

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи

на ступінь вищої освіти бакалавр

**на тему: «ПРОЕКТУВАННЯ ТА РОЗРОБКА WEB-
ДОДАТКУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ МОВОЮ
ПРОГРАМУВАННЯ PHP»**

Виконав: студент 4 курсу, групи ПД-44

спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва
спеціальності)

Полонський Д.Н.

(прізвище та ініціали)

Керівник

Яскевич В.О.

(прізвище та
ініціали)

Рецензент

(прізвище та
ініціали)

Нормоконтроль

(прізвище та
ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В. Негоденко

«_____» _____ 2022 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ
СТУДЕНТУ ПОЛОНСЬКОМУ ДАНИЛУ
НАЗАРОВИЧУ

1. Тема роботи: «Проектування та розробка web-додатку для електронної комерції мовою програмування PHP»

Керівник роботи Яскевич Владислав Олександрович, кандидат технічних наук затверджені наказом вищого навчального закладу від — «18» лютого 2022 року №.

2. Строк подання студентом роботи 03.06.2022

3. Вхідні дані до роботи:

3.1. Середовище розробки Visual Studio Code

3.2. Структура веб додатку.

3.3. Науково-технічна література, пов'язана з розробкою.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

- 4.1. Аналіз області досліджень
- 4.2. Вибір технологій і середовища
- 4.3. Розробка веб-додатку

5. Перелік матеріалу

- 5.1.1. Популярність арбітражу трафіку
- 5.1.2. Переваги та недоліки додатків для арбітражу
- 5.1.3. Програмні засоби реалізації
- 5.1.4. Огляд можливостей
- 5.1.5. Апробація результатів досліджень

6. Дата видачі завдання 11.04.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04.22 – 14.04.22	
2	Аналіз існуючих аналогів	15.04.22 – 17.04.22	
3	Дослідження програмних засобів	18.04.22 – 24.05.22	
4	Моделювання об'єкту	26.04.22 – 05.05.22	
5	Розробка функціоналу веб додатку	05.05.22 – 07.05.22	
6	Вступ, висновки, реферат	07.05.22 – 10.05.22	
7	Розробка презентації застосунку	11.05.22 – 15.06.22	
8	Попередній захист роботи	16.05.22 – 01.06.22	
9	Подання роботи в деканат	03.06.22	

Студент

Керівник роботи

РЕФЕРАТ

Текстова частина бакалаврської роботи 46 с., 36 рис., 19 джерел.

Ключові слова: розробка, додаток, арбітраж, PHP, canvas, гра, магазин

Об'єкт дослідження – електронної комерції.

Предмет дослідження – web-додаток для електронної комерції мовою програмування PHP

Мета роботи – збільшення клієнтської бази на основі знижкової системи за допомогою canvas гри

Методи дослідження – Теоретичні методи класифікації, аналізу. Емпіричні методи.

Наукова новизна даної роботи полягає в наступному:

В роботі виконано аналіз існуючих аналогів платформ для арбітражу трафіку. Встановлено їх переваги та недоліки. В результаті аналізу було визначено основні проблеми додатків аналогів та потреби користувачів. Проаналізовано мови програмування та їх можливості. PHP та JavaScript, середовища розробки VisualCode, трекера Keitaro для вхідного WEB та WAP трафіку.

Розроблено логіку, алгоритм та базовий функціонал додатку, описано чому до нього були поставлені саме такі завдання.

Галузь використання – додаток має комерційну основу, тому буде застосовуватися для окремої компанії

ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1. Web технології та арбітраж.....	10
1.2. Огляд аналогів.....	15
1.2.1. CryptoEngine	15
1.2.2. TeslaX	16
1.2.3. QuantumCode	17
1.2.4. Формування вимог до додатку	18
1.2.5. Висновки до першого розділу	18
РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ	19
2.1. Обрані технології розробки.....	19
2.1.1. JavaScript та canvas	19
2.1.2. CANVAS	21
2.1.3. Keitaro	24
2.1.4. HTML	30
2.1.5. CSS	32
2.1.6. PHP	33
2.2. Середовище розробки.....	38
2.2.1. Visual Studio Code	38
2.3. Проектування додатку.....	39
РОЗДІЛ 3. РОЗРОБКА ПЛАТФОРМИ	42
3.1. Загальні відмоті.....	42
3.2. Серверна частина.....	43
3.3. Canvas гра.....	47
3.4. Реєстрація клієнтів.....	50
3.5. Тестування.....	51
3.5.1. Інтерфейсу користувача	51
3.5.2. Функціональне тестування	52
3.6. Висновки до третього розділу.....	52
ВИСНОВКИ	53
ПЕРЕЛІК ПОСИЛАНЬ	54
ДОДАТОК А	55
ДОДАТОК Б	61

ВСТУП

Вже довгий час люди користуються сайтами, це може бути як сайти пошти, отримання інформації, сайти знайомств, творчі сайти, сайти "щось купити, щось продати".

Останнім часом, останніми роками виросла популярність інтернет-магазинів. Це простіше, швидше, іноді можна потрапити на великі знижки.

Ще відносно недавно з'явилися так звані трекери. Трекери застосовуються арбітражниками, рекламодавцями та вебмайстрами для формування та розподілу потоків цільового трафіку за рахунок численних фільтрів та інструментів редіректу. Також дає можливість ділити трафік країнами, фільтрації гео, провайдеру, мови браузера, реферальним посиланням, useragent, ключових слів та багато іншого, забезпечуючи конфіденційність ваших даних з рекламних кампаній.

Саме думка про нові, знижкові системи і штовхнула кампанію яка запропонувала мені написати для них невеликий проект, впровадити в їхній сайт форму заявки до якої підв'язати знижкову систему, маючи на увазі гру, в якій в залежності від набраних балів ти отримуєш знижку, а зв'язок з трекером гарантує постійний моніторинг відвідуваності та кліків користувачів для покращення та оптимізації трафіку.

Головне завдання в тому, щоб зробити проект простим та ефективним, а також залучити клієнтів, з огляду на скільки конкурентів зараз є. Плавна переходячи у проблеми, основна проблема інтернет-магазинів – конкуренти. Зараз знайти інтернет-магазин можна на кожному кроці, у когось кращий товар, у когось великі знижки, у когось часті заводи і завжди є щось нове, тому щоб мати якусь конкурентну здатність і виділятися на тлі інших, було прийнято рішення впровадити у сайт саме гру, особисто я такого ніде не бачив, це буде щось нове, що може залучити, порадувати та набрати аудиторію.

Актуальність обраної теми. Актуальність дипломної роботи полягає в тому, щоб прив'язувати до себе більше клієнтів та збільшити продажі за рахунок знижкової системи у вигляді гри при реєстрації. На сьогоднішній день розробка веб-додатків набирають все більшої популярності і стають невід'ємною частиною, які повинен мати розробник для успішного працевлаштування. І саме розробка власного сайту. Якщо веб-сайт для організації розробляє веб-студія або розробник, то клієнт отримує на нього всі права, стає його власником. У випадку з безкоштовними сайтами цього не відбувається — вони належать до сервісів, з підтримкою яких були зроблені. Насправді це означає, що сервіс може вимагати дотримання певних правил, може вносити будь-які видозміни і навіть видаляти сайт без погодження з користувачем.

