.1 Вибір середовища проєктування

Для розробки даного проєкту мають використовуватися засоби, які відзначаються наступними можливостями:

- веброзробка: оскільки користувачі мають отримувати доступ до роботи з системою віддалено, то найзручніший засіб для роботи з різними пристроями одночасно – це засоби веброзробки;
- штучний інтелект: програмний застосунок має одночасно реалізовувати і запропонований в розділі 2 роботи метод;
- підтримка автоматичного виконання прогнозування за результатами фандрайзингу проєктів за день.

Мова програмування Python використовується в наступних областях:

- веб-розробка: можна використовувати Python у веб-розробці, при цьому може бути багато вебфреймворків на основі Python (Django, Flask, Bottle тощо);
- програми з графічним інтерфейсом: можна розробити графічний інтерфейс користувача, використовуючи модулі Python, такі як Tkinter, PyQt, Kivy тощо;
- наука про дані: більшість дослідників даних віддають перевагу Python за його простоту та бібліотеки, такі як Pandas, Numpy, Matplotlib тощо;
- машинне навчання: проєктування машинних моделей за допомогою Python – одне з найпоширеніших застосувань Python, існують різні механізми машинного навчання, такі як Tensorflow, Scikit Learn, Spark тощо;
- автоматизація дій: можна автоматизувати дії за допомогою сценаріїв Python, наприклад, перейменування великої кількості файлів, очищення диска;
- розробка ігор: можна розробляти ігри за допомогою Python, для цього існують такі фреймворки, як Pygame, Pyglet, PyOpenGL тощо;

– витягання даних у веб: можна використовувати Python для витягання даних з Інтернету (з вебсторінок), при цьому такі модулі як requests, bs4, selenium тощо, допомагають отримувати вебвміст за допомогою Python;

– обробка зображення: модулі python, такі як Scikit-Image, OpenCV, Pillow тощо, використовуються для обробки зображень [19].

Враховуючи необхідність виконання веброзробки, було розглянуто дві альтернативи для реалізації програмного забезпечення – мови програмування PHP і Python. Через необхідність саме реалізації повноцінного застосунку (вебзастосунку) було відкинута мова R. Результати порівняння зведено до табл. 3.1.

Таблиця 3.1 – Вибір мови програмування

Показник	PHP	Python
Основна область застосування	Веброзробка	Загальне призначення, зокрема широко використовується в штучному інтелекті, роботі з даними
Фреймворки веброзробки	+	+
Машинне навчання	Наявні бібліотеки, документація мінімальна	Набір бібліотек, велика спільнота, широка документація
Відкритий вихідний текст	+	+

Загалом мови PHP і Python є конкурентними, але враховуючи, що в процесі має бути реалізовано запропонований в роботі метод, то важливою є не тільки можливість веброзробки, але і відносно простого виконання

машинного навчання. РНР підтримує машинне навчання, але має обмежену документацію, маленьку спільноту, що робить цей процес складнішим. Цей фактор є критичним, адже розробка даного проєкту дуже обмежена в часі за вимогами календарного плану. Тому в результаті було обрано мову програмування Python для реалізації проєкту.

3.2 Визначення функціональних вимог до програмного забезпечення

Задля реалізації ідеї фандрайзингу проєктів на основі проведеного аналізу існуючих програмних засобів у розділі 1 розроблюване програмне забезпечення повинно реалізовувати виконання наступних функціональних вимог:

- створення нових проєктів;
- перегляд власних проєктів;
- перегляд детальних даних проєкту;
- перегляд вкладів у проєкт;
- внесення вкладу на фінансування проєкту;
- редагування проєктів;
- перегляд переліку активних проєктів;
- перегляд переліку завершених проєктів;
- перегляд успішних проєктів;
- пошук проєктів;
- перегляд історії власних вкладів;
- внесення грошей на власний рахунок;
- авторизація облікового запису;
- керування правами доступу користувачів;
- реєстрація облікового запису.

Всіх користувачів, які взаємодіють з програмою, можна розділити на 3 групи:

- адміністратори;
- автори проєктів;
- фандрайзери.

Адміністратори можуть взаємодіяти з програмою шляхом взаємодії зі всіма наявними функціональними можливостями.

Відповідну діаграму прецедентів для роботи авторів проєктів з системою представлено на рис. 3.1.

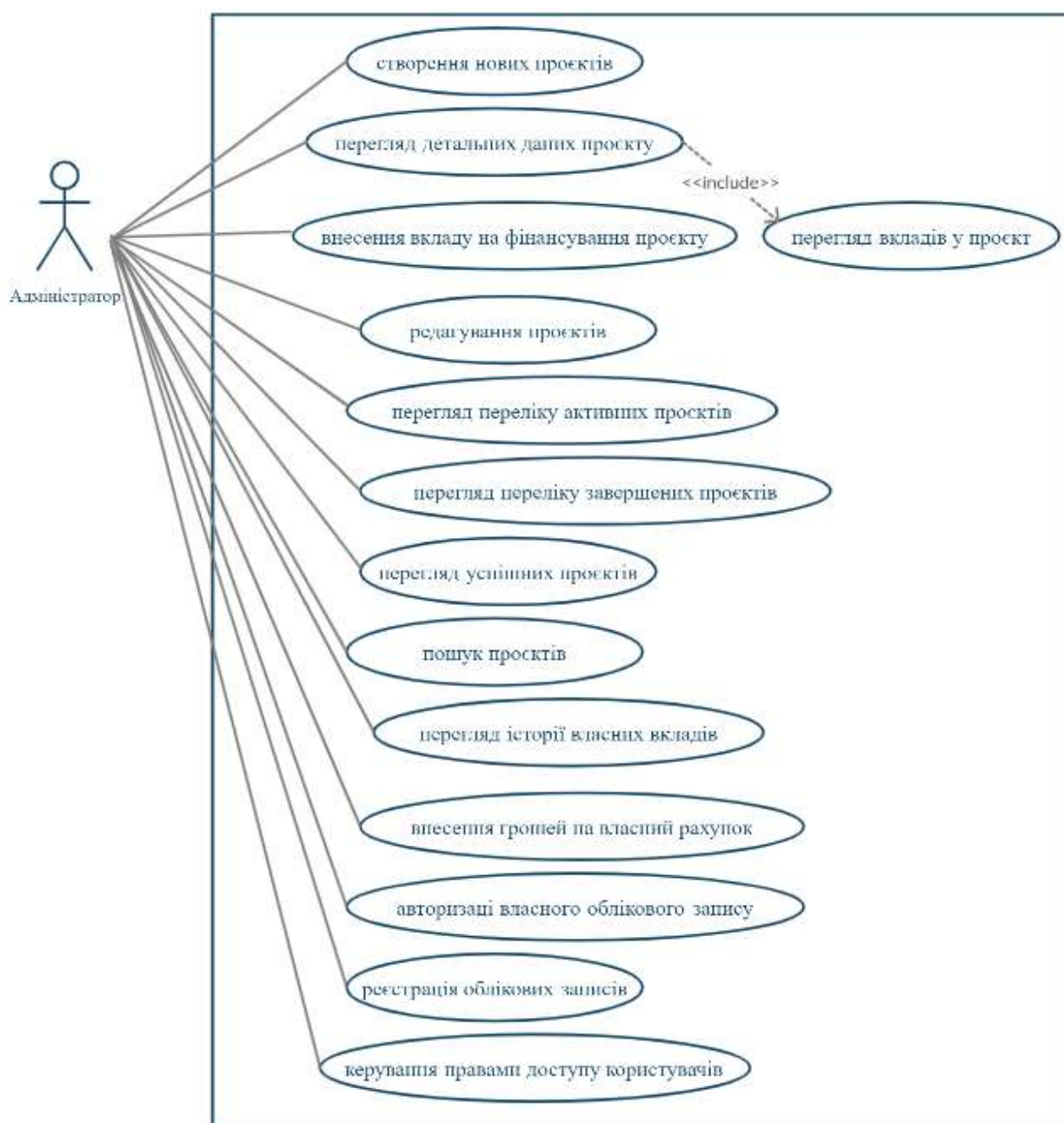


Рисунок 3.1 – Діаграма прецедентів для роботи адміністратора з системою

Автори проєктів можуть виконувати:

- створення нових проєктів;
- перегляд власних проєктів;
- перегляд детальних даних проєкту;
- перегляд вкладів у проєкт;
- внесення вкладу на фінансування проєкту;
- редагування власних проєктів;
- перегляд переліку активних проєктів;
- перегляд переліку завершених проєктів;
- перегляд успішних проєктів;
- пошук проєктів;
- перегляд історії власних вкладів;
- внесення грошей на власний рахунок;
- авторизація облікового запису.

Відповідну діаграму прецедентів для роботи авторів проєктів з системою представлено на рис. 3.2.

Тобто реєструвати облікові записи авторів проєктів можуть тільки адміністратори. Це необхідно для забезпечення попереднього узгодження проєктів для запобігання публікації будь-яких матеріалів через вебзастосунок.

Фандрайзерам доступні наступні функціональні можливості:

- перегляд детальних даних проєкту;
- перегляд вкладів у проєкт;
- внесення вкладу на фінансування проєкту;
- перегляд переліку активних проєктів;
- перегляд переліку завершених проєктів;
- перегляд успішних проєктів;
- пошук проєктів;
- перегляд історії власних вкладів;
- внесення грошей на власний рахунок;

- авторизація облікового запису;
- реєстрація облікового запису.

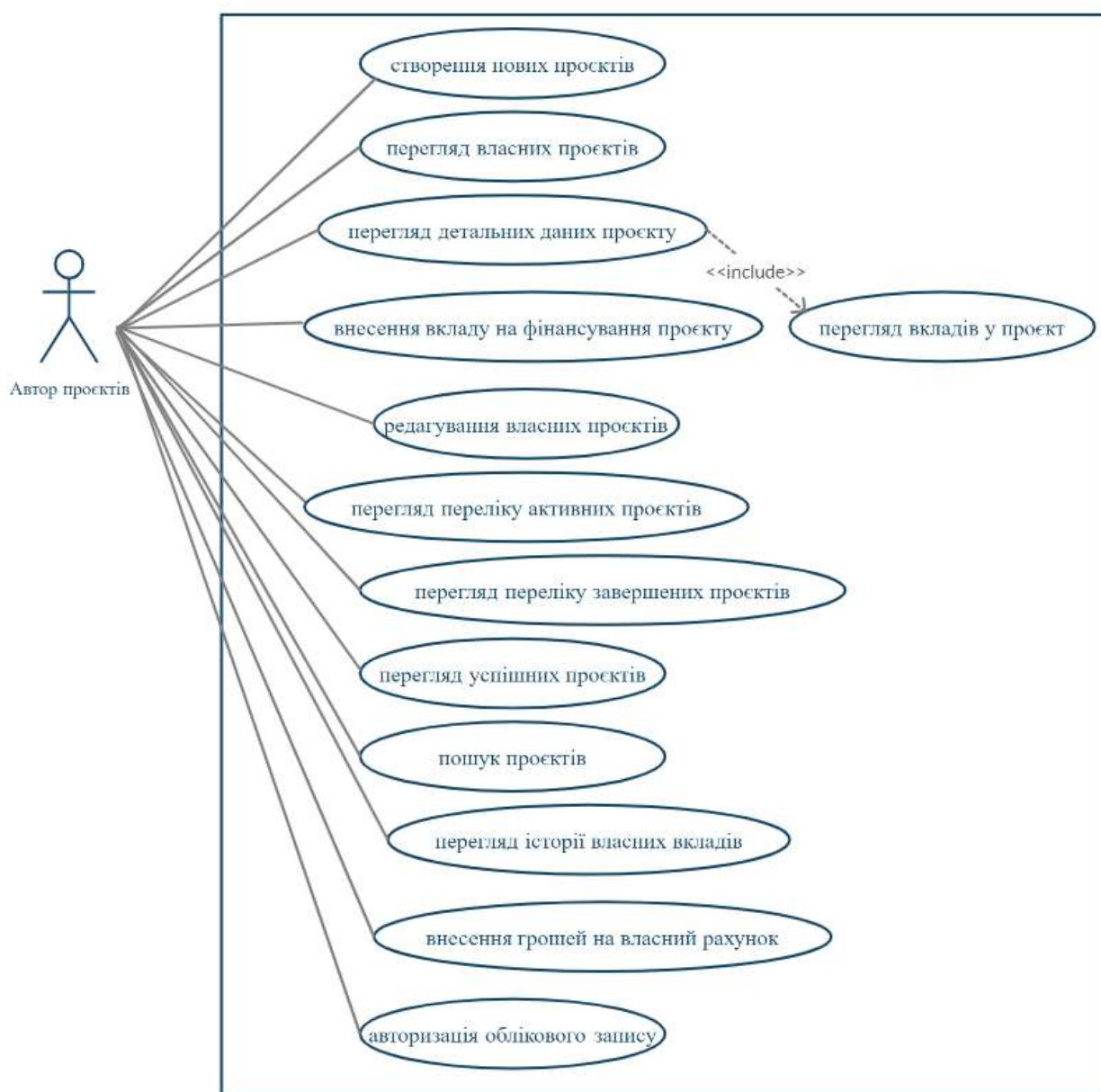


Рисунок 3.2 – Діаграма прецедентів для роботи автора проєктів з системою

Відповідну діаграму прецедентів для роботи фандрайзерів з системою представлено на рис. 3.3.

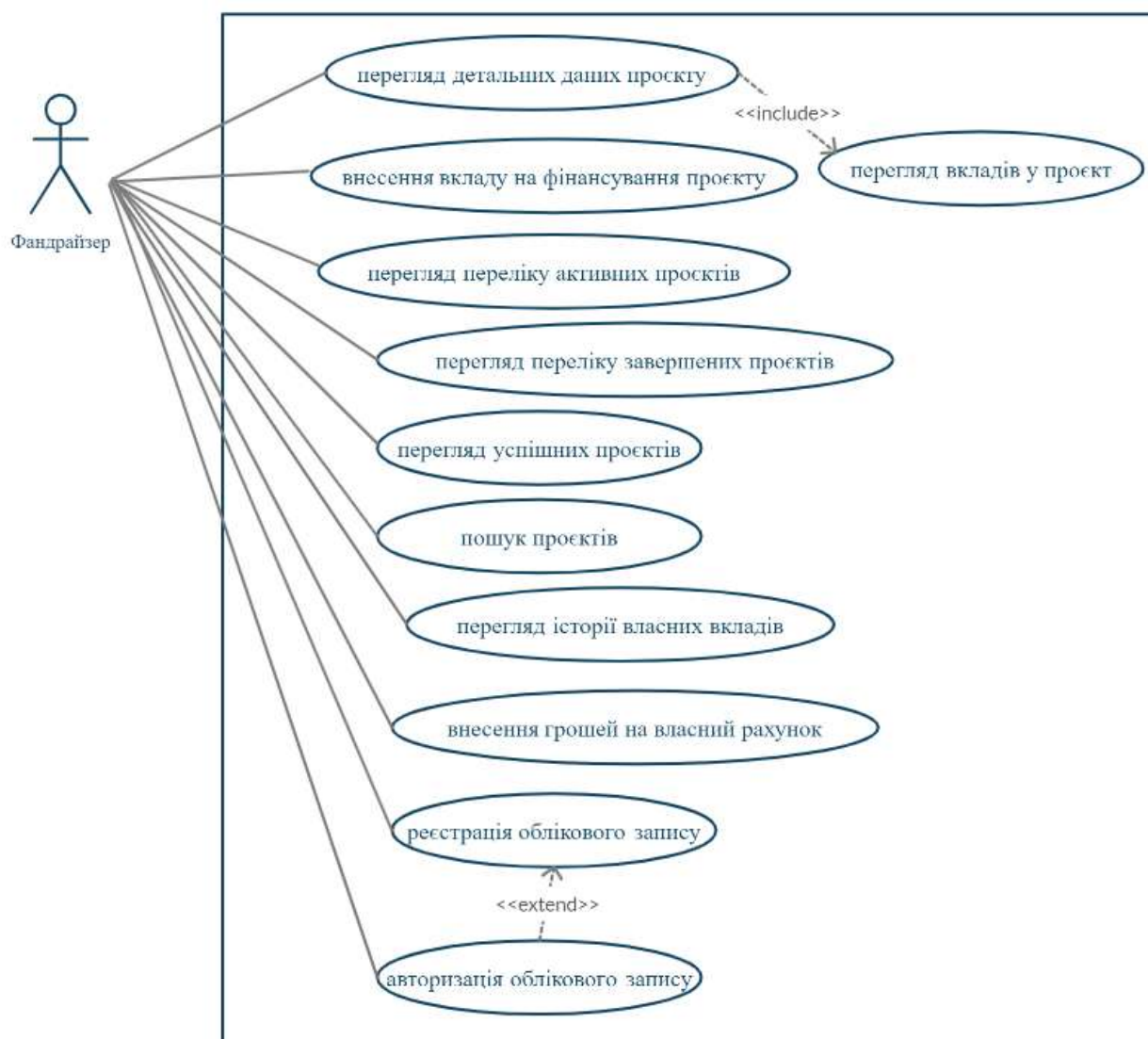


Рисунок 3.3 – Діаграма прецедентів для роботи фандрайзера з системою

3.3 Проєктування бази даних

Для роботи з даними було виконано проєктування бази даних.

Основними сутностями бази даних (рис. 3.4) є:

- project – таблиця, в якій містяться дані про всі проєкти, фандрайзинг яких виконується;
- user – користувачі, які взаємодіють з вебзастосунком;
- contributors – учасники проєктів;
- payment – транзакції з перелічення коштів користувачами на проєкти.

