

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	8
ВСТУП.....	9
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ В ОБЛАСТІ ПІДТРИМКИ ФУНКЦІОНАЛЬНОСТІ КОНТЕНТУ ВЕБ-РЕСУРСУ	10
1.1 Теоретичні відомості про краудфандінг	10
1.2 Аналіз сучасних онлайн-систем краудфандінгу.....	12
2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОСТІ ВЕБ- РЕСУРСУ.....	21
2.1 Вибір програмних засобів для реалізації проекту	21
2.1.1 Переваги та недоліки PHP.....	21
2.1.2 Огляд СУБД MySQL, операторів та команд	27
2.2 Огляд Twitter API та його застосування в дипломному проєкті.....	39
2.3 Розробка структури бази даних	42
3 РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ ЗАСОБАМИ PHP, MySQL....	45
3.1 Створення бази даних та забезпечення зв'язків між її елементами.....	45
3.2 Розробка функціональних можливостей веб-ресурсу	47
3.2.1 Впровадження системи реєстрації та прав доступу.....	47
3.2.2 Меню веб-ресурсу.....	52
3.2.3 Функції додання проєкту та рекламної інформації	54
3.2.4 Статистика репостів реклами.....	56
4 ЕКОНОМІЧНА ДОЦІЛЬНІСТЬ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО ВПРОВАДЖЕННЯ.....	58
4.1 Вплив програмного забезпечення на формування потенціалу соціально - економічних систем	58
4.2 Розрахунок витрат на розробку і використання програмного продукту ...	63
4.3 Обґрунтування дохідності розробки та реалізації програмного продукту	65

					ДП.ПЗ-64.00.00.000 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.		Крохмалюк П.А.			Розробка алгоритмічного та програмного забезпечення підтримки функціональності контенту веб-ресурсу просування та репостингу рекламної інформації Пояснювальна записка	Літ.	Арк.	Акрушіє		
Перевір.		Броновський І.В.						6	76	
Реценз.						ІФНТУНГ ПЗ-09-1				
Н. Контр.		Вовк Р.Б.								
Затверд.		Юрчишин В.М.								

5 ОХОРОНА ПРАЦІ	68
5.1 Значення охорони праці в забезпеченні безпечних та здорових умов праці	68
5.2 Аналіз потенційних небезпек та шкідливих факторів виробничого середовища.....	68
5.3 Забезпечення нормальних умов праці	69
5.4 Пожежна безпека та безпека в надзвичайних ситуаціях	77
ВИСНОВОК.....	80
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА.....	81
ДОДАТКИ	
БІБЛІОГРАФІЧНА ДОВІДКА	

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

СУБД – Система управління базами даних

SQL – Structured Query Language

БД – База даних

API – Application Programming Interface

PHP – Hypertext Preprocessor

GUI – Graphical User Interface

XML – Extensible Markup Language

DOM – Document Object Model

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Інтернет вже давно став невід'ємною частиною життя більшості із нас. З кожним роком кількість його користувачів як в Україні, так і в світі невпинно зростає, а разом з нею зростають можливості глобальної мережі.

Тому створення такого веб-ресурсу обумовлено тим, що все більш популярнішим стає краудфандінг, все частіше ми бачимо це слово на заголовках інтернет видань, журналів та статей, все більше підприємців, початківців чи стартаперів розглядають краудфандінг як якісно нове, безризикове джерело фінансування своїх проєктів. Ця обставина супроводжується все більш інтенсивним розвитком інтернет-майданчиків, де весь світ допомагає реалізувати людям їх ідеї.

Метою мого дипломного проєкту є розробка саме такого веб-ресурсу, який дозволить людям, в яких немає коштів, реалізувати свої проєкти. Більшість таких краудфандингових ресурсів фінансуються благодійними внесками людей, яких зацікавив проєкт. Проєкти на нашому веб-ресурсі будуть фінансуватися рекламодавцями, які розміщуватимуть рекламну інформацію на веб-ресурсі, а користувачі будуть репостити цю рекламну інформацію в соціальній мережі.

Отже, такий веб-ресурс буде служити не тільки як краудфандинговий інтернет-майданчик, але і як ресурс, який дозволить рекламодавцям через користувачів соціальних мереж рекламувати свої товари.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ В ОБЛАСТІ ПІДТРИМКИ ФУНКЦІОНАЛЬНОСТІ КОНТЕНТУ ВЕБ-РЕСУРСУ

1.1 Теоретичні відомості про краудфандінг

Краудфандінг являє собою особливий вид фінансування за допомогою збору коштів на реалізацію будь-якого проекту через Інтернет, наприклад, створення фільму, книги, музичного альбому, комп'ютерної гри, як правило, творчої продукції. Краудфандінг не є благодійністю. Кожен спонсор, передаючи гроші, отримує натомість певний бонус, наприклад, диск з фільмом, книгу з автографом або запрошення на знімальний майданчик. Часто крауд-інвестор отримує те, що не можна купити в магазині, наприклад, пам'ятні речі та емоції.

Між краудфандінгом та мікрофінансуванням існує принципова різниця. У випадку з мікрофінансуванням мова йде про кошти на розширення мікроскопічного бізнесу, який ніколи не зацікавить банки, по-перше, зважаючи на незначний обсяг коштів, які необхідно залучити, а по-друге, зважаючи на високі ризики і в основному відсутність будь-яких гарантій. Адже мікrokредити, як правило, беруть підприємці, у яких не вистачає коштів на розвиток свого малого бізнесу.

Якщо говорити про краудфандінг, то тут справа стосується більш творчих, тонких та інноваційних проектів, які, тим не менш, вимагають інвестицій. Таким чином, аудиторії мікрофінансових і краудфандінгових організацій абсолютно різні і обидві вони поки що нецікаві банкам та фінансовим компаніям. Проте в найближчій перспективі конкурентами краудфандінгу можуть стати інвестиційні фонди. Це відбудеться, коли інвестиційний краудфандінг набере обертів, і підприємці зможуть відразу звернутися до майбутніх покупців або приватних інвесторів, обходячи класичні способи залучення коштів у нові компанії або просто ідеї, які динамічно розвиваються. Експерти припускають, що надалі краудфандінгові фонди зможуть створювати і самі банки. Однак про те, коли підприємці зможуть звернутися до краудфандінгу, складно визначити.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Поки що краудфандінгом користуються переважно молоді підприємці, представники ІТ-індустрії та люди творчих професій. Якщо виключити новинні ресурси та мікроблоги, куди користувачі копіюють контент з інших соціальних медіа, то перші п'ять ресурсів – це популярні блоги, орієнтовані в тому числі на зарубіжну аудиторію, ВКонтакте – найбільш популярна соціальна мережа рунету, а Хабрахабр – закрите співтовариство ІТ-професіоналів.

Як і в будь-якій новій галузі, основні проблеми краудфандінга полягають у донесенні інформації до потрібної аудиторії. Якщо говорити про потенційних спонсорів, то потрібна аудиторія – це прогресивно мислячі люди, які цікавляться новинками в технологіях, культурі та розвагах. Крім цього, даній категорії людей важливо брати участь у чомусь цікавому і важливому або, більше того, бути причетним. Що стосується авторів проектів, то тут мова може йти практично про будь-яку людину або організацію, які здатні створити щось незвичайне і зацікавити публіку.

Теоретично, краудфандінг, крім інструменту для збору коштів, також виступає інструментом для аналізу попиту на створюваний продукт, і навіть інструментом для піару. Механізм простий – чим більше коштів вдалося залучити, тим вищий потенційний попит, чим більше заявив про себе, поки збирав кошти, тим більший так званий піар-вихлоп. Але знову ж таки, тут слід врахувати популярність автора проекту.

Звичайно, якщо говорити про проблеми краудфандінга і якщо при цьому копати глибоко, то варто згадати і низьку фінансову грамотність населення, і відсутність інтересу до культури, і відсутність закону про краудфандінг, якого поки що немає навіть в США. У січні 2013 року в США повинен був вступити в силу закон про форсований запуск бізнес-стартапів або JOBS Act. Саме цей закон повинен був легалізувати краудфандінг і дозволити американцям не просто жертвувати гроші на цікаві їм проекти, але й інвестувати їх, тобто вкладати з метою отримання прибутку. Однак його прийняття не відбулося.

На думку аналітиків, у країнах, що розвиваються, поки ще не дозріла потреба в легалізації інвестування на основі краудфандінгу. В основному за

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

допомогою цієї моделі зараз залучають кошти соціальні проекти, що не спрямовані на отримання прибутку і не придатні для народного інвестування. Експерти не очікують легалізації акціонерного краудфандінгу на горизонті 2-х-3-х років. Проте у Європі краудфандінг вже узаконено. Якщо він стане коли-небудь законним і у нас, то можна буде очікувати появи великої кількості організацій, що спеціалізуються на цьому виді фінансування і стартапів, а також спрощення процесу створення бізнесу для людей, що мають на руках тільки хорошу ідею й нічого більше.

Основні проблеми краудфандінга пов'язані, по-перше, з недостатньо розвиненою інфраструктурою, а по-друге, з відсутністю інтересу з боку можливих вкладників, що обумовлено низькою фінансовою грамотністю населення. До основних труднощів можна віднести те, що не повсюдно є хороший інтернет, а також не всі знають, як платити банківськими картками в інтернеті. Разом з тим, у країнах, що розвиваються, краудфандінг має гарні перспективи для розвитку, враховуючи ще ненасичений ринок соціальних медіа, які є головним інструментом у цьому методі фінансування.

Основним ризиком краудфандінга багато експертів вважають шахрайство, оскільки під виглядом благодійної ініціативи або цікавого проекту може бути організований збір коштів, які в підсумку не дійдуть до заявленого адресата і осядуть в руках зловмисників.

1.2 Аналіз сучасних онлайн-систем краудфандінгу

У 1997 році шанувальники британської рок-групи Marillion без будь-якого залучення самої групи організували і провели інтернет-кампанію по збору коштів для фінансування музичного туру групи по всій території США. Їм вдалося зібрати \$60 000. Пізніше група використовували такий метод для запису і просування кількох своїх альбомів, зокрема Anoraknophobia, Marbles і Happiness Is the Road.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Заснована в США компанія ArtistShare (2000/2001) задокументована як перший краудфандинговий сайт для музики. Слідом за ним з'явилися такі сайти, як Sellaband (2006), SliceThePie (2007), Hyper Funding (2008), IndieGoGo (2008), Pledge Music (2009), Kickstarter (2009), RocketHub (2009), FundaGeek (2011) і in the UK Sponsume (2010), PleaseFund.Us (2011), Authr.com (2012) і OnSetStart (2012).

Краудфандинг в кіноіндустрії був започаткований підприємцем Еріком Бауманом із запуском FilmVenture.com в 2002 році. А через 2 роки французькі підприємці та виробники Бенджамін Помера і Гійом Колбок з компанії Гійом Корп. почали компанію зі збору пожертв в інтернеті у серпні 2004 року, щоб дозняти свій фільм *Demain la Veille* (Очікування вчора). Протягом трьох тижнів, їм вдалося назбирати \$50000, що дозволило їм знімати своє кіно. Це було перше структуроване інтернет краудфандингове фінансування ініціативи із використанням присвяченого фінансуванню веб-сайту, а також різними пропозиціями-подяками для своїх жертводавців: бонуси, DVD або навіть присутність на зйомках.

Чотири місяці по тому, на іншій стороні Атлантики, компанія Spanner Films почала виробництво свого документального фільму про зміну клімату 'The Age of Stupid' (Епоха дурнів). Команді, очолюваній Френні Армстронгом успішно вдалося зібрати більш, ніж \$900 000 впродовж п'яти років (з грудня 2004 року до 2009 рік, дата випуску), для виробництва і просування фільму. Вся знімальна команда працювала за дуже низьку заробітну плату, але окрім цього отримувала також краудфандингові «акції». Відповідно до умов контракту краудфандингу інвесторам і членам знімальної команди один раз на рік протягом десяти років з моменту виходу фільму виплачується дивіденди.

Morton Valence була однією з перших порівняно маловідомих груп, що самостійно вступила у краудфандинг без використання спеціальних сайтів таких, як sellaband.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

The Cosmonaut є ще одним прикладом краудфандінгу в кіноіндустрії: їхня кампанія “Врятуй The Cosmonaut” лише за перший тиждень залучила €130,000. Протягом ж усього часу фільм лише зібрав €300,000 пожертв.

The Professional Contractors Group, торгова асоціація для фрілансерів у Великобританії була заснована в інтернеті в 1999, коли Енді Уайт зробив заклик до 2000 підрядників пожертвувати £50 щоб назбирати £100,000, які необхідні для заснування такого типу організації. Через 5 днів 2002 людей перерахували гроші і організація народилася. Сьогодні її членами є більше 14 000 британців.

А станом на даний час, суми, які можна залучити через краудфандінг значно зросли. У 2012 році проекти створення відеоігор Double Fine Adventure і Wasteland 2 зібрали по \$3 336 371 і \$3 040 299 через краудфандінговий сайт Kickstarter. Внески зробили більше 87 000 людей.

Kickstarter — це сайт фінансування творчих проектів за схемою краудфандінгу. Kickstarter фінансує різноманітні проекти, у 13-ох категоріях: мистецтво, комікс, танець, дизайн, мода, фільми і відео, їжа, відеоігри, музика, фотографія, видавництво, технологія, театр. За 3 роки існування більш ніж 1 800 000 людей своїми пожертвами повністю успішно профінансували понад 20 000 проектів, зібравши більше \$200 000 000.

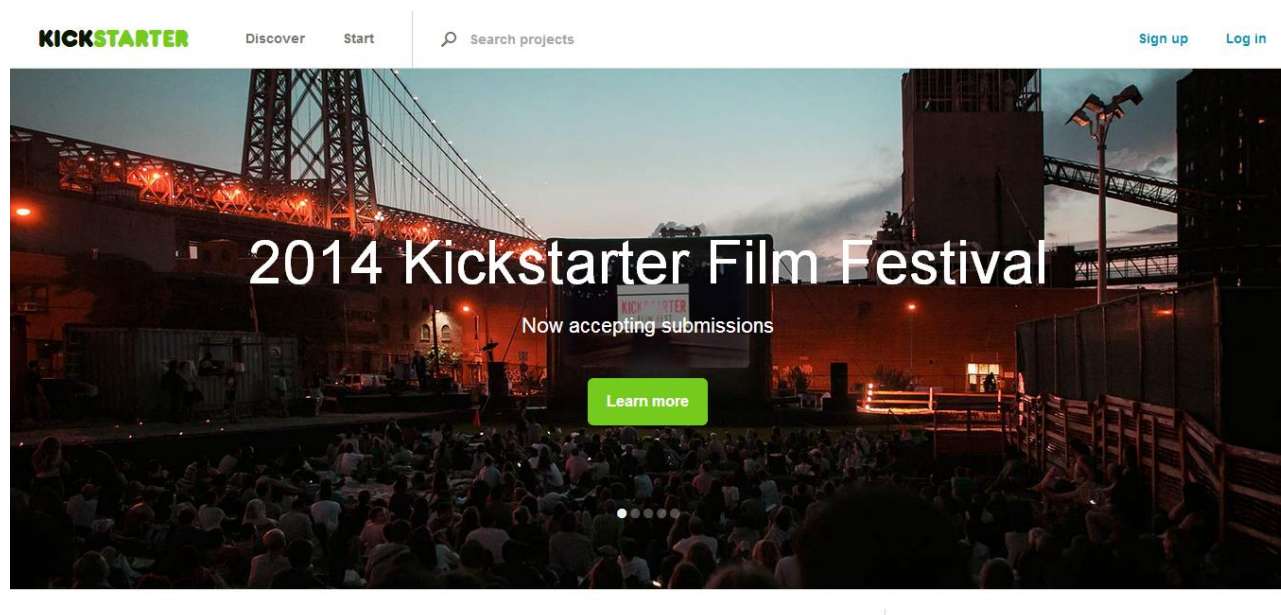


Рисунок 1.1 – Головна сторінка ресурсу kickstarter.com

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Kickstarter полегшує збір коштів, створивши модель, яка може бути кращою за традиційні способи інвестування. Той, хто хоче отримати фінансування, повинен зареєструватися і розмістити опис проекту на Kickstarter. Kickstarter містить рекомендації для проектів, які будуть прийняті.

Власник проекту повинен вказати останній термін (макс. 90 днів) і мінімальну суму коштів, яку необхідно зібрати. Якщо проект не зібрав потрібну кількість коштів до кінця терміну, то гроші повертаються жертводавцям. Гроші збираються за допомогою Amazon Payments, ініціатору проекту потрібно мати рахунок в американському банку.

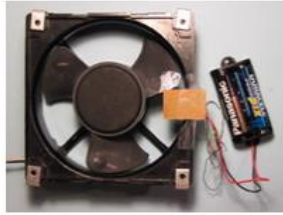
Kickstarter бере 5% від залучених коштів; Amazon стягує додаткові 3-5%. На відміну від багатьох форумів по збору коштів або інвестицій, Kickstarter не претендує на право власності на проекти і роботи, яку вони виробляють. Тим не менш, проекти, здійснювані на сайті, зберігаються і доступні для громадськості. Після того, як фінансування проектів завершується завантажена інформація і матеріали не можуть бути відредаговані або видалені з сайту.

