**ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

**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.

Переважна більшість компаній чи осіб, що займаються комерційною діяльністю мають власні веб-сайти, які допомагають повідомити широкому загалу про їхню діяльність. Раніше для здійснення певної покупки потрібно було їхати в магазин і вже там на місці вибирати потрібний товар. Не менше часу забирало замовлення необхідних послуг. Ці малоприємні процедури стали набагато простішими з появою комерційних сайтів. На їхніх сторінках можна знайти контактну інформацію бізнес структури, дізнатися вид її діяльності, ознайомитися з асортиментом товарів та послуг, які компанія може запропонувати своїм потенційним клієнтам, а також зробити замовлення чи зв`язатися з безпосередніми представниками підприємства не виходячи з дому чи офісу, що дозволить значно зекономити час та сили обох сторін . Метою даної роботи є створення програмного шаблону сайту електронної комерції, який міститиме інформацію про компанію, а також список товарів та послуг, які вона надає. Крім того, сайт надаватиме можливість не тільки ознайомлення з діяльністю компанії, а й замовлення потрібних товарів і послуг.

аний програмний продукт не матиме аналогів сереДд вже існуючих рішень, що повинно забезпечити його високу популярність серед компаній, що

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

* 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 мікрокредити, 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.

Поки що краудфандінгом користуються переважно молоді підприємці, представники IT-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 IT-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 форсованний 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.

* 1. **Аналіз сучасних онлайн-систем краудфандингу**

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.

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 Stupig (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.

The Cosmonaut є ще одним прикладом краундфандингу 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 краудфандинг значно зросли. У 2012 році проекти створення відеоігор Double Fine Adventure і Wasteland 2 зібрали по $3 336 371 і $3 040 299 через краудфандинговий сайт Kickstarter. Внески зробили більше 87 000 людей.

Kickstarter — це сайт фінансування творчих проектів за схемою краудфандингу. 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.