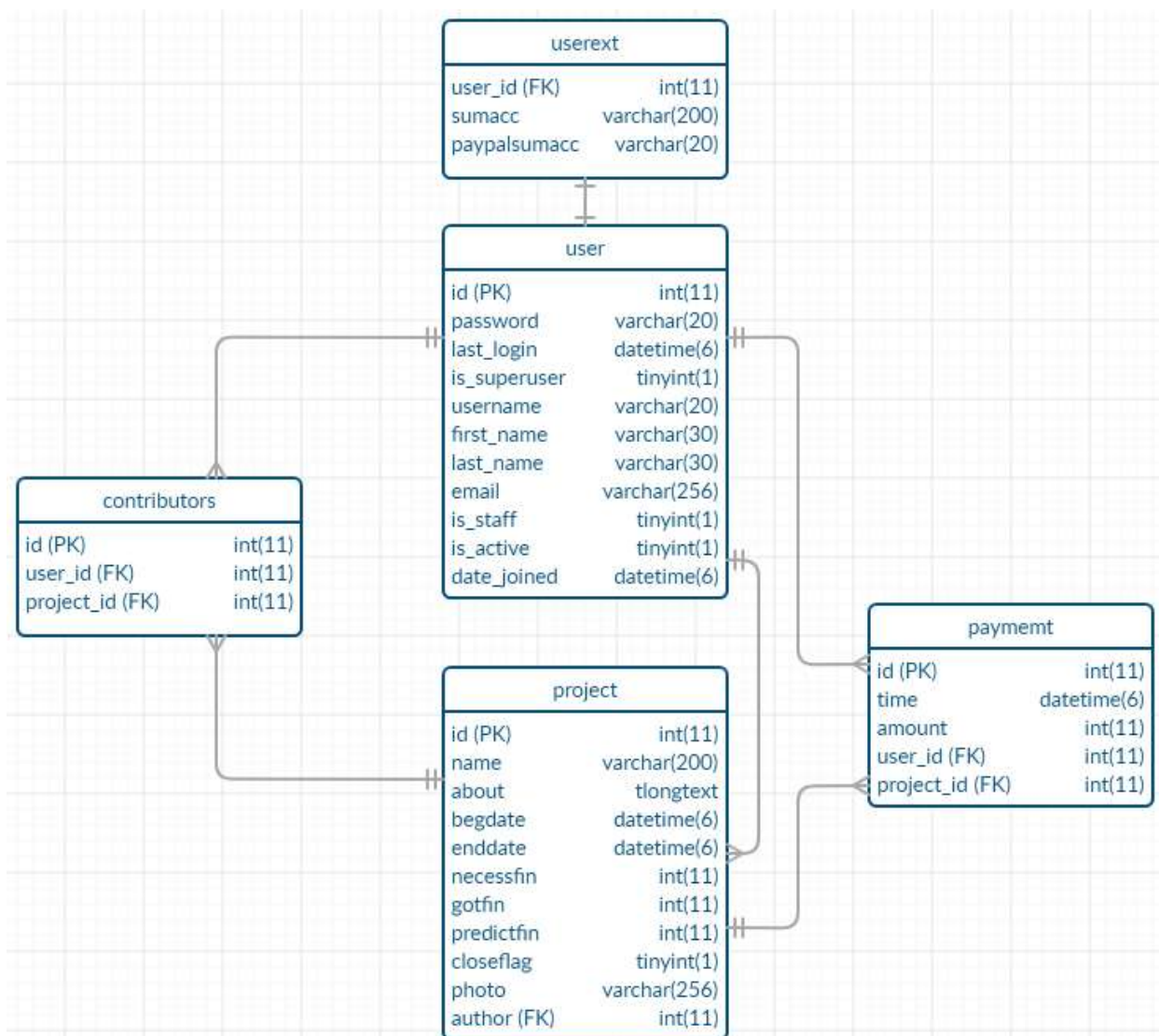


Рисунок 3.4 – Схема бази даних

Таблиця project містить в якості атрибутів:

- id – номер проєкту;
- name – назва проєкту;
- about – опис проєкту;
- begdate – дата створення проєкту;
- enddate – дата завершення збору коштів на проєкт;
- necessfin – сума, необхідна для проєкту;
- gotfin – зібрані кошти на проєкт;
- closeflag – активність проєкту (продовжується збір коштів або ні);
- author_id – користувач, який створив проєкт;

– predictfin – прогнозована сума коштів, які вдасться зібрати на проєкт (результат роботи запропонованого методу).

Таблиця user містить в якості атрибутів:

- id – номер облікового запису;
- password – пароль;
- last_login – час і дата останньої авторизації;
- is_superuser – адміністратор або ні;
- username – логін;
- first_name – ім'я;
- last_name – прізвище;
- email – електронна пошта;
- is_staff – менеджер вебзастосунку або звичайний користувач;
- is_active – активність облікового запису (можна авторизуватися або ні);
- date_joined – час і дата реєстрації.

Таблиця contributors містить в якості атрибутів:

- id – номер запису;
- project_id – номер проєкту;
- user_id – обліковий запис користувача.

Таблиця payment містить в якості атрибутів:

- id – номер транзакції;
- time – час виконання транзакції;
- project_id – номер проєкту;
- user_id – обліковий запис користувача;
- amount – сума коштів.

3.4 Висновки за розділом 3

За результатами виконання третього розділу було повністю виконано проєктування програмного забезпечення у вигляді вебзастосунку. Результати

включають в себе наступні:

- обрано мову програмування Python;
- визначено функціональні вимоги до програмного забезпечення;
- виконано проєктування бази даних на основі використання системи керування базами даних SQLite.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Структура розробленої програмної системи

Програмну систему було реалізовано на основі об'єднання вебзастосунку з підсистемою прогнозування результатів фандрайзингу проєктів.

Підсистема прогнозування результатів фандрайзингу проєктів реалізує запропонований в роботі методі, в основі свого функціонування використовує бібліотеку Keras.

Вебзастосунок відповідно є реалізацією на основі використання вебфреймворку Django та мови програмування Python.

Відповідну діаграму розгортання представлено на рис. 4.1.

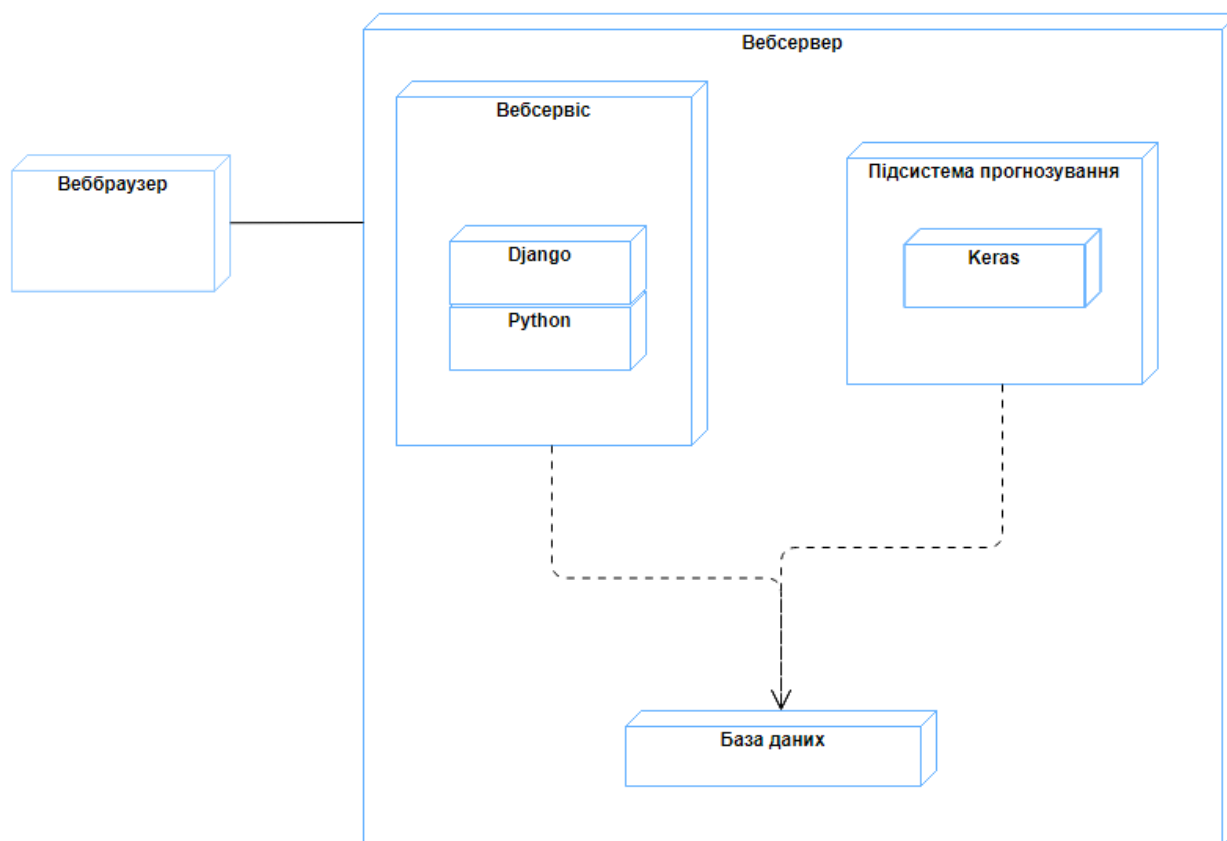


Рисунок 4.1 – Діаграма розгортання

Вебсервіс та підсистема на вебсервері функціонують окремо, але взаємодія між ними відбувається через базу даних. Підсистема результати прогнозування, отримані кожен день, зберігає в базу даних, а вебсервіс запитує їх з бази даних та виводить автору проєкту.

Структура вебзастосунку в результаті включає в себе наступні файли та директорії:

- manage.py – файл, в якому міститься скрипт, за допомогою якого реалізується керування вебзастосунком (створення застосунку, робота з базою даних, запуск серверу);

- db.sqlite3 – файл бази даних, керування якою відбувається за допомогою системи SQLite;

- ProjectFundraiser – директорія з файлами веб-застосунку.

У свою чергу директорія ProjectFundraiser містить наступні файли:

- _init_.py – файл, який необхідний для ідентифікації директорії в якості модуля Python;

- admin.py – файл скрипту адміністративної панелі, в якому реалізується реєстрація моделей вебзастосунку;

- apps.py – файл скрипту, що реєструє застосунок в Django;

- models.py – файл скрипту з моделями вебзастосунку;

- settings.py – файл скрипту, що визначає налаштування вебзастосунку;

- urls.py – файл скрипту, що визначає асоціації URL-адрес з представленнями;

- views.py – файл скрипту з представленнями вебзастосунку.

Окрім того до директорії ProjectFundraiser входить директорія templates. До її складу в свою чергу входять зокрема наступні файли:

- history.html – вебсторінка перегляду історії вкладів;

- index.html – базова вебсторінка, яка забезпечує доступ до переліку проєктів;

- login.html – вебсторінка авторизації;

- projects.html – вебсторінка перегляду переліку проєктів;
- project.html – вебсторінка перегляду детальних даних про проєкт;
- upload.html – вебсторінка внесення коштів на рахунок користувача;
- blocks – директорія, що містить вебсторінки, які визначають блок меню (menu.html), блок заголовку (head.html) та нижній блок сторінки разом з пошуком (footer.html).

Відповідно підсистема прогнозування результатів фандрайзингу проєктів побудована на основі наступних модулів:

- PFpreprocess.py – файл, де реалізується підготовка даних для навчання та отримання результату прогнозу;
- PFPredictionModel.py – файл, який реалізує створення і за необхідності навчання моделі в межах запропонованого методу прогнозування результатів фандрайзингу проєктів;
- PFautoloader.py – файл, який забезпечує подовбу ініціалізацію роботи запропонованого методу на основі витягання даних з бази даних та їх підготовки засобами PFpreprocess.py та безпосередньо прогнозування засобами PFPredictionModel.py.

4.2 Основні рішення щодо реалізації вебзастосунку

В основі побудови вебзастосунку лежать розроблені моделі, класи яких перетворюються на таблиці бази даних. Усі моделі знаходяться в модулі models.py.

Під час визначення моделі кожна з них наслідується від класу models.Model.

Відповідно створення полів виконується шляхом визначення типу, до якого належить ця характеристика (наприклад, IntegerField визначає поле з цілим значенням, DateTimeField – поле з датою і часом).

Для створення зовнішніх ключів призначений ForeignKey, який визначає зв'язок з відповідною таблицею, а також спосіб вилучення пов'язаних даних.

Зокрема створення моделі платежів, що визначають окремі вклади в проєкти, відбувається наступним чином:

```
class Payment(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    project = models.ForeignKey(Project, on_delete=models.CASCADE)
    amount = models.IntegerField(default=0)
    time = models.DateTimeField()
```

Для створення бази даних та відповідних таблиць потрібно виконати послідовно наступні команди:

```
python /projectfundraiser/manage.py makemigrations
python /projectfundraiser/manage.py migrate
```

У модулі urls.py визначено зв'язок URL-адрес з відповідними представленнями вебзастосунку, зокрема адреси з додаванням «/success» будуть призводити до отримання результату, який забезпечується представленням success:

```
url(r'^success', views.success, name='success')
```

Для перегляду історії платежів також вказується номер облікового запису користувача до «/history», відповідний шаблон виглядає наступним чином:

```
url(r'^history/(?P<userid>[0-9]+)/$', views.history, name='history')
```

Для забезпечення цих зв'язків у модулі `views.py` було реалізовано наступні представлення, кожне з яких визначається відповідною функцією.

Представлення `index(request)` визначає представлення, яке виконується за замовчуванням (при введенні адреси вебсервера). При цьому виконується переадресація на сторінку, що формується з використанням шаблону `index.html`.

Представлення `success(request)` призводить до формування переліку успішних проєктів. Для цього спочатку визначається перелік таких проєктів шляхом відфільтровування з загального набору проєктів тих, які зібрали під час фандрайзингу більшу суму за заплановану:

```
projectss = Project.objects.filter(necessfin__lte=F('gotfin'))
```

Далі відповідно повертається шаблонна сторінка `projectfundraiser/projects.html`, яка дозволяє заповнити набір проєктів будь-яким набором (у даному випадку успішних), а також передається назва «успішних» для формування назви сторінки «Перелік успішних проєктів»:

```
return render(request, 'projectfundraiser/projects.html', {'projects':  
projectss, 'info': 'успішних'})
```

Представлення `active(request)` працює шляхом відбору проєктів, дата завершення фандрайзингу для яких не пізніша за поточну та для яких не визначено користувачем прапор завершення. Окрім того до кожного проєкту набору додається ще властивість `togo`, у якій визначається, скільки саме днів лишилося до завершення фандрайзингу. Вся ці інформація разом з часиною назви, що відповідає успішним проєктам, передається в шаблон з переліком проєктів.

Представлення `closed(request)` формує перелік завершених проєктів, тобто проєктів для яких дата завершення прострочена або для яких вже визначено прапор завершення.

Представлення `search(request)` забезпечує отримання результатів пошуку, витягаючи параметри пошуку з запиту `request`, переданих з форми методом POST. При цьому пошук відбувається за введеним запитом, який перевіряється на вміст в назві об'єкта описі проєкту. В результаті також повертається сторінка `projects.html`, але перелік проєктів заповнений результатами пошуку, а до назви додається відповідна частина.

Представлення `history(request, userid)` призначено для заповнення платежів, які було виконано заданим користувачем `userid`. Результат повертається через шаблон `history.html`, при цьому в нього передається перелік платежів, для кожного з яких додатково визначається назва проєкту, фінансування якого він виконував:

```
payments = Payment.objects.filter(user=userid)
projects = Project.objects.order_by('-begdate')
tr = []
for t in payments:
    for p in projects:
        if p.id==t.project_id:
            t.name=p.name
    tr.append(t)
return render(request, 'projectfundraiser/history.html', {'payments': tr})
```

Представлення `own(request, userid)` працює з переліком проєктів, аналогічно описаним вище представленням, заповнюючи його переліком проєктів, які було створено користувачем `userid`.

Представлення `project(request, projectid)` дозволяє отримати детальну інформацію за проєктом з номером `projectid`, повертаючи всю отриману

інформацію шляхом заповнення даними шаблону `project.html`. До даного шаблону передається інформація про проєкт, вклади, які було виконано в цей проєкт, а також учасників проєкту. Для кожного вкладу (платежу) додатково додається інформація про те, користувачем з яким логіном було виконано даний вклад.

Представлення `finance(request, userid, projectid)` реалізує внесення вкладу, розмір якого передається через дані запиту `request` з відповідної форми методом POST, на рахунок проєкту `projectid` користувачем `userid`. Для цього реалізується перевірка, що грошей достатньо на рахунку користувача в застосунку, після чого виконується занесення відповідних даних. В результаті повертається оновлена сторінка деталізованого перегляду даних проєкту за запитом `/project/id`.

Представлення `close_project(request, project_id)` забезпечує закриття фандрайзингу проєкту з номером `project_id`. Виконання цієї дії можливо тільки у випадку, якщо на рахунку проєкту вже зібрано суму не меншу за очікувану. У результаті також повертається оновлена сторінка деталізованого перегляду даних проєкту за запитом `/project/id`.

Представлення `load(request)` виконує переведення заданої суми через метод POST з відповідної форми з рахунку PayPal на рахунок користувача у застосунку. Для реалізації саме переведення використовується реалізований на основі модуля `django-paypal` модуль `paypalpay`.

Представлення `logout_route(request)` використовується для виходу з власного облікового запису з перенаправленням на базову сторінку.

Представлення `login_route(request)` забезпечує авторизацію облікового запису на основі переданих методом POST параметрів облікового запису.

В якості шаблону для створення HTML-сторінок використано вільний шаблон `It Next` [20].

Шаблон `projects.html` дозволяє відтворити перелік проєктів у вигляді вебсторінки. Для цього забезпечується динамічне формування сторінки шляхом підставлення даних, які було отримано з представлення. Наприклад,

формування назви сторінки відбувається шляхом включення частини назви, яка визначає з якого саме представлення було отримано перелік проєктів:

```
<h1 class="namewebpage">Перелік {{info}} проєктів</h1>
```

Фрагмент заповнення даними проєктів:

```
{% if projects %}
{% for project in projects %}
<!--елементи HTML -->
<div class="center">
    <h4><a href="/project/{{ project.id }}">{{ project.name }}</a></h4>
</div>
{% endfor %}
{% else %}
<p>
    На даний момент проєкти в цій категорії відсутні.
</p>
<!--елементи HTML -->
{% endif %}
```

4.3 Основні рішення щодо реалізації підсистеми прогнозування результатів фандрайзингу проєктів

Центральною одиницею організації підсистеми прогнозування результатів фандрайзингу проєктів є модуль `PFPredictionModel.py`, який відповідає за створення моделі на основі нейронної мережі LSTM, що передбачено в запропонованому методі.