Веб-програмування — являється однією з найбільш перспективних напрямлень для розвитку. Будь-який бізнес, який має власний веб-додаток, за замовчуванням буде більш доступним, ніж його конкуренти.

У сучасному суспільстві, заснованому на технології, навички програмування відкривають двері до добре оплачуваної роботи та успішної кар'єри. Отже, потрібно оволодіти певними навичками програмування. Як і вивчення будь-якої іншої мови, робота з кодом вимагає регулярного оволодіння. Але людям не завжди легко знайти час для самостійної практики. Тому дуже корисно використовувати спеціальні веб-додатки для вивчення програмування.

Сучасний веб-додаток — це унікальна розробка, орієнтована на вирішення певних завдань

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Web технології та арбітраж

Web технології поділяють на послідовні ітерації інтернету. Web - це всесвітня павутина (WWW), основна система пошуку інформації в Інтернеті. Зараз поділяють web на три частини:

1. Веб 1.0
2. Веб 2.0
3. Веб 3.0

Веб 1.0 У жовтні 1990 року Бернерс-Лі написав три фундаментальні технології, які стали основою Інтернету, включаючи найперший редактор/браузер веб-сторінки (WorldWideWeb.app). До середини 1990-х з появою таких веб-браузерів, як Netscape Navigator, почалася ера Web 1.0. Це епоха отримання статичних веб-сторінок із серверів, далека від гладкого вмісту, який сьогодні сприймається як належне. Більшість користувачів Інтернету в той час були в захваті від нових функцій, таких як електронна пошта та новини в режимі реального часу. Створення контенту все ще знаходиться на початковому етапі, і користувачі мають мало можливостей брати участь в інтерактивному програмуванні, хоча це покращилося, оскільки онлайн-банкінг і комерція стали популярнішими.

Веб 2.0 означає зміну парадигми використання Інтернету. За останні 15-20 років програмні веб-сторінки Web 1.0 були повністю замінені інтерактивністю Web 2.0, соціальними мережами та контентом, створеним користувачами. Web 2.0 дає вам миттєвий доступ до вмісту, створеного користувачами від мільйонів людей по всьому світу; це безпрецедентне охоплення призвело до вибуху такого вмісту в останні роки.

Експоненційне зростання Web 2.0 було зумовлено такими ключовими інноваціями, як мобільний доступ до Інтернету та соціальні мережі, а також поширенням потужних мобільних пристроїв. У другому десятилітті цього століття

ці розробки домінували в додатках, які значно розширили інтерактивність і корисність в Інтернеті, наприклад Facebook, Instagram, TikTok, Twitter, YouTube, і це лише деякі з них.

Феноменальне зростання доходів цих домінуючих платформ дозволило багатьом компаніям, орієнтованим на Web 2.0, таким як Apple, Amazon, Google, Meta (раніше Facebook) і Netflix

Веб 3.0 — це наступна ітерація або фаза еволюції Інтернету, потенційно настільки ж руйнівна і представляє таку ж значну зміну парадигми, як і Web 2.0. Web 3.0 заснований на відкритості та більшій корисності для користувачів.

У Web 3.0 комп'ютери зможуть розуміти інформацію так само, як і люди, використовуючи технології, засновані на концепціях семантичної мережі та обробки природної мови. Web 3.0 також використовуватиме машинне навчання, частину штучного інтелекту (ШІ), яке використовує дані та алгоритми для моделювання того, як люди навчаються, поступово підвищуючи їх точність. Ці можливості дозволять комп'ютерам досягати швидших та відповідніших результатів у багатьох сферах, таких як розробка ліків та нових матеріалів, а не лише цільова реклама, що є основною частиною поточних зусиль.

Використовуючи аналогію з фільмами, якщо Web 1.0 представляє епоху чорно-білих фільмів, то Web 2.0 буде епоєю кольорового/основного 3D, тоді як Web 3.0 був би зануренням у метавсесвіт.[1]

Арбітраж

Дня початку потрібно зрозуміти всі важливі аспекти і засвоїти їх значення.

Креатив – поєднання картинки, заголовка або відео, на який піде клік користувача.

Джерело трафіку – поділяють його на дві категорії:[2]

1. Класичні
 - a. Цільова реклама
 - b. Контекстна реклама
2. Альтернативні (нестандартні)
 - a. Пуш-трафік(Push-сповіщення)

в. Розсилки, спам

Офер – сайт на якому проходить реєстрація.

Зв'язка – сукупність підходів, дій та умов при роботі з конкретним офером.

Трекер – необхідний інструмент для кожного арбітражника. Він допомагає відстежити конверсії, оцінити ефективність рекламної кампанії, розділяти трафік та багато іншого.

П'ять основних речей для роботи з трафіком:

1. Акаунт в соціальній мережі на який буде «крутити» реклама
2. Налаштування рекламного кабінету у соціальній мережі
3. Домен через який вся ця реклама проходитиме
4. Трекер де відстежуватимуться дії користувачів
5. Вибір зв'язки. У зв'язку входить
 - a. Креатив
 - b. Джерело трафіку
 - c. Офер

Тепер потрібно визначитися із джерелом трафіку.

Цільова реклама – це спосіб залучити ЦА(цільова аудиторія), зібравши профіль ЦА – стать, приблизний вік, інтереси, географічне розташування — і ввівши ці дані в налаштування реклами. Всю інформацію ввічливо надають рекламодавцям великі агрегатори трафіку (в основному соціальні мережі). Це Facebook (Instagram), Reddit та багато інших ресурсів. Немає фіксованого номера, але будь-яке джерело може надати від п'ятдесяти мільйонів користувачів. Між тим, віджимання мають менший потік, але, як правило, вважаються альтернативою.

Таргетування має велику перевагу, оскільки забезпечує найбільш цільову аудиторію – ціль може навіть вибрати операційну систему пристрою.

Основні параметри націлювання:

1. Розташування
2. Вік
3. Стать
4. Інтереси

5. Місце навчання, роботи, служби
6. Сімейний стан, наявність дітей, освіти
7. Дата народження (через пару днів, сьогодні, був нещодавно і т.д.)
8. Операційна система (Android, IOS, Windows та ін.);
9. Марка пристрою

Контекстна реклама – це спосіб охоплення користувачів за допомогою їх короткострокових і довгострокових запитів. Наприклад, покажіть оголошення лише там, де релевантне вашій кампанії. Основна тематична мережа – Google Ads. Наступні плейсменти (місце розміщення) для реклами:

Пошук – ваші оголошення будуть показані у видачі пошукової системи за запитами, що відповідають тематиці реклами.

Сайти – ваші оголошення будуть показані на сайтах, підключених до даних контекстних мереж та відповідних тематиці реклами.

Крім того, Google дозволяє розміщувати контекст ще в додатках та на YouTube. У цьому випадку також враховуються нинішні (запит прямо зараз) та довгострокові (запити, що передаються) інтереси.