Кожен проект мусить проходити модерацію, і спостерігається тенденція що організатори здебільшого відбирають тільки потенційно найуспішніші проекти. Котрі обов'язково мають відео, обіцяють своїм інвесторам найбільші нагороди, бенефіти, віддяки. Наприклад, готовий примірник чи декілька примірників гри, коміксу, книги; гравіювання ім'я інвестора тощо. Також на сайті дуже важко шукати неуспішні проекти, що не здобули фінансування. Саме цим частково і пояснюється величезний показник успішно профінансованих проектів. Згідно з цими позиціями сайт надміру сконцентрований на успішності і є надто комерціалізованим, тому виступає швидше біржею чи магазином передоплати продуктів мистецтва, які сподобалися.

Іншим пунктом критики є принцип «Все або нічого», за яким гроші повертаються, якщо проект не назбирав 100% потрібних коштів. Це спонукає авторів самим дофінансовувати свої проекти. За приблизними підрахунками точкою неповернення є 25%. Якщо проект їх назбирав, то імовірність цілковитого фінансування і успішності проекту становить 90%.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Staff Picks



Low Voltage Metal Sensor for use with Arduino type boards
by Micro Oscillator, Inc.

Low Voltage Metal Sensor directly compatible with Arduino type computers for Robotics, & Motor Control, WITHOUT USING MAGNETS!

Philadelphia, PA

123% funded \$1,845 pledged 6 days to go



Unique wooden handbag. Dutch Design. Handmade.
by Leon de Haan

A flexible, elegant and strong wooden handbag for women. Easily stores your tablet and cell phone and looks gorgeous!

Amsterdam, Netherlands

36% funded €7,379 pledged 20 days to go



Bowls of Love: Paleo Soups for the Seasons
by Ali Rakowski

Bowls of Love is a Paleo soup cookbook full of colorful, delicious, and healthy recipes for all of the seasons and people in your life.

Stamford, CT

103% funded \$10,382 pledged 22 days to go

[See more staff picks](#)

▶ Video mode

★ Featured

Advanced **NEW**

Staff Picks

Popular

Recently Launched

Ending Soon

Most Funded

Curated Pages

🔖 Categories

Art

Comics

Dance

Design

Fashion

Film & Video

Food

Games

Music

Photography

Рисунок 1.2 – Сторінка з проектами ресурсу kickstarter.com

ВЕЛИКА ІДЕЯ є найбільшою краудфандинговою платформою в Україні. Вона побудована за принципом learn and do. В розділі сайту Практики, відслідковуються актуальні тренди та висвітлюються міждисциплінарні рішення, поєднуючи теорію та практику з ідеями соціальних змін. Спільнокошт - онлайнвий інструмент фінансування проектів, який пропонує соціальним інноваторам та творчим людям звернутися за підтримкою до аудиторії, яка зацікавлена в реалізації їхнього проекту. Цікаво, що редизайн сайту Велика Ідея, в результаті якого мав би запрацювати інструмент краудфандингу, був профінансований людьми на суму 38 260 гривень. У жовтні 2012 року відбувся офіційний старт спільнокошту.

Змн.	Арк.	№ докум.	Підпис	Дата

ДП.ПЗ-64.00.00.000 ПЗ

Арк.

16

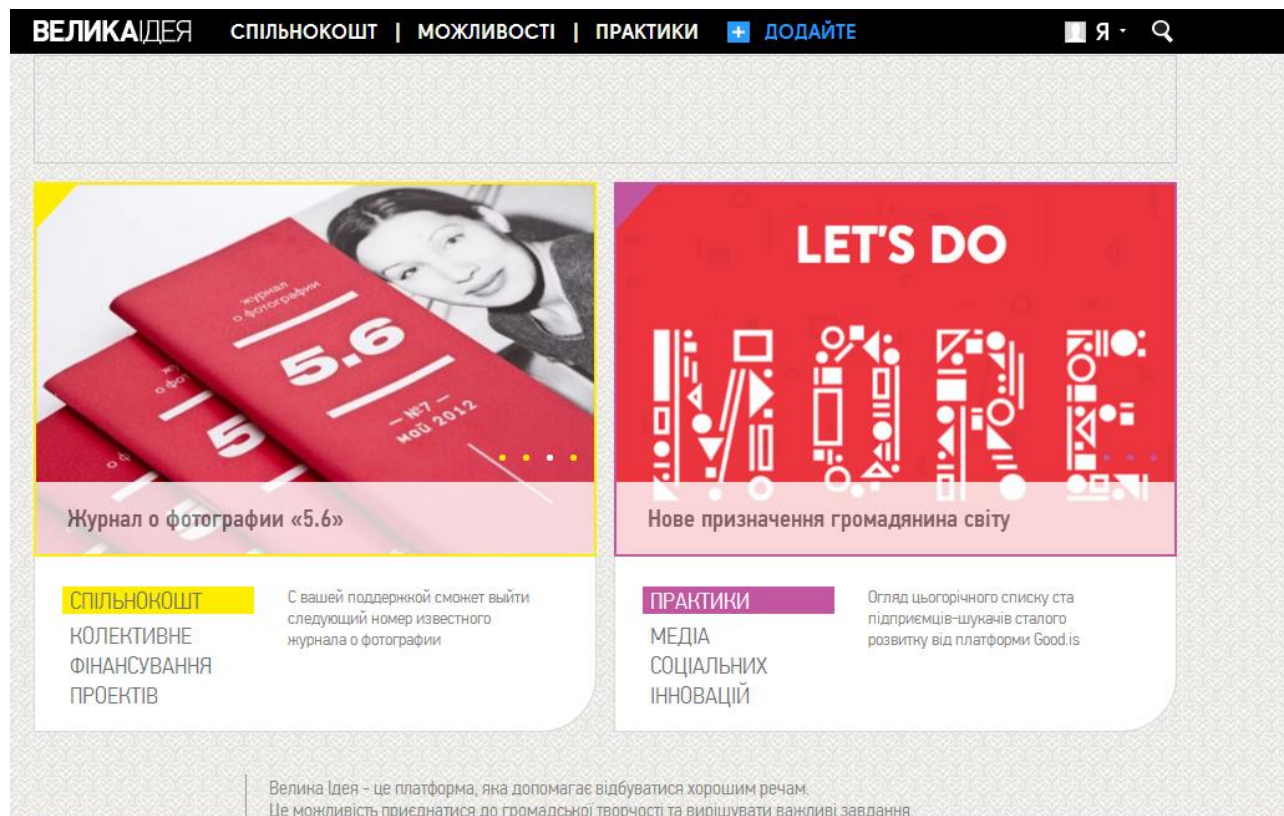


Рисунок 1.3 – Головна сторінка ресурсу bigggidea.com

Спільнокошт пропонує людям взятися за здійснення власних проєктів, розраховуючи на матеріальну підтримку друзів, колег та товариств, зацікавлених у позитивних змінах. Механізм Спільнокошт захищає ваші добродійні наміри, діючи за принципом «все або нічого»: тільки той проєкт, що зібрав 100% потрібного бюджету до завершення зазначеного терміну, отримає кошти на здійснення. Якщо ж проєкт не зміг вчасно зібрати потрібну суму, всі кошти повернуться добродійникам.

BIGIDEA
СПІЛЬНОКОШТ | МОЖЛИВОСТІ | ПРАКТИКИ
+ ДОДАЙТЕ
Я

екологія

Зробимо Україну чистою

РОМАН БРИТАНЧУК

Наймасштабніше громадське прибирання в Україні, яке щороку об'єднує тих, хто дбає про довкілля

101%
101%
ПІДТРИМАНО

50328 грн
50328 грн
ЗІБРАНО

13.04.2014
13.04.2014
ПРОФІНАНСОВАНО

медіа

Підтримай Громадське

КОМАНДА ГРОМАДСЬКОГО

Допомогти Громадському забезпечити безперебійну роботу у 2014 році

124%
124%
ПІДТРИМАНО

1243381 грн
1243381 грн
ЗІБРАНО

06.04.2014
06.04.2014
ПРОФІНАНСОВАНО

Рисунок 1.4 – Розділ ресурсу biggggidea.com - Спільнокошт

Boomstarter – це платформа, орієнтована на залучення фінансування в креативні проекти через краудфандінг.

Тисячі людей готові підтримати авторські починання в таких областях, як музика, фільми, ігри, мистецтво, нові технології, промисловий дизайн і інших. Безліч успішних проектів є яскравим підтвердженням цього.

Абсолютно новий підхід до реалізації творчого потенціалу та до залучення фінансування через краудфандінг.

Цікаві оригінальні, новаторські, сучасні ідеї, які можуть бути корисні для суспільства. Ресурс надає людям можливість підтримати цікаві їм проекти, а натомість отримати унікальні продукти, бонуси або незвичайний досвід.

Тільки повне фінансування проектів. Чому? Таким чином виключаються можливі негативні наслідки: очевидно, що нестача коштів може призвести до

неякісної реалізації проекту. Якщо проект не приваблює потрібну суму грошей, то всі зібрані кошти моментально повертаються спонсорам.

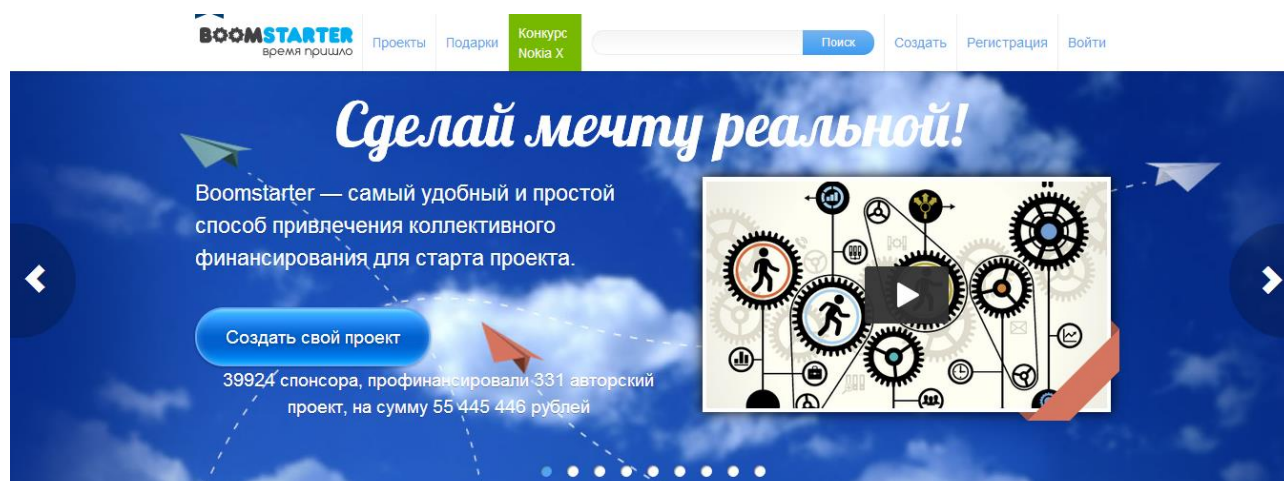


Рисунок 1.5 – Головна сторінка ресурсу boomstarter.ru



Рисунок 1.6 – Сторінка з описом вибраного проекту

Платформа створена для самих різних проектів: великих і малих, серйозних і не дуже, традиційних та експериментальних. Будь-хто може реалізувати свою

Змн.	Арк.	№ докум.	Підпис	Дата

ДП.ПЗ-64.00.00.000 ПЗ

Арк.

19

ідею, будь то запис музичного альбому, зйомка фільму або створення нового приладу. Потрібні тільки фантазія, бажання і трохи таланту.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОСТІ ВЕБ-РЕСУРСУ

2.1 Вибір програмних засобів для реалізації проекту

2.1.1 Переваги та недоліки PHP

Можливості PHP дуже великі. Головним чином, область застосування PHP сфокусована на написання скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI. Наприклад, обробляти дані форм, генерувати динамічні сторінки, посилати і приймати cookies. Але PHP здатний виконувати і безліч інших завдань. Існують три основні області, де використовується PHP.

- Створення скриптів для виконання на стороні сервера. PHP найбільш широко використовується саме таким чином. Все, що вам знадобиться, це парсер PHP (у вигляді програми CGI або серверного модуля), веб-сервер і браузер. Щоб ви могли переглядати результати виконання PHP-скриптів в браузері, вам потрібен працюючий веб-сервер і встановлений PHP.

- Створення скриптів для виконання в командному рядку. Ви можете створити PHP-скрипт, здатний запускатися незалежно від веб-сервера і браузера. Все, що вам буде потрібно - парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою cron (на платформах *nix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в задачах простої обробки текстів.

- Створення додатків GUI, що виконуються на стороні клієнта. Можливо, PHP є не найкращою мовою для створення подібних додатків, але, якщо ви дуже добре знаєте PHP і хотіли б використовувати деякі його можливості в своїх клієнт-додатках, ви можете використовувати PHP-GTK для цього. Подібним чином Ви можете створювати і крос-платформні додатки. PHP-GTK є розширенням PHP і не поставляється разом з дистрибутивом PHP.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

PHP доступний для більшості операційних систем, включаючи Linux, багато модифікації Unix (такі, як HP-UX, Solaris і OpenBSD), Microsoft Windows, Mac OS X, RISC OS, та багатьох інших. (Існує навіть версія PHP для OS/2. Невідомо, правда, наскільки відповідна нинішнім реаліям). Також в PHP включена підтримка більшості сучасних веб-серверів, таких, як Apache, Microsoft Internet Information Server, Personal Web Server, серверів Netscape та iPlanet, сервера Oreilly Website Pro, Caudium, Xitami, OmniHTTPd та багатьох інших. Для більшості серверів PHP поставляється в якості модуля, для інших, що підтримують стандарт CGI, PHP може функціонувати як процесор CGI.

Таким чином, вибираючи PHP, ви отримуєте свободу вибору операційної системи та веб-сервера. Крім того, у вас з'являється вибір між використанням процедурного або об'єктно-орієнтованого програмування або ж їх поєднання.

PHP здатний не лише видавати HTML. Можливості PHP включають формування зображень, файлів PDF і навіть роликів Flash (з використанням libswf і Ming), створених "на льоту". PHP також здатний видавати будь-які текстові дані, такі, як XHTML та інші XML-файли. PHP здатний здійснювати автоматичну генерацію таких файлів і зберігати їх у файловій системі вашого сервера, замість того, щоб віддавати клієнту, організовуючи, таким чином, кеш динамічного змісту, розташований на стороні сервера.

Одним із значних переваг PHP є підтримка широкого кола баз даних. Створення скрипта, що використовує бази даних - дуже просто. В даний час PHP підтримує наступні бази даних: MySQL, Adabas D, Ingres, Oracle (OCI7 і OCI8), dBase, InterBase, Ovrimos, Empress, FrontBase, PostgreSQL, FilePro (тільки читання), mSQL, Solid, Hyperwave, Direct MS-SQL, Sybase, IBM DB2, Velocis, Informix, ODBC, Unix dbm.

Також в PHP включена підтримка DBX для роботи на абстрактному рівні, так що ви можете працювати з будь-якою базою даних, що використовують DBX. Крім того, PHP підтримує ODBC (Open Database Connection standard), таким чином, ви можете працювати з будь-якою базою даних, що підтримує цей всесвітньо визнаний стандарт.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

PHP також підтримує "спілкування" з іншими сервісами з використанням таких протоколів, як LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) і багатьох інших. Крім того, ви отримуєте можливість працювати з мережевими сокетом "безпосередньо". PHP підтримує стандарт обміну складними структурами даних WDDX. Звертаючи увагу на взаємодію між різними мовами, слід згадати про підтримку об'єктів Java і можливості їх використання в якості об'єктів PHP. Для доступу до віддалених об'єктів ви можете використовувати розширення CORBA.

PHP включає засоби обробки текстової інформації, починаючи з регулярних виразів Perl або POSIX Extended і закінчуючи парсером документів XML. Для парсинга XML використовуються стандарти SAX і DOM. Для перетворення документів XML ви можете використовувати розширення XSLT.

Останнім по порядку, але не за значенням, є підтримка багатьох інших розширень, таких, як функції пошукової машини mnoGoSearch, функції IRC Gateway, функції для роботи зі стиснутими файлами (gzip, bz2), функції календарних обчислень, функції перекладу і багато іншого.

PHP може запропонувати дуже багато переваг перед іншими мовами.

Головним чинником мови PHP є практичність. PHP повинна надати програмісту інструмент для швидкого і ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками:

- традиційністю
- простотою
- ефективністю
- безпекою
- гнучкістю

Існує ще одна «характеристика», яка робить PHP особливо привабливим: він розповсюджується безкоштовно! Причому, з відкритими вихідними кодами (Open Source).

Традиційність. Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкції мови запозичені з Cі, Perl.

Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP - мова, що поєднує достоїнства Perl, Cі і спеціально націлена на роботу в Інтернеті, мова з універсальним (правда, за деякими застереженнями) і ясним синтаксисом. І хоча PHP є досить молодого мовою, вона знайшла таку популярність серед web-програмістів, що на даний момент є мало чи не найпопулярнішою мовою для створення web-додатків (скриптів).

Простота. Сценарій PHP може складатися з 10 000 рядків або з одного рядка - все залежить від специфіки вашої задачі. Вам не доведеться довантажувати бібліотеки, вказувати спеціальні параметри компіляції або щонебудь в цьому дусі. Механізм PHP просто починає виконувати код після першої послідовності (`<?php`) і продовжує виконання до того моменту, коли вона зустріне парну послідовність (`?>`). Якщо код має правильний синтаксис, він виконується в точності так, як вказав програміст.