В основі реалізації модуля функція `getLSTMmodel(x = [], y = [])`, яка за результатами своєї роботи повертає навчану модель. Параметри `x` та `y`

визначають вхідну та вихідну навчальну вибірки, які використовуються для навчання створеної моделі, якщо її не було створено і навчано раніше. Якщо модель вже була створена і навчана, то потрібно передати порожні значення параметрів, в результаті параметри нейронної мережі будуть завантажені з файлів і отримана таким чином модель буде повернена за результатами роботи функції.

Безпосередньо створення моделі було реалізовано на основі двох LSTM-шарів, кожен з яких складається з 4 блоків. До кожного шару для уникнення перенавчання застосовується шар Dropout, на якому буде відкинуто 20 %. Після цього застосовується Dense-шар, який створює вихід, що складається з 1 вихідного блоку, в якому міститься значення зібраної за фандрайзинговою компанією суми. В результаті код, який відповідає за описану логіку:

```
model = Sequential()
model.add(LSTM(4, return_sequences=True, input_shape=(7, 1)))
model.add(Dropout(0.2))
model.add(LSTM(4))
model.add(Dropout(0.2))
model.add(Dense(1))
```

Під час навчання обчислюється середньо-квадратична похибка, навчання реалізується протягом 1000 епох, застосовується оптимізатор Адама:

```
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x, y, epochs=1000, batch_size=8, verbose=2)
```

Відповідно якщо вже збережена навчана модель, то виконується завантаження ваг і інших параметрів моделі з відповідних файлів.

В модулі PFpreprocess.py виконується передоброблення даних шляхом їх нормалізації, а також формування наборів даних для навчання мережі, після чого виконується виклик модуля створення моделі з передачею отриманих значень у функцію та збереженням моделі.

В модулі PFautoloader.py забезпечується отримання даних з бази даних, після чого відбувається передоброблення цих даних, отримання моделі та передача їх в модель для розрахунку вихідного значення, тобто сумарного обсягу зібраних коштів на фінансування проєкту.

4.4 Висновки за розділом 4

За результатами виконання четвертого розділу було описано структуру створеної програмної системи разом з виділенням реалізованих програмних одиниць, окрім того детально описано процес створення вебзастосунку в межах програмної системи та підсистеми прогнозування результатів фандрайзингу проєктів, що реалізує представлений в роботі метод.

5 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

5.1 Призначення програми

Програмне забезпечення призначено для виконання фандрайзингу проєктів. Для цього звичайні користувачі можуть переглядати наявні проєкти та виділяти кошти з власного рахунку на їх фінансування. Відповідно автори проєктів можуть вносити дані щодо власного проєкту, спостерігати за збором коштів, отримувати прогноз за результатами збору коштів за перший тиждень загальних результатів збору коштів.

Для зручності взаємодії різних користувачів та різних типів користувачів програмне забезпечення створене у вигляді вебзастосунку.

5.2 Умови виконання програми

Для клієнтської сторони мінімальними вимогами для забезпечення звернення до серверу є вимоги, які висуваються до роботи сучасних браузерів.

Технічні засоби, наявність яких має бути обов'язково забезпечена на сервері:

- процесор, який має щонайменше 4 ядра та частоту не меншу за 3,1 ГГц (або можливість прискорення до 3,1 ГГц);
- оперативна пам'ять обсягом не менше ніж 8 Гб;
- жорсткий диск загальним обсягом не менше ніж 1 Тб, а з обсягом вільного простору не менше ніж 10 Гб;
- монітор та відеокарта з роздільною здатністю не менше ніж 1920x1080.

Окрім того на сервері має бути обов'язково встановлено наступні програмні бібліотеки та модулі:

- django;
- keras;

- django-paypal і sched;
- python.

5.3 Виконання програми

Для запуску на виконання вебзастосунку на сервері потрібно виконати наступну команду:

```
python ProjectFundraiser runserver
```

Далі в браузері потрібно ввести назву вебсервера, виконуючи таким чином до нього звернення, тоді з'явиться базова сторінка вебзастосунку (рис. 5.1).

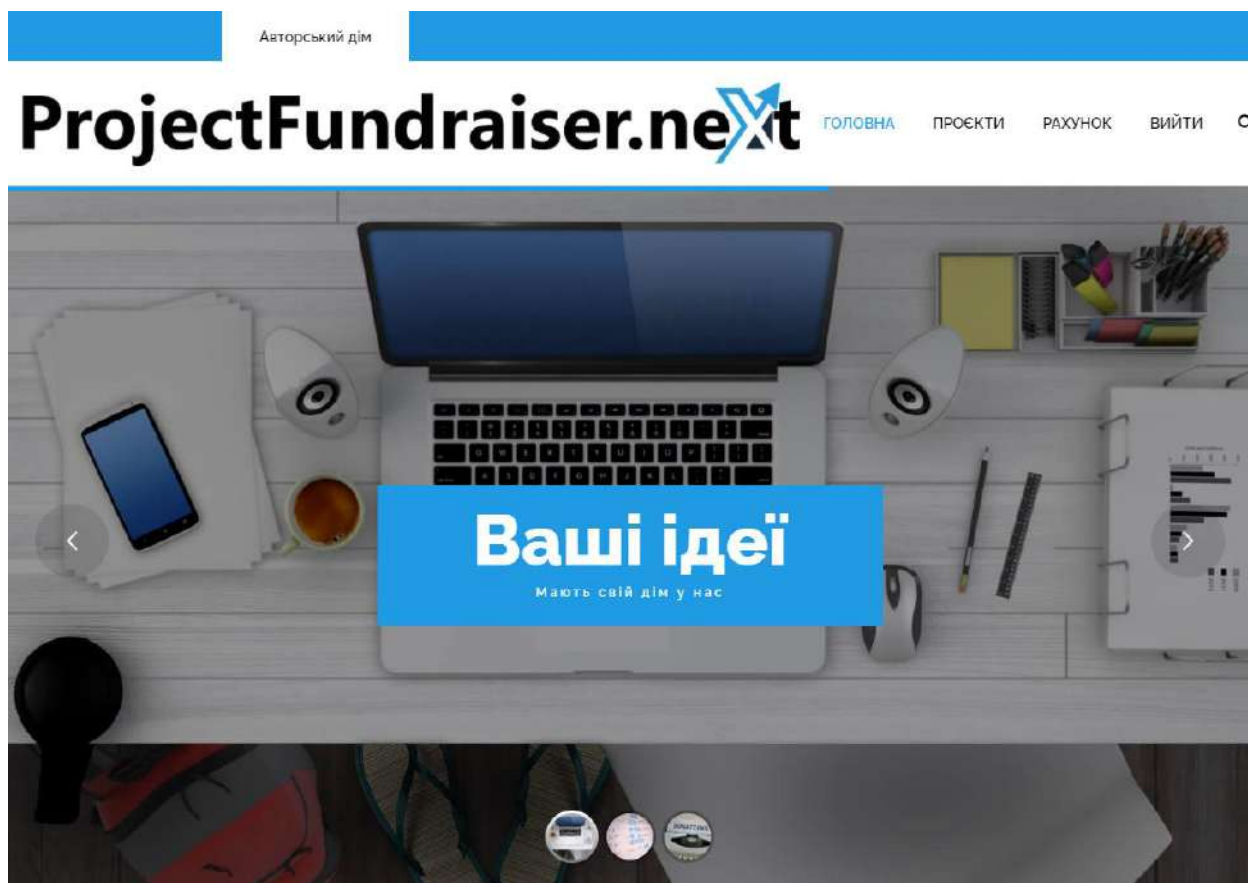


Рисунок 5.1 – Головна сторінка вебзастосунку

Окрім каруселі з зображеннями дана сторінка містить також меню, що складається з таких пунктів:

- «Головна» – базова сторінка вебзастосунку (рис. 5.1);
- «Проекти» – меню, що містить підпункти, і забезпечує доступ до переліку проектів;
- «Рахунок» – меню, що містить підпункти, і визначає операції з фінансовим рахунком користувача;
- «Увійти/Вийти» – пункт меню, що дозволяє вийти з авторизованого облікового запису або увійти в нього, якщо користувач не авторизований.

Пункт меню «Проекти» складається з наступних підпунктів:

- «Активні» – містить всі проекти, збір коштів на які ще не було завершено на поточний момент;
- «Завершені» – містить всі проекти, збір коштів на які було завершено автором проекту або для яких вийшов дедлайн;
- «Успішні» – містить всі проекти, фандрайзинг яких було успішно завершено (зібрана сума не менша за заплановану);
- «Власні» – містить всі проекти, які було створено авторизованим в даний момент користувачем (або не відображається, якщо користувач не авторизований).

Окремим пунктом меню є пункт «Авторський дім», роботу якого розглянуто пізніше.

Приклад вигляду сторінки, що відображається за підпунктом меню «Активні», приведено на рис. 5.2. Для кожного проекту відображається його зображення, назва, наводиться зібрана на фінансування сума та запланована сума, а також кількість днів, що лишилась до завершення фандрайзингу проекту.

Окрім того рядок для пошуку (позначений міткою «Пошук») дозволяє одразу знайти потрібний проект.

Авторський дім

ProjectFundraiser.net

ГОЛОВНА

ПРОЄКТИ

РАХУНОК

ВИЙТИ

Перелік активних проєктів

Головна

>

Проекти



Моніторинг якості повітря в Запоріжжі

зібрано \$0 з \$800

залишилось днів: 41



Створення Центру звуку

зібрано \$765 з \$8000

залишилось днів: 1



Запахи на дотик

зібрано \$98 з \$7000

залишилось днів: 56

Пошук

Шукати



Захисні щитки для медиків

зібрано \$812 з \$5000

залишилось днів: 1



Простори для птахів

зібрано \$470 з \$450

залишилось днів: 7

Рисунок 5.2 – Перелік активних проєктів

На рис. 5.3 наведено приклад сторінки з відображенням переліку успішних проєктів. Ця сторінка дозволяє ознайомитись з успішними заявками на фандрайзинг.

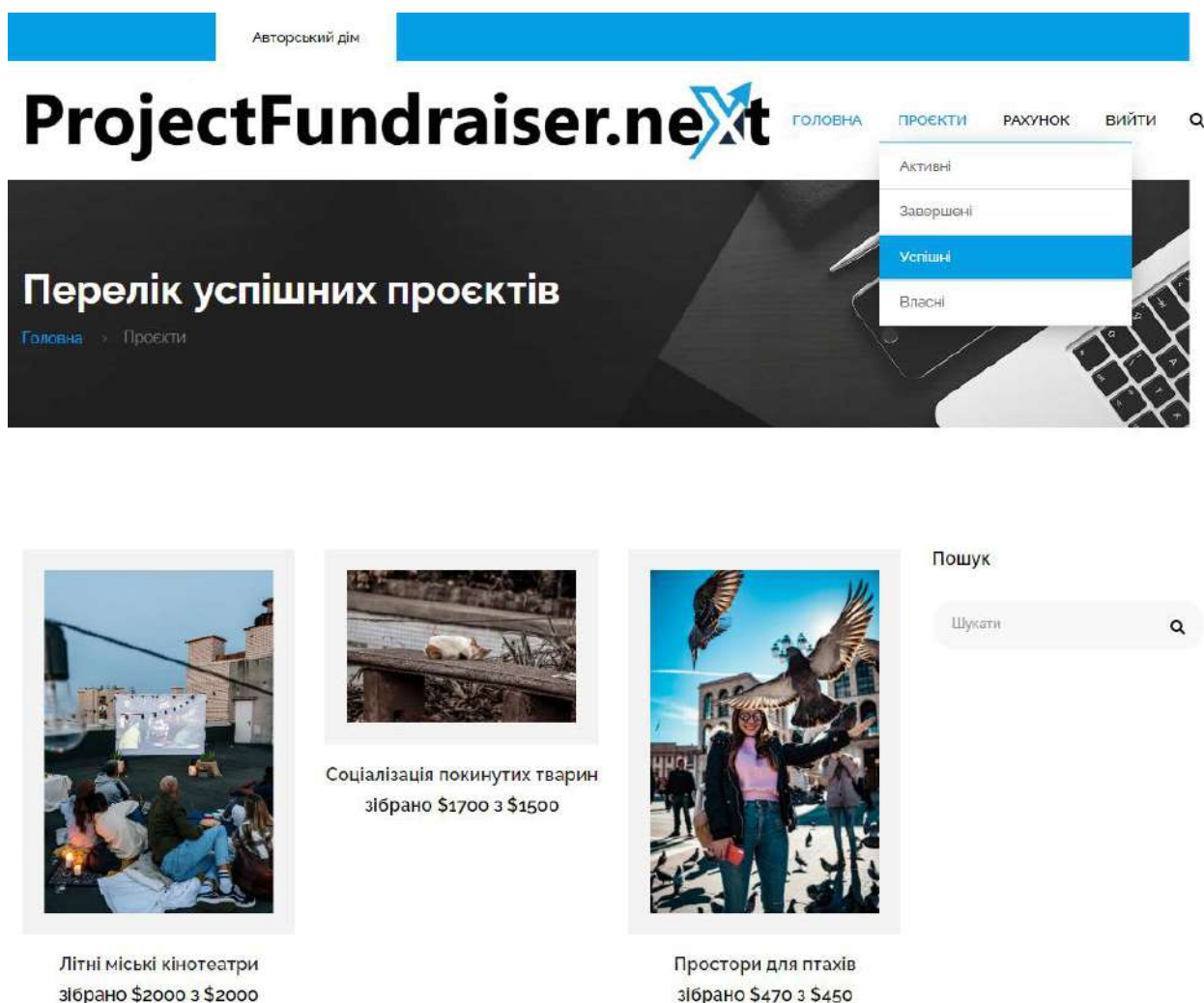


Рисунок 5.3 – Перелік успішних проєктів

З будь-якого переліку проєктів (успішних, активних, власних, завершених) можна обрати один з проєктів (натиснути на його назву), після чого відкриється сторінка з переглядом детальних даних обраного проєкту (рис. 5.4).

Дана сторінка включає зображення, прив'язане до проєкту, назву проєкту, інформацію про стан проєкту (у випадку рис. 5.4 проєкт завершено), дату і час завершення фандрайзингу проєкту, зібрану суму на проєкт, заплановану суму, а також 3 вкладки:

- «Опис» – містить текст опису проєкту;
- «Учасники» – містить перелік учасників проєкту (ті, хто можуть працювати над внесенням даних проєкту);
- «Внески» – містить перелік внесків на фінансування проєкту.



Літні міські кінотеатри

Проект вже завершено

Дата завершення: Трав. 14, 2020, 12:00

Зібрано \$2000 з \$2000

Опис Учасники Внески

Літні міські кінотеатри є однією з точок розвитку культурних подій в місті. Такі точки не потребують великого фінансування, при цьому дозволяють достатньо легко привертати увагу, оскільки таку точку можна розмістити тимчасово на майже будь-якій площі міста, на відміну від побудови стабільних літніх театрів, як це виконувалося декілька десятиків років тому. Для демонстрації можуть використовуватися фільми з безкоштовним показом, відповідно витрати на один такий проєкт достатньо низькі. На першому етапі заплановано створення трьох літніх міських кінотеатрів в місті Запоріжжя. Кожен кінотеатр передбачає закупівлю проєктору, звукового обладнання, стільців та створення фонду кінофільмів.

Рисунок 5.4 – Перегляд детальних даних завершеного проєкту

У випадку рис. 5.4 проєкт вже завершений, тобто за ним приводиться вся фактична інформація.

Якщо проєкт активний, користувач авторизований, то він може внести вклад на фінансування проєкту (рис. 5.5).



Пошук

Шукати



Моніторинг якості повітря в Запоріжжі

Фандрайзинг активний

Дата завершення: Груд. 25, 2020, 10:00

Зібрано \$10 з \$800

1

ВНЕСТИ СУМУ

Опис

Учасники

Внески

Контроль якості повітря має бути в руках громади.

Ми плануємо встановити станції контролю якості повітря в місті Запоріжжя, які дозволять визначати поточне забруднення, при цьому робити це будь-якої миті зі впевненістю в отриманих результатах. Планом проєкту передбачено встановлення 6 таких станцій в різних районах міста.

Окрім того буде розроблено програмну систему, що дозволить спостерігати за даними онлайн, а також зберігати історію даних.

Відслідковування за стаціонарними точками дозволить визначати вплив подій, які відбуваються в місті, а також просто мати впевненість в якості повітря під час прогулянки.

Рисунок 5.5 – Перегляд детальних даних активного проєкту

Якщо дедлайн ще не вийшов, а зібрана сума коштів вже більша ніж запланована, то сам автор може компанію з фандрайзингу завершити (рис. 5.6).

Авторський дім

ProjectFundraiser.next

ГОЛОВНА

ПРОЕКТИ

РАХУНОК

ВИЙТИ

Детальніше про проєкт

Головна

>

Детальніше



Пошук

Шукати

Моніторинг якості повітря в Запоріжжі

Фандрайзинг активний

Дата завершення: Груд. 25, 2020, 10:00

ЗАВЕРШИТИ ЗБІР

Зібрано \$810 з \$800

Очікувана сума: 904\$

1

ВНЕСТИ СУМУ

Опис

Учасники

Внески

Контроль якості повітря має бути в руках громади.

Ми плануємо встановити станції контролю якості повітря в місті Запоріжжя, які дозволять визначати поточне забруднення, при цьому робити це будь-якої миті зі впевненістю в отриманих результатах. Планом проєкту передбачено астанавлення 6 таких станцій в різних районах міста.

Окрім того буде розроблено програмну систему, що дозволить спостерігати за даними онлайн, а також зберігати історію даних.

Відслідковування за стаціонарними точками дозволить визначати вплив подій, які відбуваються в місті, а також просто мати впевненість в якості повітря під час прогулянки.

Рисунок 5.6 – Перегляд детальних даних активного успішного проєкту

Для завершення збору призначена кнопка «Завершити збір», яка дозволяє перетворити проєкт на проєкт зі статусом «завершений», після цього кнопка «Внести суму» разом зі всією відповідною формою стає неактивною для авторизованих користувачів.

Якщо на сторінку деталізованого перегляду інформації про проєкт заходить його автор, при чому це відбувається після семи перших днів збору коштів, то для нього відображається також прогноз зібраної суми коштів на фінансування проєкту. Для цього під зібраною сумою призначено поле «Очікувана сума». Розрахунок очікуваної суми відбувається кожного дня для всіх проєктів, для яких поточний день був сьомим у тривалості фінансування.