На рисунку 1.1 можна помітити типи запитів

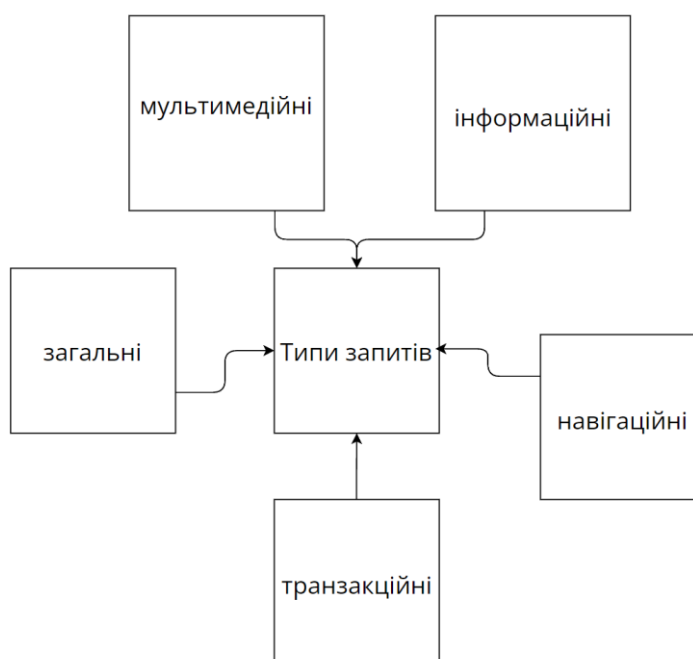


Рис. 1.1. – Типи запитів для Google Ads

Для контекстної реклами найважливішу роль грають ключові слова — запити, якими користувачі шукають вирішення своїх питань.

Пуші в арбітражі трафіку. Push-ads — це невеликі рекламні матеріали, що складаються з невеликих зображень, заголовків і короткого тексту. Зазвичай воно з'являється в нижньому правому куті ПК або у вигляді повідомлення на телефоні користувача.

Під час входу на веб-сайт користувачів часто просять «дозволити доступ до повідомлень та надіслати відповідні оновлення на веб-сайт». Погоджуючись з цим, читачі дозволяють сайту надсилати повідомлення на ваш пристрій. Звичайно, дев'яносто дев'ять відсотків матимуть не інформацію про веб-сайт, а рекламу, оскільки власник джерела підключив веб-сайт до push-мережі. Мережа платить власникам за будь-які перегляди або кліки рекламних оголошень, надісланих від імені їхнього веб-сайту.

Водночас як власник джерела push-мережі, веб-майстер виступає як джерело трафіку. Вони купують рекламу та мають можливість надсилати повідомлення з потрібними зображеннями та вмістом, а також партнерськими посиланнями, які вони надають.

Тизерові мережі в арбітражі трафіку

Тизер – це відомий тип реклами, що складається із зображення та невеликої кількості тексту. Тизери можна знайти на різних веб-сайтах.

Розсилки. Це електронні листи – ними користуються як компанії, так і веб-майстри. Цей лист може містити заклик до дії, рекламну пропозицію або інформаційний матеріал про організацію події на сайті для збільшення залучення. Клієнтська база становить понад 5 мільярдів користувачів по всьому світу. Ціна листа коливається не більше однієї гривні. Крім того, багато послуг самі є основою для деяких послуг. Якщо ні, їх можна легко придбати.

Висновок

Джерел трафіку для арбітражу маса, і щоб зрозуміти, де працювати комфортніше, доведеться перепробувати все. Можна почати з класичних джерел –

великих соцмереж, – усвідомити механіку роботи, після цього протестувати менш комфортні плейсменти. Постійні випробування – запорука тріумфу

1.2. Огляд аналогів

1.2.1. CryptoEngine

Першим аналогом для розгляду було обрано веб-сайт CryptoEngine. Сайт є одним із лідерів на ринку платформ для інвестування у криптовалюту.

Завдяки тому що платформа існує з 2015 року, в неї є доволі велика база клієнтів перевірених часом та довіра(8 269 клієнтів за рік).

На рисунку 1.2 зображено інтерфейс CryptoEngine.

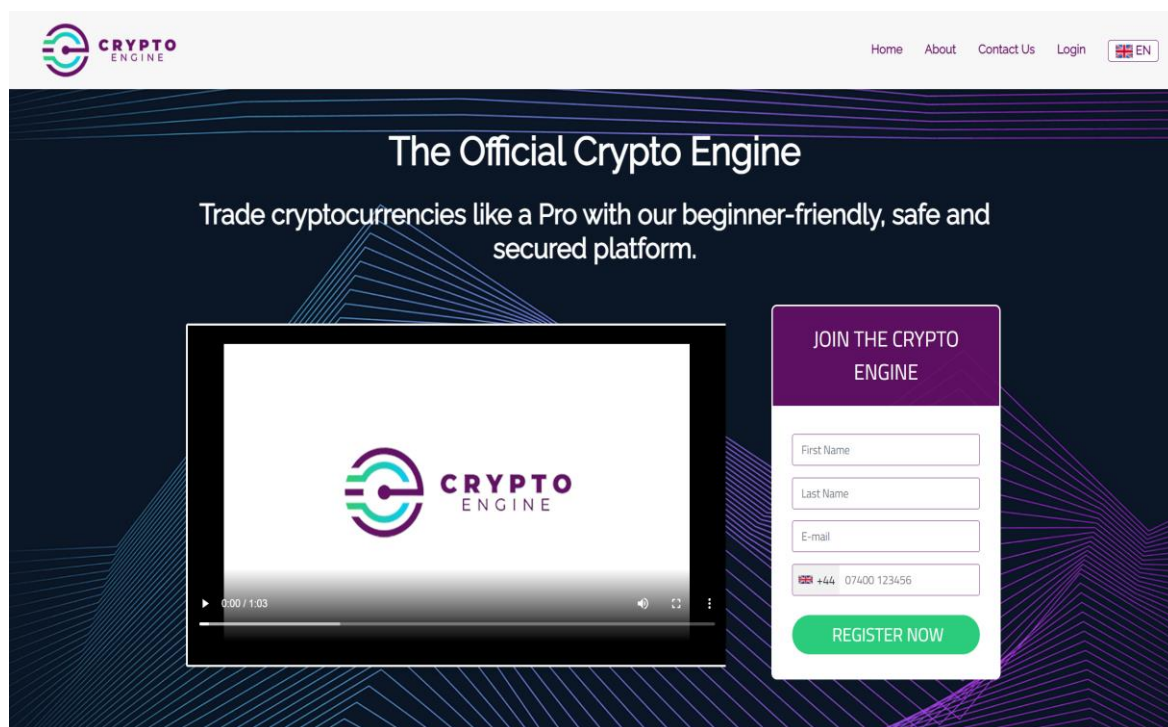


Рис. 1.2. – Інтерфейс CryptoEngine

Головним недоліком цього додатку було визначено відсутність так званої мультимови тільки під Велику Британію.

Також мінусом платформи є відсутність сторінки дякую, після реєстрації, форма повертає тебе просто на те місце, звідки ти тикнув на рекламу.

Таблиця 1 – Переваги та недоліки платформи CryptoEngine

Переваги	Недоліки
Присмний інтерфейс	Відсутність сторінки дякую
Написано все грамотно та красиво	
Перевірена часом клієнтська база	

1.2.2. TeslaX

Ця платформа вийшла відносно недавно. TeslaX вийшла в світ 30 квітня 2021 року. Але це не заважає займати почесні місця та мати велику клієнтську базу (10 315 клієнтів за рік).