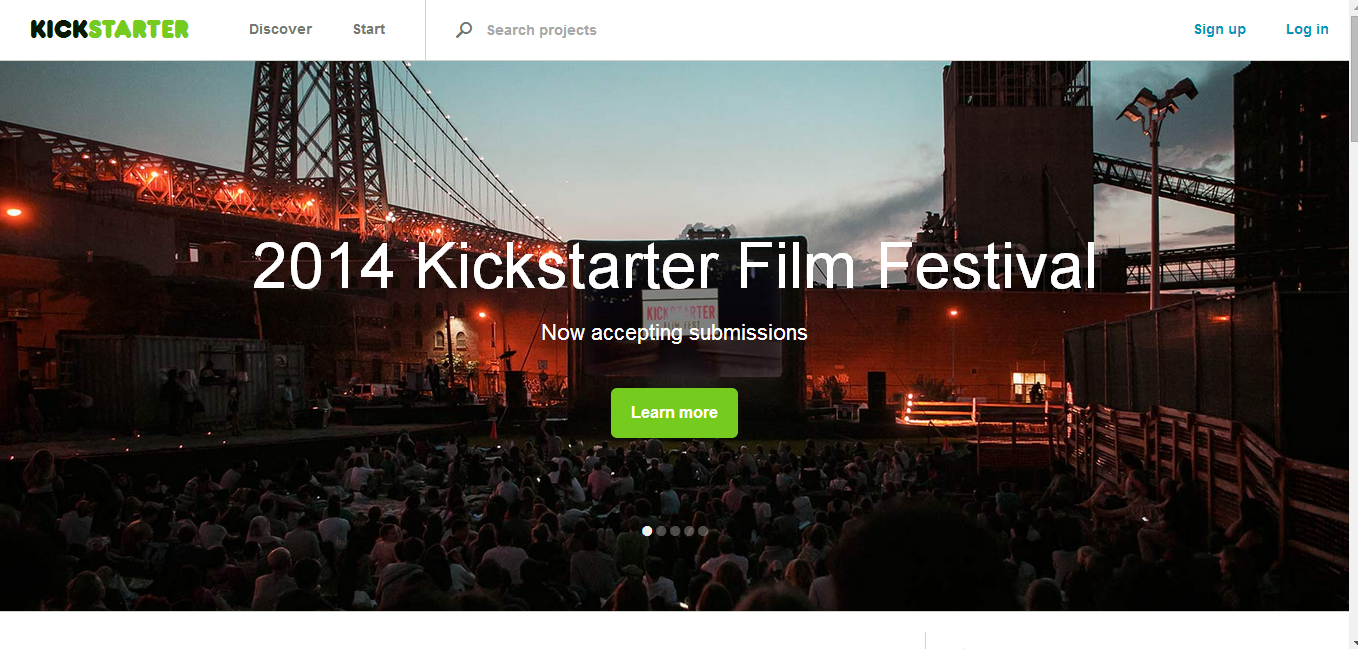


Рисунок 1.1 – Головна сторінка ресурсу kickstarter.com

Kickstarter полегшує збір коштів, створивши модель, яка може бути кращою за традиційні способи інвестування[3]. Той, хто хоче отримати фінансування, повинен зареєструватися і розмістити опис проекту на Kickstarter. Kickstarter містить рекомендації[4] які проекти 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[5], ініціатору проекту потрібно мати рахунок в американському банку.

Kickstarter бере 5% від залучених коштів; Amazon стягує додаткові 3-5%[6]. 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[7].

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 упішності 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 упішності проекту становить 90%.

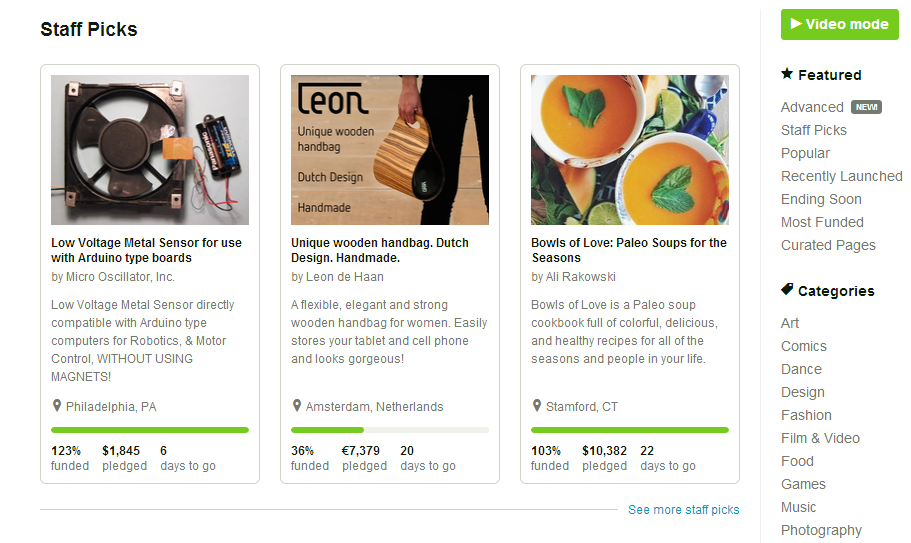


Рисунок 1.2 – Сторінка з проектами ресурсу kickstarter.com

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 - онлайновий 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.