PHP - мова, яка може бути вбудована безпосередньо в html-код сторінок, які, у свою чергу будуть коректно оброблятися PHP-інтерпретатором. Ми можемо використовувати PHP для написання CGI-сценаріїв і позбутися від безлічі незручних операторів виведення тексту. Ми можемо залучати PHP для формування HTML-документів, позбавившись від безлічі викликів зовнішніх сценаріїв.

Велика різноманітність функцій PHP позбавлять вас від написання багаторядкових призначених для користувача функцій на C або Pascal.

Ефективність. Ефективність є виключно важливим чинником при програмуванні для розрахованих на багато користувачів середовищ, до числа яких належить і web.

Дуже важлива перевага PHP полягає в його «движку». «Движок» PHP не є ні компілятором, ні інтерпретатором. Він є транслуючим інтерпретатором.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

Такий пристрій «движка» РНР дозволяє обробляти сценарії з достатньо високою швидкістю.

За деякими оцінками, більшість РНР-сценаріїв (особливо не дуже великих розмірів) обробляються швидше аналогічних їм програм, написаних на Perl. Проте, щоб не робили розробники РНР, виконувані файли відкомпільований будуть працювати значно швидше - в десятки, а іноді і в сотні разів. Але продуктивність РНР цілком достатня для створення цілком серйозних web-додатків

Безпека. РНР надає в розпорядження розробників та адміністраторів гнучкі та ефективні засоби безпеки, які умовно поділяються на дві категорії: засоби системного рівня та засоби рівня програми.

Засоби безпеки системного рівня:

У РНР реалізовані механізми безпеки, що знаходяться під управлінням адміністраторів; при правильному налаштуванні РНР це забезпечує максимальну свободу дій і безпеку. РНР може працювати в так званому безпечному режимі (safe mode), який обмежує можливості застосування РНР користувачами по ряду важливих показників. Наприклад, можна обмежити максимальний час виконання та використання пам'яті (неконтрольована витрата пам'яті негативно впливає на швидкодію сервера). За аналогією з cgi-bin адміністратор також може встановлювати обмеження на каталоги, в яких користувач може переглядати та виконувати сценарії РНР, а також використовувати сценарії РНР для перегляду конфіденційної інформації на сервері (наприклад, файлу passwd).

Засоби безпеки рівня програми:

У стандартний набір функцій РНР входить ряд надійних механізмів шифрування. РНР також сумісний з багатьма додатками незалежних фірм, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції (e-commerce). Інша перевага полягає в тому, що вихідний текст сценаріїв РНР не можна переглянути у браузері, оскільки сценарій компілюється до його відправлення за запитом користувача. Реалізація РНР на стороні сервера

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

запобігає викрадення нетривіальних сценаріїв користувачами, знань яких вистачає хоча б для виконання команди View Source.

Гнучкість. Оскільки PHP є вбудовуваною (embedded) мовою, вона відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча PHP зазвичай рекомендується використовувати у поєднанні з HTML, вона з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови. Крім того, добре структуровані програми PHP легко розширюються в міру необхідності (втім, це відноситься до всіх основних мов програмування).

Немає проблем із залежністю від браузерів, оскільки перед відправкою клієнту сценарії PHP повністю компілюються на стороні сервера. По суті, сценарії PHP можуть передаватися будь-яким пристроям із браузерами, включаючи стільникові телефони, електронні записні книжки, пейджери і портативні комп'ютери, не кажучи вже про традиційні ПК. Програмісти, які розробляють допоміжні утиліти, можуть виконувати PHP-код в режимі командного рядка.

Оскільки PHP не містить коду, орієнтованого на конкретний web-сервер, користувачі не обмежуються певними серверами (можливо, незнайомими для них). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold і Zeus - PHP працює на всіх перерахованих серверах. Оскільки ці сервери працюють на різних платформах, PHP в цілому є платформо-незалежним мовою і існує на таких платформах, як UNIX, Solaris, FreeBSD і Windows 95/98/NT/2000/XP/2003.

Нарешті, можливості PHP дозволяють програмісту працювати із зовнішніми компонентами, такими як Enterprise Java Beans або COM-об'єкти Win32. Завдяки цим новим можливостям PHP займає гідне місце серед сучасних технологій і забезпечує масштабування проектів до необхідних меж.

Безкоштовне розповсюдження. Стратегія Open Source, і розповсюдження початкових текстів програм в масах, зробило поза сумнівом благодетельний вплив на багато проектів, в першу чергу - Linux, хоча і успіх проекту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

створення PHP, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником в розвитку проекту PHP.

Прийняття стратегії Open Source і безкоштовне розповсюдження початкових текстів PHP надало неоціниму послугу користувачам. До того ж, чуйне співтовариство користувачів PHP є свого роду «колективною службою підтримки», і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання.

2.1.2 Огляд СУБД MySQL, операторів та команд

MySQL - це одна з найпопулярніших і найпоширеніших СУБД (система управління базами даних) в інтернеті. Вона не призначена для роботи з великими обсягами інформації, але її застосування ідеально для інтернет сайтів, як невеликих, так і досить великих.

MySQL відрізняється хорошою швидкістю роботи, надійністю, гнучкістю. Робота з нею, як правило, не викликає великих труднощів. Підтримка сервера MySQL автоматично включається в поставку PHP.

Структуровані мова запитів SQL дозволяє робити різні операції з базами даних: створювати таблиці, поміщати, оновлювати і видаляти з них дані, виробляти запити з таблиць тощо. Далі ми послідовно розглянемо всі ці оператори.

Оператор SELECT має наступну структуру:

```
SELECT [STRAIGHT_JOIN]
      [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
      [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
[HIGH_PRIORITY]
      [DISTINCT | DISTINCTROW | ALL]
select_expression, ...
      [INTO {OUTFILE | DUMPFILE} 'file_name' export_options]
      [FROM table_references
      [WHERE where_definition]
      [GROUP BY {unsigned_integer | col_name | formula} [ASC |
DESC], ...]
```

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

```

[HAVING where_definition]
[ORDER BY {unsigned_integer | col_name | formula} [ASC |
DESC], ...]
[LIMIT [offset,] rows]
[PROCEDURE procedure_name]
[FOR UPDATE | LOCK IN SHARE MODE]]

```

SELECT застосовується для вилучення рядків, вибраних з однієї або декількох таблиць. Вираз `select_expression` задає стовпці, в яких необхідно проводити вибірку. Крім того, оператор SELECT можна використовувати для витягання рядків, обчислених без посилання якусь таблицю. Наприклад:

```

mysql> SELECT 1 + 1;
      - > 2

```

При вказівці ключових слів слід точно дотримуватися порядку, вказаний вище. Наприклад, вираз HAVING повинне розташовуватися після всіх виразів GROUP BY і перед усіма виразами ORDER BY .

Використовуючи ключове слово AS , висловом у SELECT можна призначити назву. Псевдонім використовується в якості імені стовпця в даному вираженні і може застосовуватися в ORDER BY або HAVING . Наприклад:

```

mysql> SELECT CONCAT (last_name, ', ', first_name) AS full_name
      FROM mytable ORDER BY full_name;

```

Псевдоніми стовпців не можна використовувати у виразі WHERE , оскільки знаходяться в стовпцях величини на момент виконання WHERE можуть бути ще не визначені.

Вираз FROM `table_references` задає таблиці, з яких належить витягувати рядка. Якщо вказано ім'я більш ніж однієї таблиці, слід виконати об'єднання.

```

table_name [[AS] alias] [USE INDEX (key_list)] [IGNORE INDEX (key_list)]

```

Оператор SELECT може бути представлений у формі SELECT ... INTO OUTFILE 'file_name' . Цей різновид команди здійснює запис вибраних рядків у файл, зазначений у `file_name` . Цей файл створюється на сервері і до цього не повинен існувати (таким чином, крім іншого, запобігається руйнування таблиць і файлів, таких як ``etc / passwd``). Для використання цієї форми команди SELECT необхідні привілеї FILE . Форма SELECT ... INTO OUTFILE головним

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

чином призначена для виконання дуже швидкого дампа таблиці на серверному комп'ютері. Команду SELECT ... INTO OUTFILE не можна застосовувати, якщо необхідно створити результуючий файл на іншому хості, відмінному від серверного. У такому випадку для генерації потрібного файлу замість цієї команди слід використовувати деяку клієнтську програму на зразок mysqldump - tab або mysql-e "SELECT ..." > Outfile . Команда SELECT ... INTO OUTFILE є додатковою по відношенню до LOAD DATA INFILE;

У версії MySQL 3.23.12 можна вказувати, які саме індекси (ключі) MySQL повинен застосовувати для добування інформації з таблиці. Це корисно, якщо оператор EXPLAIN (виводить інформацію про структуру та порядок виконання запиту SELECT), показує, що MySQL використовує хибний індекс. Якщо потрібно, щоб для пошуку запису в таблиці застосовувався тільки один із зазначених індексів, слід задати значення цього індексу в USE INDEX (key_list). Альтернативний вираз IGNORE INDEX (key_list) забороняє використання в MySQL даного конкретного індексу. Вирази USE / IGNORE KEY є синонімами для USE / IGNORE INDEX .

Посилання на таблицю можна замінити псевдонімом, використовуючи tbl_name [AS] alias_name:

```
mysql> SELECT t1.name, t2.salary FROM employee AS t1, info AS
t2
        WHERE t1.name = t2.name;
mysql> SELECT t1.name, t2.salary FROM employee t1, info t2
        WHERE t1.name = t2.name;
```

MySQL підтримує наступний синтаксис оператора JOIN при використанні в командах SELECT :

```
table_reference, table_reference
table_reference [CROSS] JOIN table_reference
table_reference INNER JOIN table_reference join_condition
table_reference STRAIGHT_JOIN table_reference
table_reference LEFT [OUTER] JOIN table_reference
join_condition
table_reference LEFT [OUTER] JOIN table_reference
```



```

table_reference NATURAL [LEFT [OUTER]] JOIN table_reference
{Oj table_reference LEFT OUTER JOIN table_reference ON
conditional_expr}
table_reference RIGHT [OUTER] JOIN table_reference
join_condition
table_reference RIGHT [OUTER] JOIN table_reference
table_reference NATURAL [RIGHT [OUTER]] JOIN table_reference
де table_reference визначено, як:
table_name [[AS] alias] [USE INDEX (key_list)] [IGNORE INDEX
(key_list)]
i join_condition визначено, як:
ON conditional_expr |
USING (column_list)

```

Ніколи не слід вказувати в частині ON які б то не було умови, що накладають обмеження на рядки в наборі результатів. Якщо необхідно вказати, які рядки повинні бути присутніми в результаті, слід зробити це у вираженні WHERE .

Необхідно враховувати, що у версіях до 3.23.17 оператор INNER JOIN не приймає параметр join_condition !

Наявність останньої з наведених вище конструкцій вираження LEFT OUTER JOIN обумовлено лише вимогами сумісності з ODBC:

Замість посилання на таблицю може використовуватися псевдонім, який присвоюється за допомогою виразів tbl_name AS alias_name або tbl_name alias_name :

```

mysql> SELECT t1.name, t2.salary FROM employee AS t1, info AS
t2
WHERE t1.name = t2.name;

```

Умовний оператор ON являє собою умова в будь-якій формі з числа тих, які можна використовувати у виразі WHERE .

Якщо запис для правої таблиці в частинах ON або USING в LEFT JOIN не знайдено, то для даної таблиці використовується рядок, в якій всі стовпці

встановлені в NULL . Цю можливість можна застосовувати для знаходження результатів в таблиці, яка не має еквівалента в іншій таблиці:

```
mysql> SELECT table1. * FROM table1
      LEFT JOIN table2 ON table1.id = table2.id
      WHERE table2.id IS NULL;
```

Цей приклад знаходить всі рядки в таблиці table1 з величиною id , яка не присутня в таблиці table2 (тобто всі рядки в table1 , для яких немає відповідних рядків у table2). Звичайно, це передбачає, що table2.id оголошений як NOT NULL.

USING (column_list) служить для зазначення списку стовпців, які повинні існувати в обох таблицях. Такий вираз USING , як:

```
A LEFT JOIN B USING (C1, C2, C3, ...)
```

семантично ідентично вираженню ON , наприклад:

```
A.C1 = B.C1 AND A.C2 = B.C2 AND A.C3 = B.C3, ...
```

Вираз NATURAL [LEFT] JOIN для двох таблиць визначається так, щоб воно було семантичним еквівалентом INNER JOIN або LEFT JOIN з виразом USING , в якому зазначені всі стовпці, наявні в обох таблицях.

INNER JOIN і ,(кома) є семантичними еквівалентами. Обидва здійснюють повне об'єднання використовуваних таблиць. Спосіб зв'язування таблиць зазвичай задається в умові WHERE .

RIGHT JOIN працює аналогічно LEFT JOIN . Для збереження переносимості коду між різними базами даних рекомендується замість RIGHT JOIN використовувати LEFT JOIN .

STRAIGHT_JOIN ідентично JOIN , за винятком того, що ліва таблиця завжди читається раніше правої. Це вираз може використовуватися для тих (небагатьох) випадків, коли оптимізатор об'єднання своєму розпорядженні таблиці в неправильному порядку.

Починаючи з версії MySQL 3.23.12, можна давати MySQL вказівки про те, який індекс повинен використовуватися при витяганні інформації з таблиці. Ця можливість корисна, якщо оператор EXPLAIN (виводить інформацію про структуру та порядок виконання запиту SELECT), показує, що MySQL

використовує хибний індекс. Задаючи значення індексу в USE INDEX (key_list), можна змусити MySQL застосовувати для пошуку запису тільки один із зазначених індексів. Альтернативне вираз IGNORE INDEX (key_list) забороняє використання в MySQL даного конкретного індексу. Вирази USE / IGNORE KEY є синонімами для USE / IGNORE INDEX .

Оператор INSERT має наступну структуру:

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
      [INTO] tbl_name [(col_name, ...)]
      VALUES (expression, ...), (...), ...
```

або

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
      [INTO] tbl_name [(col_name, ...)]
      SELECT ...
```

або

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
      [INTO] tbl_name
      SET col_name = expression, col_name = expression, ...
```

Оператор INSERT вставляє нових рядків в існуючу таблицю. Форма даної команди INSERT ... VALUES вставляє рядки відповідно до точно вказаними в команді значеннями. Форма INSERT ... SELECT вставляє рядки, обрані з іншої таблиці або таблиць. Форма INSERT ... VALUES зі списком з декількох значень підтримується у версії MySQL 3.22.5 і пізніших. Синтаксис вираження col_name = expression підтримується у версії MySQL 3.22.10 і пізніших.

tbl_name задає таблицю, в яку повинні бути внесені рядки. Стовпці, для яких задані величини в команді, вказуються в списку імен стовпців або в частині SET:

Якщо не вказано список стовпців для INSERT ... VALUES або INSERT ... SELECT , то величини для всіх стовпців повинні бути визначені в списку VALUES () або в результаті роботи SELECT . Якщо порядок стовпців у таблиці невідомий, для його отримання можна використовувати DESCRIBE tbl_name .

Будь-який стовпець, для якого явно не вказано значення, буде встановлений у своє значення за замовчуванням. Наприклад, якщо в заданому списку стовпців

не вказані всі стовпці в даній таблиці, то чи не згадані стовпці встановлюються у свої значення за замовчуванням. Установка значень за замовчуванням описується в розділі section 6.5.3 Синтаксис оператора CREATE TABLE . В MySQL завжди передбачено значення за замовчуванням для кожного поля. Ця вимога створена MySQL, щоб забезпечити можливість роботи як з таблицями, що підтримують транзакції, так і з таблицями, що не підтримують їх. Точка зору розробників полягає в тому, що перевірка вмісту полів повинна проводитися додатком, а не сервером баз даних.

Вираз expression може відноситься до будь-якому стовпцю, який раніше був внесений до списку значень. Наприклад, можна вказати наступне:

```
mysql> INSERT INTO tbl_name (col1, col2) VALUES (15, col1 * 2);
```

Але не можна вказати:

```
mysql> INSERT INTO tbl_name (col1, col2) VALUES (col2 * 2, 15);
```

Якщо вказується ключове слово LOW_PRIORITY , то виконання даної команди INSERT буде затримано доти, поки інші клієнти не завершать читання цієї таблиці. У цьому випадку дана клієнт повинен очікувати, поки дана команда вставки не будуть завершені, що в разі інтенсивного використання таблиці може зажадати значного часу. На противагу цьому команда INSERT DELAYED дозволяє даному клієнтові продовжувати операцію відразу. Синтаксис оператора INSERT DELAYED . Слід зазначити, що показчик LOW_PRIORITY зазвичай не використовується з таблицями MyISAM , оскільки при його вказівці стають неможливими паралельні вставки.

Якщо в команді INSERT з рядками, що мають багато значень, вказується ключове слово IGNORE , то всі рядки, що мають дублюються ключі PRIMARY або UNIQUE у цій таблиці, будуть проігноровані і не будуть внесені. Якщо не вказувати IGNORE , то дана операція вставки припиняється при виявленні рядка, що має дублюється значення існуючого ключа. Кількість рядків, внесених в дану таблицю, можна визначити за допомогою функції C API mysql_info () .

Якщо MySQL був налаштований з використанням опції DONT_USE_DEFAULT_FIELDS , то команда INSERT буде генерувати помилку, якщо явно не вказати величини для всіх стовпців, які вимагають допустимих значень - NULL.

За допомогою функції mysql_insert_id можна знайти величину, застосовану для стовпця AUTO_INCREMENT.

Якщо задається команда INSERT ... SELECT або INSERT ... VALUES зі списками з декількох значень, то для отримання інформації про даний запиті можна використовувати функцію C API mysql_info () . Формат цієї інформаційної рядка наведено нижче:

Records: 100 Duplicates: 0 Warnings: 0

Duplicates показує кількість рядків, які не могли бути внесені, оскільки вони дублювали б значення деяких існуючих унікальних індексів. Показчик Warnings показує число спроб внести величину в стовпець, який з якої-небудь причини опинився проблематичним. Попередження виникають при виконанні будь-якого з таких умов:

Внесення NULL в стовпець, який був оголошений, як NOT NULL . Даний стовпець встановлюється в значення, задане за замовчуванням.

Установка числового стовпця в значення, що лежить за межами його допустимого діапазону. Дана величина буде скорочуватися до відповідної кінцевої точки цього діапазону.

Занесення до числової стовпець такої величини, як '10 .34 а ' . Кінцеві дані видаляються і вноситься тільки числова частина. Якщо величина зовсім не має сенсу як число, то стовпець встановлюється у 0 .

Внесення у стовпці типу CHAR , VARCHAR , TEXT або BLOB рядка, яка перевершує максимальну довжину стовпчика. Дана величина буде скорочуватися до максимальної довжини стовпчика.

Внесення в стовпець дати або часу рядки, неприпустимою для даного типу стовпця. Цей стовпець встановлюється у нульову величину, відповідну даному типу.

Оператор UPDATE має наступну структуру:

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name
      SET col_name1 = expr1 [, col_name2 = expr2, ...]
      [WHERE where_definition]
      [LIMIT #]
```

Оператор UPDATE оновлює стовпці відповідно до їх новими значеннями в рядках існуючої таблиці. У виразі SET вказується, які саме стовпці слід модифікувати і які величини повинні бути в них встановлені. У виразі WHERE, якщо воно присутнє, задається, які рядки підлягають оновленню. В інших випадках оновлюються всі рядки. Якщо задано вираз ORDER BY, то рядки будуть оновлюватися у зазначеному в ньому порядку.