Розрахунок відбувається за допомогою скрипта, який автоматизує запуск скрипта розрахунку. Скрипт виконується кожного дня о 00:00.

Для того щоб запустити в системі скрипт перший раз, виконується команда:

```
python PFautoloader.py
```

Після цього звернення буде виконуватися кожного дня. Для реалізації автозвернень використовується модуль sched.

Вкладка «Внески» дозволяє переглянути суми та авторів внесків на фінансування проєкту, які вже були виконані (рис. 5.7).

Підпункти меню «Рахунок» реалізують звернення до власного рахунку.

Підпункт «Поповнити» дозволяє переглянути суму на рахунку користувача в застосунку, яку він використовує для фінансування, та перевести з власного PayPal-рахунку суму на рахунок у даному вебзастосунку.

Підпункт «Вклади» дозволяє переглянути, які проєкти та в яку мить, авторизований користувач вже фінансував раніше.

Авториський дім

ProjectFundraiser.next

ГОЛОВНА

ПРОЕКТИ

РАХУНОК

ВИЙТИ

Детальніше про проєкт

ГОЛОВНА

Детальніше



Пошук

Шукати

Створення Центру звуку

Фандрайзинг активний

Дата завершення: Лист. 16, 2020, 12:00

Зібрано \$865 з \$8000

Очікувана сума: 923\$

1

ВНЄСТИ СУМУ

Опис

Учасники

Внески

SERHII	Жовт. 10, 2020, 18:29	100\$
BEK	Жовт. 17, 2020, 14:53	55\$
ANDREW	Жовт. 19, 2020, 13:17	12\$
USER11	Жовт. 21, 2020, 18:07	80\$
IGOR	Жовт. 25, 2020, 22:07	90\$
ANDREW	Жовт. 28, 2020, 11:13	100\$
LUGO	Лист. 1, 2020, 19:11	150\$
PETER	Лист. 6, 2020, 13:18	80\$
USER12	Лист. 9, 2020, 11:44	100\$
PLAYER	Лист. 14, 2020, 12:17	138\$

Рисунок 5.7 – Перегляд внесків на фінансування проєкту

На рис. 5.8 приведено приклад перегляду власних вкладів у проекти.

На рис. 5.9 наведено форму визначення параметрів пошуку проектів.

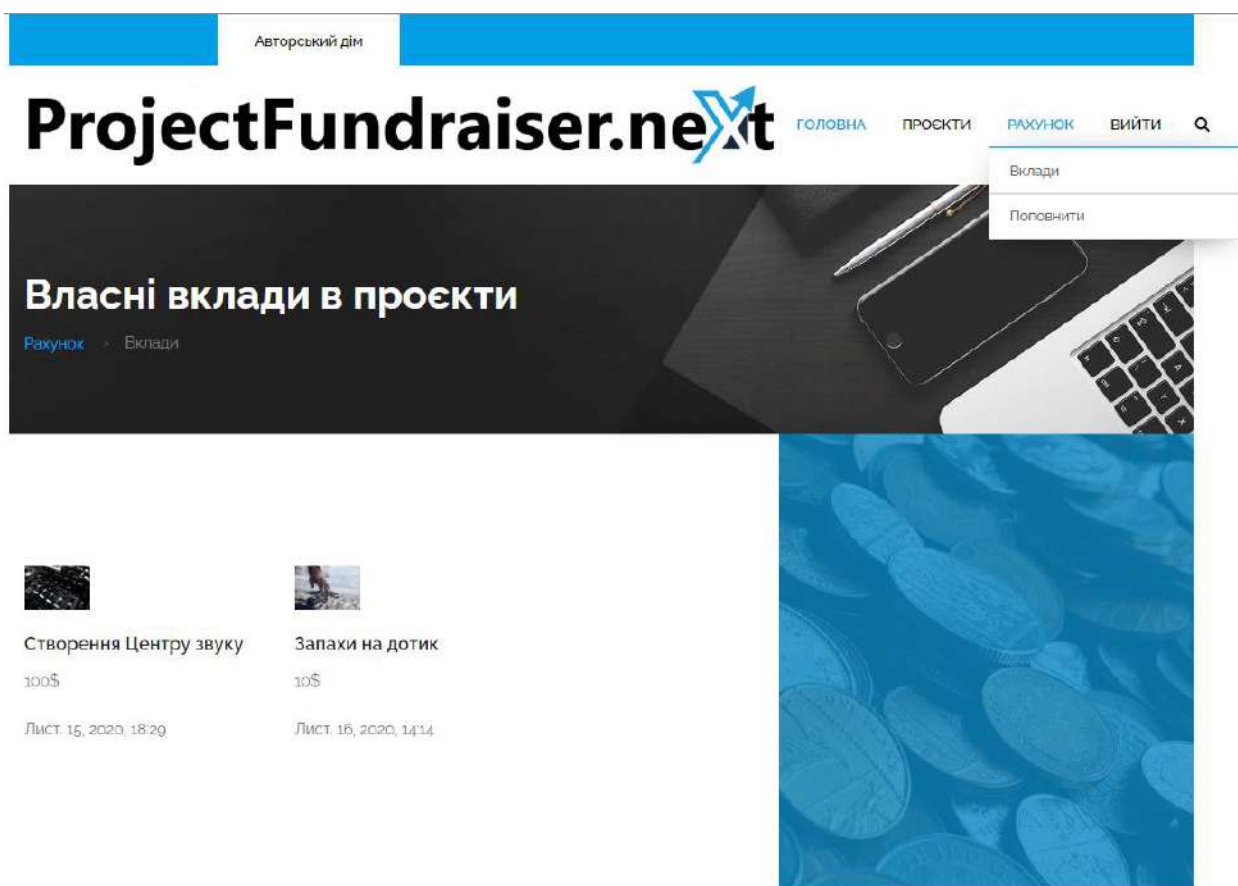


Рисунок 5.8 – Перегляд власних вкладів в проекти

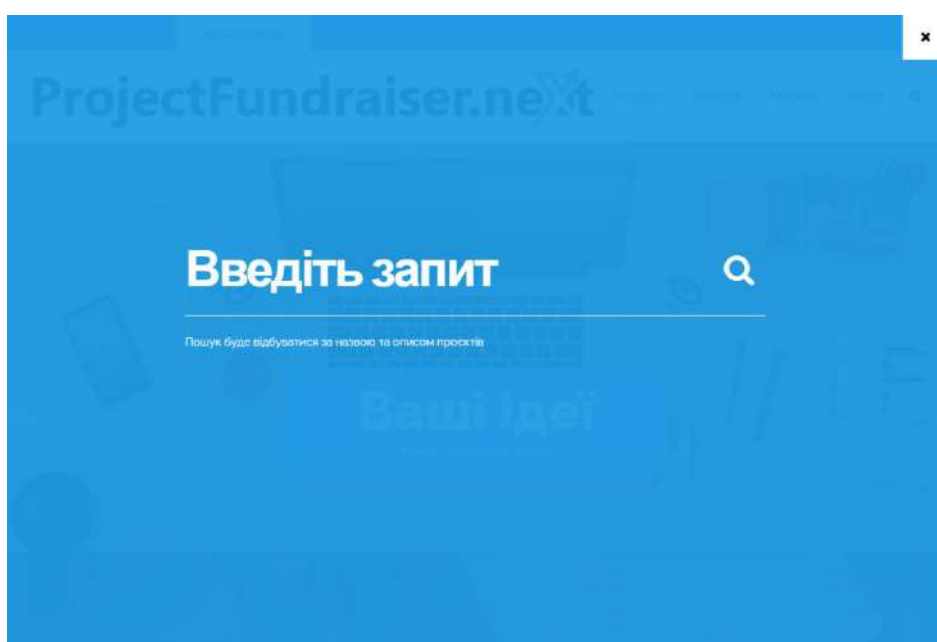


Рисунок 5.9 – Пошук проектів

Дана форма виводиться під час вибору іконки лупи з меню вебзастосунку.

Пошук також може відбуватися за рахунок поля «Пошук» зі всіх варіантів сторінок перегляду переліку проєктів, а також деталізованого перегляду даних проєкту.

На рис. 5.10 наведено приклад видачі результатів пошуку за запитом «Запахи».

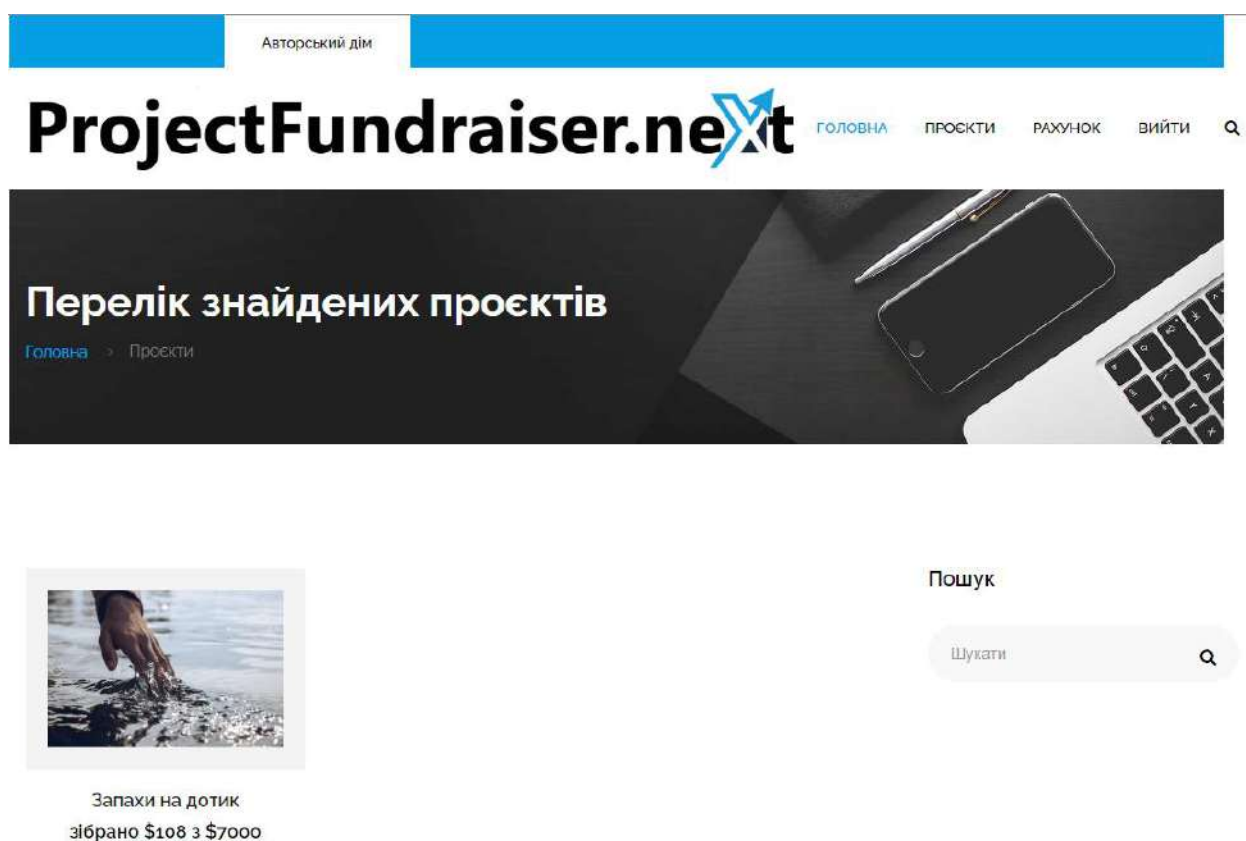


Рисунок 5.10 – Результати пошуку

При відкритті розділу «Авторський дім» відкривається адміністративна панель, де необхідно авторизуватися, якщо користувач не увійшов у свій адміністраторський або авторський обліковий запис.

На рис. 5.11 приведено приклад адміністративної панелі для користувача, який має частину адміністраторських функцій, але для нього заборонено дії з транзакціями. Тобто він може виконувати керування

обліковими записами користувачів, керувати власними проектами, але не може керувати фінансовими операціями.



Рисунок 5.11 – Адміністративна панель

За допомогою даного меню можна створювати нові облікові записи, проекти, а також змінювати існуючі. Відповідно автор проектів має доступ тільки до власних проектів та створення нових проектів.

Під час створення проекту (рис. 5.12) автор має вказати:

- назву проекту в поле «Name»;
- опис проекту в поле «About»;
- дату і час запуску фандрайзингу в поле «Creation date»;
- дату і час завершення фандрайзингу в поле «Enddate»;
- суму, яку необхідно зібрати, в поле «Total sum»;
- поточну зібрану суму в поле «Gathered sum» (якщо відбувався збір коштів іншими засобами попередньо);
- стан проекту в поле «Is closed» (таким чином можна звершити проект, призначено для редагування проектів через панель адміністратора);
- власника проекту в поле «Author» (якщо проект створює не сам автор);
- учасників проекту в списку «Contributors» (можна обрати декількох), що визначає, хто з користувачів може змінювати дані проекту окрім автора;
- зображення в поле «Photo».

Django administration
WELCOME, SERHII. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Kick > Projects > Add project

Add project

Name:

About:

Begdate:
 Date: 2020-10-18 Today
 Time: 13:46:09 Now

Enddate:
 Date: 2020-11-18 Today
 Time: 14:00:00 Now

Necessfin:

Gotfin:

Predictfin:

☐ Closeflag

Author: serhii

Contributors:
 serhii

Hold down "Control", or "Command" on a Mac, to select more than one.

/static/projectfundraiser: Файл не вибрано

Рисунок 5.12 – Створення проєкту

На рис. 5.13 продемонстровано редагування прав облікового запису. При цьому можна визначити, які саме дії користувач зможе виконувати у вебзастосунку.

Django administration
WELCOME, SERHII / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home / Authentication and Authorization / Users / serhii

Change user
HISTORY

Username: serhii
Required. 30 characters or fewer. Letters, digits and @/./+/-/_ only.

Password: algorithm: pbkdf2_sha256 iterations: 24000 salt: XWRAZr***** hash: d2Dnb0*****
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name:
Last name:
Email address:

Permissions

☒ Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☒ Staff status
Designates whether the user can log into this admin site.

☒ Superuser status
Designates that this user has all permissions without explicitly assigning them.

Groups:

Available groups ?

Choose all

Chosen groups ?

Remove all

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

Available user permissions ?

Choose all

Chosen user permissions ?

Remove all

Specify permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

Important dates

Last login:
Date: 2020-09-28 Today
Time: 18:47:21 Now

Date joined:
Date: 2020-09-28 Today
Time: 18:47:04 Now

Delete
Save and add another
Save and continue editing
SAVE

Рисунок 5.13 – Керування обліковим записом

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

6.1 Планування розробки програмного продукту

Метою магістерської роботи є розроблення програмного забезпечення прогнозування результатів фандрайзингу проєктів для визначення політики рекламування проєктів на платформі зі своєчасним привертанням уваги до них.

Програмне забезпечення призначено для автоматичного виконання визначення результатів фандрайзингу, тобто прогнозування результатів фандрайзингу проєктів на певний період.

Відповідно на даний момент в компанії дані функції виконуються експертом.

Відповідно експерт отримує зарплатню за таку роботу 16000 гривень за місяць, а використання для автоматизації даного процесу такого програмного забезпечення дозволить скоротити 0,75 його ставки.

В розробці брали участь програміст з заробітною платнею 6000 гривень та консультант з заробітною платнею 8500 гривень. На розробку проєкту було витрачено 2 місяці.

Тривалість робіт визначають за формулою (6.1):

$$T_{\text{ц}} = \frac{Q}{R}, \quad (6.1)$$

де $T_{\text{ц}}$ – тривалість циклу, днів;

Q – трудомісткість, люд.-дн.;

R – кількість виконавців, чол. [21].

Характеристика робіт по розробці програми загалом представлена в табл. 6.1.

Таблиця 6.1 – Характеристика робіт по розробці програми

Найменування робіт	Трудомісткість		Виконавці	Тривалість, днів
	люд.- дні	% до підсумку		
Аналіз програмного забезпечення	10	14,70	Програміст Консультант	5
Визначення вимог до програмного продукту	10	14,70	Програміст Консультант	5
Проектування структури програми	20	29,41	Програміст Консультант	10
Розробка схеми функціонування програми	10	14,70	Програміст Консультант	5
Створення програмного коду	10	14,70	Програміст	10
Тестування налагодження програми	5	7,35	Програміст	5
Створення програмної документації	3	4,43	Програміст	3
Разом	68	100		43

Зведений графік планування розробки програмного продукту представлено в табл. 6.2, при цьому в верхньому рядку позначений програміст, а в нижньому – аналітик.

Таблиця 6.2 – Зведений графік планування розробки програмного продукту

Найменування робіт	Календарний період, дні									
	1.10	5.10	12.10	19.10	26.10	2.11	9.11	16.11	23.11	30.11
Аналіз програмного забезпечення	2	3								
	2	3								
Визначення вимог до програмного продукту		2	3							
		2	3							
Проектування структури програми			2	5	3					
			2	5	3					
Розробка схеми функціонування програми					2	3				
					2	3				
Створення програмного коду						2	5	3		
Тестування налагодження програми								2	3	
Створення програмної документації									2	1

6.2 Розрахунок основної заробітної платні

Розрахунок основної заробітної платні працівників представлено в табл. 6.3.

Таблиця 6.3 – Розрахунок основної заробітної платні

Посада	Кількість осіб	Місячний оклад, грн.	Кількість місяців роботи	Сума заробітної платні, грн.
Програміст	1	6000	2	12000
Консультант	1	8500	1	8500
Разом	2			20500

6.3 Розрахунок додаткової заробітної платні

Розрахунок додаткової заробітної платні розраховується за формулою (6.2):

$$ЗП_{\text{додаткова}} = ЗП_{\text{основна}} \cdot 10\%, \quad (6.2)$$

де $ЗП_{\text{основна}}$ – основна заробітна платня;

$ЗП_{\text{додаткова}}$ – додаткова заробітна платня [21].