На рисунку 1.3 зображено інтерфейс TeslaX.

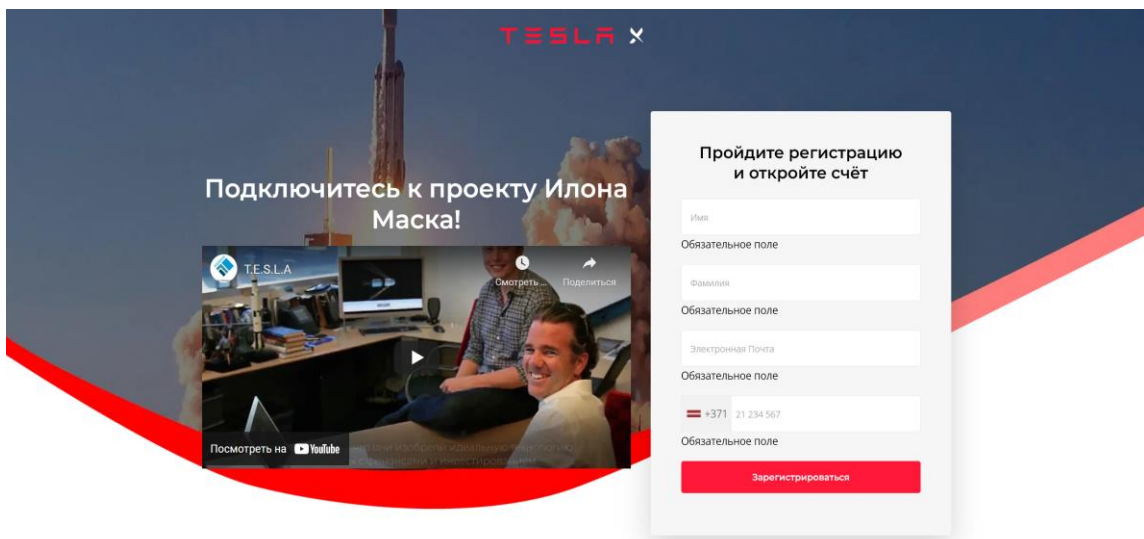


Рис. 1.3. – інтерфейс TeslaX

Додаток також націлений на криптовалюту, але з деякими змінностями від CryptoEngine.

Це проект від Ілона Маска. Це бот, в який ти вносиш кошти, а він за тебе торгує на біржі.

Позитивними сторонами сервісу можна визначити велику базу. За рік більше ніж десять тисяч клієнтів.

Таблиця 2 – Переваги та недоліки платформи TeslaX

Переваги	Недоліки
Новітня розробка у вигляді бота, який за тебе торгує	Відсутність сторінки дякую
Велика база клієнтів = великому прибутку	Відсутність підтримки мобільної версії(тільки ПК)

1.2.3. QuantumCode

QuantumCode – платформа була заснована зовсім недавно, в січні 2022 року. Але вийшла, на мою думку, та на звіти статистики зовсім не тим, що потрібно людям, майже за пів року лише 233 користувача.

На рисунку 1.4 зображено інтерфейс QuantumCode.

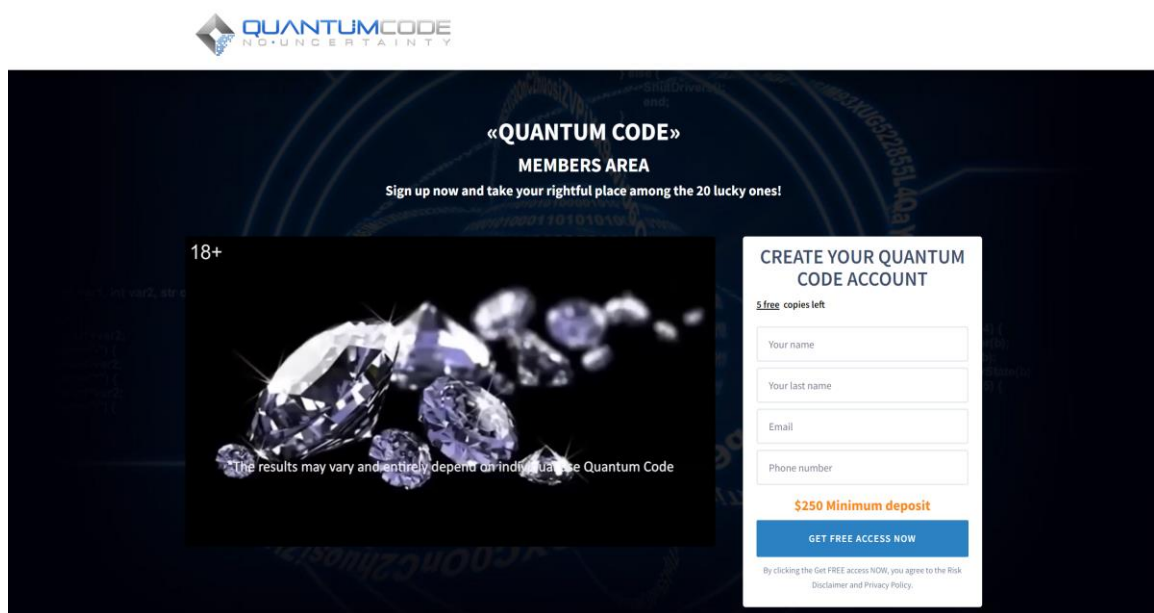


Рис. 1.4. – інтерфейс QuantumCode

Що дуже бісить в цьому офері, та що можна вважати суттєвим недоліком – швидкість завантаження, особливо при повільному інтернеті цей офер завантажується вічність.

Також немає сторінки дякую за реєстрацію, це на мою думку відштовхує користувачів.

Також відео, яке йде цілих 40 хвилин, поки подивишся відео що собою являє цей офер, то вже забудеш що тут робиш. Ще на додаток до цього всього для того щоб користуватися цим проектом – треба вносити депозит в розмірі 250\$.

Таблиця 3 – Переваги та недоліки платформи QuantumCode

Переваги	Недоліки
Підтримка мобільної версії	Для користування платформою треба внести депозит 250\$, це дуже багато
	Мінімальна швидкість завантаження(>4 секунд при дуже швидкому інтернеті), дуже дратує

1.2.4. Формування вимог до додатку

Виходячи з результатів аналізу додатків аналогів можна сформулювати наступні вимоги для розроблювального додатку:

1. Додаток повинен мати мобільну версію
2. Додаток повинен завантажуватися швидко
3. Додаток повинен мати велику клієнтську базу щоб мати змогу конкурувати з іншими додатками

1.2.5. Висновки до першого розділу

В першому розділі були розглянуті наступні теми: веб технологій, що таке арбітраж, його всі напрямлення та аспекти, існуючі аналоги платформ, виявлено їх недоліки та переваги.

Жоден з розглянутих аналогів не мав таких переваг, як залучення клієнтів будь-яким чином для збільшення клієнтської бази. Саме тому розробка цієї платформи для арбітражу є актуальною.

РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ

2.1. Обрані технології розробки

2.1.1. JavaScript та canvas

JavaScript — це мова скриптів, що використовується для розробки вбудованих програм, що виконуються як на стороні замовника, так і на стороні сервера.

На рисунку 2.1 зображено зразок коду JavaScript.

```
<script>
  document.querySelector('#phone').addEventListener('keyup', function() {
    this.value = this.value.replace(/^[^d]/g, '');
  });
</script>
```

Рис. 2.1. – Фрагмент коду JavaScript

Чому саме JavaScript?

Якщо ви вже знаєте іншу мову програмування, то вивчити JavaScript буде значно легше, ніж вивчити цю мову. Осягати свою першу мову програмування завжди складніше за все, тому що коли ви осягаєте 2-ї та подальші мови, що використовують аналогічний жанр програмування, ви вже розумієте жанр програмування, і вам просто необхідно дізнатися, як нова мова визначає свій особливий синтакси.

Відмінності у жанрах мови програмування. Мови програмування мають різні жанри. Якщо мова, яку ви вже знаєте, має такий самий жанр або парадигму, як і JavaScript, вивчити JavaScript буде досить легко. JavaScript підтримує два жанри: процедурний чи об'єктно-орієнтований. Якщо ви вже знаєте процедурну або об'єктно-орієнтовану мову, вам буде легко навчитися писати JavaScript таким же методом.

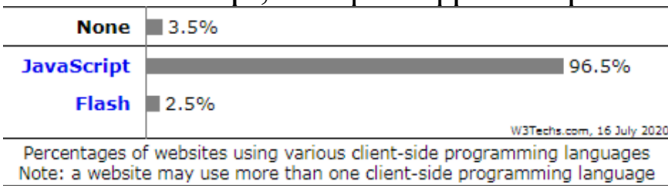
Багато в чому JavaScript є однією з найпримітивніших мов програмування для розуміння як вашої першої мови. Так як вона працює як інтерпретована мова у веб-браузері, означає, що ви можете легко написати навіть найпростіший код,

написавши його за невеликими скибочками та перевіряючи його у веб-браузері під час роботи. Навіть маленькі фрагменти JavaScript можуть бути придатні для вдосконалення веб-сторінки, тому ви можете стати продуктивними практично відразу.

Зазвичай JavaScript додається на веб сторінку між тегами або завантажується з сервера у вигляді файлу з розширенням « .js ».

JavaScript – динамічна, скриптована мова високого рівня. Найчастіше мова використовується для створення інтерактивних веб-сторінок.

Таблиця 4 – Переваги та недоліки JavaScript

Переваги	Недоліки
Підтримка скриптів всіма браузерами.	Немає підтримки віддаленого доступу. Тому мову неможливо використовувати для мережних додатків.
Повна інтеграція з версткою сторінок (HTML+CSS) та серверною частиною (backend).	Немає можливості раннього виявлення помилок. Усі недоліки виявляються вже на етапі роботи
<p>Майже у всіх сайтах використовується JavaScript, або його фреймворк</p>  <p>Percentages of websites using various client-side programming languages Note: a website may use more than one client-side programming language</p>	
Легкість освоєння	

Можливість додавання коду на сторінку. Є два можливих варіанти підключення

Перший спосіб підключення зображений на рисунку 2.2

```
<script>  
  console.log('Hi');  
</script>
```

Рис. 2.2. – Перший спосіб підключення JavaScript.

Другий спосіб підключення зображений на рисунку 2.3.

```
<script src="intlTelInput.js"></script>
```

Рис. 2.3. – Другий спосіб підключення JavaScript.

2.1.2. CANVAS

Canvas – це елемент HTML, який може працювати як полотно для скриптів (зазвичай JavaScript) для малювання. Таким чином, наприклад, можна створювати графіки, об'єднувати фотографії та створювати примітивну (і не дуже примітивну) анімацію.

Базовий API Canvas HTML5 включає 2D-контент, що дозволяє програмісту малювати різні фігури, відображати текст і відображати зображення природно в певній області вікна браузера. Можна використати колір, обертання, маніпуляції із пікселями, а також різні типи ліній, кривих, прямокутників та заливок для додавання фігур, тексту та зображень, які ви розміщуєте.[3]

На рисунку 2.4 зображено фрагмент коду Canvas

```

preload(callback) {
  let loaded = 0;
  let required = Object.keys(this.sprites).length;
  required += Object.keys(this.sounds).length;

  let onResourceLoad = () => {
    ++loaded;
    if (loaded >= required) {
      callback();
    }
  };

  this.preloadSprites(onResourceLoad);
  this.preloadAudio(onResourceLoad);
},

```

Рис. 2.4. – Фрагмент коду Canvas

Для чого використовується Canvas HTML?

Елемент HTML Canvas можна використовувати для багатьох речей, для створення яких раніше потрібно було використовувати вбудовані програми, наприклад Flash:

1. Динамічна графіка
2. Офлайн та онлайн ігри
3. Анімації
4. Інтерактивне відео та аудіо

Основна причина, чому люди застосовують елемент Canvas, полягає в тому, що легко перетворити звичайну веб-сторінку на динамічну веб-програму, а потім перетворити це на мобільний додаток для застосування на телефонах і планшетах.

Так, для повного розуміння підтримки треба перевірити чи буде працювати Canvas в браузері чи ні.[4]

У розділ <body> сторінки HTML додаємо тег <canvas>. Нам потрібне посилання на об'єкт Canvas, щоб ми знали, де відображати виклики Canvas API, які ми будемо робити за допомогою JavaScript.

Ми почнемо з визначення нової змінної під назвою theCanvas, яка міститиме посилання на об'єкт Canvas.

Далі ми отримуємо посилання на `canvasMain`, викликаючи функцію `getElementById()` документа та передаючи ім'я `canvasMain`, яке ми визначаємо як ідентифікатор `<canvas>` тегу, який ми створили на сторінці HTML

Тепер, коли у нас є посилання на елемент `canvas` на сторінці HTML, нам потрібно перевірити, чи містить він контекст. Контекст `Canvas` — це поверхня малювання, визначена веб-браузером для підтримки `Canvas`. Простіше кажучи, якщо немає контексту, немає і полотна.