Якщо вказується ключове слово LOW_PRIORITY, то виконання даної команди UPDATE затримується доти, поки інші клієнти не завершать читання цієї таблиці.

Якщо вказується ключове слово IGNORE, то команда поновлення не буде перервана, навіть якщо при оновленні виникне помилка дублювання ключів. Рядки, через які виникають конфліктні ситуації, оновлені не будуть.

Якщо доступ до колонки із зазначеного вираження здійснюється по аргументу tbl_name, то команда UPDATE використовує для цього стовпця його поточне значення. Наприклад, наступна команда встановлює стовпець age в значення, на одиницю більше його поточної величини:

```
mysql> UPDATE persondata SET age = age + 1;
```

Значення команда UPDATE присвоює зліва направо. Наприклад, наступна команда дублює стовпець age, потім інкрементує його:

```
mysql> UPDATE persondata SET age = age * 2, age = age + 1;
```

Якщо стовпець встановлюється в його поточне значення, то MySQL помічає це і не оновлює його.

Команда UPDATE повертає кількість фактично змінених рядків. У версії MySQL 3.22 і більш пізніх функція C API mysql_info() повертає кількість рядків, які були знайдені і оновлені, і кількість попереджень, що мали місце при виконанні UPDATE.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

У версії MySQL 3.23 можна використовувати LIMIT # , щоб переконатися, що було змінено тільки задану кількість рядків.

Оператор DELETE має наступну структуру:

```
DELETE [LOW_PRIORITY | QUICK] FROM table_name
      [WHERE where_definition]
      [ORDER BY ...]
      [LIMIT rows]
```

або

```
DELETE [LOW_PRIORITY | QUICK] table_name [. *] [, Table_name
[. *] ...]
      FROM table-references
      [WHERE where_definition]
```

або

```
DELETE [LOW_PRIORITY | QUICK]
      FROM table_name [. *], [Table_name [. *] ...]
      USING table-references
      [WHERE where_definition]
```

Оператор DELETE видаляє з таблиці table_name рядки, що задовольняють заданим в where_definition умовам, і повертає число видалених записів.

Якщо оператор DELETE запускається без визначення WHERE , то видаляються всі рядки. При роботі в режимі AUTOCOMMIT це буде аналогічно використанню оператора TRUNCATE . В MySQL 3.23 оператор DELETE без визначення WHERE поверне нуль як число видалених записів.

Якщо дійсно необхідно знати число видалених записів при видаленні всіх рядків, і якщо припустимі втрати в швидкості, то можна використовувати команду DELETE в такій формі:

```
mysql> DELETE FROM table_name WHERE 1> 0;
```

Слід враховувати, що ця форма працює набагато повільніше, ніж DELETE FROM table_name без вираження WHERE , оскільки рядки видаляються по черзі по одній.

Якщо вказано ключове слово `LOW_PRIORITY` , виконання даної команди `DELETE` буде затримано доти, поки інші клієнти не завершать читання цієї таблиці.

Якщо заданий параметр `QUICK` , то вказівник таблиці при виконанні видалення не об'єднуватиметься індекси - у деяких випадках це може прискорити дану операцію.

У таблицях `MyISAM` видалені записи зберігаються у зв'язаному списку, а наступні операції `INSERT` повторно використовують місця, де розташовувалися видалені записи. Щоб повернути не використаний простір і зменшити розмір файлів, можна застосувати команду `OPTIMIZE TABLE` або утиліту `myisamchk` для реорганізації таблиць. Команда `OPTIMIZE TABLE` простіше, але утиліта `myisamchk` працює швидше.

Перший з числа наведених на початку даного розділу багатотабличних формат команди `DELETE` підтримується, починаючи з `MySQL 4.0.0`. Другий багатотабличних формат підтримується, починаючи з `MySQL 4.0.2`.

Ідея полягає в тому, що видаляються тільки збережені рядки з таблиць, перераховані перед виразами `FROM` або `USING` . Це дозволяє видаляти одночасно рядки з декількох таблиць, а також використовувати для пошуку додаткові таблиці.

Символи `.` * після імен таблиць потрібні тільки для сумісності з `Access`:

```
DELETE t1, t2 FROM t1, t2, t3 WHERE t1.id = t2.id AND t2.id = t3.id
```

або

```
DELETE FROM t1, t2 USING t1, t2, t3 WHERE t1.id = t2.id AND t2.id = t3.id
```

У попередньому випадку просто видалені збігаються рядки з таблиць `t1` і `t2`

Вираз `ORDER BY` і використання декількох таблиць в команді `DELETE` підтримується в `MySQL 4.0`.

Якщо застосовується вираз `ORDER BY` , то рядки будуть видалені в зазначеному порядку. Насправді цей вислів корисно тільки в поєднанні з `LIMIT`.
Наприклад:

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37


```
DELETE FROM somelog
      WHERE user = 'jcole'
      ORDER BY timestamp
      LIMIT 1
```

Даний оператор видалить найстаріший запис (за timestamp), в рядку який вказаний у вираженні WHERE .

Специфічна для MySQL опція LIMIT для команди DELETE вказує серверу максимальну кількість рядків, які слід видалити до повернення управління клієнту. Ця опція може використовуватися для гарантії того, що дана команда DELETE не зажадає занадто багато часу для виконання. Можна просто повторювати команду DELETE до тих пір, поки кількість вилучених рядків буде менша, ніж величина LIMIT.

Незважаючи на те, що останній стандарт SQL прийнятий в 1992 році, на сьогоднішній день немає жодної СУБД, де б він повністю виконувався. Більш того, в різних базах даних частина операцій здійснюється по-різному. Ми будемо дотримуватися діалекту SQL характерного для СУБД MySQL тому не всі запити можуть виконуватися для інших баз даних.

Важливим чинником є її безкоштовність. MySQL поширюється на умовах загальної ліцензії GNU (GPL, GNU Public License).

Раніше для довготривалого зберігання інформації ми працювали з файлами: поміщали в них деяку кількість рядків, а потім витягували їх для подальшої роботи. Завдання тривалого зберігання інформації дуже часто зустрічається в програмуванні Web-додатків: підрахунок відвідувачів в лічильнику, зберігання повідомлень у форумі, віддалене управління змістом інформації на сайті і т.д.

Тим часом, професійні прийоми роботи з файлами дуже трудомісткі: необхідно піклується про приміщення в них інформації, про її сортуванні, видалення, при цьому не потрібно забувати, що всі ці дії будуть відбуватися на сервері хост-провайдера, де з дуже великою ймовірністю коштує один з варіантів Unix - отже, потрібно так само піклується про права доступу до файлів і їх розміщенні. При цьому обсяг коду значно зростає, і зробити помилку в програмі дуже просто.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Всі ці проблеми вирішує використання бази даних. Бази даних самі дбають про безпеку інформації та її сортування і дозволяють витягувати і розміщувати інформацію за допомогою однієї строчки. Код з використанням бази даних виходить більш компактним, і налагоджувати його набагато легше. Крім того, не потрібно забувати і про швидкість - вибірка інформації з бази даних відбувається значно швидше, ніж з файлів.

Додаток на РНР, що використовує для зберігання інформації базу даних (зокрема MySQL) завжди працює швидше додатка, побудованого на файлах. Справа в тому, що бази даних написані на мові С++, і написати на РНР програму, яка працювала б з жорстким диском ефективніше бази даних - завдання нерозв'язне за визначенням, оскільки програми на РНР в принципі працюють повільніше, ніж програми на С++, так як РНР - інтерпретатор, а С++ - компілятор.

Таким чином, основне достоїнство бази даних полягає в тому, що вона бере на себе всю роботу з жорстким диском і робить це дуже ефективно.

2.2 Огляд Twitter API та його застосування в дипломному проєкті

Twitter є інформаційною мережею і комунікаційним механізмом, який виробляє більше 200 мільйонів твітів на день. Платформа Twitter пропонує доступ до цих даних, за допомогою API. Кожен інтерфейс API представляє собою доступ до Twitter, і дозволяє розробникам розвивати і розширювати свої додатки. Важливо відзначити, що API-інтерфейси Twitter постійно розвиваються.

Twitter for Websites. Twitter для веб-сайтів (TFW) являє собою набір інструментів, який дозволяє легко інтегрувати Twitter на веб-сайтах. TFW ідеально підходить для розробників сайтів, які бажають швидко і легко інтегрувати самі основні функції Twitter. Це включає в себе такі інструменти, як кнопка Tweet, яка дозволяє користувачеві миттєво відправити твіт повідомлення на свою сторінку та поділитись ним з друзями.

Search API. API пошуку призначено для здійснення користувачем функції пошуку тексту певного змісту у Twitter. Search API може включати пошук твітів по певними ключовими словами, пошук твітів, які посилаються на конкретного користувача, або пошук твітів від конкретного користувача.

REST API. API REST дозволяє розробникам отримати доступ до деяких з основних примітивів Twitter включаючи графіки, поновлення статусу та інформації користувача. Якщо ви створюєте додаток, який використовує базові об'єкти Twitter, то це API для вас. Це API також надає розробникам безліч можливостей інтеграції для взаємодії з Twitter. Через API REST, користувач може створювати і відправляти твіти в Twitter, відповідати на замітки в соціальних мережах та багато іншого.

Streaming API. Потокowe API реального часу на зразок Twitter Firehose. Це API для розробників, які інтенсивно працюють з даними. Потокowe API завдяки своїй швидкості дозволяє обробляти велику кількість запитів та даних.

Відмінності між Streaming і REST. Підключення до потокового API вимагає підтримки постійного відкритого HTTP з'єднання. Для прикладу розглянемо веб-додаток, який приймає запити користувачів, робить один або декілька запитів, щоб передати в API, формує і друкує результат для користувача, в якості відповіді на початковий запит користувача:

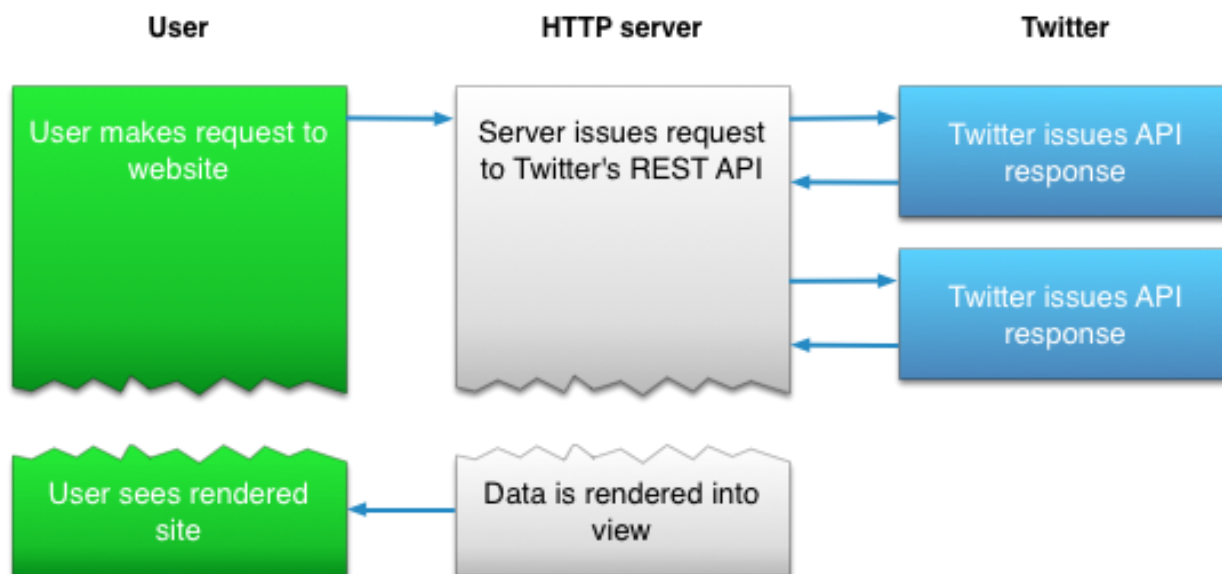


Рисунок 2.1 – Діаграма роботи REST API

Додаток, який підключається до Streaming API, не може встановити з'єднання у відповідь на запит користувача, як показано в наведеному вище прикладі. Замість цього код для підтримки з'єднання Streaming, як правило, виконуються в процесі окремо від процесу, який обробляє HTTP запити:

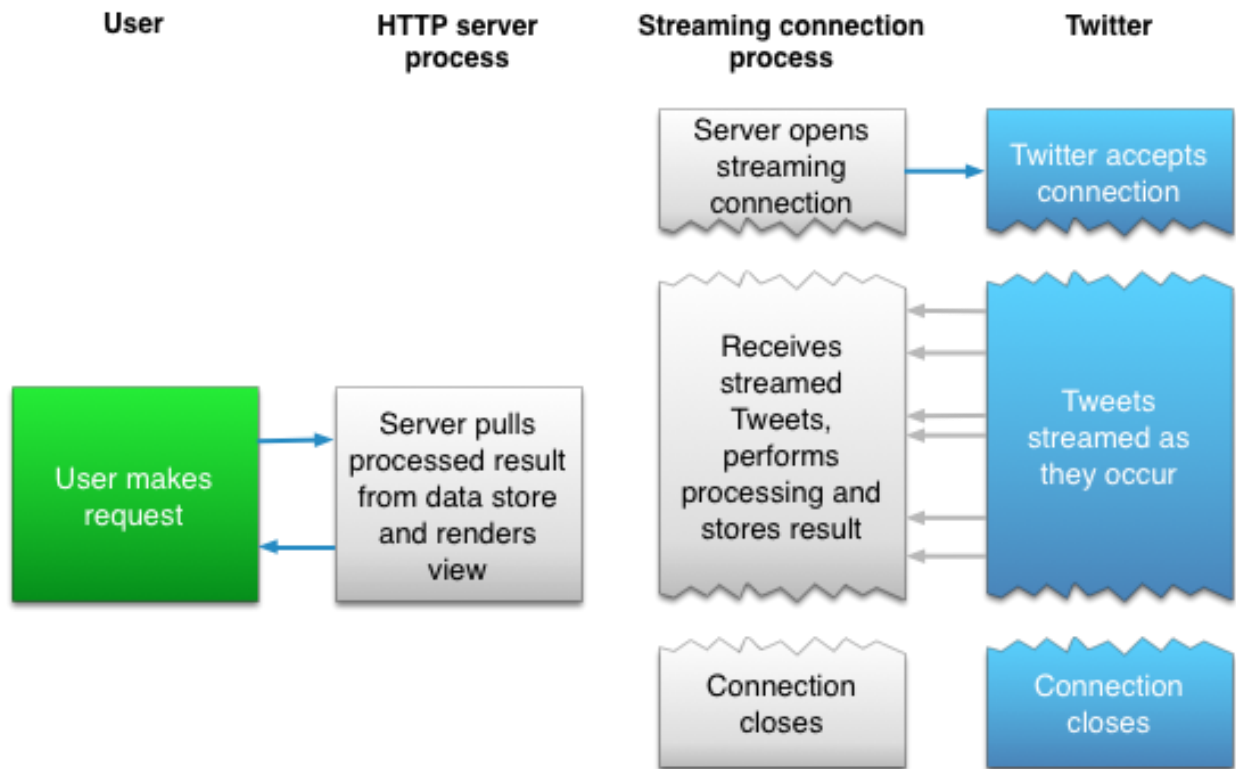


Рисунок 2.2 – Діаграма роботи Streaming API

Процес Streaming API отримує вхідні твіти та виконує фільтрацію, необхідну перед збереженням результату в сховищі даних. Процес обробляє HTTP запити, формує результат у відповідь на запити користувачів. Хоча ця модель є більш складною, ніж у першому прикладі, але тут Tweet дані передаються в реальному часі, що забезпечує інтеграцію Streaming API для багатьох типів додатків.