Підставивши значення додаткової заробітної плати в формулу, отримуємо:

$$ЗП_{\text{додаткова}} = 20500 \cdot 0,1 = 2050 \text{ грн.}$$

6.4 Відрахування на соціальні заходи

Розрахунок на відрахування на соціальні заходи розраховується за формулою (6.3):

$$\text{Відрахування на соціальні заходи} = (ЗП_{\text{основна}} + ЗП_{\text{додаткова}}) \cdot 22\% \text{ [21]}. \quad (6.3)$$

Підставивши значення в формулу, отримуємо:

Відрахування на соціальні заходи = $(20500 + 2050) \cdot 0,22 = 4961$ грн.

6.5 Визначення витрат на матеріали

Використовується 3 найменування матеріалів: диск CD-R – 5 грн.; картридж – 150 грн. і папір – 100 грн. (1 упаковка).

Витрати на матеріали розраховують за формулою (6.4):

$$M = \sum_{i=1}^n (\text{Ц}_i \cdot N_i \cdot (1 + K_{\text{т.з.}}) - \text{Ц}_{io} \cdot N_{io}), \quad (6.4)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.;

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

Ц_i – ціна i -го найменування матеріалу, напівфабрикату і комплектуючого, грн.;

N_i – необхідність в i -му матеріалі, напівфабрикаті і комплектуючому;

Ц_{io} – ціна зворотних відходів i -го найменування матеріалу, грн.;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів і комплектуючих [21].

Визначимо відповідні значення:

$$\text{Ц}_{io} = 0, N_{io} = 0, K_{\text{тз}} = 0,05,$$

$$M = (1 + 0,05) \cdot (5 + 150 + 100) = 267,75 \text{ грн.}$$

Витрати на матеріали становлять 267,75 грн.

6.6 Витрати на спеціальне обладнання

Амортизація обладнання за формулою (6.5):

$$A = \Phi_6 \cdot \frac{H_a}{100}, \quad (6.5)$$

де Φ_6 – балансова вартість обчислювальної техніки, грн.;

H_a – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки, % [21].

Балансова вартість обчислювальної техніки становить 3000 грн., тоді:

$$A = 3000 \cdot 0,25 = 750 \text{ грн.}$$

Витрати на електроенергію визначають за формулою (6.6):

$$C_e = N_n \cdot \Phi_{ef} \cdot K_{зв} \cdot K_{зм} \cdot C_e, \quad (6.6)$$

де N_n – номінальна потужність електронно-обчислювальної машини (ЕОМ), кВт;

Φ_{ef} – річний ефективний фонд часу роботи ЕОМ, машино-год;

$K_{зв}$ – середній коефіцієнт завантаження за часом;

$K_{зм}$ – коефіцієнт завантаження по потужності;

C_e – ціна одного кВт · год електроенергії, грн. / (кВт · год) [21].

Обчислимо:

$$C_e = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 410,18 \text{ грн.}$$

Витрати за статтею «Заробітна плата обслуговуючих робітників та відрахування на соціальне страхування в інші фонди» визначають за формулою (6.7):

$$ЗП_{\text{обсл}} = ФЗП_{\text{Г}} \cdot (1 + K_{\text{відрах}}) \cdot \frac{t_{\text{обсл}}}{\Phi_{\text{еф.обсл}}}, \quad (6.7)$$

де $ФЗП_{\text{Г}}$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн.;

$K_{\text{відрах}}$ – коефіцієнт, що враховує відрахування на соціальне страхування і в інші фонди;

$t_{\text{обсл}}$ – час протягом року, необхідний на технічне обслуговування ЕОМ, год./рік;

$\Phi_{\text{еф.обсл}}$ – річний ефективний фонд часу обслуговуючого персоналу, год./рік [21].

Заробітна плата обслуговуючого персоналу = 3723 грн./місяць.

Обчислимо:

$$ЗП_{\text{обсл}} = 44676 \cdot (1 + 0,22) \cdot \frac{12}{1750} = 373,75 \text{ грн.}$$

Кошторис витрат на утримання і експлуатацію устаткування представлено в табл. 6.4.

Витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів визначаються за наступною формулою (6.8):

$$C_{\text{МО}} = P_{\text{екс}} \cdot t_{\text{МО}}, \quad (6.8)$$

де $C_{\text{МО}}$ – витрати на оплату машинного часу, грн.;

$P_{\text{екс}}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн./машино-год.;

$t_{\text{мо}}$ – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино-год. [21].

Тепер на основі описаних вище виразів виконаємо відповідні обчислення:

$$P_{\text{екс}} = \frac{1705,13}{1800} = 0,95 \text{ грн. машино} - \text{год.},$$

$$C_{\text{мо}} = 0,95 \cdot 344 = 326,8 \text{ грн.}$$

Таблиця 6.4 – Кошторис витрат на утримання і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація обладнання	750 грн.
Експлуатація обладнання (крім витрат на поточний ремонт)	410,18 грн.
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальне страхування і в інші фонди	373,75 грн.
Поточний ремонт обладнання	90 грн.
Інші витрати	81,2 грн.
Разом	1705,13 грн.

6.7 Розрахунок накладних витрат

Калькуляція кошторисної вартості робіт з розробки програми представлена в табл. 6.5.

Таблиця 6.5 – Калькуляція кошторисної вартості робіт з розробки програми.

Найменування статей витрат	Сума, грн.	Питома вага до підсумку, %
Основна заробітна плата	20500	36,19
Додаткова заробітна плата	2050	3,62
Відрахування на соціальне страхування та інше	4961	8,76
Матеріали та комплектація	267,75	0,47
Витрати на спеціальне обладнання	326,8	0,58
Накладні витрати	8200	14,48
Разом (S_{rp})	36305,55	-
Прибуток	10 891,67	19,23
Разом	47 197,22	-
Податок на додану вартість	9439,44	16,67
Разом	56 636,66	100

6.8 Економічна ефективність

Очікується продати 10 копій по 20000 грн., що становить 200000 грн. з урахуванням ризику 50% надходження від продажів складе 100000 грн., що більше ніж сума програми, в якій вже врахована рентабельність 30 %. Таким чином, можна вважати розробку магістерської роботи економічно доцільною. Річна економія на зарплаті аналітика з урахуванням відрахувань на єдиний соціальний внесок становить 175 680 гривень.

Таким чином, витрати окупаються за одну третину року (нормативний термін окупності для соціальної сфери 6,67 року) і можна зробити висновок, що дана робота економічно доцільна.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

У магістерській роботі проводиться дослідження, розробка та програмна реалізація методів прогнозування результатів фандрайзингу проєктів.

При роботі постійно або періодично діють наступні небезпечні і шкідливі чинники:

- невідповідність ергономічних характеристик устаткування нормованим величинам внаслідок нераціональної організації робочого місця;
- можливість ураження електрострумом внаслідок небезпечного рівня напруги в електричному ланцюзі, тому у приміщенні можуть виникнути аварійні ситуації: замикання, загоряння проводки, поразка електрострумом, що може привести до електротравм або летальних наслідків;
- неправильні дії персоналу під час надзвичайних ситуацій внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків [22].

7.2 Заходи по забезпеченню безпеки

Забезпечення безпеки праці, організації оптимальних умов праці, а також захисту навколишнього середовища досягається шляхом дотримання норм і проведення заходів, передбачених законодавчими актами України.

У розділі 4 Державних санітарних правил і норм 3.3.2.007-98 «Гігієнічні вимоги до організації праці з візуальними дисплейними терміналами електронно-обчислювальних машин» передбачені гігієнічні вимоги до організації устаткування робочих місць візуальних дисплейних терміналів (ВДТ) ЕОМ і персональних ЕОМ [22]. Конструкція робочого стола повинна відповідати сучасним вимогам ергономіки й забезпечувати оптимальне розміщення на робочій поверхні використовуваного

устаткування (дисплея, клавіатури, принтера) і документів: приймаємо робочу поверхню з розмірами 1400x800 [23].

Велике значення надається ергономічним та естетичним вимогам. Цього домоглися за допомогою колірної рішення приміщення й устаткування з урахуванням вимог ГОСТ 17.2.032-78 «Робоче місце під час виконання робіт сидячи» [24] – у приміщеннях стіни пофарбовані фарбами пастельних тонів.

Для розробників програм із застосуванням ЕОМ слід призначати регламентовану перерву для відпочинку тривалістю 15 хвилин через кожен годину роботи за ВДТ [22].

Згідно правил улаштування електроустановок приміщення виробництва відноситься до категорії «Приміщення без підвищеної небезпеки» [25].

Відповідно до правил улаштування електроустановок захисне заземлення 4 Ом застосовується у всіх електроустановках змінного струму під напругою 220 В і вище й постійного струму 440 В і вище, у приміщеннях з підвищеною небезпекою заземлення використовується в електроустановках змінного струму напругою 42 В та постійного струму 110 В, захисному заземленню підлягають корпуси трансформаторів, оболонки кабелів, металеві огороження електроустановок відповідно до ГОСТ 12.1.030-81 «Електробезпека. Захисне заземлення. Занулення» [26].

7.3 Заходи по забезпеченню промислової санітарії та гігієни праці

Оптимальні норми температури повітря в робочій зоні виробничих приміщень у холодний і перехідний періоди року при II категорії робіт згідно Державних санітарних норм 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень – 20...23 °С, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек. У теплий період року температура – 22...25 °С, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек [27].

У приміщенні забезпечений приплив свіжого повітря, кількість якого визначається техніко-економічним розрахунком і вибором схеми системи вентиляції, мінімальна витрата повітря визначається з розрахунку $50...60 \text{ м}^3/\text{ч}$ на одну людину, але не менш дворазового повітрообміну в годину [28].

При невеликому забрудненні зовнішнього повітря кондиціонування приміщень здійснюється із перемінними витратами зовнішнього повітря й рециркуляційного [28].

Розрахуємо систему загального рівномірного штучного освітлення в адміністративному приміщенні [29].

Приймаємо розміри приміщення:

- довжина $A = 11 \text{ м}$;
- ширина $B = 7 \text{ м}$;
- висота $H = 5 \text{ м}$;
- висота робочої поверхні складає $h = 0,7 \text{ м}$.

Коефіцієнти відбиття поверхонь приміщення: колір стелі, стін - сірий, незначне пиловиділення, $\rho_c = 50\%$; $\rho_{ст} = 30\%$; $\rho_{п} = 10\%$,

Розряд зорової роботи – III б.

Згідно з Державними будівельними нормами В.2.5-28-2018 «Природне і штучне освітлення. Норми проєктування» рівень нормованого загального освітлення в приміщенні нормуються як $E_n = 400 \text{ Лк}$.

Обираємо світильник типу ЛПП, призначений для приміщень висотою до 6 м. $IP=65$ – ступінь захисту світильника, $[L/h] = 1,4$ – числове значення коефіцієнта світильника.

Коефіцієнт запасу k_z враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо $k_z = 1,6$.

Коефіцієнт нерівномірності (мінімального) освітлення z (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Розрахуємо кількість рядів світильників у приміщенні за формулою (7.1):

$$N_p = \frac{B}{(H-h_p) \cdot (L/h)} = 1,16 \approx 2. \quad (7.1)$$

Визначимо максимально припустиму відстань між рядами світильників за формулою (7.2):

$$L_{max} = \frac{B}{N_p}, \quad (7.2)$$

де B – ширина приміщення;

N_p – кількість рядів.

Розрахуємо:

$$L_{max} = \frac{7}{1,16 \approx 2} = 3,5 \text{ м.}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (7.3):

$$h = \frac{L_{max}}{(L/h)}. \quad (7.3)$$

Розрахуємо:

$$h = \frac{3,5}{1,4} = 2,5 \text{ м.}$$

Знайдемо висоту звисання світильника від стелі за формулою (7.4):

$$h_3 = H - h_p - h, \quad (7.4)$$

де H – висота приміщення;

h_p – висота робочої поверхні;

h – висота підвісу світильника над робочою поверхнею.

Розрахуємо:

$$h_3 = 5 - 0,7 - 2,5 = 1,8 \text{ м.}$$

Чисельне значення індексу приміщення визначаємо за рівнянням (7.5):

$$i = \frac{A \cdot B}{h \cdot (A+B)}, \quad (7.5)$$

де A – довжина приміщення, м;

B – ширина приміщення, м;

h – висота розміщення світильників над робочою поверхнею, м.

Розрахуємо:

$$i = \frac{11 \cdot 7}{2,5 \cdot (11+7)} = 1,71.$$

Значення коефіцієнта використання світлового потоку η вибирається в залежності від виду джерела світла, типу обраного світильника, коефіцієнтів відбиття поверхонь приміщення (ρ_c , $\rho_{ст}$, $\rho_{п}$) та індексу приміщення. В нашому випадку $\eta = 54\%$.

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні за формулою (7.6):

$$\Phi_{\Sigma} = \frac{(E_n \cdot S \cdot k_3 \cdot z)}{\eta}, \quad (7.6)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, *лм*;

E_n – нормоване значення освітленості, *лк*;

S – площа освітлюваної поверхні, *м²*;

k_3 – коефіцієнт запасу;

z – коефіцієнт нерівномірності (мінімальної) освітленості;

η – коефіцієнт використання світлового потоку.

Розрахуємо:

$$\Phi_{\Sigma} = \frac{400 \cdot 11 \cdot 7 \cdot 1,6 \cdot 1,1}{0,54} = 100385 \text{ лм.}$$

Визначаємо умовну загальну кількість світильників у приміщенні за формулою (7.7):

$$N^* = \frac{AB}{L_{max}^2}. \quad (7.7)$$

Розрахуємо:

$$N^* = \frac{11 \cdot 7}{3,5 \cdot 3,5} = 6,3 \approx 8 \text{ шт.}$$

Розрахуємо світловий потік умовного джерела світла за формулою (7.8):

$$\Phi_{\text{л}} = \frac{\Phi_{\Sigma}}{N_{\text{л}}} \text{ лм,} \quad (7.8)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, *лм*;

$N_{\text{л}}$ – загальна кількість ламп у приміщенні, *шт.*

При цьому за формулою (7.9):

$$N_{\text{л}} = N^* \cdot n, \quad (7.9)$$

де n – кількість ламп у світильнику (в ЛПП світильнику знаходиться 2 лампи).

Розрахуємо:

$$N_{\text{л}} = 8 \cdot 2 = 16 \text{ шт.},$$

$$\Phi_{\text{л}} = \frac{100385}{16} = 6274 \text{ лм.}$$

Обираємо лампу ЛБ:

- $P_{\text{л}} = 40 \text{ Вт};$
- $\Phi_{\text{л}} = 5800 \text{ лм.}$

Розрахуємо:

$$m = \frac{6274}{5800} = 1,081.$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні за формулою (7.10):

$$N = N^* \cdot m. \quad (7.10)$$

Розрахуємо:

$$N = 8 \cdot 1.081 \approx 8 \text{ шт.}$$

Фактична кількість ламп у приміщенні за формулою (7.11):

$$N_{\phi\lambda} = N_{\phi} \cdot n. \quad (7.11)$$

Розрахуємо:

$$N_{\phi\lambda} = 8 \cdot 2 = 16 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість за формулою (7.12):

$$E_p = \frac{\Phi_{\lambda} N_{\phi\lambda} \eta}{S k_{\Sigma} z}, \text{ лк.} \quad (7.12)$$

Розрахуємо:

$$E_p = \frac{5800 \cdot 16 \cdot 0,54}{11 \cdot 7 \cdot 1,6 \cdot 1,1} = 370 \text{ лк.}$$

Тобто виконується умова $E_p = (-10\% \dots +20\%) \cdot E_n$, лк. На основі розрахунків для освітлення приміщення площею 77 м² встановлено 8 світильників, розташованих у 2 рядах по 4 світильники у кожному ряді, що дозволяє отримати освітленість на робочому місці 400 лк відповідно до Державних будівельних норм В.2.5-28-2018 «Природне і штучне освітлення. Норми проектування». Згідно з Державними санітарними нормами З.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» під настільні шумливі апарати можна підкладати м'які килимки із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, – прокладки з м'якої гуми, повсті, товщиною 6-8 мм [30].

7.4 Заходи по забезпеченню пожежної безпеки

Дані приміщення належать до категорії «Д» з пожежної небезпеки, тому відповідно до вимог Державних будівельних норм В.1.1-7:2016

«Пожежна безпека об'єктів будівництва. Загальні вимоги» воно має II ступінь вогнестійкості [32]. Згідно вимог Державних будівельних норм В.2.5-56:2014 «Системи протипожежного захисту», в приміщенні обладнаному ПК з ВДТ встановлена система пожежної й охоронної сигналізації «Фотон-ОФІС», яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика [33].

Оскільки приміщення (дослідницької лабораторії, конструкторського бюро тощо), що обладнане ПК з ВДТ має площу 77 м², тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 4 штуки (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг. і більше, на 20 м² площі приміщення), відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м [34].

7.5 Заходи по забезпеченню безпеки у надзвичайних ситуаціях

Укриття населення у захисних спорудах цивільного захисту недотримання якого може призвести до механічних травм, зараження хімічними викидами або летального наслідку, укриття населення в захисних спорудах – це комплекс заходів із завчасним будівництвом захисних споруд, а також пристосуванням наявних приміщень для захисту населення та підтримання їх у готовності до використання [35].

Укриття населення в захисних спорудах є надійним способом захисту від уражаючих факторів ядерної, хімічної, бактеріологічної, звичайної зброї, у разі аварій і деяких стихійних лих (ураганів, снігових заносів) [36].

ВИСНОВКИ

Виконано огляд проблеми фандрайзингу з наведенням основних понять. Виконано огляд програмних засобів, які використовуються для фандрайзингу проєктів. Визначено, що вони не мають засобів прогнозування результатів збору коштів на проєкти у своєму складі.

За результатами проведеного аналізу методів, які використовуються для прогнозування під час фандрайзингу проєктів, було виділено, що на даний момент прогнозування результатів фандрайзингу проєктів виконується за допомогою використання регресійної моделі. У той же час актуальним рішенням для розв'язання ширшої проблеми, прогнозування часових рядів, є нейронні мережі, тому було запропоновано використовувати даний механізм для виконання прогнозування результатів фандрайзингу проєктів.