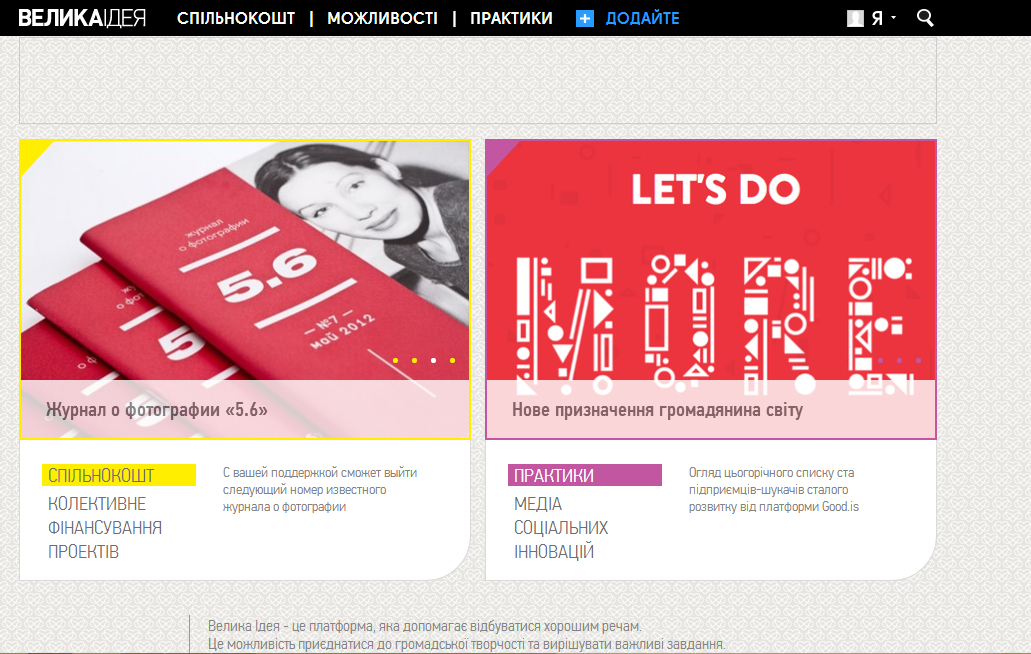


Рисунок 1.3 – Головна сторінка ресурсу biggggidea.com

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

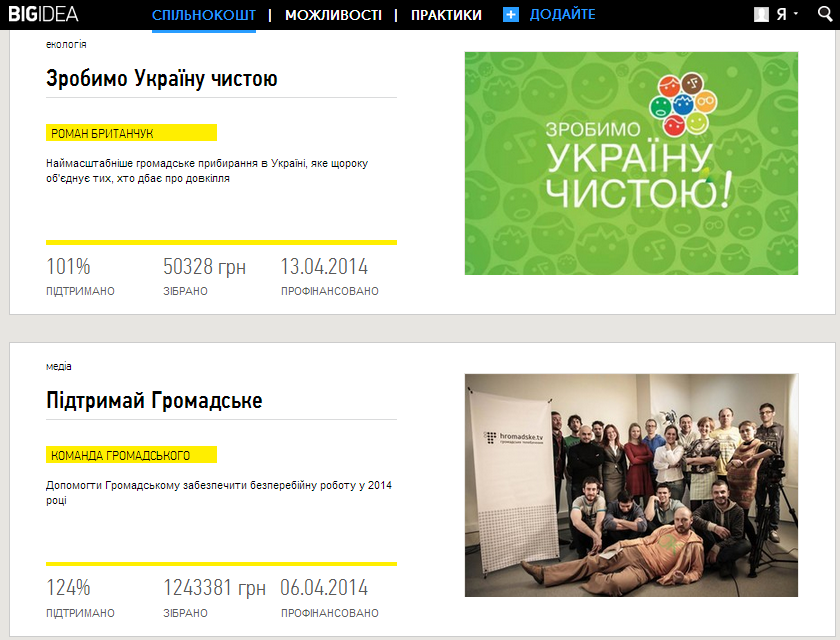


Рисунок 1.4 – Розділ ресурсу biggggidea.com - Спільнокошт

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.