2.3 Розробка структури бази даних

Для того, щоб спроектувати структуру бази даних необхідно спочатку чітко визначити які функції має виконувати наш проект.

- Авторизація користувачів
- Реєстрація користувачів
- Додавання проектів
- Видалення проектів
- Редагування проектів
- Додавання реклами
- Забезпечення прав доступу

На основі цих критеріїв спроектуємо базу даних, яка буде містити інформацію про зареєстрованих користувачів, проекти, рекламну інформацію та записи репостів. Розроблена структура бази даних показана на рисунку 2.3

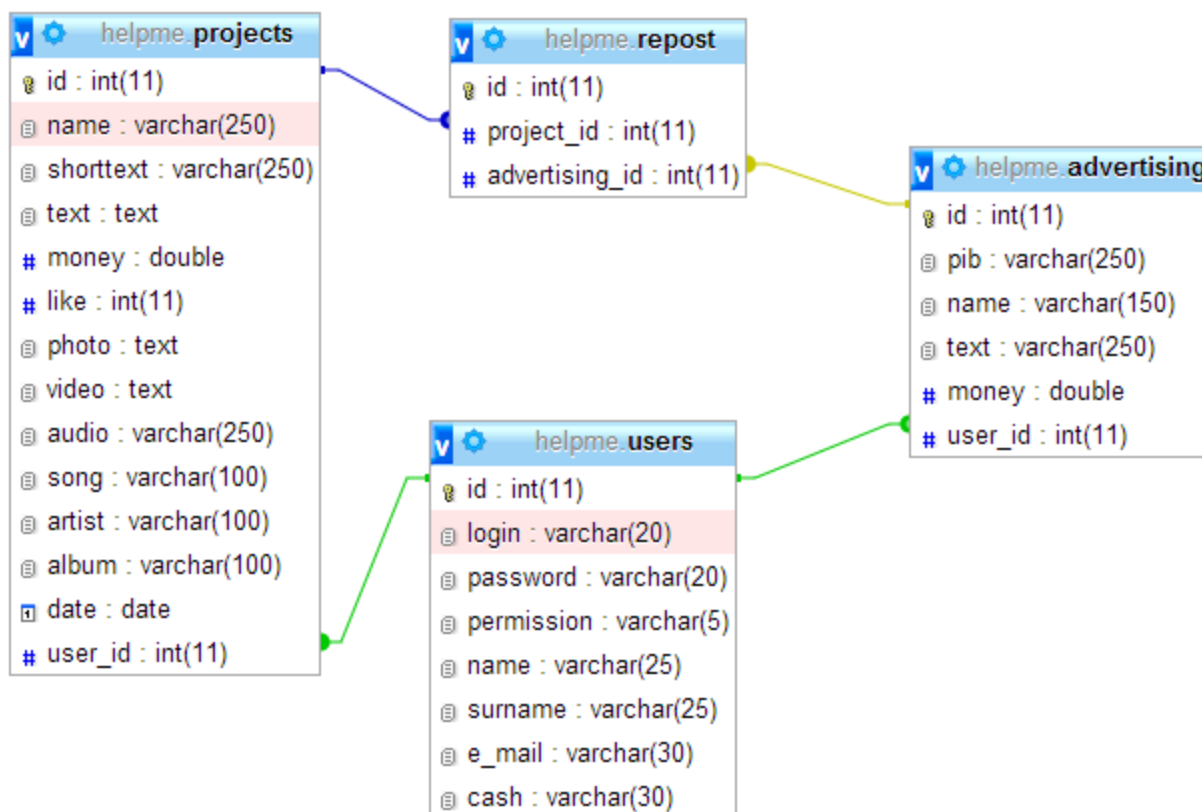


Рисунок 2.3 – Розроблена структура бази даних проекту

Таблиця projects містить інформацію про проекти створені користувачами. В ній зберігаються такі дані як: назва проекту, короткий опис проекту, повний опис проекту, дата створення проекту, також необхідна сума коштів. В полях photo, video, audio зберігаються посилання на розміщені мультимедійні файли, якщо такі є.

Таблиця 2.1 – Проекти

Назва поля	Тип поля	Призначення
id	int(11)	ключове поле
name	varchar(250)	назва проекту
shorttext	varchar(250)	короткий опис
text	text	повний опис
money	double	необхідна сума коштів
like	int(11)	лайки
photo	text	посилання на фото
video	text	код вставки відео
audio	varchar(250)	посилання на аудіо файл
song	varchar(100)	назва пісні
artist	varchar(100)	виконавець
album	varchar(100)	альбом
date	date	дата створення проекту
user_id	int(11)	зв'язок з користувачами

Таблиця users містить дані про зареєстрованих користувачів веб-ресурсу, такі як: прізвище, ім'я, логін і пароль для авторизації на сайті. В полі permission міститься інформація про привілеї користувача (права доступу). Поле e_mail зберігає електронну адресу зареєстрованого користувача. В поле cash записується номер картки для розрахунку.

Таблиця 2.2 – Користувачі

Назва поля	Тип поля	Призначення
id	int(11)	ключове поле
login	varchar(20)	логін
password	varchar(20)	пароль

Продовження таблиці 2.2

permission	varchar(5)	права доступу
name	varchar(25)	ім'я
surname	varchar(25)	прізвище
e_mail	varchar(30)	e-mail
cash	varchar(30)	рахунок

Таблиця advertising зберігає рекламну інформацію. Поле pib ПБ рекламодавця, поле name це заголовок реклами, text поле в якому вказаний повний текст реклами, money - поле де вказується ціна реклами за один зроблений репост, user_id – містить id адміністратора, який додав рекламу на сайт.

Таблиця 2.3 – Реклама

Назва поля	Тип поля	Призначення
id	int(11)	ключове поле
pib	varchar(250)	ПБ
name	varchar(150)	заголовок реклами
text	varchar(250)	текст реклами
money	double	Ціна за 1 репост
user_id	int(11)	зв'язок з користувачами

Таблиця repost хоча і не велика, але не менш важлива. Тут тільки два поля, які зв'язані з таблицями проекти та реклама. Після кожного зробленого репосту сюди буде записуватись id проекту з якого зроблено репост та id реклами, текст якої містився в репості

Таблиця 2.4 – Реклама

Назва поля	Тип поля	Призначення
id	int(11)	ключове поле
project_id	int(11)	зв'язок з проектами
advertising_id	int(11)	зв'язок з рекламою

3 РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ ЗАСОБАМИ PHP, MySQL

3.1 Створення бази даних та забезпечення зв'язків між її елементами

Для створення бази даних вибрано MySQL та систему управління phpMyAdmin. MyISAM є потужним двигуном для таблиць і встановлений він по замовчуванню. Для нашої системи важлива підтримка зовнішнього ключа, тому MyISAM замінений на InnoDB.

На відміну від таблиць MyISAM, де для кожної таблиці створюється один файл даних, дані InnoDB в налаштуваннях за замовчуванням зберігаються у великих спільно використовуваних файлах (змінити це можна за допомогою налаштувань опції `innodb_file_per_table`), що дозволяє використовувати посторінковий кеш сторінок бази даних. Формат даних InnoDB забезпечує надійне зберігання даних за рахунок транзакційності та блокування даних на рівні рядка.

phpMyAdmin — веб-застосунок з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування СКБД MySQL. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних. Ця програма користується великою популярністю у веб-розробників, оскільки дозволяє керувати СКБД MySQL без безпосереднього вводу SQL команд через дружній інтерфейс і з будь-якого комп'ютера під'єданого до інтернету без необхідності встановлення додаткового програмного забезпечення.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

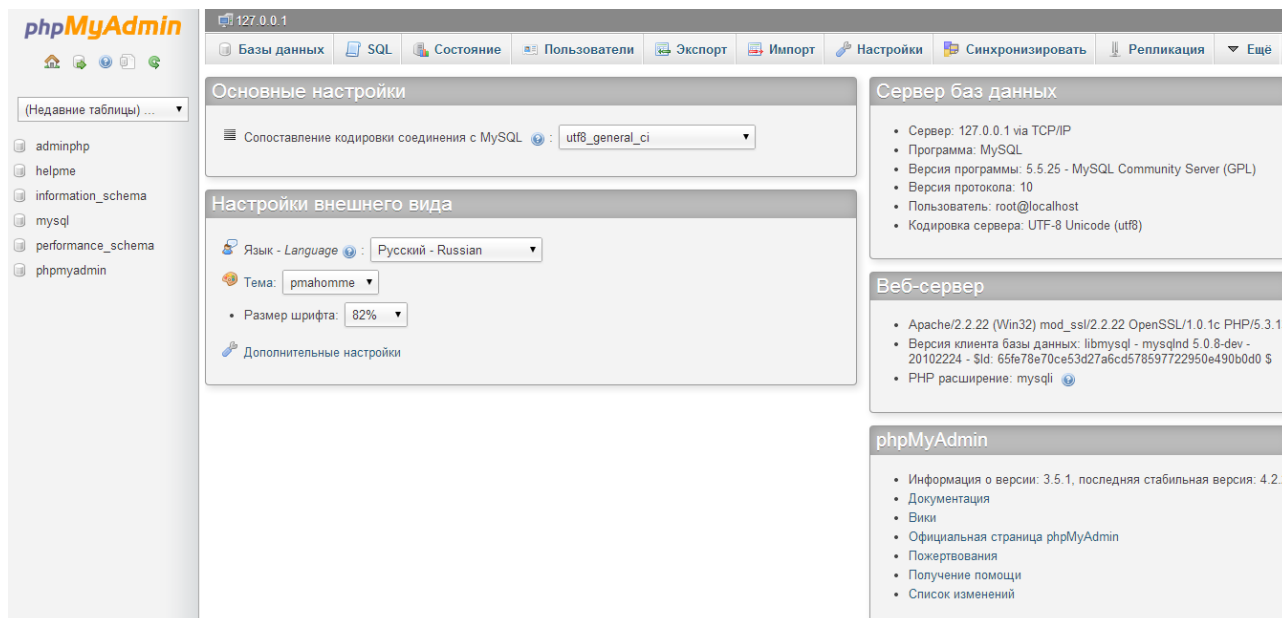


Рисунок 3.1 – Головна сторінка веб-застосунка phpMyAdmin

На сьогоднішній день phpMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, з огляду на всі нововведення СКБД MySQL. Переважна більшість українських провайдерів використовують цей застосунок як панель керування для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

Працювати в phpMyAdmin можна як у візуальному режимі, так і за допомогою SQL-запитів. Наступний код показує можливість створення таблиці projects та проведення зв'язків через SQL-запит:

```
CREATE TABLE IF NOT EXISTS `projects` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(250) NOT NULL,
  `shorttext` varchar(250) NOT NULL,
  `text` text NOT NULL,
  `money` double NOT NULL,
  `like` int(11) NOT NULL,
  `photo` text NOT NULL,
  `video` text NOT NULL,
  `audio` varchar(250) NOT NULL,
  `song` varchar(100) NOT NULL,
  `artist` varchar(100) NOT NULL,
```

```

        `album` varchar(100) NOT NULL,
        `date` date NOT NULL,
        `user_id` int(11) NOT NULL,
        PRIMARY KEY (`id`),
        KEY `user_id` (`user_id`),
        KEY `user_id_2` (`user_id`),
        KEY `user_id_3` (`user_id`)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5
    ALTER TABLE `projects`
        ADD CONSTRAINT `projects_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`);

```

Запустивши запит на виконання, в базі даних створиться нова таблиця з іменем projects (проекти), а також створяться зв'язок між таблицею projects та таблицею users.

Зв'язки потрібні для того, щоб використовувати дані однієї таблиці в іншій та забезпечити цілісність даних при виконанні запитів.

3.2 Розробка функціональних можливостей веб-ресурсу

3.2.1 Впровадження системи реєстрації та прав доступу

Першим етапом розробки є створення реєстрації користувачів. В цьому етапі нам потрібно розробити такі функції як: реєстрація, вхід в систему та надання прав доступу.

Реєстрація нового користувача здійснюється через форму реєстрації. Для успішної реєстрації користувача в базі даних потрібно заповнити всі поля, які позначені як обов'язкові для заповнення та натиснути кнопку Зареєструватись. Після чого буде виконано php-код перевірки логіна на наявність в базі даних. Якщо такий логін уже зареєстрований в базі даних, то веб-сервер повідомить нам про це. А якщо такого логіна немає, то відбудеться перевірка на правильність заповнення полів, після чого користувача буде добавлено в базу даних.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

Реєстрація

Якщо ви ще не зареєстровані, пройдіть будь ласка реєстрацію. Всі поля обов'язкові до заповнення

Логін *

Імя *

Прізвище *

Пароль *

Email *

Номер банківської картки

Зареєструватись

Рисунок 3.2 – Форма реєстрації користувача

Наведений нижче код обробляє введені на формі реєстрації дані та добавляє їх в базу даних:

```
<?php
session_start();
$login = $_POST['login'];
$password = $_POST['password'];
$name = $_POST['name'];
$surname = $_POST['surname'];
$email = $_POST['email'];
$cash = $_POST['cash'];
//якщо не заповнені поля
```

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

```

        if (empty($login) or empty($password) or empty($name) or
empty($surname) or empty($email)) {
            $_SESSION['empty_string'] = "yes";
            header("Location: ../../registration.php");
        } else {
            //видаляєм лишні символи
            $login = stripslashes($login);
            $login = htmlspecialchars($login);
            $password = stripslashes($password);
            $password = htmlspecialchars($password);
            //видаляєм пропуски
            $login = trim($login);
            $password = trim($password);
            //підключення до бд
            include("../db_connect/db_connect.php");
            //перевіряєм чи є такий логін в бд
            $result = mysql_query("SELECT id FROM users WHERE
login='$login'");
            $myrow = mysql_fetch_array($result);
            if (!empty($myrow['id'])) {
                $_SESSION['bad_login'] = "yes";
                header("Location: ../../registration.php");
            } else {
                //якщо немає, то додаємо дані в бд
                $result2 = mysql_query("INSERT INTO users
(login,password,permission,name,surname,e_mail,cash)
VALUES('$login','$password','user','$name','$surname','$email','$c
ash')");
                //перевіряєм чи дані додані в бд
                if ($result2 == 'TRUE') {
                    header("Location: ../../index.php");
                } else {
                    $_SESSION['error'] = "yes";
                    header("Location: ../../registration.php");
                    //якщо помилка
                }
            }
        }
    }

```

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

```
}
}
```

Права доступу до функцій веб-ресурсу є двох типів: користувач та адміністратор. Всім зареєстрованим користувачам автоматично надаються права користувача. Надати права адміністратора може тільки людина яка має доступ до бази даних.

Людина з правами користувача, тобто та яка зареєстрована на сайті, може додавати проекти та редагувати їх. Адміністратор має більше прав. Крім додавання та редагування проектів він може додавати та редагувати рекламу на сайті, а також в адміністратора є можливість видалення записів.

Після того як користувач зареєструвався він може зайти на сайт через форму входу:

Вхід

Якщо ви ще не зареєстровані, пройдіть будь ласка [реєстрацію](#).

Логін *

Пароль *

Ввійти

Рисунок 3.3 – Форма входу на сайт

Користувач повинен ввести свій логін та пароль та натиснути кнопку Ввійти, після чого буде виконаний наступний фрагмент коду, який перевіряє наявність логіна в базі даних:

```
<?php
session_start();
if (isset($_POST['login'])) {
    $login = $_POST['login'];
    if ($login == '') {
```

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

```

        unset($login);
    }
}
if (isset($_POST['password'])) {
    $password = $_POST['password'];
    if ($password == '') {
        unset($password);
    }
}
// якщо логін або пароль = "", то повертаємось на сторінку
вводу
if (empty($login) or empty($password)) {
    $_SESSION['empty_string'] == "yes";
    header("Location: ../../login.php");
}
//підключення до бази даних
include("../../db_connect/db_connect.php");
//звернення до бд
$result = mysql_query("SELECT * FROM users WHERE login =
'$login'");
$myrow = mysql_fetch_array($result);
if (empty($myrow['password'])) {
    //якщо користувача з таким паролем не існує
    $_SESSION['bad_login'] = "yes";
    header("Location: ../../login.php");
} else {
    //якщо такий користувач є
    if ($myrow['password'] == $password) {
        //якщо паролі співпадають
        $_SESSION['id'] = $myrow['id'];
        $_SESSION['login'] = $myrow['login'];
        $_SESSION['password'] = $myrow['password'];
        $_SESSION['permission'] = $myrow['permission'];
        header("Location: ../../index.php");
    } else {
        //якщо паролі не сходяться

```

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

```

        $_SESSION['bad_login'] = "yes";
        header("Location: ../../login.php");
    }
}

```

Якщо логін та пароль були введені правильно та пройшли перевірку, то дані користувача будуть добавлені в сесію. Вони потрібні для подальшої роботи з функціями сайту.