Запропоновано метод прогнозування результатів фандрайзингу проєктів, який ґрунтується на використанні нейронних мереж з довгою короткостроковою пам'яттю, використовуючи в якості вхідних ознак суми зібраних коштів на проєкт за перші дні збору, що дозволяє визначити підсумкову зібрану на проєкт суму коштів, визначаючи тим самим політику для роботи над фандрайзинговою компанією на ранніх етапах.

Виконано проєктування програмного забезпечення у вигляді вебзастосунку мовою програмування Python. Визначено функціональні вимоги до програмного забезпечення, структуру вебзастосунку. Виконано проєктування бази даних на основі використання системи SQLite.

Реалізовано вебзастосунок, який дозволяє виконувати створення нових проєктів та редагування існуючих, перегляд власних проєктів, детальних даних проєкту, вкладів у проєкт, внесення вкладів на фінансування проєктів, перегляд переліку активних, завершених, успішних проєктів, пошук проєктів, перегляд історії власних вкладів, внесення грошей на власний рахунок та безпосередньо прогнозування результатів фандрайзингу за динамікою перших 7 днів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Что такое фандрайзинг? [Электрон. ресурс]. – Режим доступа : <https://intalent.pro/article/fandrayzing.html-0>
2. Соколова, А.М. Фандрайзингова діяльність [Текст] : сутність, особливості та перспективи розвитку / А.М. Соколова // Економіка та держава. – 2013. – № 1. – С. 13-16.
3. Огородник, В. О. Фандрайзинг – інструмент активації соціально-відповідальної взаємодії суб'єктів суспільних відносин [Текст] / В.О. Огородник // Економіка і організація управління. – 2014. – № 1 (17). – С. 197-201.
4. Фандрайзинг [Электрон. ресурс]. – Режим доступа : http://www.marketch.ru/marketing_dictionary/marketing_terms_f/fundraising/
5. Фандрайзинг в сфере культуры [Электрон. ресурс]. – Режим доступа : <https://okulture24.ru/fandraizing-v-sfere-kultury/>
6. Онлайн фандрайзинг [Електрон. ресурс] : як залучити кошти не виходячи із мережі. – Режим доступу : <https://gurt.org.ua/news/informator/53859/>
7. Airfunding | Helping everyone, With everyone [Electronic resource]. – Access mode : <https://www.airfunding.net/>
8. 7 Best Crowdfunding Platforms for 2020 [Electronic resource]. – Access mode : <https://medium.com/@Airfunding/7-best-crowdfunding-platforms-for-2020-152120f121e4>
9. GoFundMe [Electronic resource] : #1 Fundraiser Platform – Crowdfund Online. – Access mode : <https://www.gofundme.com/>
10. Tracking the Dynamics in Crowdfunding [Electronic resource] / H. Zhao, H. Zhang, Y. Ge, Q. Liu, E. Chen, H. Li, L. Wu // KDD'17, August 13–17, 2017, Halifax, NS, Canada. – 2017. – Pp. 625-634. – Access mode : <http://dx.doi.org/10.1145/3097983.3098030>

11. Yeh, J.-Y. A machine learning approach to predict the success of crowdfunding fintech project [Electronic resource] / J.-Y. Yeh, C.-H. Chen // Journal of Enterprise Information Management. – 2020. – Access mode : <https://doi.org/10.1108/JEIM-01-2019-0017>

12. Predicting Startup Crowdfunding Success through Longitudinal Social Engagement Analysis [Electronic resource] / Qizhen Zhang, Tengyuan Ye, Meryem Essaidi, Shivani Agarwal, Vincent Liu, Boon Thau Loo // CIKM'17 : Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, November 6–10, 2017, Singapore. – Singapore : Association for Computing Machinery, 2017. – Pp. 1937-1946. – Access mode : <https://doi.org/10.1145/3132847.3132908>

13. Kindler, A. Early prediction of the outcome of Project Fundraiserstarter campaigns [Electronic resource] : is the success due to virality? / Alex Kindler, Michael Golosovsky & Sorin Solomon // Palgrave Communications. – 2019. – Volume 5. – Access mode : <https://www.nature.com/articles/s41599-019-0261-6>

14. Верзунов, С.Н. Применение глубоких нейронных сетей для краткосрочного прогноза дальности видимости [Текст] / С.Н. Верзунов // Проблемы автоматизации и управления. – 2019. – № 1 (36). – С. 118-130.

15. Глайборода, М. Нейрон, персептрон, кошка [Электрон. ресурс] : фундаментальные отличия нейросетей. Часть 2 / М. Глайборода. – Режим доступа : <https://blog.heyml.com/нейрон-персептрон-кошка-фундаментальные-отличия-нейросетей-часть-2-efafc7c089f6>

16. Временной ряд – модель LSTM [Электрон. ресурс]. – Режим доступа : <https://coderlessons.com/tutorials/mashinnoe-obuchenie/izuchit-vremennyye-riady/vremennoi-riad-model-lstm>

17. Алжеев, А.В. Сравнительный анализ прогнозных моделей ARIMA и LSTM на примере акций российских компаний [Текст] / А.В. Алжеев, Р.А. Кочкаров // Финансы : теория и практика. – 2020. – № 1, Т. 24. – С. 14-23.

18. Weather Based Day-Ahead and Week-Ahead Load Forecasting using Deep Recurrent Neural Network [Text] / Mingzhe Zou, Duo Fang, Gareth Harrison, Sasa Djokic // Conference: 2019 IEEE 5th International forum on Research and Technology for Society and Industry (RTSI). – Firenze : IEEE, 2019. – Pp. 341-346.
19. Python Language advantages and application [Electronic resource]. – Access mode : <https://www.tutorialspoint.com/python-language-advantages-and-application>
20. IT Next Free Website Template [Electronic resource]. – Access mode : <https://www.free-css.com/free-css-templates/page256/it-next>
21. Остапенко, В.В. Методичні вказівки по економічному обґрунтуванню дипломних проєктів для студентів спеціальностей 7.080403 «Програмне забезпечення автоматизованих систем», 7.080402 «Інформаційні технології проєктування» і 7.091501 «Комп'ютерні та інтелектуальні системи і мережі» всіх форм навчання [Текст] / В.В. Остапенко, Е.М. Касьян, Є.М. Сердюк. – Запоріжжя : ЗНТУ, 2009. – 22 с.
22. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. НПАОП 0.00-7.15-18. Державні санітарні правила і норми [Текст]. – [Чинний від 2018-02-14]. – Харків : Форт, 2018. – 48 с.
23. Жидецький, В.Ц. Основи охорони праці [Текст] : підручник / В. Ц. Жидецький. – 5-те вид., доп. – К. : Знання, 2014. – 373 с.
24. ГОСТ 17.2.032-78 Рабочее место при выполнении работ сидя. Система стандартов безопасности труда [Текст]. – [Чинний від 1979-01-01]. – М. : ИПК Издательство стандартов, 2001. – 7 с.
25. Правила улаштування електроустановок [Текст] : Нормативне виробничо-практичне видання : вид. 5-те, перероблене й доповнене : затв. М-вом енергетики та вугільної промисловості України 20.06.14. – [Чинний від 2014-11-20]. – Х. : Міненерговугілля України, 2014. – 793 с.

26. Правила улаштування електроустановок [Текст] : ПУЕ-2017. – На заміну ПУЕ-86. – [Чинний від 2017-08-21]. – К. : Міненерговугілля України, 2017. – 617 с.
27. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [Текст]. – [Чинний від 1999-12-01]. – Харків : Форт, 2000. – 24 с.
28. ДБН В.2.5-28:2018. Природне та штучне освітлення [Текст]. – [Чинний від 2019-03-01]. – К. : Мінрегіон України, 2019. – 137 с.
29. Охорона праці [Текст] : навч. посібник / З. М. Яремко, С. В. Тимошук, О. І. Третяк, Р. М. Ковтун ; за ред. проф. З. М. Яремка. – Львів : Видавничий центр ЛНУ імені Івана Франка, 2010. – 374 с.
30. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку [Текст]. – [Чинний від 1999-12-01]. – К. : МОЗ України, 1999. – 24 с.
31. Правила пожежної безпеки в Україні [Текст]. – Харків : Форт, 2015. – 124 с.
32. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст]. – [Чинний від 2017-01-31]. – Київ : Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.
33. НПАОП 40.1-1.32-01 Правила будови електроустановок. Електрообладнання спеціальних установок [Текст]. – [Чинний від 2002-01-01]. – Харків : Індустрія, 2010. – 118 с.
34. Законодавство про охорону праці. Том 1 [Текст]. – Харків : Форт, 2013. – 368 с.
35. Стеблюк, М.І. Цивільна оборона та цивільний захист [Текст] : підручник / М.І. Стеблюк. – К. : Знання-Прес, 2007. – 487 с.
36. Шоботов, В. М. Цивільна оборона [Текст] : навчальний посібник / В. М. Шоботов. – 2-ге вид., перероб. – К. : Центр навчальної літератури, 2006. – 438 с.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення має бути розроблено у вигляді вебзастосунку, який повинен з одного боку забезпечувати підтримку роботи площадки з фандрайзингу проєктів, а з іншого боку – надавати математичну основу реалізації прогнозування результатів фандрайзингу проєктів.

A.1 Підстави для розробки

Підставою для розробки є завдання на магістерську роботу за темою “Дослідження, розробка та програмна реалізація методів прогнозування результатів фандрайзингу проєктів”, затверджене наказом № 301 від 3 листопада 2020 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програмне забезпечення призначено для підтримки роботи платформи з фандрайзингу проєктів, за допомогою якої користувачі можуть докладатися до фінансування проєктів, які беруть участь в конкурсі.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програма, що розробляється, повинна забезпечувати виконання таких функцій:

- створення нових проєктів;
- перегляд власних проєктів;
- перегляд детальних даних проєкту;
- перегляд вкладів у проєкт;
- внесення вкладу на фінансування проєкту;

- редагування проєктів;
- перегляд переліку активних проєктів;
- перегляд переліку завершених проєктів;
- перегляд успішних проєктів;
- пошук проєктів;
- перегляд історії власних вкладів;
- внесення грошей на власний рахунок;
- авторизація облікового запису;
- керування правами доступу користувачів;
- реєстрація облікового запису.

Вхідними даними є дані облікового запису користувача, дані проєктів, дані власного PayPal рахунку, динаміка фандрайзингу проєктів.

Вихідними даними є прогноз результатів фандрайзингу проєктів, накопичені дані проєктів, результати виконання окремих операцій.

A.3.2 Вимоги до надійності

Вимоги до надійності полягають в контролі за винятковими ситуаціями (зокрема коректності введення даних шляхом, наприклад, забезпечення неможливості введення текстових даних в поле дати). Окрім того сюди належать також вимоги до забезпечення доступу користувачів, коли доступ до власних проєктів можуть отримати тільки авторизовані користувачі, а всі інші бачать тільки загальні результати.

A.3.3 Умови експлуатації

Базовою вимогою запуску програмного забезпечення на сервері є наявність наступних програмних бібліотек та модулів:

- django;
- keras;

- django-paypal;
- python.

Відповідно якщо програмне забезпечення не встановлено, то запустити вебзастосунок на сервері не вдасться. Для клієнтської частини окремих вимог не висувається: для цього достатня наявність браузеру та мережевого підключення.

A.3.4 Вимоги до складу та параметрів технічних засобів

Технічні засоби, наявність яких має бути забезпечена на сервері (для клієнта мінімальними вимогами є вимоги, які висуваються до роботи сучасних браузерів):

- процесор, який має щонайменше 4 ядра та частоту не меншу за 3,1 ГГц (або можливість прискорення до 3,1 ГГц);
- оперативна пам'ять обсягом не менше ніж 8 Гб;
- жорсткий диск загальним обсягом не менше ніж 1 Тб, а з обсягом вільного простору не менше ніж 10 Гб;
- монітор та відеокарта з роздільною здатністю не менше ніж 1920x1080.

A.3.5 Вимоги до транспортування та збереження

Збереження програмного забезпечення виконується на сервері, відповідно вимоги є вимогами до роботи сервера. Транспортування може виконуватися за допомогою флеш-носіїв або віддалено.

A.3.6 Вимоги до програмної документації

У межах програмної документації за роботою передбачено оформлення технічного завдання та тексту програми.

A.3.7 Вимоги до маркування та упакування

Матеріали за пояснювальною запискою мають надаватися на компакт-диску з позначкою про автора роботи на ньому.

Безпосередньо програмний продукт може розповсюджуватися через віддалений сервер, на якому розташовується сервера частина. Всі клієнти відповідно звертаються до програми віддалено.

A.4 Стадії та етапи розробки

Етапи виконання магістерської роботи:

- постановка завдання роботи;
- аналіз та дослідження предметної області;
- розробка архітектури програми;
- розробка програми;
- тестування та експериментальне дослідження програми;
- оформлення пояснювальної записки та документів до неї, нормоконтроль та рецензування;
- захист роботи.

A.5 Порядок контролю та приймання

Магістерська робота повинна бути виконана у відповідності з календарним планом.

Магістерська робота має бути прийнята консультантами відповідних розділів та керівником роботи загалом, після чого затверджена завідуючим кафедрою та захищена на засіданні екзаменаційної комісії.

ДОДАТОК Б
Текст програми

Б.1 Текст файлу menu.html

```
{% load staticfiles %}
<!-- створено з використанням вільного шаблону It Next
https://www.free-css.com/free-css-templates/page256/it-next -->
<!-- анімація завантаження -->
<div class="waiting">  </div>
<!-- кінець завантаження -->
<!-- заголовок -->
<header id="menubase" class="menus">
  <!-- верхній рядок заголовку -->
  <div class="upmenu">
    <div class="lego">
      <div class="row">
        <div class="minicol">
          <div class="info">
            <div class="adminpanel"> <a class="whiteinfo"
href="/admin">Авторський дім</a> </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <!-- кінець верхнього рядка заголовку -->
  <!-- меню -->
  <div class="menusec">
    <div class="lego">
      <div class="row">
        <div class="tradrowsec">
          <!-- визначення логотипу -->
          <div class="logo"> <a href="/"></a> </div>
          <!-- завершення логотипу -->
        </div>
      </div>
    </div>
    <!-- початок меню -->
    <div class="begmenu">
      <div id="menuparts">
        <ul class="spisik">
          <!-- Домашня сторінка -->
          <li> <a href="/">
            Головна
          </a>
```

```

</li>
<!-- Випадаюче меню для роботи з проєктами -->
<li> <a href="/projects">
    Проєкти
  </a>
  <ul>
    <li><a href="/active">
        Активні
      </a>
    </li>
    <li><a href="/closed">
        Завершені
      </a>
    </li>
    <li><a href="/success">
        Успішні
      </a>
    </li>
    {% if user.is_authenticated %}
    <li><a href="/own/{{ user.id }}">
        Власні
      </a>
    </li>
    {% endif %}
  </ul>
</li>
<!-- меню роботи з рахунком авторизованого користувача -->
{% if user.is_authenticated %}
<li> <a href="/load">
    Рахунок
  </a>
  <ul>
    <li>
      <a href="/history">
        Вклади
      </a>
    </li>
    <li>
      <a href="/load">
        Поповнити
      </a>
    </li>
  </ul>
</li>
{% endif %}

```

```

        {% if user.is_authenticated %}
        <li>
            <a href="/logout">
                Вийти
            </a>
        </li>
        {% else %}
        <li>
            <a href="/login">
                Увійти
            </a>
        </li>
        {% endif %}
    </ul>
</div>
<div class="findpic">
    <ul>
        <li>
            <li><a href="#" data-toggle="modal" data-
target="#findform"><i class="findf" aria-hidden="true"></i></a></li>
            </li>
        </ul>
    </div>
</div>
<!-- завершення меню -->
</div>
</div>
</div>
<!-- завершення нижнього рядка заголовку -->
</header>

```

Б.2 Текст файлу projects.html

```

{% load staticfiles %}
<html>
    <head>
        <title>
            ProjectFundraiser
        </title>

        {% include 'projectfundraiser/blocks/head.html' %}
    </head>

```

```

<body>
  <div class="lego">
    {% include 'projectfundraiser/blocks/menu.html' %}
  <div id="infoplacer" class="infoplacersec">
    <div class="lego">
      <div class="row">
        <div class="sec12">
          <div class="full">
            <div class="nameplace">
              <div class="nameplpos">
                <h1 class="namewebpage">Перелік {{info}} проєктів</h1>
                <ol class="routepage">
                  <li><a href="index.html">Головна</a></li>
                  <li class="active">Проекти</li>
                </ol>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  <div class="projectlist">
    <div class="lego">
      <div class="row">
        <div class="sec9">
          <div class="row">
            {% if projects %}
            {% for project in projects %}
              <div class="oneprojsec">
                <div class="projall">
                  <div class="projphoto">  </div>
                  <div class="projinfo">
                    <div class="center">
                      <h4><a href="/project/{{ project.id }}">{{ project.name
}}</a></h4>
                    </div>
                  </div>
                <div class="projsum">
                  <p><span class="sum">зібрано ${{ project.gotfin }} з ${{
project.necessfin }}</span></p>
                  {% if info == "активних" %}
                  <p>залишилось днів: {{project.togo}}</p><br>
                  {% endif %}

```

```
{% if project.closeflag %} - завершено автором {% endif %}  
    </div>  
  </div>  
</div>  
</div>  
      {% endfor %}  
  
      {% else %}  
        <p>  
          На даний момент проєкти в цій категорії відсутні.  
        </p>  
      {% endif %}
```

```
</div>  
</div>  
  
<div class="sec3">  
  <div class="finder">  
    <div class="finderpos">  
      <h4>Пошук</h4>  
      <div class="findform">  
        <div class="inpfindform">  
          <form action="/search" method="post" id="projsearchform" class="search-  
global">  
            {% csrf_token %}  
              <input class="inpstyle" placeholder="Шукати" type="text"  
name="searchtext">  
                <span class="btnpos">  
                  <button type="submit"><i class="findf" aria-  
hidden="true"></i></button>  
                </span>  
              </form>  
            </div>  
          </div>  
        </div>  
      </div>  
    </div>  
  </div>  
  {% include 'projectfundraiser/blocks/footer.html' %}  
</body>  
</html>
```