На рисунку 2.5 зображено фрагмент коду перевірки на підтримку `Canvas`.

```
<canvas id="canvasMain" width="500" height="300">
  Ваш браузер не підтримує HTML5 Canvas.
</canvas>

var theCanvas = document.getElementById("canvasMain");

if (!theCanvas || !theCanvas.getContext) {
  return;
}
```

Рис. 2.5. – Перевірка підтримки `Canvas`

Наведу невеликий приклад роботи з `Canvas`:

```
<!DOCTYPE html>
<html>

<body>
  <style>
    body {
      background-color: bisque;
    }

    #Canvas {
      background-color: black;
      display: block;
    }
  </style>

```

```
margin-left: auto;
margin-right: auto;
}
</style>
<canvas id="Canvas" width="200" height="200" style="border:1px solid
#000000;">
</canvas>

</body>

</html>
```

На рисунку 2.5 Зображено результат виконання Canvas

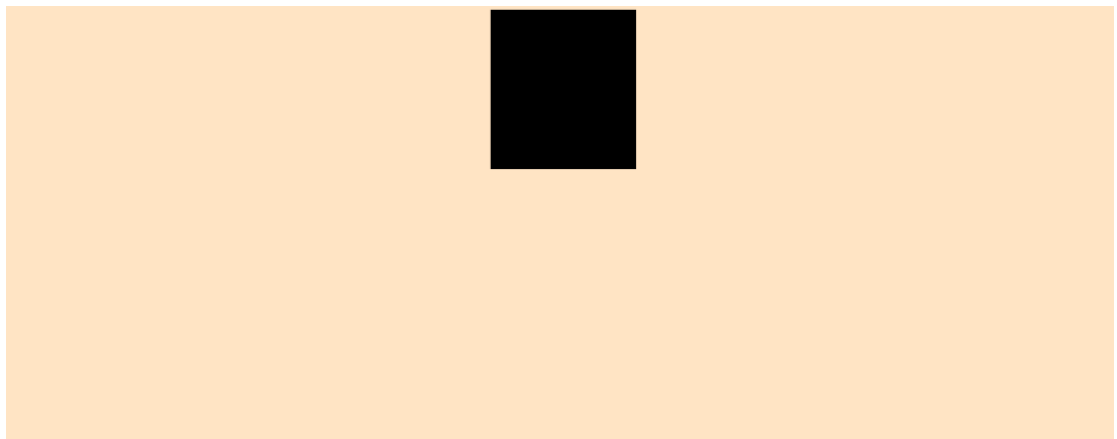


Рис. 2.6. – Результат виконання Canvas

2.1.3. Keitaro

Keitaro – повноцінний трекер, що надає широкий набір можливостей для роботи із трафіком. Наперед хочу попросити вибачення що буду багато замазувати скріншоти, це конфіденційна інформація, і я давав згоду на її нерозголошення. Дякую.[7]

На рисунку 2.7 Зображено головна сторінка Keitaro

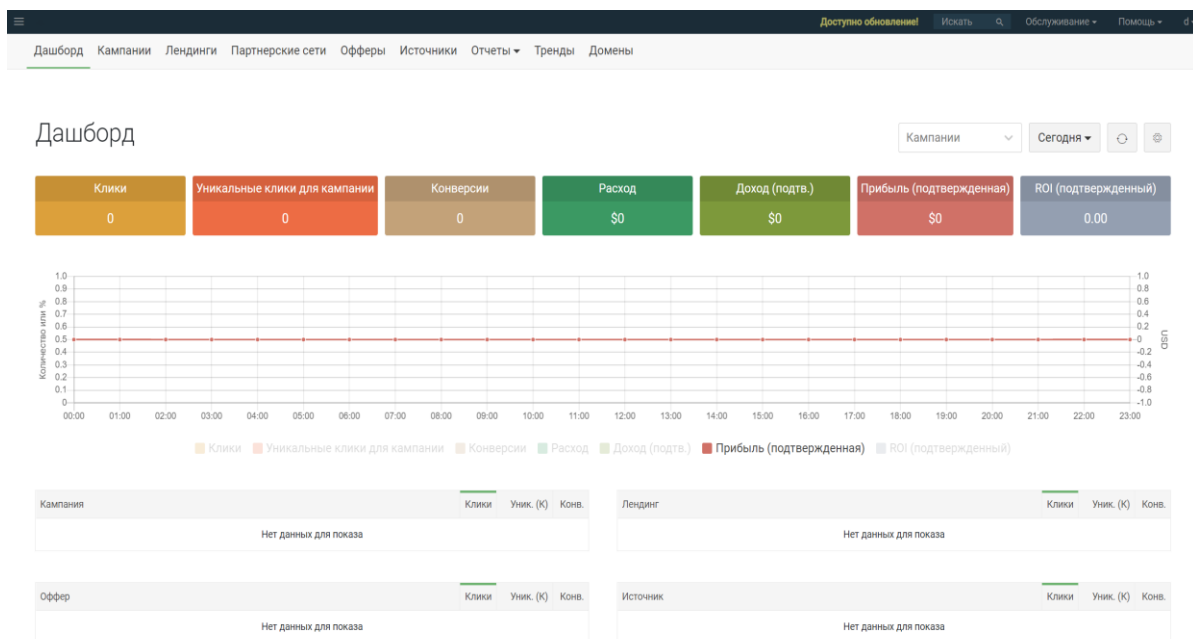


Рис. 2.7. – Головна сторінка Keitaro(дашборд)

Про гнучкість даного трекера можна розмовляти рік. Пробіжусь по головним можливостям.

На рисунку 2.8 Зображено вкладку кампанії

ID	Название	Группа	Клики	Уник. (К)	Конв.	Продажи	CR (продажи)	Доход (подтв.)	Расход	Прибыль (подтв.)	Время проведенное
1414		AK	149	126	7	0	0%	\$0	\$0	\$0	0с.
1907		DM	30	25	2	0	0%	\$0	\$0	\$0	1м. 26с.
1979		VBB	20	12	2	0	0%	\$0	\$0	\$0	29с.
1		Без группы	0	0	0	0	0%	\$0	\$0	\$0	0с.
2		Без группы	0	0	0	0	0%	\$0	\$0	\$0	0с.
3		Без группы	0	0	0	0	0%	\$0	\$0	\$0	0с.

Рис. 2.8. – Кампанії

В цій вкладці ми створюємо кампанію на яку буде йти трафік. Тут ми бачимо найголовніші параметри «кліки» – це скільки людей зробило клік на кампанії, «унікальні кліки» – це клік людини з однієї IP адреси. Тобто, якщо взяти першу кампанію. То 122 – це клік зробили 122 унікальні людини, а загалом 145 кліків, тобто декілька людей зробили пару кліків к однієї IP адреси. Далі «конверсія», в даному випадку це означає скільки людей зареєструвались, «продажі» – це скільки людей зробили покупку чи інвестування в проект.

Як створюється кампанія? Що є в налаштуваннях трекера?

На рисунку 2.9 Зображено вкладку налаштування кампаній

The screenshot shows a dashboard for managing campaigns. The top navigation bar includes: Дашборд, Кампанії, Лендинги, Партнерські сети, Оффери, Источники, Отчеты, Тренды, Домены. Below this is a search bar with 'test campaign' and a status indicator 'Кампания обновлена 19 минут назад' along with 'Сохранено', 'Лог', and a menu icon.