3.2.2 Меню веб-ресурсу

Меню сайту складається з чотирьох пунктів: Проекти, FAQ, Ввійти на сайт та Контакти.

FAQ в містить такі підпункти як: Про сервіс та Як це працює. В свою чергу пункт меню Ввійти на сайт містить два підпункти: Вхід та Реєстрація.

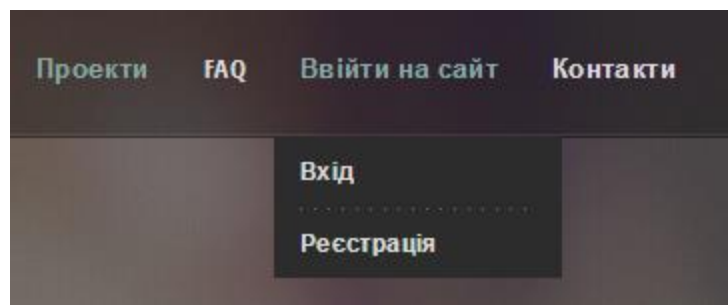


Рисунок 3.4 – Меню веб-ресурсу

Потрібно зазначити те, що пункт меню Ввійти на сайт змінюється після того, як користувач або адміністратор авторизувався. Його назва змінюється на логін користувача, а підпункти змінюються в залежності від прав доступу.

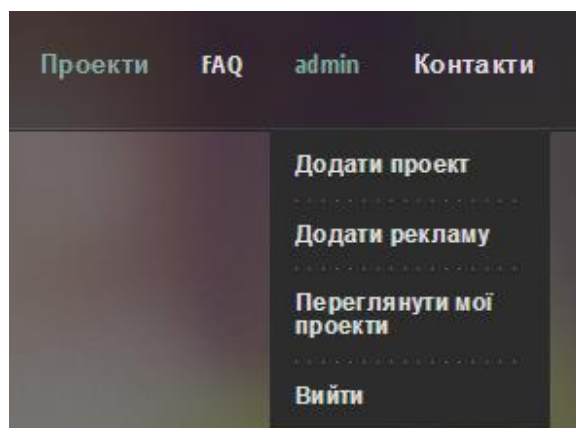


Рисунок 3.5 – Вигляд меню після авторизації

Наступний фрагмент коду визначає який формат меню потрібно вивести на сайті:

```
<?
session_start();
function menu()
{
    $login = $_SESSION['login'];
    $password = $_SESSION['password'];
    $permission = $_SESSION['permission'];

    if (empty($login) && empty($password)) {
        print <<<HERE
            //вивід загального меню
    HERE;
    } elseif (!empty($login) && !empty($password)) {
        print <<<HERE
            //вивід меню для авторизованих користувачів
        ...
    HERE;
        if ($permission == "admin") {
            //вивід меню адміністратора
            ...
        HERE;
        } elseif ($permission == "user") {
            print <<<HERE
                //вивід меню користувача
                ...
            }
        print <<<HERE
        ...
    HERE;
    }
}
```


3.2.3 Функції додання проекту та рекламної інформації

Доступ до функції додання реклами є тільки у адміністратора. Рекламна інформація додається в базу даних через спеціальну форму:

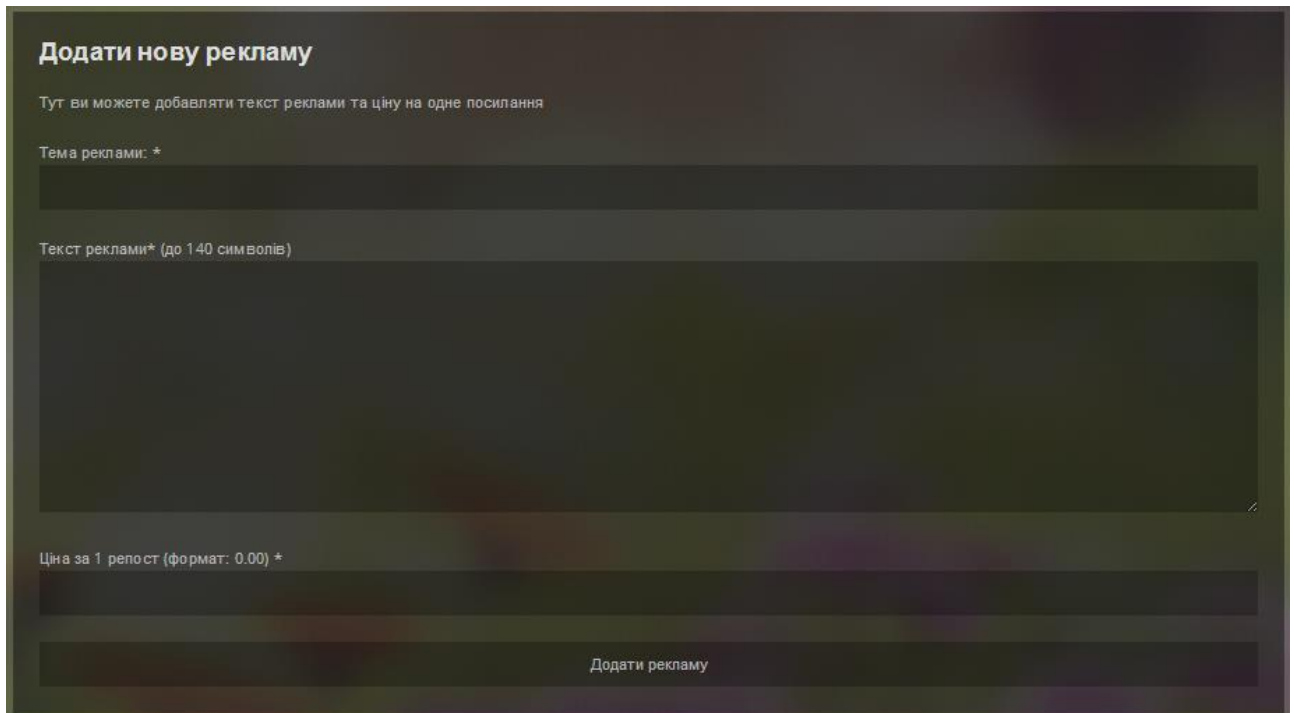


Рисунок 3.6 – Форма додання реклами

Після відправки на сервер форми з даними виконується php-код, який додає рекламу в базу даних:

```
<?php
session_start();
include("../php/db_connect/db_connect.php");
$name = $_POST['name'];
$text = $_POST['text'];
$money = $_POST['money'];
if(!empty($name) && !empty($text) && !empty($money)){
    if($_SESSION['login'] != ""){
        $sql = "INSERT INTO advertising (name, text, money,
user_id) VALUES ('".$name."', '".$text."', ".$money.",
".$_SESSION['id'].")";
        mysql_query($sql);
        if(mysql_error() == ""){
```

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

```

        header("Location: ../index.php");
    } else {
        $_SESSION['error'] = "yes";
        header("Location: ../add-advertising.php");
    }
} else {
    $_SESSION['admin_error'] = "yes";
    header("Location: ../add-advertising.php");
}
} else {
    $_SESSION['empty_string'] = "yes";
    header("Location: ../add-advertising.php");
}
}

```

Додати проект на сайт може як адміністратор, так і зареєстрований користувач. В проект можна додати мультимедійні фрагменти такі як: фото, відео або аудіо. Дані проекту додаються в базу таким же методом як і додання реклами, крім додання мультимедійних файлів. Ці файли додаються через спеціальну модальну форму.

Рисунок 3.7 – Модальна форма загрузки аудіо файла на сервер

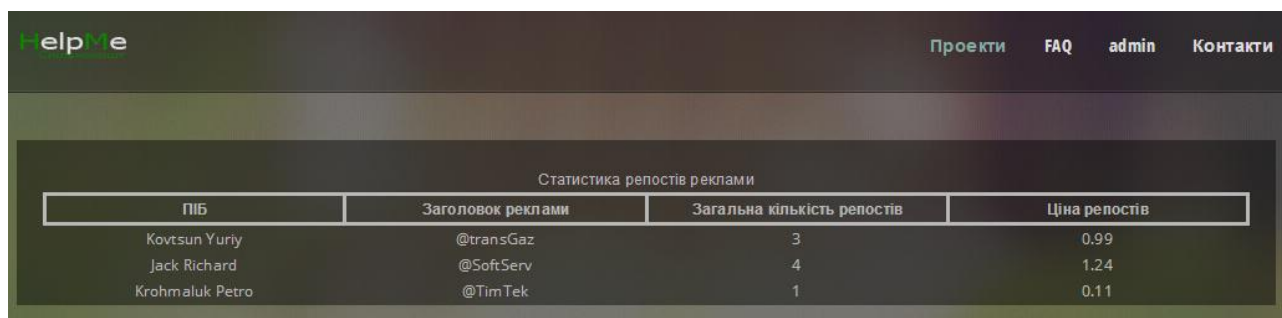
Після того, як користувач вибрав аудіо файл і натиснув кнопку Зберегти, відбудеться загрузка цього файлу на сервер. Наступний фрагмент коду демонструє загрузку файла на сервер:

```
<?php
//Тека в яку буде загрузатись файл
$uploadaddir = './files/';
$uploadfile =
$uploadaddir.basename($_FILES['uploadfile']['name']);
// Копіюєм файл із тимчасової теки
if (copy($_FILES['uploadfile']['tmp_name'], $uploadfile))
{
echo "<h3>Файл успішно загрузено на сервер</h3>";
}
else {
echo "<h3>Помилка! Не вдалося загрузити файл на сервер </h3>";
exit;
}
```

Якщо файл було успішно загрузено на сервер то на сторінці додавання проекту буде виведене повідомлення, якщо ж відбулася помилка при загрузці, то буде виведено повідомлення про помилку.

3.2.4 Статистика репостів реклами

Для того, щоб можна було відстежувати репости реклами – розроблено спеціальну сторінку статистики, на якій відображається таблиця з такими даними як: ПІБ рекламодавця, заголовок реклами, загальна кількість репостів реклами та ціна цих репостів.



The screenshot shows a web interface with a dark theme. At the top, there is a navigation bar with links: 'Проекти', 'FAQ', 'admin', and 'Контакти'. Below this, a section titled 'Статистика репостів реклами' contains a table with the following data:

ПІБ	Заголовок реклами	Загальна кількість репостів	Ціна репостів
Kovtsun Yuriy	@transGaz	3	0.99
Jack Richard	@SoftServ	4	1.24
Krohmaluk Petro	@TimTek	1	0.11

Рисунок 3.8 – Сторінка статистики репостів реклами

Таблиця показана на рисунку 3.8 формується через SQL-запит до бази даних, після чого виводиться на сторінку наступним php-кодом:

```
<?php
include("php/db_connect/db_connect.php");

$sql = "SELECT advertising.pib, advertising.name,
advertising.id, advertising.money, COUNT(repost.advertising_id) as
count FROM advertising INNER JOIN repost ON advertising.id =
repost.advertising_id GROUP BY advertising.id";

$result = mysql_query($sql);
while($row = mysql_fetch_array($result)){
    $price = $row['count'] * $row['money'];
    print "<tr>";
    print "<td>".$row['pib']. "</td>";
    print "<td>".$row['name']. "</td>";
    print "<td>".$row['count']. "</td>";
    print "<td>".$price. "</td>";
    print "</tr>";
}
?>
```

Таким чином сформувавши таку таблицю ми можемо побачити скільки зроблено репостів кожної із реклам та загальну вартість цих репостів.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

4 ЕКОНОМІЧНА ДОЦІЛЬНІСТЬ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО ВПРОВАДЖЕННЯ

4.1 Вплив програмного забезпечення на формування потенціалу соціально- економічних систем

Прихильники краудфандінгу стверджують, що він допомагає хорошим ідеям, які не відповідають формам традиційного сприйняття фінансистів, пробитися і отримати гроші завдяки мудрості натовпу. Якщо проект втягнувся, тобто стартував збір коштів, то це для підприємця означає ще більше шансів на успіх, адже його потенційні майбутні клієнти долучаються до творення бізнесу і в добавок до цього ще рекламують цей проект з уст в уста.

Краудфандінг являє собою особливий вид фінансування за допомогою збору коштів на реалізацію будь-якого проекту через Інтернет, наприклад, створення фільму, книги, музичного альбому, комп'ютерної гри, як правило, творчої продукції. Краудфандінг не є благодійністю. Кожен спонсор, передаючи гроші, отримує натомість певний бонус, наприклад, диск з фільмом, книгу з автографом або запрошення на знімальний майданчик. Часто крауд-інвестор отримує те, що не можна купити в магазині, наприклад, пам'ятні речі та емоції.

Між краудфандінгом та мікрофінансуванням існує принципова різниця. У випадку з мікрофінансуванням мова йде про кошти на розширення мікроскопічного бізнесу, який ніколи не зацікавить банки, по-перше, зважаючи на незначний обсяг коштів, які необхідно залучити, а по-друге, зважаючи на високі ризики і в основному відсутність будь-яких гарантій. Адже мікrokредити, як правило, беруть підприємці, у яких не вистачає коштів на розвиток свого малого бізнесу.

Якщо говорити про краудфандінг, то тут справа стосується більш творчих, тонких та інноваційних проектів, які, тим не менш, вимагають інвестицій. Таким чином, аудиторії мікрофінансових і краудфандінгових організацій абсолютно різні і обидві вони поки що нецікаві банкам та фінансовим компаніям. Проте в

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

найближчій перспективі конкурентами краудфандінгу можуть стати інвестиційні фонди. Це відбудеться, коли інвестиційний краудфандінг набере обертів, і підприємці зможуть відразу звернутися до майбутніх покупців або приватних інвесторів, обходячи класичні способи залучення коштів у нові компанії або просто ідеї, які динамічно розвиваються. Експерти припускають, що надалі краудфандінгові фонди зможуть створювати і самі банки. Однак про те, коли підприємці зможуть звернутися до краудфандінгу, складно визначити.

Поки що краудфандінгом користуються переважно молоді підприємці, представники ІТ-індустрії та люди творчих професій. Якщо виключити новинні ресурси та мікроблоги, куди користувачі копіюють контент з інших соціальних медіа, то перші п'ять ресурсів – це популярні блоги, орієнтовані в тому числі на зарубіжну аудиторію, ВКонтакте – найбільш популярна соціальна мережа рунету, а Хабрахабр – закрите співтовариство ІТ-професіоналів.

Як і в будь-якій новій галузі, основні проблеми краудфандінга полягають у донесенні інформації до потрібної аудиторії. Якщо говорити про потенційних спонсорів, то потрібна аудиторія – це прогресивно мислячі люди, які цікавляться новинками в технологіях, культурі та розвагах. Крім цього, даній категорії людей важливо брати участь у чомусь цікавому і важливому або, більше того, бути причетним. Що стосується авторів проектів, то тут мова може йти практично про будь-яку людину або організацію, які здатні створити щось незвичайне і зацікавити публіку.

Теоретично, краудфандінг, крім інструменту для збору коштів, також виступає інструментом для аналізу попиту на створюваний продукт, і навіть інструментом для піару. Механізм простий – чим більше коштів вдалося залучити, тим вищий потенційний попит, чим більше заявив про себе, поки збирав кошти, тим більший так званий піар-вихлоп. Але знову ж таки, тут слід врахувати популярність автора проекту.

Звичайно, якщо говорити про проблеми краудфандінга і якщо при цьому копати глибоко, то варто згадати і низьку фінансову грамотність населення, і відсутність інтересу до культури, і відсутність закону про краудфандінг, якого

поки що немає навіть в США. У січні 2013 року в США повинен був вступити в силу закон про форсований запуск бізнес-стартапів або JOBS Act. Саме цей закон повинен був легалізувати краудфандінг і дозволити американцям не просто жертвувати гроші на цікаві їм проекти, але й інвестувати їх, тобто вкладати з метою отримання прибутку. Однак його прийняття не відбулося.

На думку аналітиків, у країнах, що розвиваються, поки ще не дозріла потреба в легалізації інвестування на основі краудфандінгу. В основному за допомогою цієї моделі зараз залучають кошти соціальні проекти, що не спрямовані на отримання прибутку і не придатні для народного інвестування. Експерти не очікують легалізації акціонерного краудфандінгу на горизонті 2-х-3-х років. Проте у Європі краудфандінг вже узаконено. Якщо він стане коли-небудь законним і у нас, то можна буде очікувати появи великої кількості організацій, що спеціалізуються на цьому виді фінансування і стартапів, а також спрощення процесу створення бізнесу для людей, що мають на руках тільки хорошу ідею й нічого більше.

Основні проблеми краудфандінга пов'язані, по-перше, з недостатньо розвиненою інфраструктурою, а по-друге, з відсутністю інтересу з боку можливих вкладників, що обумовлено низькою фінансовою грамотністю населення. До основних труднощів можна віднести те, що не повсюдно є хороший інтернет, а також не всі знають, як платити банківськими картками в інтернеті. Разом з тим, у країнах, що розвиваються, краудфандінг має гарні перспективи для розвитку, враховуючи ще ненасичений ринок соціальних медіа, які є головним інструментом у цьому методі фінансування.

Основним ризиком краудфандінга багато експертів вважають шахрайство, оскільки під виглядом благодійної ініціативи або цікавого проекту може бути організований збір коштів, які в підсумку не дійдуть до заявленого адресата і осядуть в руках зловмисників.