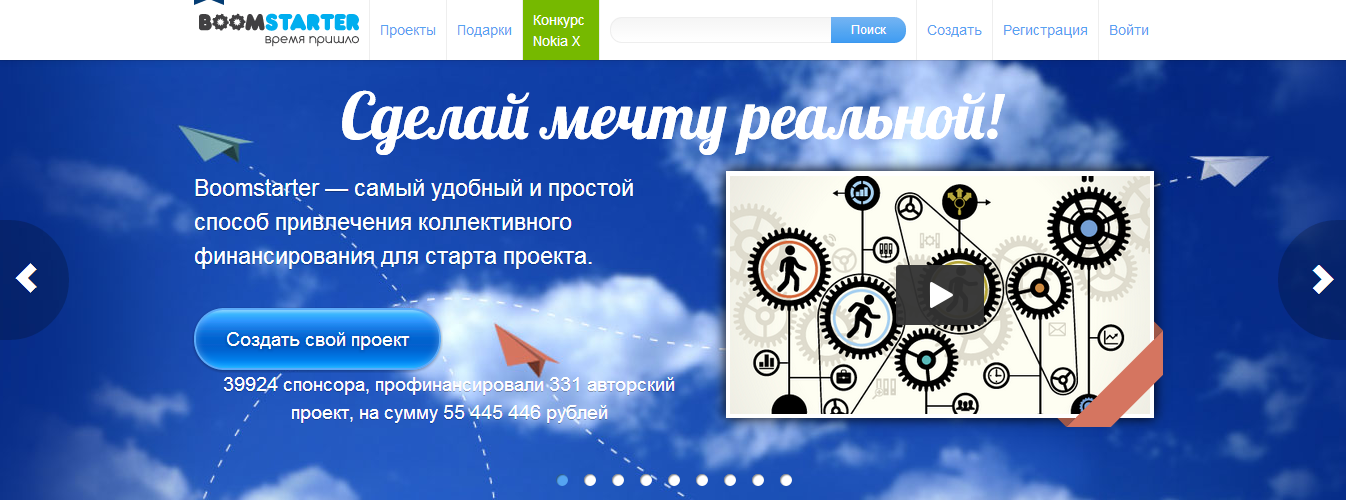


Рисунок 1.5 – Головна сторінка ресурсу boomstarter.ru



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

**2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОСТІ ВЕБ-РЕСУРСУ**

**2.1 Вибір програмних засобів для реалізації проекту**

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](http://site-konstruktor.com.ua/php/mozhlyvosti_php) (у вигляді програми CGI або серверного модуля), [веб-сервер](http://site-konstruktor.com.ua/php/mozhlyvosti_php) 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](http://site-konstruktor.com.ua/php/mozhlyvosti_php). Ви можете створити PHP-скрипт, здатний запускатися незалежно від веб-сервера і браузера. Все, що вам буде потрібно - [парсер PHP](http://site-konstruktor.com.ua/php/mozhlyvosti_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.

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), створюваних "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.

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 сокетами "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](http://www.php.net/) є практичність. PHP повинна надати програмісту інструмент для швидкого і ефективного вирішення поставлених завдань. Практичний характер РНР обумовлений п'ятьма важливими характеристиками:

* традиційністю
* простотою
* ефективністю
* безпекою
* гнучкістю

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

*Традиційність*. Мова РНР здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкції мови запозичені з Сі, Perl.

Код РНР дуже схожий на той, який зустрічається в типових програмах на С або Pascal. Це помітно знижує початкові зусилля при вивченні РНР. PHP - мова, що поєднує достоїнства Perl, Сі і спеціально націлена на роботу в Інтернеті, мова з універсальним (правда, за деякими застереженнями) і ясним синтаксисом. І хоча PHP є досить молодою 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 РНР 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 що-небудь в цьому дусі. Механізм РНР 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»](http://site-konstruktor.com.ua/php/engine). «Движок» PHP не є ні компілятором, ні інтерпретатором. Він є транслюючим інтерпретатором. Такий пристрій «движка» PHP дозволяє обробляти сценарії з достатньо високою швидкістю.

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

*Безпека*. РНР надає в розпорядження розробників та адміністраторів гнучкі та ефективні [засоби безпеки](http://site-konstruktor.com.ua/php/perevagy_php), 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 запобігає викрадення нетривіальних сценаріїв користувачами, знань яких вистачає хоча б для виконання команди View Source.