Б.3 Текст файлу project.html

```
{% load staticfiles %}
<html>
  <head>
    <title>
      ProjectFundraiser
    </title>

    {% include 'projectfundraiser/blocks/head.html' %}
  </head>
  <body>
    <div class="lego">
      {% include 'projectfundraiser/blocks/menu.html' %}
    <div id="infoplacer" class="infoplacersec">
      <div class="lego">
        <div class="row">
          <div class="sec12">
            <div class="full">
              <div class="nameplace">
                <div class="nameplpos">
                  <h1 class="namewebpage">Детальніше про проєкт</h1>
                  <ol class="routepage">
                    <li><a href="/">Головна</a></li>
                    <li class="active">Детальніше</li>
                  </ol>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="detprojsec">
      <div class="lego">
        <div class="row">
          <div class="sec9">
            <div class="row">
              <div class="projinfo">
                <div class="imgimfo">
                  <div class="imgpos">  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

<div class="detprojstyle">
  <div class="projheadstyle">
    <h2>{{project.name}}</h2>
    {% if project.closeflag %}
    Проект вже завершено<br><br>
    {% else %}
    Фандрайзинг активний<br><br>
    {% endif %}
    Дата завершення: {{ project.enddate }}

    {% if not project.closeflag and project.author.id == user.id
and project.gotfin >= project.necessfin %}
    <a href="/project/{{ project.id }}/close"
class="redbtn">Завершити збір</a>
    {% endif %}
  </div>
  <div class="projdetailpos"> <span class="money">Зібрано
${{ project.gotfin }} з ${{ project.necessfin }}</span> </div>
  {% if project.predictfin and project.author.id == user.id %}
  <div class="projdetailpos"> <span
class="money">Очікувана сума: {{ project.predictfin }}$</span> </div>
  {% endif %}
  <div class="projdetaction">
    {% if not project.closeflag %}
    <form class="finform"
action="/project/{{ user.id }}/{{ project.id }}/finance" method="post">
      {% csrf_token %}
      <div class="quantity">
        <input name="moneyam" value="1"
class="inpamount" size="2" type="text" pattern="^[0-9]+$">
      </div>
      <button type="submit" class="paybtn">Внести
суму</button>
    </form>
    {% endif %}

    <div class="tableall">
      <div class="tablesec">
        <ul class="tableheaders" role="tablist">
          <li class="tablehead"> <a class="tableheadl" data-
toggle="tab" href="#about">Опис</a> </li>
          <li class="tablehead"> <a class="tableheadl" data-
toggle="tab" href="#contributors">Учасники</a> </li>
          <li class="tablehead"> <a class="tableheadl active"
data-toggle="tab" href="#fund">Внески</a> </li>

```

```

</ul>

<div class="tab-content">
  <div id="about" class="nonacttable">
    <p>{{project.about|linebreaks}}</p>
  </div>

  <div id="fund" class="acttable">
    <div class="tablepos">
      {% for f in funds %}
      <div class="sec8">
        <h5>{{f.name}}</h5>
        <p><span class="datetimest">{{f.time}}</span></p>
        <p class="msg">{{f.amount}}$</p><br>
      </div>
      {% endfor %}

    </div>
  </div>
  <div id="contributors" class="nonacttable">
    <div class="tablepos">
      <p>{{project.about|linebreaks}}</p>
    </div>
  </div>
</div>
</div>
</div>
</div>
<div class="sec3">
  <div class="finder">
    <div class="finderpos">
      <h4>Пошук</h4>
      <div class="findform">
        <div class="inpfindform">
          <input class="inpstyle" placeholder="Шукати"
type="text">

          <span class="btnpos">
            <button type="submit"><i class="findf" aria-
hidden="true"></i></button>
          </span>

```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>

    </div>
    {% include 'projectfundraiser/blocks/footer.html' %}
</body>
</html>

```

Б.4 Текст файла settings.py

```

import os

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

SECRET_KEY = ...

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'projectfundraiser.apps.ProjectfundraiserConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.staticfiles',
    'django.contrib.messages'
]

MIDDLEWARE_CLASSES = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.auth.middleware.SessionAuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',

```

```

'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'ProjectFundraiser.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.contrib.auth.context_processors.auth',
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.messages.context_processors.messages'
            ],
        },
    ],
]

WSGI_APPLICATION = 'ProjectFundraiser.wsgi.application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'db'
    }
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {

```

```

        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

```

```
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'Europe/Kiev'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

```
STATIC_URL = '/static/'
```

Б.5 Текст файлу models.py

```

from django.db import models
from django.contrib.auth.models import User

class UserExt(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    sumacc = models.IntegerField(default=0)
    paypalsumacc = models.TextField()

class Project(models.Model):
    name = models.CharField(max_length=300)
    about = models.TextField()
    begdate = models.DateTimeField()
    enddate = models.DateTimeField()
    necessfin = models.IntegerField()
    gotfin = models.IntegerField()
    predictfin = models.IntegerField()
    closeflag = models.BooleanField(default=False)
    author = models.ForeignKey(User, on_delete=models.CASCADE,
related_name='author')
    contributors = models.ManyToManyField(User)
    photo = models.ImageField('/static/projectfundraiser')
    def __str__(self):

```

```
return self.name.encode('utf8')
```

```
class Payment(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    project = models.ForeignKey(Project, on_delete=models.CASCADE)
    amount = models.IntegerField(default=0)
    time = models.DateTimeField()
```

Б.6 Текст файла admin.py

```
from django.contrib import admin
from .models import Project, Payment
```

```
admin.site.register(Project)
admin.site.register(Payment)
```

Б.7 Текст файла views.py

```
# -*- coding: utf-8 -*-

from .models import Project
from .models import Payment
from .models import User
from .models import UserExt
from django.contrib.auth import authenticate
from django.contrib.auth import login
from django.contrib.auth import logout
from django.contrib import messages
from django.shortcuts import get_object_or_404
from django.shortcuts import redirect
from django.shortcuts import render
from django.utils import timezone
from django.db.models import F
from django.db.models import Q
import paypalpay
import datetime

def index(request):
    return render(request, 'projectfundraiser/index.html'))
```

```

def success(request):
    projectss = Project.objects.filter(necessfin__lte=F('gotfin'))
    return render(request, 'projectfundraiser/projects.html', {'projects':
projectss, 'info': 'успішних'})

def active(request):
    projectsa =
Project.objects.filter(Q(enddate__gte=datetime.date.today())&Q(closeflag=False))
    pa = []
    for p in projectsa:
        t=str(p.enddate).split('-')
        tt=t[2].split(' ')
        print(tt[0])
        p.togo = (datetime.date(int(t[0]),int(t[1]),int(tt[0]))-
datetime.date.today()).days
        pa.append(p)
    return render(request, 'projectfundraiser/projects.html', {'projects': pa,
'info': 'активних'})

def closed(request):
    projectsc =
Project.objects.filter(Q(closeflag=True)|Q(enddate__lte=datetime.date.today()))
    return render(request, 'projectfundraiser/projects.html', {'projects':
projectsc, 'info': 'завершених'})

def search(request):
    if request.method == 'GET':
        return render(request, 'projectfundraiser/projects.html',{'projects': [],
'info': '-'})
    elif request.method == 'POST':
        proj =
Project.objects.filter(Q(name__contains=request.POST['searchtext'])|Q(about__co
ntains=request.POST['searchtext']))
        return render(request, 'projectfundraiser/projects.html', {'projects':
proj, 'info': 'знайдених'})

def history(request, userid):
    payments = Payment.objects.filter(user=userid)
    projects = Project.objects.order_by('-begdate')
    tr = []
    for t in payments:
        for p in projects:
            if p.id==t.project_id:
                t.name=p.name

```

```

        tr.append(t)
    return render(request, 'projectfundraiser/history.html', {'payments': tr})

def own(request, userid):
    projectso = Project.objects.filter(author=userid)
    return render(request, 'projectfundraiser/projects.html', {'projects':
projectso, 'info': 'ВЛАСНИХ'})

def project(request, projectid):
    projobj = get_object_or_404(Project, pk=projectid)

    if projobj.enddate <= timezone.now() and not projobj.closeflag:
        projobj.closeflag = True
        projobj.save()

    for Payment in projobj.Payments:
        Payment.user.UserExt.sumacc += int(Payment.amount)
        Payment.user.UserExt.save()

    funds = Payment.objects.filter(project_id=projectid)
    f2 = []
    for f in funds:
        usall = User.objects.order_by('-id')
        for ua in usall:
            if ua.id == f.user_id:
                f.name = ua.username
                f2.append(f)
                print(f.project_id)
    return render(request, 'projectfundraiser/project.html', {
        'project': projobj,
        'funds': f2,
        'contributors': the_project.contributors.all()
    })

def finance(request, userid, projectid):
    projobj = get_object_or_404(Project, pk=projectid)
    fundsum = int(request.POST['moneyam'])
    if fundsum <= request.user.UserExt.sumacc:
        t = Payment()
        t.user = request.user
        t.project = projobj
        t.amount = fundsum
        t.time = timezone.now()
        t.save()
        projobj.gotfin += fundsum

```

```

        projobj.save()
        request.user.UserExt.sumacc -= fundsum
        request.user.UserExt.save()
        messages.add_message(request, messages.SUCCESS, 'Суму було
успішно внесено!')
    else:
        messages.add_message(request, messages.ERROR, 'Не вистачає
грошей на рахунку. Операцію не вдалося виконати')

    return redirect('/project/{}'.format(projobj.id))

```

```

def close_project(request, project_id):
    projobj = get_object_or_404(Project, pk=project_id)
    if projobj.gotfin >= projobj.necessfin:
        projobj.closeflag = True
        projobj.save()
        messages.add_message(request, messages.SUCCESS, '\uПроведення
фандрайзингу було завершено. Операція виконана успішно')
    else:
        messages.add_message(request, messages.ERROR, '\uФандрайзинг
не може бути завершено зараз. Операцію не виконано')

    return redirect('/project/{}'.format(projobj.id))

```

```

def load(request):
    if request.method == 'GET':
        return render(request, 'projectfundraiser/load.html')
    elif request.method == 'POST':
        loadsum = int(request.POST['sum'])
        res = paypalpay.paypalload(loadsum, user.paypalsumacc)
        if res:
            try:
                request.user.UserExt.sumacc += loadsum
                request.user.UserExt.save()
            except:
                u = UserExt()
                u.sumacc = loadsum
                u.user = request.user
                u.save()
                request.user.UserExt = u
                messages.add_message(request, messages.SUCCESS, 'Успішно
додано ${{}} до рахунку'.format(load_sum))
        else:

```

```

        messages.add_message(request, messages.ERROR, "\u041e\u043f\u0435\u0440\u0430\u0446\u0456\u044e \u043d\u0435
        виконано')
    return redirect('/')

```

```

def logout_route(request):
    logout(request)
    return redirect('/')

```

```

def login_route(request):
    if request.method == 'GET':
        return render(request, 'projectfundraiser/login.html')
    elif request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(username=username, password=password)
        if user is not None:
            if user.is_active:
                login(request, user)
                return redirect('/')
            else:
                messages.add_message(request, messages.ERROR, 'Даний
                обліковий запис не є активним')
                return redirect('/login')
        else:
            messages.add_message(request, messages.ERROR, 'Невірні дані')
            return redirect('/login')

```

Б.8 Текст файлу urls.py

```

from django.conf.urls import url
from . import views

```

```

urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^success', views.success, name='success'),
    url(r'^closed', views.closed, name='closed'),
    url(r'^active', views.active, name='active'),
    url(r'^own/(?P<userid>[0-9]+)/$', views.own, name='own'),
    url(r'^search', views.search, name='search'),
    url(r'^load/$', views.load, name='load'),
    url(r'^history/(?P<userid>[0-9]+)/$', views.history, name='history'),

```

```

        url(r'^project/(?P<projectid>[0-9]+)/$',          views.project,
name='detailedproject'),
        url(r'^project/(?P<userid>[0-9]+)/(?P<projectid>[0-9]+)/finance$',
views.finance, name='financeproject'),
        url(r'^project/(?P<projectid>[0-9]+)/endfund$',    views.endfund,
name='fundingend'),
        url(r'^logout/$', views.logout_route, name='logout'),
        url(r'^login/$', views.login_route, name='login')
    ]

```

Б.9 Текст файла PFPredictionModel.py

```

from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.models import model_from_json

def getLSTMmodel(x = [],y = []):
    model = None
    if not os.path.isfile('./proffund.json') and x and y:
        model = Sequential()
        model.add(LSTM(4, return_sequences=True, input_shape=(7, 1)))
        model.add(Dropout(0.2))
        model.add(LSTM(4))
        model.add(Dropout(0.2))
        model.add(Dense(1))
        model.compile(loss='mean_squared_error', optimizer='adam')
        model.fit(x, y, epochs=1000, batch_size=8, verbose=2)
        rm = model.to_json()
        f = open('./proffund.json', 'w')
        f.write(rm)
        model.save_weights('./weights.h5')
    else:
        f = open('./proffund.json', 'r')
        lm = f.read()
        rm = model_from_json(lm)
        rm.load_weights('./weights.h5')
        rm.compile(loss='mean_squared_error', optimizer='adam')
        model = rm
    return model

```

ДОДАТОК В
Слайди презентації

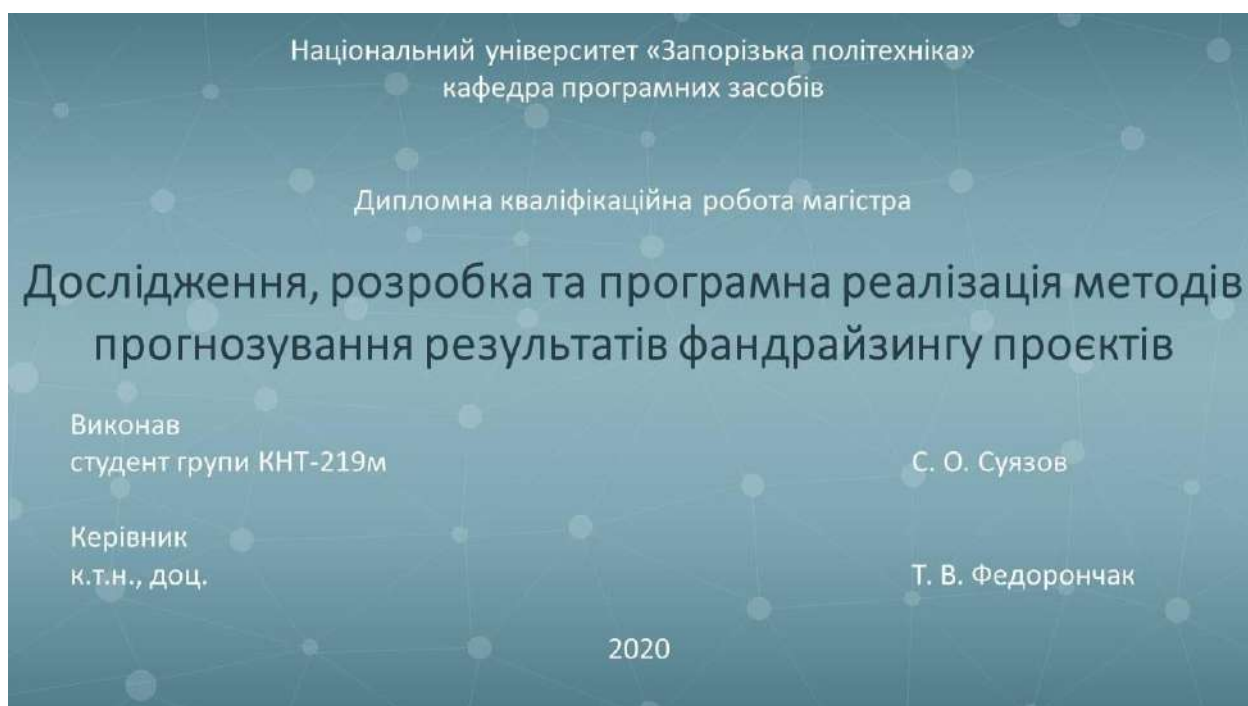


Рисунок В.1 – Слайд 1

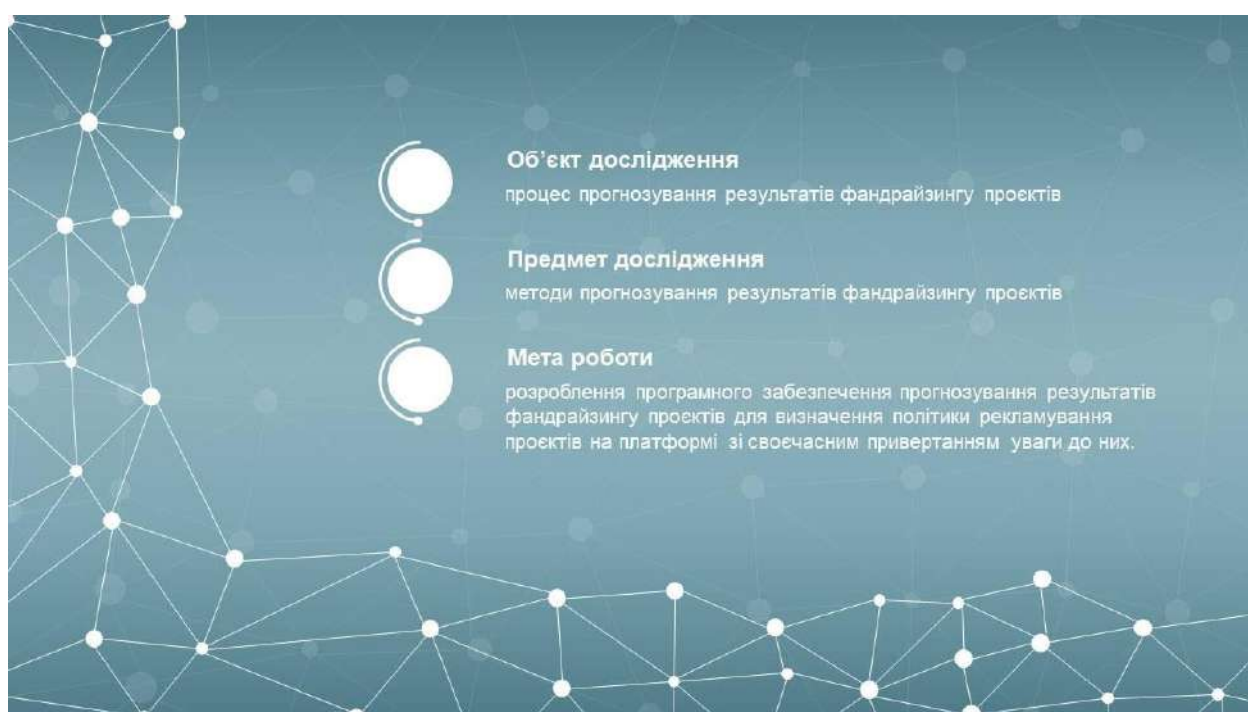


Рисунок В.2 – Слайд 2



Рисунок В.3 – Слайд 3

Порівняння існуючих методів прогнозування/класифікації результатів фандрайзингу проєктів

Критерій порівняння/робота	Tracking the Dynamics in Crowdfunding [Electronic resource] / H. Zhao, H. Zhang, Y. Ge, Q. Liu, E. Chen, H. Li, L. Wu // KDD'17, August 13–17, 2017, Halifax, NS, Canada. – 2017. – Pp. 625-634.	Yeh, J.-Y. A machine learning approach to predict the success of crowdfunding fintech project [Electronic resource] / J.-Y. Yeh, C.-H. Chen // Journal of Enterprise Information Management. – 2020. – Access mode : https://doi.org/10.1108/JEIM-01-2019-0017	Predicting Startup Crowdfunding Success through Longitudinal Social Engagement Analysis [Electronic resource] / Qizhen Zhang, Tengyuan Ye, Meryem Essaidi, Shilvani Agarwal, Vincent Liu, Boon Thau Loo // CIKM'17 : Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, November 6–10, 2017, Singapore. – Singapore : Association for Computing Machinery, 2017. – Pp. 1937-1946. – Access mode : https://doi.org/10.1145/3132847.3132908
Проблема	Прогнозування результатів фандрайзингу проєктів	Прогнозування успішності фандрайзингу проєктів	Прогнозування успішності фандрайзингу проєктів
Тип задачі	Прогнозування	Класифікація	Класифікація
Часовий інтервал	Наступний день	Завершення кампанії	Завершення кампанії
Модель	Регресійна модель	Ансамблі нейронних мереж	Метод к-найближчих сусідів

Рисунок В.4 – Слайд 4



Рисунок В.5 – Слайд 5

Запропонований метод прогнозування результатів фандрайзингу проєктів

На першому етапі виконується підготовка даних. Під час підготовки даних вся множина наявних проєктів має бути представлена у вигляді записів $v(1), v(2), \dots, v(t-d-1), v(t-d), v(t)$. Зі всіх таких наявних записів мають бути виділені навчальна та тестова вибірка. Для кожного запису має бути виконана нормалізація даних.