The main configuration area is divided into several sections:

- Настройки:** Домен (По умолчанию), Идентификатор (QYbCRT), Группа (TECH DP), Источник, Ротация потоков (Сплит-тест потоков), Привязать пользователей (Потоки, Лендинги, Офферы), Модель затрат (CPC (Оплата за клик)), Стоимость (0.0000 USD (\$)), % потерь (0.00 %), Уникальность (IP и UA), Срок уникальности (24 часов), Click API Token (fmrqgmfxc3qxyvmrvcvn5ghbfnp3gh).
- Интеграция:** Создать поток, Уравнять.
- Summary Table:**

	Вес	Клики	Уник.	Боты
Stream 1	100	0	0	0
Stream 2	100	0	0	0

Рис. 2.9. – Налаштування кампанії

Тут можна розділити на дві частини, на ліву і на праву відповідно. Ліва сторона відповідає та технічні налаштування. Туди входить:

1. Назва
2. Група(ідентифікатор користувача) до якого відноситься кампанія
3. Модель Витрат
 - a. CPC (оплата за клік)
 - b. CPA (оплата за залучення)
 - c. CPS (оплата за продаж)
4. Задати виплату вручну, або через автоматичний генератор в кодї
5. Унікальність користувача (в даному випадку рахується на унікальний IP та за унікальність, якщо користувач с UA (України))
6. Термін унікальності (скільки часу унікальність по IP буде жити у користувача), якщо через 24 години та сама людина відвідає сайт, вона знову буде унікальна
7. Click API Token (тут знаходиться 32 символи, із яких генеруються 10 символів унікального токена користувача, як тільки він потрапить на сайт)

Права сторона відповідає за потоки, які зв'язуються між собою трекером.[8]
Туди входить IM KLO – це, мовою арбітражника, клоака, яка блокує ботів, щоб не псували статистику. Розрізняють два поняття:є

1. Біла сторінка (та сторінка, яку ми вказуємо, якщо до нас «стукається» бот)
2. Чорна сторінка (та сторінка, яку ми вказуємо, якщо до нас «стукається» реальна людина)

На рисунку 2.10 Зображено IM KLO

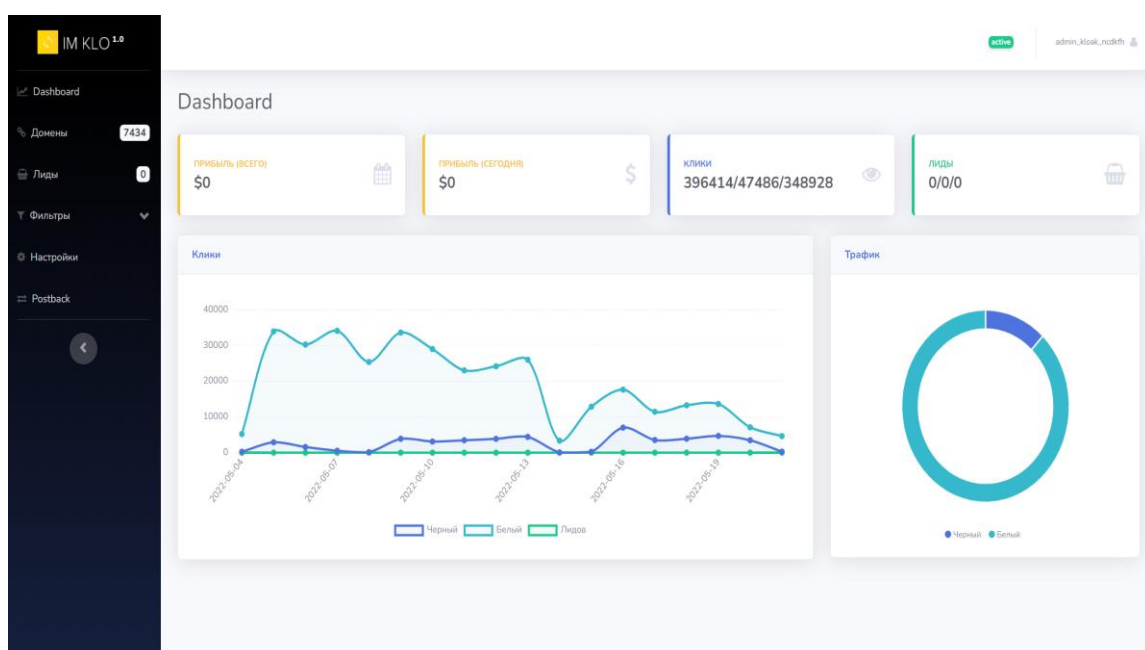


Рис. 2.10. – IM KLO

Також туди входять офери, про них на наступній сторінці.

На рисунку 2.11 Зображено вкладку офери

Дашборд Кампании Лендинги Партнерские сети Офферы Источники Отчеты Тренды Домены

Создать Группы Метрики 12

Группы Партнерские сети Состояние Сегодня

ID	Название	Сеть	Клики	Лиды	Продажи	CR	CPC	Доход (подтв.)	Расход	Прибыль (подтв.)	ROI	Уник. (П)
747			0	0	0	0%	\$0	\$0	\$0	\$0	0%	0
745			0	0	0	0%	\$0	\$0	\$0	\$0	0%	0
744			0	0	0	0%	\$0	\$0	\$0	\$0	0%	0
743			0	0	0	0%	\$0	\$0	\$0	\$0	0%	0
742			0	0	0	0%	\$0	\$0	\$0	\$0	0%	0
741			0	0	0	0%	\$0	\$0	\$0	\$0	0%	0

Рис. 2.11. – Офери

Тут як раз заводяться офери(магазин чи проект для інвестування). Сюди завантажується сайт index.html або index.php та трекер це зчитує, інтерпретує та виводить користувачу.

На рисунку 2.12 Зображено вкладку звіти → конверсії

IP	Страна	Тип устройств	Модель устройства	Время конверсии	Браузер	Лендинг
85.158.76.101	Латвия	Мобильный ...	LG K50S	21 мая 2022, 16:19	Facebook	Не задан
188.147.38.252	Польша	Мобильный ...	Samsung SM-A125F	21 мая 2022, 15:59	Chrome Mobile	PL Maximizer Short
139.47.81.50	Испания	Мобильный ...	Huawei P20 Lite	21 мая 2022, 15:52	Instagram App	ES Elomundo Tesla X short
91.129.104.106	Эстония	Мобильный ...	Xiaomi Redmi Note 8 Pro	21 мая 2022, 11:48	Facebook	Не задан
46.109.206.18	Латвия	Мобильный ...	iPhone	21 мая 2022, 10:19	Facebook	Не задан
83.26.186.91	Польша	Мобильный ...		21 мая 2022, 09:13	Facebook	PL Maximizer Short
80.235.35.56	Эстония	Мобильный ...	Xiaomi Redmi Note 8	21 мая 2022, 08:11	Facebook	Не задан
90.131.36.165	Литва	Мобильный ...	Xiaomi Redmi Note 8 Pro	21 мая 2022, 07:52	Facebook	Не задан
80.235.36.18	Эстония	Десктоп		21 мая 2022, 07:04	Microsoft Edge	Не задан
91.129.101.101	Эстония	Мобильный ...	Samsung SM-A226B	21 мая 2022, 07:02	Facebook	Не задан
90.167.7.51	Испания	Мобильный ...	Samsung GALAXY S20+ 5G	21 мая 2022, 05:27	Facebook	ES Elomundo Tesla X short

Рис. 2.12. – Звіти → конверсії

Це найголовніша вкладка, сюди приходять користувачі. Тут ми бачимо з якої кампанії прийшла людина, на який офер вона потрапила, отримали дохід з неї, чи ні, всі данні які вона ввела (пошта, телефон, ім'я, прізвище, а subid – це як раз той унікальний ідентифікатор про який писав вище, по ньому трекер і розуміє де і як відобразити користувача). Також ми бачимо IP користувача, країну звідки прийшов користувач, модель та тип пристрою, коли зареєструвався та через яким браузером користується(щоб ми могли оптимізувати сайти під більшість використовуваних браузерів).