Краудфандінгу також не вистачає харизматичних євангелістів, які зможуть привернути до цього процесу увагу людей. Поки ж їхню роль відіграють один з

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

засновників найбільшої краудфандінгової платформи Kickstarter Чарльз Адлер і режисер Гарі Хаствіт, який фінансує з її допомогою свої фільми.

Незважаючи на всі складності, краудфандінг активно розвивається і залучає все більше проектів і спонсорів. Найвідоміший американський краудфандінговий майданчик Kickstarter.com за 2012 рік зібрав близько \$320 млн, що, за оцінками експертів, дало йому виручку приблизно \$16 млн. Це, безперечно, хороший результат, але будь-яка велика фінансова організація заробляє значно більше. З іншого боку, тенденція дуже позитивна – кількість відвідувачів Kickstarter за останній рік збільшилася вдвічі, а краудфандінг стає помітним і резонансним явищем.

Для стартування збору коштів обов'язково повинна бути задекларована мета, визначена ціна її досягнення, а обрахунок усіх витрат і процес збору мають бути відкриті для публіки у вільному доступі.

Суть краудфандінга в сучасному розумінні цього слова: люди скидаються грошима не на абстрактну допомогу далеким ближнім, а на деякий потрібний їм самим продукт, послугу або товар. Наприклад, така підтримка з усього світу п'ятого по відвідуваності сайту всесвітньої павутини - Вікіпедії, безкоштовної відкритої онлайн-енциклопедії. Вікіпедія налічує більше 37 мільйонів статей на 256 мовах, робиться руками добровольців, число яких обчислюється сотнями тисяч, і міститься виключно на добровільні пожертвування. При цьому будь-хто, хто переведе гроші в «Фонд Вікімедіа», прекрасно знає, навіщо він це зробив заради того, щоб енциклопедія продовжувала працювати безкоштовно, продовжувала рости і розширюватися, і не розміщувала на своїх сторінках рекламу. По суті справи, він жертвує гроші, щоб мати вільний доступ до інформації без зайвих проблем.

В даному проекті потрібно розробити систему веб-ресурсу просування та репостингу рекламної інформації. Продукт розрахований на звичайних користувачів, які ще не мають свого власного сайту, але їм потрібна площадка де вони можуть зібрати кошти на свій проект. Також він розрахований на рекламодавців, які мають рекламну інформацію, і на користувачів соціальних

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

мереж, які будуть цю рекламу поширювати в соціальних мережах. Разом з детальною інформацією на ресурсі буде надана можливість в режимі реального часу репостити інформацію, спілкуватися з користувачами, які роблять свої проекти, запитувати, дискутувати, та допомагати.

Серед користувачів такого сайту будуть переважно студенти та молоді люди, які мають постійний доступ до інтернету та активно ведуть або хочуть розробляти, якісь свої перспективні проекти, але на даний час у них немає засобів для їх реалізації. На фоні наявних краудфандінгових сайтів у мережі наш проекти виділяється саме тим, що не люди будуть фінансувати проекти, а саме рекламодавці за допомогою розміщеної реклами в соціальних мережах.

На основі проведених досліджень в сфері веб-ресурсів даного напрямку робляться висновки, які допомагають обґрунтувати рентабельність розробки та купівлі програмного забезпечення.

Створення такого продукту обумовлено тим, що все більш популярнішим стає краудфандінг, все частіше ми бачимо це слово на заголовках інтернет видань, журналів та статей, все більше підприємців, початківців чи стартаперів розглядають краудфандінг як якісно нове, безризикове джерело фінансування своїх проектів. Ця обставина супроводжується все більш інтенсивним розвитком інтернет-майданчиків, де весь світ допомагає реалізувати людям їх ідеї.

В розробці даного програмного продукту моїм завданням є забезпечення роботи функціональної частини веб-ресурсу. Всі функції веб-ресурсу виконуються на сервері, після чого результат виконання повертається до клієнтської частини, тобто до користувацького інтерфейсу.

У моїй частині роботи було покладено за мету розробити тільки клієнтську частину проекту. Як правило, клієнтська частина додатків спілкується з користувачем за допомогою користувацького інтерфейсу, формуючи параметри користувацького запиту, після чого запит відправляється на сервер. Серверна частина приймає запит, виконує всі необхідні обчислення і відправляє результат назад до клієнта.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

Як і будь-який інший продукт нашу систему потрібно обґрунтувати з точки зору економічної доцільності, як і для власника, так і для розробників. Потрібно показати наскільки швидко окуповуються капітальні вкладення власника сайту та який економічний ефект можна очікувати від впровадження даної системи.

4.2 Розрахунок витрат на розробку і використання програмного продукту

Економічне обґрунтування розробки та впровадження програми будемо здійснювати на аналізі таких економічних показників:

- S_{po} – сумарні витрати на розробку програмного забезпечення;
- e_z – експлуатаційні витрати.

Розрахунок відповідних коефіцієнтів проводиться з врахуванням того, що варіаційні задачі діагностування раніше виконувались вручну.

Сумарні витрати на розробку програмного забезпечення S_{po} визначаються за формулою:

$$S_{po} = \sum_i t_{po_i} \cdot B_{po_i} \cdot [(1 + \omega_d) \cdot (1 + \omega_c) + \omega_n] + t_{mo} \cdot e_z, \quad (4.1)$$

де t_{po_i} - час, що витрачається на розробку даної програми працівником i -ої кваліфікації, люд.-міс;

B_{po_i} - основна заробітна плата розробника i -ої кваліфікації, грн/міс;

ω_d - коефіцієнт, що враховує додаткову заробітну плату розробникам програми, у відсотках від основної заробітної плати;

ω_c - коефіцієнт, що враховує нарахування на заробітну плату, у відсотках від основної та додаткової заробітної плати;

ω_n - коефіцієнт, що враховує накладні витрати установи, в якій розробляється ця програма, у відсотках до основної заробітної плати розробника;

t_{MO} - машинний час ЕОМ, необхідний для налагоджування даної програми, машино-год;

e_z - експлуатаційні витрати, що припадають на 1 год машинного часу.

Значення коефіцієнтів $\omega_o = 0.2$; $\omega_c = 0.37$; $\omega_n = 0.54$; $p_o = 0.16$.

Нехай $t_{po_i} = 2$ люд.-міс, а $B_{po_i} = 3300$ грн. Експлуатаційні витрати, що припадають на 1 год машинного часу, можуть бути визначені за витратою електроенергії

$$e_z = P_{cn} \cdot C_{вод} \quad (4.2)$$

де $P_{cn} = 80$ Вт - споживана потужність ЕОМ;

$C_{вод} = 0.85$ - вартість 1 кВт/год електроенергії для підприємств.

Отже, за (5.3)

$$e_z = 0.08 \cdot 0.85 = 0.068 \text{ грн/год.}$$

Необхідний час налагодження програми становить 180 машино-год.

Сумарні витрати на розробку програмного забезпечення складуть:

$$S_{po} = 2 \cdot 3300 \cdot ((1 + 0.2) \cdot (1 + 0.37) + 0.54) + 180 \cdot 0.068 = 14426, 64 \text{ грн.}$$

Річні поточні затрати, пов'язані з експлуатацією програмного продукту, визначаються за формулою:

$$Z_{екс} = Z_{п}, \quad (4.3)$$

де $Z_{п}$ – затрати на оплату роботи адміністратора веб-ресурсу з відрахуваннями в соціальні служби, грн.

Витрати на оплату праці адміністратора веб-ресурсу з відрахуваннями на соціальні потреби визначаються виходячи з часу рішення управлінського завдання на ЕОМ за формулою:

$$Z_{п} = k \times Z_{зп} \times (1 + \omega_c), \quad (4.4)$$

де $Z_{зп}$ – разовий гонорар за роботу адміністратора, грн;

k – щоквартальні витрати на адміністрування ресурсу, рік;

ω_c – коефіцієнт, що враховує нарахування заробітної плати, в процентах від основної та додаткової заробітної плати;

$$З_{\Pi} = 4 * 1200 * (1 + 0,375) = 6600 \text{ грн.}$$

Отже, витрати на розробку програмного продукту становлять 14426,64 грн,

4.3 Обґрунтування дохідності розробки та реалізації програмного продукту

Загальний показник доходу формуються з кількості потенційних користувачів нашого веб-ресурсу та потенційного доходу від банерів та розміщеної реклами на сайті.

Складемо порівняльну характеристику показників для визначення приросту умовного прибутку, Пу:

Таблиця 4.1 – Порівняльна характеристика показників

Характеристика показника	Базовий	Плановий	Абсолютне відхилення прибутку
К-сть потенційних користувачів, чол	1000	700	
Ціна реклами, грн/рік	30	20	
Дохід від реклами, грн	3000	14000	17 000
Потенційний дохід від банерів, грн	-	1200	1 200
Приріст умовного прибутку, грн			18 200

Отже, Пу = 18 200 грн.

Загальні витрати на розробку та реалізацію програмного продукту:

$$З_{\text{в}} = S_{\text{po}} + З_{\Pi} \quad (4.5)$$

$$З_{\text{в}} = 14426,64 + 6600 = 21026,64 \text{ грн.}$$

Використавши дані з розрахунків вище, визначимо річний економічний ефект:

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

$$E_p = \Pi_y - 3п - E_n * S_{po}, \quad (4.6)$$

де E_n - показник нормативної економічної ефективності, величина якого залежить від прийнятого для підприємства рівня віддачі від капіталовкладень (як правило, 15%, тобто $E_n = 0,15$).

$$E_n = 0,18$$

$$E_p = 18200 - 6600 - 0,18 * 14426,64 = 9003 \text{ грн.}$$

Як бачимо умовний прибуток буде вищим, ніж сумарні затрати на розробку веб-системи, тому можна сказати, що проект буде прибутковим. Для встановлення точки дохідності доцільно розрахувати $T_{ок}$:

$$T_{ок} = \frac{S_{po}}{\Pi_y} \quad (4.7)$$

$$T_{ок} = 14426,64 / 18200 = 0.79 \text{ рік.}$$

Визначивши термін окупності, можна спостерігати що за перший рік, саме уже за 9 місяців після впровадження програмного продукту, сумарний прибуток вже буде переважати над затратами. Уже після 3 кварталів ми зможемо спостерігати надходження прибутку, що і являє собою окупність проекту. В таблиці 4.2 наведені зведені економічні показники системи.

Таблиця 4.2 - Зведені економічні показники розробки системи

№п/п	Показник	Розмірність	Значення
1	Вартість ПЗ	грн	14426
2	Річні затрати пов'язані з експлуатацією	грн	6600
3	Приріст умовного прибутку	грн	18200
4	Річний економічний ефект	грн	9003
5	Термін окупності інвестицій	рік	0,79

На основі інформації наведеної в цьому розділі можна зробити висновок, що розробка та впровадження даного програмного продукту є економічно

доцільним. Термін окупності капітальних вкладень є доволі низьким, що дозволяє за короткий час отримувати відповідний дохід.

Водночас, підтримка програми потребує значних ресурсів і постійної зайнятості від розробника, проте, вираховані економічні показники свідчать про те, що розробка такого програмного продукту є економічно доцільно.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

5 ОХОРОНА ПРАЦІ

5.1 Значення охорони праці в забезпеченні безпечних та здорових умов праці

Охорона праці – це система правил і заходів, які забезпечують безпечну роботу на даному виробництві.

Охорона здоров'я працівників, забезпечення безпеки умов праці, ліквідація професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства. Звертається увага на необхідність широкого застосування прогресивних форм наукової організації праці, зведення до мінімуму ручної, малокваліфікованої праці, створення обстановки, що виключає професійні захворювання і виробничий травматизм.

На робочому місці повинні бути передбачені заходи захисту від можливого впливу небезпечних і шкідливих факторів виробництва. Рівні цих чинників не повинні перевищувати граничних значень, обумовлених правовими, технічними та санітарно-технічними нормами. Ці нормативні документи зобов'язують до створення на робочому місці умов праці, при яких вплив небезпечних і шкідливих чинників на працюючих або усунуто зовсім, або знаходиться в допустимих межах.

5.2 Аналіз потенційних небезпек та шкідливих факторів виробничого середовища

Науково-технічний прогрес вніс серйозні зміни в умови виробничої діяльності робітників розумової праці. Їх праця стала більш інтенсивним, напруженим, які вимагають значних витрат розумової, емоційної і фізичної енергії. Це зажадало комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

В даний час комп'ютерна техніка широко застосовується у всіх областях діяльності людини. При роботі з комп'ютером людина піддається дії ряду небезпечних і шкідливих виробничих факторів: електромагнітних полів (діапазон радіочастот: ВЧ, УВЧ і СВЧ), інфрачервоного і іонізуючого випромінювань, шуму і вібрації, статичної електрики і ін.

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ. Велике значення має раціональна конструкція і розташування елементів робочого місця, що важливо для підтримки оптимальної робочої пози людини-оператора.

У процесі роботи з комп'ютером необхідно дотримувати правильний режим праці та відпочинку. В іншому випадку у персоналу наголошуються значна напруга зорового апарату з появою скарг на незадоволеність роботою, головні болі, дратівливість, порушення сну, втому і хворобливі відчуття в очах, в поясниці, в області шиї і руках.

5.3 Забезпечення нормальних умов праці

Робоче місце користувача ЕОМ має відповідати всім вимогам щодо ергономічності та розташування елементів робочого місця згідно ДСанПІН 3.3.2.007-98 , характеру та особливостей трудової діяльності.

Конструкція робочого місця повинна передбачати підтримання оптимального робочого положення тіла програміста з такими ергономічними характеристиками: ступні ніг – на підлозі, або на підставці для ніг, стегна – в горизонтальній площині, передпліччя – вертикально, лікті – під кутом 70 - 90 градусів до вертикальної площини, зап'ястя зігнуті під кутом не більше 20 градусів відносно горизонтальної площини, нахил голови – 15 - 20 градусів відносно вертикальної площини.

Дисплей та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітно-цифрових знаків та символів.

Розташування екрану відео терміналу має забезпечувати зручність зорового спостереження у вертикальній площині під кутом ± 30 градусів від лінії зору працівника.

Клавіатуру слід розміщувати на поверхні столу, або на спеціальній робочій і такій, яка регулюється за висотою поверхні, окремо від столу на відстані 100 - 300 мм від краю, ближчого до працівника. Кут нахилу клавіатури має бути в межах 5 - 15 градусів.

Правильно спроектоване і виконане виробниче освітлення покращує умови зорової роботи, знижує стомлюваність, сприяє підвищенню продуктивності праці, благотворно впливає на виробниче середовище, надаючи позитивну психологічну дію на працюючого, підвищує безпеку праці і знижує травматизм.

Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань, тому такий важливий правильний розрахунок освітленості.

Існує три види освітлення - природне, штучне і поєднане (природне і штучне разом).

Природне освітлення - освітлення приміщень денним світлом, що потрапляє через світлові прорізи в зовнішніх огорожуючих конструкціях приміщення. Природне освітлення характеризується тим, що змінюється в широких межах залежно від часу дня, пори року, характеру області і ряду інших чинників.

Штучне освітлення застосовується при роботі в темний час доби і вдень, коли не вдається забезпечити нормовані значення коефіцієнта природного освітлення (похмура погода, короткий світловий день). Освітлення, при якому

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

недостатнє за нормами природне освітлення доповнюється штучним, називається змішаним освітленням.

Штучне освітлення підрозділяється на робоче, аварійне, евакуаційне, охоронне. Робоче освітлення, у свою чергу, може бути загальним або комбінованим. Загальне - освітлення, при якому світильники розміщуються у верхній зоні приміщення рівномірно, або, як розташоване устаткування.

Комбіноване - освітлення, при якому до загального додається місцеве освітлення.

Ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими, оскільки яскраве світло в районі периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності.

Велику роль в забезпеченні нормальних умов праці відіграють санітарно-гігієнічні показники робочої зони.

Параметри мікроклімату можуть мінятися в широких межах, у той час як необхідною умовою життєдіяльності людини є підтримка постійності температури тіла завдяки терморегуляції, тобто здатності організму регулювати віддачу тепла в навколишнє середовище. Принцип нормування мікроклімату - створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Обчислювальна техніка є джерелом істотних тепловиділень, що може привести до підвищення температури і зниження відносної вологості в приміщенні. У приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. У санітарних нормах СН-245-71 встановлені величини параметрів мікроклімату, що створюють комфортні умови. Ці норми встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення.

Мікрокліматичні умови виробничих приміщень характеризуються такими показниками, як температура повітря, відносна вологість повітря, швидкість руху повітря, інтенсивність теплового (інфрачервоного) опромінення,

температура поверхні. Оптимальні величини температури, відносної вологості та швидкості руху повітря в робочій зоні(ДСН 3.3.6.042-99) виробничих приміщень наведені в таблиці 5.2.

Таблиця 5.2 – Оптимальні величини температури, відносної вологості та швидкості руху повітря в робочій зоні

Характеристика приміщень	Категорія робіт	Період року	Параметр мікроклімату	Величина
Приміщення з ЕОМ, кондиціонером та радіомодулем	Іа - легка	Холодний	Температура повітря в приміщенні Відносна вологість Швидкість руху повітря	22...24 °C 60...40 % 0,1 м/с
		Теплий	Температура повітря в приміщенні Відносна вологість Швидкість руху повітря	23...25 °C 60...40 % 0,1 м/с

Регулювання цих показників здійснюється за допомогою організаційних та технічних засобів, які проводяться з метою забезпечення комфортних умов праці. До цих засобів відносяться опалювання, кондиціонування, вентиляція.