На другому етапі реалізується створення нейронної мережі архітектури LSTM для моделювання залежності

$$v^c(t) = f(v(t-d), v(t-d-1), \dots, v(1)),$$

де $v^c(t)$ – загальна сума, зібрана в процесі фандрайзингу проєкту, тобто сума, зібрана в момент часу t , що визначає тривалість реалізації фандрайзингу;

$v(t-d)$ – сума коштів, яка була зібрана на реалізацію проєкту в момент часу $t-d$, тобто в останній момент часу, який використовується для подальшого прогнозування;

$v(t-d-1)$ – сума коштів, яку було зібрано в попередній момент $t-d-1$;

$v(1)$ – сума коштів, яку було зібрано в перший день фандрайзингу;

f – функція, яка моделює відповідну залежність.

На наступному етапі виконується навчання моделі на основі створеної навчальної вибірки.

На останньому етапі реалізується прогнозування на основі навченої моделі.

Рисунок В.6 – Слайд 6

Вибір мови програмування

Показник		RНР	Python
01	Основна область застосування	Веброзробка	Загальне призначення, зокрема широко використовується в штучному інтелекті, роботі з даними
02	Фреймворки веброботки	+	+
03	Машинне навчання	Наявні бібліотеки, документація мінімальна	Набір бібліотек, велика спільнота, широка документація
04	Відкритий вихідний текст	+	+

Рисунок В.7 – Слайд 7

Діаграми прецедентів

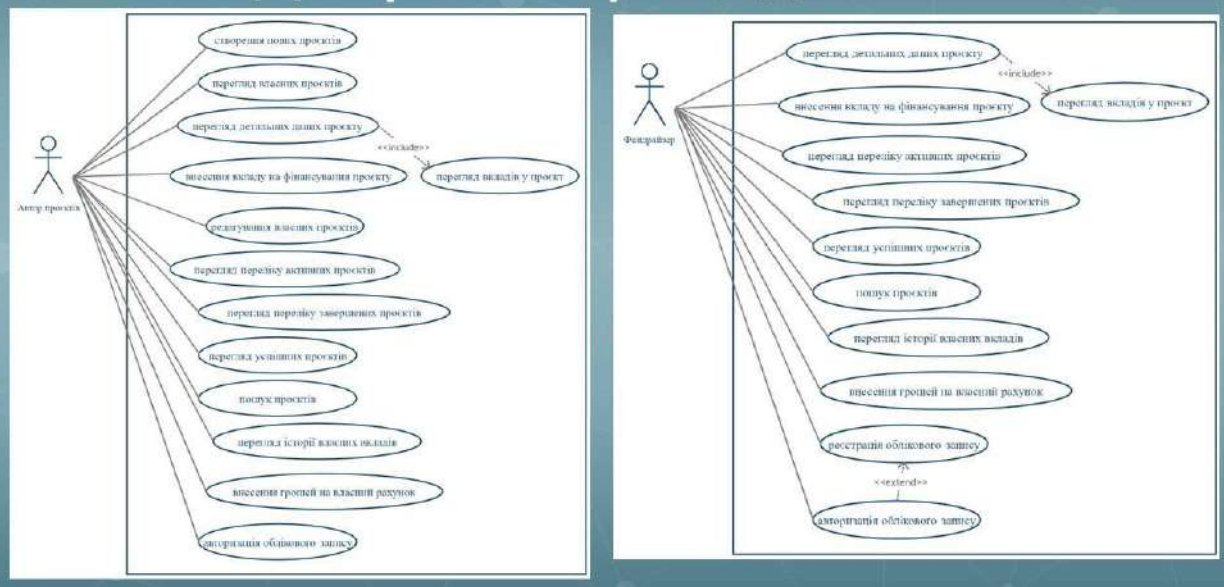


Рисунок В.8 – Слайд 8

Схема бази даних

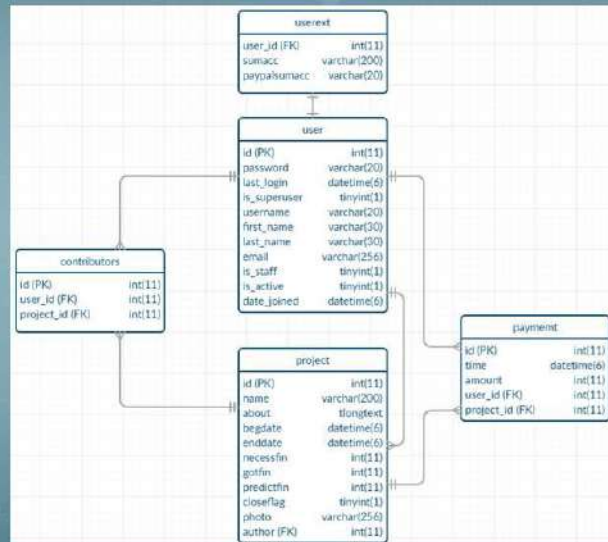


Рисунок В.9 – Слайд 9

Діаграма розгортання

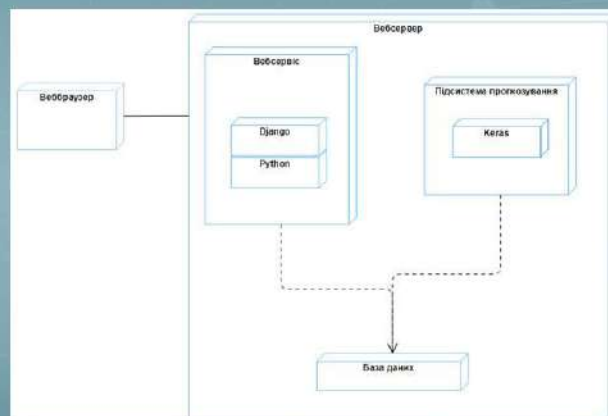


Рисунок В.10 – Слайд 10

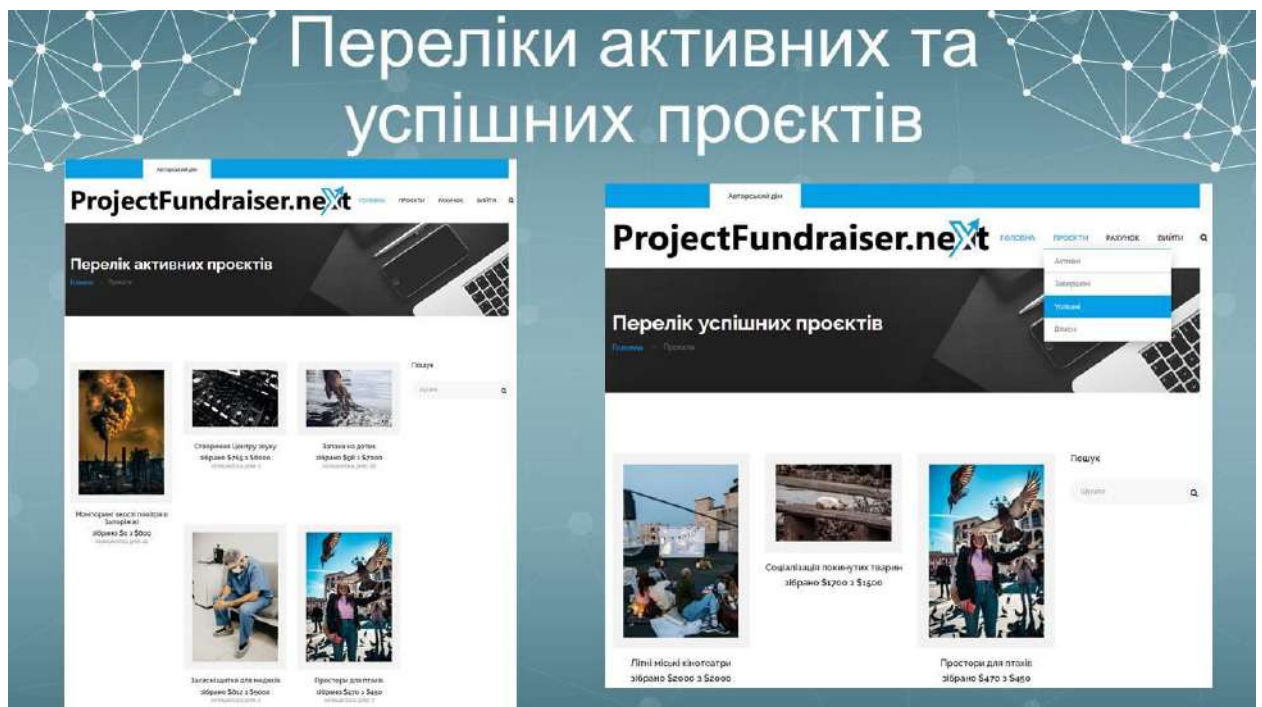


Рисунок В.11 – Слайд 11

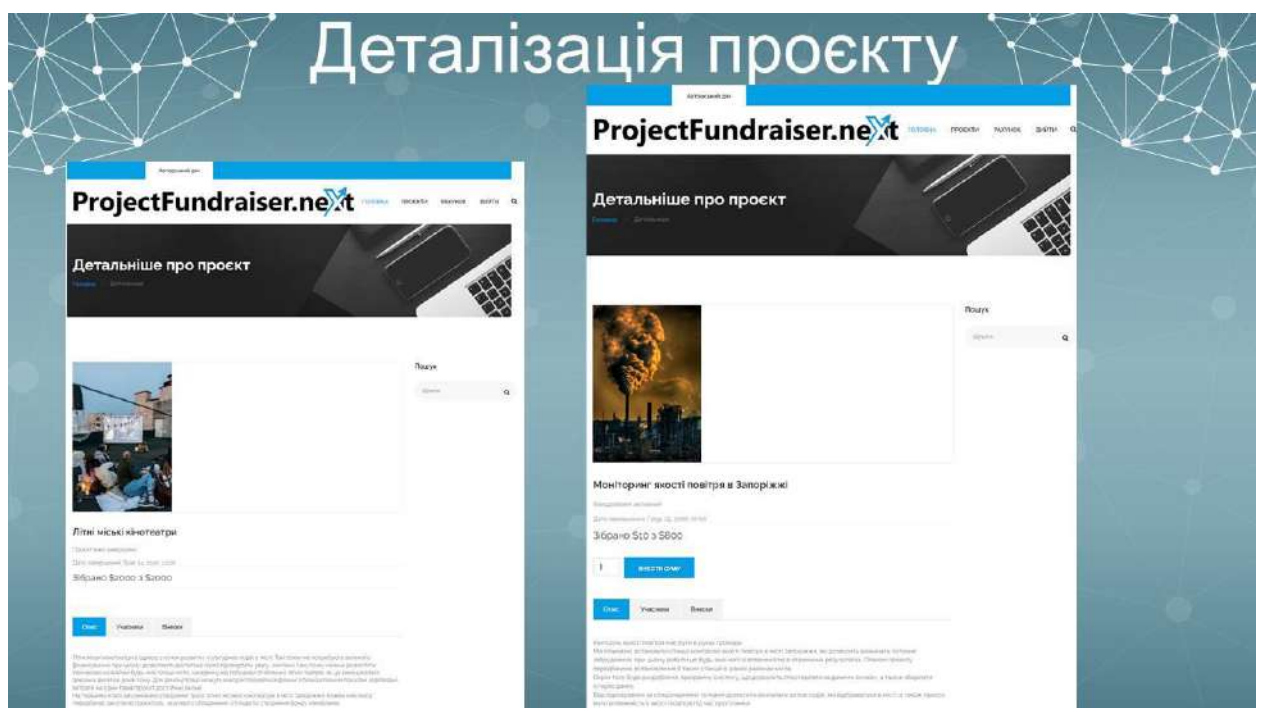


Рисунок В.12 – Слайд 12

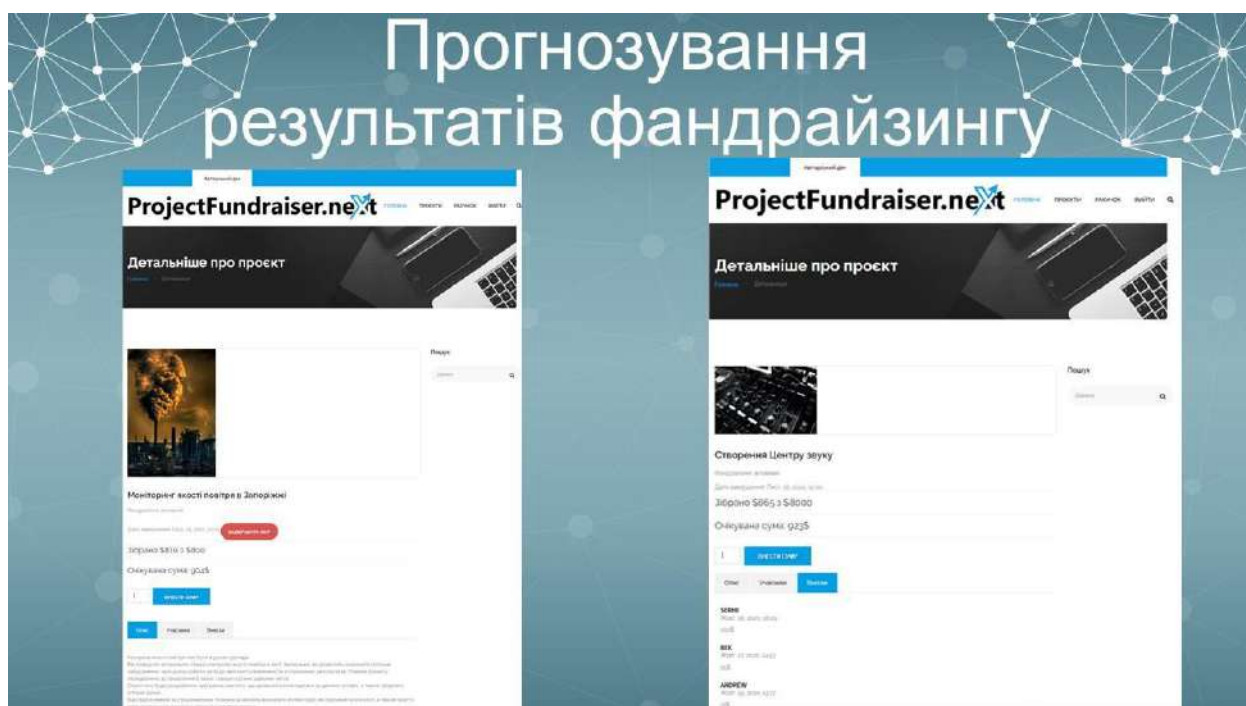


Рисунок В.13 – Слайд 13

Висновки

Запропоновано метод прогнозування результатів фандрайзингу проєктів, який ґрунтується на використанні нейронних мереж з довгою короткостроковою пам'яттю, використовуючи в якості вхідних ознак суми зібраних коштів на проєкт за перші дні збору, що дозволяє визначити підсумкову зібрану на проєкт суму коштів, визначаючи тим самим політику для роботи над фандрайзинговою компанією на ранніх етапах.

Виконано проєктування програмного забезпечення у вигляді вебзастосунку мовою програмування Python. Визначено функціональні вимоги до програмного забезпечення, структуру вебзастосунку. Виконано проєктування бази даних на основі використання системи SQLite.

Реалізовано вебзастосунок, який дозволяє виконувати створення нових проєктів та редагування існуючих, перегляд власних проєктів, детальних даних проєкту, вкладів у проєкт, внесення вкладів на фінансування проєктів, перегляд переліку активних, завершених, успішних проєктів, пошук проєктів, перегляд історії власних вкладів, внесення грошей на власний рахунок та безпосередньо прогнозування результатів фандрайзингу за динамікою перших 7 днів.

Рисунок В.14 – Слайд 14