*Гнучкість*. Оскільки РНР є вбудовуваною (embedded) мовою, вона відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча РНР 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 РНР легко розширюються в міру необхідності (втім, це відноситься до всіх основних мов програмування).

Немає проблем із залежністю від браузерів, оскільки перед відправкою клієнту сценарії РНР повністю компілюються на стороні сервера. По суті, сценарії РНР 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 РНР-код в [режимі командного рядка.](http://site-konstruktor.com.ua/php/perevagy_php)

Оскільки РНР 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 - РНР працює на всіх перерахованих серверах. Оскільки ці сервери працюють на різних платформах, РНР 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 РНР дозволяють програмісту працювати із зовнішніми компонентами, такими як Enterprise Java Beans або СОМ-об'єкти Win32. Завдяки цим новим можливостям РНР 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 благотворний вплив на багато проектів, в першу чергу - Linux, хоча і успіх проекту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії створення РНР, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником в розвитку проекту РНР.

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

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

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, видаляння, 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.

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 C++, і написати на PHP програму, яка працювала б з жорстким диском ефективніше бази даних - завдання нерозв'язна за визначенням, оскільки програми на PHP в принципі працюють повільніше, ніж програми на C++, 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 проекті**

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 повідомленя 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*. Потокове API реального часу на зразок Twitter Firehose. Це API для розробників, які інтенсивно працюють з даними. Потокове 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:

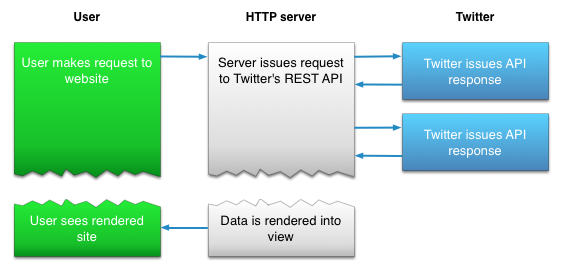


Рисунок 2.1 – Приклад роботи REST API

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:

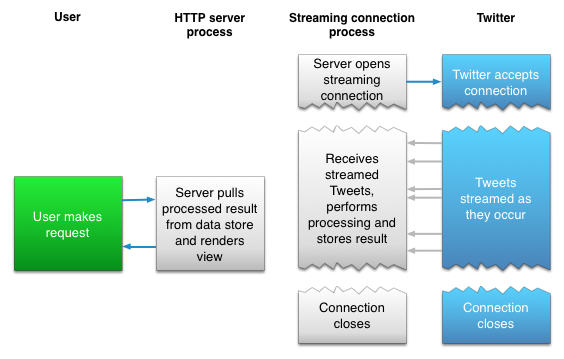


Рисунок 2.2 – Приклад роботи Streaming API

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

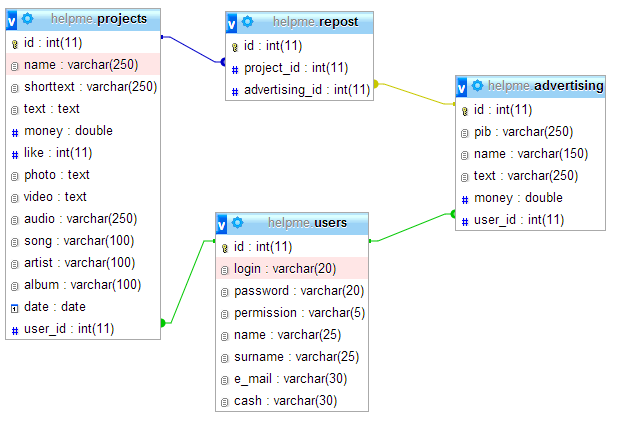


Рисунок 2.3 – Розроблена структура бази даних проекту

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 привілегії 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 |
| 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 ПІБ 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) | ПІБ |
| 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 проекта 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 |