Забезпечення даних параметрів мікроклімату забезпечуються з допомогою приладів зволоження або штучної іонізації, або кондиціонування повітря.

Потрібно дуже ретельно підбирати та встановлювати кондиціонери, оскільки неправильно вибране місце розташування може заважати нормальним умовам праці. Щоб запобігти прямому потоку повітря на програміста потрібно встановлювати спеціальні контейнери на кондиціонер, які перенаправляють потік вгору.

В таблиці 5.4 розглянуто характеристики кондиціонера, який використовується в офісному приміщенні – Samsung AR18HQFSAWKN.

Таблиця 5.4 – Характеристики кондиціонера Samsung AR18HQFSAWKN.

Характеристики по холоду:	
Площа обслуговування	55 м ²
Продуктивність при охолодженні	5,2 кВт

Характеристики по теплу:	
Площа обслуговування	55 м ²
Продуктивність при охолодженні	5,8 кВт
Технічні характеристики внутрішнього блоку:	
Рівень шуму (мін-макс), дБ (А)	29-40
Технічні характеристики зовнішнього блоку:	
Рівень шуму (макс.), дБ (А)	52

Системи вентиляції, на відміну від кондиціонерів, які все ж таки не є предметами першої необхідності, встановлюються у всіх офісних і житлових будівлях. Наявність вентиляційних систем є просто необхідністю, а вимоги до їх технічних характеристик мають силу закону.

У першу чергу при виборі обладнання для системи вентиляції, необхідно розрахувати наступні параметри:

- Продуктивність по повітрю (м³/год).
- Допустимий рівень шуму (дБ).
- Швидкість потоку повітря у повітропроводах (м/с).
- Робочий тиск (Па).
- Потужність калорифера (кВт).

Розрахуємо необхідну кількість повітря для офісу площею 70 м² та висотою стелі 3 м, в якому працює 5 програмістів.

При значній кількості людей, котрі знаходяться в приміщенні, необхідну кількість повітря визначимо за формулою

$$L = I \cdot n_d \text{ м}^3/\text{ГОД}, \quad (5.1)$$

де I – мінімум подання повітря на одну людину (працівника) відповідно до санітарних норм (при об'ємі приміщення, що припадає на одного працівника, більше 20 м³, $I = 20 \text{ м}^3/\text{год}$);

n_d – кількість людей, котрі одночасно знаходяться в приміщенні, $n_d = 5$.

Отже,

$$L = 20 \cdot 5 = 100 \text{ м}^3/\text{ГОД}.$$

Для приміщень, де немає шкідливих виділень (або кількість їх незначна), приплив (витяжку) повітря можна визначити за кратністю повітрообміну k – відношення об'єму вентиляційного повітря L (м³/год) до об'єму приміщення V (м³). Звідки

$$k = \frac{100}{70} = 1.43 \text{ м}^3/\text{год}$$

Оскільки кратність повітрообміну є малою, то немає потреби у штучній вентиляції. Розрахуємо природну вентиляцію та визначимо її ефективність. Фактичний повітрообмін у відділі здійснюється за допомогою природної вентиляції як неорганізовано – через різні нещільності у віконних і дверних прорізах, так і організовано – через квартиру у віконному прорізі.

Розглянемо приміщення, в якому працюють програмісти (Рисунок 5.1). Розміри дверей – 2 м на 1.5 м, квартирки – 1 м на 0.5 м. Відповідно значення $F_{\text{кв}}$ рівне 0,5 м², а $S_{\text{дв}}$ – 3 м². Центр квартирки знаходиться на висоті 2.5 м, звідси h дорівнює 1.5 м.

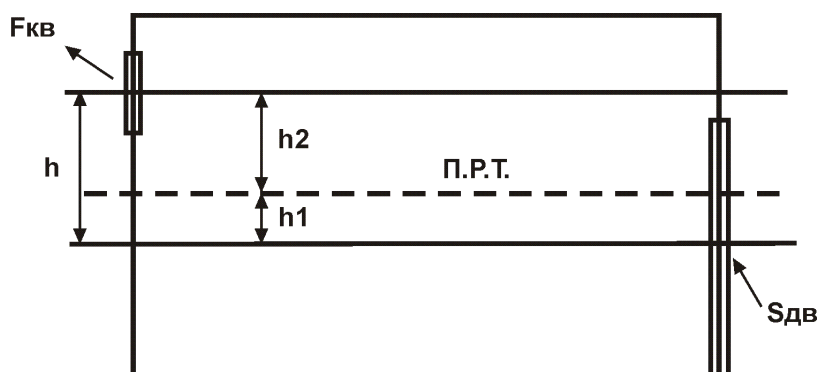


Рисунок 5.1 – Схема розрахунку природної вентиляції

Фактичний повітрообмін $L_{\text{ф}}$, м³/год., обчислюється за формулою:

$$L_{\text{ф}} = F_{\text{кв}} \cdot V_{\text{п}} \cdot \mu \cdot 3600, \quad (5.2)$$

де $F_{\text{кв}}$ – площа квартирки, через яку буде виходити повітря, м²;

$V_{\text{п}}$ – швидкість виходу повітря через квартиру, м/с. Її можна розрахувати за формулою:

$$V = \sqrt{\frac{2g \cdot \Delta H_2}{\gamma_{\text{вн}}}} \quad (5.3)$$

де g – прискорення вільного падіння, $g = 9,8$ м/с²;

ΔH_2 – тепловий напір, під дією якого буде виходити повітря з квартирки, кг/м²:

$$\Delta H_2 = h_2 \cdot (\gamma_3 - \gamma_{\text{вн}}) \quad (5.4)$$

де h_2 – висота від площини рівних тисків до центру квартирки (Рисунок 5.1).

Її можна визначити з наступного співвідношення: відстані від площини рівних тисків до центрів нижніх і верхніх прорізів відповідно h_1 та h_2 , обернено пропорційні квадратам площ цих прорізів $S_{\text{дв}}$ та $F_{\text{кв}}$, тобто:

$$\frac{h_1}{h_2} = \frac{F_{\text{кв}}^2}{S_{\text{дв}}^2} \quad (5.5)$$

З геометричних розмірів приміщення $h_1 + h_2 = h$,

Таким чином, з системи двох рівнянь з двома невідомими знаходимо h_2 , яке дорівнює 1,46 м

γ_3 та $\gamma_{\text{вн}}$ – відповідно об'ємна вага повітря зовні та з середини приміщення, кг/м³.

Об'ємна вага повітря визначається за формулою:

$$\gamma = 0.465 \cdot \frac{P_6}{T} \quad (5.6)$$

де P_6 – барометричний тиск, мм рт. ст., в розрахунках береться $P_6 = 750$ мм рт. ст.;

T – температура повітря, градуси Кельвіна.

Для приміщень, в яких виконується легка робота, для теплого періоду року, температура повітря повинна бути не вище $+28^\circ\text{C}$, або $T = 301\text{K}$, для холодного періоду року відповідно $t = 17^\circ\text{C}$, або $T = 290\text{K}$.

Для повітря зовні приміщення температура є наступною:

для теплого періоду: $t = 24^\circ\text{C}$, $T = 297\text{K}$;

для холодного періоду: $t = -11^\circ\text{C}$, $T = 262\text{K}$.

Для холодного періоду

Об'ємна вага повітря зовні приміщення, кг/м³ буде дорівнювати:

$$\gamma_3 = 0.465 \cdot 750 / 262 = 1.33$$

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

Об'ємна вага повітря з середини приміщення, кг/м³ буде дорівнювати:

$$\gamma_{\text{вн}} = 0,465 \cdot 750 / 290 = 1,2$$

Тепловий напір, під дією якого буде виходити повітря з квартирки, кг/м² становить:

$$\Delta H_2 = 1,46 \cdot (1,33 - 1,2) = 0,19$$

Швидкість виходу повітря через квартиру, м/с складає:

$$V_{\text{п}} = \sqrt{\frac{2 \times 9,8 \times 0,19}{1,2}} = 1,76$$

Фактичний повітрообмін $L_{\text{ф}}$, м³/год., у відділі становить:

$$L_{\text{ф}} = 0,5 \cdot 1,76 \cdot 0,5 \cdot 3600 = 1584 \text{ м}^3/\text{год}$$

Визначивши фактичний повітрообмін і порівнявши його з необхідним, можна зробити висновок про ефективність природної вентиляції у відділі. У холодний період року фактичний повітрообмін набагато більше необхідного ($1584 > 100$), що може викликати переохолодження працюючих. Для покращення ефективності природної вентиляції рекомендується скорочення часу провітрювання приміщення пропорційно перевищенню фактичного повітрообміну над необхідним.

Необхідні 100 м³/год. Фактично можна досягти за:

$$(100 \cdot 60) / 1584 = 3,8 \text{ хв}$$

Тобто для ефективної роботи природної вентиляції достатньо провітрювати приміщення протягом 3,8 хв. кожної години.

Для теплого періоду

Об'ємна вага повітря зовні приміщення, кг/м³ буде дорівнювати:

$$\gamma_3 = 0,465 \cdot 750 / 297 = 1,17$$

Об'ємна вага повітря з середини приміщення, кг/м³ буде дорівнювати:

$$\gamma_{\text{вн}} = 0,465 \cdot 750 / 301 = 1,16$$

Тепловий напір, під дією якого буде виходити повітря з квартирки, кг/м² становить:

$$\Delta H_2 = 1,46 \cdot (1,17 - 1,16) = 0,01$$

Швидкість виходу повітря через квартиру, м/с складає:

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

$$V_{\pi} = \sqrt{\frac{2 \times 9,8 \times 0,01}{1,16}} = 0,4$$

Фактичний повітрообмін L_{ϕ} , м³/год., у відділі становить:

$$L_{\phi} = 0,5 \cdot 0,4 \cdot 0,5 \cdot 3600 = 360 \text{ м}^3/\text{год}$$

Визначивши фактичний повітрообмін і порівнявши його з необхідним, можна зробити висновок про ефективність природної вентиляції у відділі. У теплий період року фактичний повітрообмін набагато більше необхідного ($360 > 100$). Для покращення ефективності природної вентиляції рекомендується скорочення часу провітрювання приміщення пропорційно перевищенню фактичного повітрообміну над необхідним.

Необхідні 100 м³/год. Фактично можна досягти за:

$$(100 \cdot 60) / 360 = 16,6 \text{ хв}$$

Тобто для ефективної роботи природної вентиляції достатньо провітрювати приміщення протягом 16,6 хв. кожної години.

5.4 Пожежна безпека та безпека в надзвичайних ситуаціях

Для всіх будівель та приміщень виробничого, складського призначення і лабораторій повинна бути визначена категорія щодо вибухопожежної та пожежної небезпеки (за НАПБ Б.03.002-2007), а також клас зони за правилами будови електроустановок, у тому числі для зовнішніх виробничих і складських ділянок, які необхідно позначати на входних дверях до приміщення, а також на межах зон всередині приміщень та ззовні.

Визначаємо категорію приміщення за вибухопожежною небезпекою.

Таблиця 5.5 – Категорії приміщень за НАПБ Б.03.002-2007

Категорія приміщення	Характеристика речовин та матеріалів, які знаходяться (обертаються) у приміщенні
А (вибухопожежонебезпечна)	Горючі гази, легкозаймисті рідини з температурою спалаху не більше 28°C у такій кількості, що можуть утворюватися вибухонебезпечні парогазоповітряні суміші, у разі спалахування яких розвивається розрахунковий надлишковий тиск вибуху в приміщенні.

Б (вибухопожежонебезпечна)	Горючі пил або волокна, ЛЗР з температурою спалаху більше 28°C, ГР у такій кількості, що можуть створювати вибухонебезпечні пилоповітряні або пароповітряні суміші, у разі спалахування яких розвивається розрахунковий надлишковий тиск вибуху в приміщенні, що перевищує 5 кПа;
В (пожежонебезпечна)	Горючі і важкогорючі рідини, тверді горючі і важкогорючі речовини й матеріали (у тому числі пил та волокна), речовини й матеріали при взаємодії з водою, киснем повітря або одне з одним здатні горіти тільки за умови, якщо приміщення, у яких вони є в наявності або обертаються, не належать до категорій А і Б;
Г	Негорючі речовини й матеріали в гарячому, розжареному або розплавленому стані, процес оброблення яких супроводжується виділенням променистого тепла, іскор та полум'я; горючі гази, рідини і тверді речовини, які спалюються або утилізуються як паливо;
Д	Негорючі речовини і матеріали в холодному стані. Дозволяється зараховувати до категорії Д приміщення, у яких розміщені ГР у системах змащування, охолодження і гідроприводу обладнання, в яких не більше 60 кг в одиниці обладнання за умов тиску не більше 0,2 МПа, кабельні електропроводки до обладнання, окремі предмети меблі на місцях.

Приміщення в якому працюють програмісти є горючі матеріали такі як: пластик, меблі, килимове, чи дерев'яне покриття підлоги, тощо. Тому таке приміщення є пожежонебезпечним, а значить тут потрібно приділяти особливу увагу пожежній безпеці. Потрібно дотримуватися загальних вимог пожежної безпеки. Для споруд та приміщень, в яких експлуатуються ЕОМ вимоги пожежної безпеки визначені Правилами пожежної безпеки в Україні, НПАОП 0.00-1.28-10 та іншими нормативними документами.

Насамперед потрібно забезпечити робоче приміщення первинними засобами пожежогасіння. Для забезпечення безпечних умов праці в організації приміщення мають бути оснащені системами автоматичної пожежної сигналізації з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунком 2 шт. на кожні 20 м² площі.

Оскільки в компаніях, які займаються розробкою програмного забезпечення немає легко займистих речовин і робота не проводиться з вибухонебезпечними речовинами, то ступінь ризику є доволі низьким.

Короткі замикання виникають в результаті порушення ізоляції частин обладнання, що проводять струм і зовнішніх механічних пошкоджень в електричних дротах, монтажних дротах, обмотках двигунів і апаратів. Ізоляція елементів, що проводять струм може пошкоджуватися при дії на неї високої температури або полум'я, інфрачервоного випромінювання, переходу напруги з первинної обмотки силового трансформатора на вторинну, при підвищених режимах навантаження (нагрів до високих температур, і як наслідок при охолодженні конденсується вода) та інші.

В даному розділі описані вимоги щодо охорони праці на підприємстві, яке займається розробкою програмного забезпечення. Наведені можливі професійні захворювання та норми, дотримання яких дозволить зменшити ризик цих захворювань, а також надасть можливість уникнути травматизму, пов'язаному з виконанням робочих обов'язків.

В розділі описані вимоги щодо облаштування робочого місця програміста, а також охарактеризовані небезпечні фактори, які діють на людину під час виконання її обов'язків. Робота програміста вимагає високої концентрації, тому відповідність усім мікрокліматичним нормам тільки підвищить його продуктивність. Розраховане значення кратності повітрообміну доводить непотрібність штучної вентиляції приміщення. Натомість, врахуваний необхідний для забезпечення ефективної природної вентиляції час провітрювання для холодного та теплого періодів.

Керівництво підприємства в свою чергу має подбати про належну електробезпеку та пожежну безпеку, щоб їхні працівники могли зосередитися на якості своїх програмних продуктів та на принесенні своїй компанії доходів.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

ВИСНОВОК

В результаті роботи над дипломним проектом було проведено розробку алгоритмічного та програмного забезпечення веб-ресурсу просування та репостингу рекламної інформації.

Також було досліджено існуючі рішення програмного забезпечення в області краудфандінгу.

Спроектовано базу даних MySQL, яка містить чотири таблиці. До всіх таблиць проведено зв'язки для забезпечення цілісності даних.

На веб-ресурсі розроблено функції авторизації та реєстрації для розділення прав доступу до функцій контенту. Розроблено функції додання проектів та рекламної інформації на веб-ресурс. Застосовано Twitter API для перевірки кількості зроблених користувачами репостів рекламної інформації в соціальній мережі twitter.com. Також розроблено сторінку ведення статистики репостів реклами.

Впровадження такого програмного продукту як інтернет-майданчика для підприємців, початківців або стартаперів дозволить безризиково реалізувати свої ідеї, а весь світ буде мати змогу допомогти їм в цьому.

					ДП.ПЗ-64.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		80

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. uk.wikipedia.org/wiki/Краудфандинг
2. <http://biggggidea.com/>
3. <https://www.kickstarter.com/>
4. <http://boomstarter.ru/>
5. <http://www.jetbrains.com/phpstorm/>
6. uk.wikipedia.org/wiki/InnoDB
7. <http://www.denwer.ru/>
8. http://site-konstruktor.com.ua/php/perevagy_php
9. <http://www.phpcoders.org.ua/>
10. <http://www.php-adminka.ru/?id=10>
11. uk.wikipedia.org/wiki/PhpMyAdmin
12. www.phpmyadmin.net/
13. <https://twitter.com/>
14. <https://dev.twitter.com/>
15. <http://cpanel.hostinger.com.ua/>
16. <http://help-me.96.lt/>
17. www.php.net/
18. www.php.su
19. <http://ru.html.net/tutorials/php/>
20. <https://dev.twitter.com/docs/api/streaming>
21. <https://dev.twitter.com/docs/api/1.1>
22. <http://www.mysql.ru/docs/man/Reference.html>