Та остання вкладка – домени.

На рисунку 2.13 Зображено вкладку домени

ID	Домен	HTTPS	Поддомены	SSL Редирект	Индексация	Кампаний	Припарковать к кампании	Статус	Статус SSL
2321	...ns.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен
2361	...ding.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен
2362	...stand.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен
2363	...ute.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен
2364	...ook.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен
2365	...p.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен
2366	...l.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен
2367	...k.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен
2368	...e.com	Да	Нет	Да	Разрешена	0	...	Подключен	Сертификат выгружен

Рис. 2.13. – Домени

Перше що ми тут бачимо – запис для DNS (IP трекера) під який IP буде «в’язатися» домен, потім назва самого домену, його налаштування, з якою кампанією він буде зв’язаний та SSL сертифікат. Він повинен бути на кожному домені, або ваш антивірус чи браузер видасть вам попередження про безпеку ваших даних, та на сайті буде весіти ось така помітка що цей сайт не є безпечним



Таблица 5 – Переваги та недоліки Keitaro

Переваги	Недоліки
Розподіл трафіку	Не дуже зручна мобільна версія
Захист посилань	
Функціональність	
Дуже швидкий	
Є дуже велика API документація	

Його особливість полягає в тому що все в одному місці і все дуже зручно.

Статистика та звіти, робота з цільовими сторінками, захист посилань, доменів, захист від ботів, різні рівні доступів (можна настроїти більше 7 видів ролей) повна інформації в API доступі, гнучкість налаштування системи під конкретного користувача

Оплата щомісяця в розмірі 70 євро – невелик, або ні це вже вирішувати кожному.

2.1.4. HTML

HTML – це мова тегів. HTML означає Hyper Text Markup Language що в перекладі – мова гіпертекстової розмітки. HTML описує структуру web-сторінки, та складається з ряду елементів, а елементи в свою чергу повідомляють браузеру що потрібно відобразити, будь то текст, посилання, картинка чи відео[6]

Найосновніші елементи HTML 5:

1. Декларація `<!DOCTYPE html>` визначає, що цей документ є документом HTML5
2. Елемент `<html>` є кореневим елементом сторінки HTML
3. Елемент `<head>` містить мета інформацію про сторінку HTML
4. Елемент `<title>` визначає заголовок сторінки HTML (який відображається в рядку заголовка браузера або на вкладці сторінки)
5. Елемент `<body>` визначає тіло документа і є контейнером для всього видимого вмісту, такого як заголовки, абзаци, зображення, гіперпосилання, таблиці, списки тощо.
6. Елемент `<h1>` визначає великий заголовок
 - a. Розрізняють 6 типів від h1 до h6, кожен наступний менший за попередній
7. Елемент `<a>` визначає посилання
8. Елемент `` визначає картинку
9. Елемент `<p>` визначає абзац

Приклад найпростішої HTML сторінки[9]

На рисунку 2.14 зображено приклад сторінки HTML

Заголовок

Параграф



Рис. 2.13. приклад сторінки HTML

```
<!DOCTYPE html>
<html>

<head>
  <title>Диплом</title>
</head>

<body>
  <style>
    img {
      width: 20%;
    }
  </style>
```

```

    <h1>Заголовок</h1>
    <a href="https://www.dut.edu.ua/ua/"></a>
    <p>Параграф</p>
    

</body>

</html>

```

Теги, як ми бачимо бувають двох видів:

1. Закритого типу

- a. <h1></h1>
- b. <a>
- c. <p></p>

2. Відкритого типу

- a. Img

Мова яка дуже щільно пов'язана з HTML це CSS

2.1.5. CSS

CSS – це мова стилю для html сторінок, розшифровується як Cascading Style Sheets (каскадні таблиці стилів). Використовується для завдання стилю сторінкам.

Переваги CSS та недоліки.

Переваги CSS:

1. Простота самої мови CSS
2. Існують різні варіанти оформлення сторінки для різних пристроїв.
3. Легкі зміни дизайну. CSS контролює відображення кількох сторінок HTML.

Недоліки CSS:

1. Різне відображення макетів у різних браузерях. Якщо браузери застаріли, вони можуть інтерпретувати ті самі дані CSS по-різному

2.1.6. PHP

PHP означає PHP: Hypertext Preprocessor. Саме так, перша буква в аббревіатурі означає всю аббревіатуру – це, мабуть, одна з найбільш заплутаних речей у мові. Я вважаю, що його не так важко зрозуміти.

PHP — це мова сценаріїв, спеціально розроблена для створення веб-сайтів. Звичайною програмою є отримання даних з бази даних SQL. Ця мова також використовується для діаграм та іншої графіки. Дизайнери можуть посылатися на зображення або створювати галерею мініатюр.

Тришки історії мови. Спочатку (вважалося) ці букви представляли особисті домашні сторінки. Перша робота була виконана Расмусом Лерддорфом і датується приблизно тисяча дев'ятсот дев'яносто четвертим роком.

Лерддорф спочатку використовував його для своїх особистих сторінок для відстеження відвідувачів. Багато додаткових функцій було додано швидко. Однак знадобилося кілька років, щоб вона стала мовою, а не просто набором інструментів. Код був випущений в середині 90-х років. Євреї Енді Гутманс і Зеєв Сураскі переробили його в тисяча дев'ятсот дев'яносто сьомому році, **щоб запустити сайт електронної комерції з використанням PHP**. Їхня версія, яка називається PHP 3.0, має більше можливостей, яких ми чекаємо і бачимо сьогодні.

Пізніше дует створив версію 4.0. У дві тисячі четвертому році відбулася ще одна серйозна редакція PHP 5.0. Однак версія 6.0 ще не випущена, а після версії 5.0 були значні покращення, включаючи видалення деяких речей, які викликали нестабільність або потенційні діри в безпеці.

PHP виконує загальні функції веб-сайту, такі як прийняття паролів, аутентифікація користувачів, керування повідомленнями на форумі та гостьовими книгами. Ще одна поширена програма – інтеграція з Facebook. Використовуючи PHP, ви навіть можете створити веб-сайт, який дозволяє користувачам створювати контент, особливо тим, хто навіть не знає основ HTML.

Основна відмінність PHP від традиційних форм HTML полягає в тому, що PHP є динамічним. Однак це теж не пропозиція. Дві мови легко поєднуються. HTML-код може бути згенерований PHP.

На рисунку 2.15 зображено приклад коду PHP

```

1  <?php
2  $a = 'fmrqgmfxc3qtxyvmnrvcvn5ghbfnp3gh';
3  $b = strlen($a);
4  echo $b;
5  ?>

```

Рис. 2.15. приклад коду PHP

Переваги PHP

1. Висока швидкість роботи
2. Простота освоєння, простий синтаксис.
3. Багатозадачність та широкі можливості

Недоліки PHP

1. Іноді нові версії PHP мають кардинальні зміни

Та трішки про структуру реєстрації та відправки даних, де вони зберігаються, як оброблюються. На рисунку 2.16. зображено основну сторінку реєстрації

Рис. 2.16. Основна сторінка реєстрації

Далі йде вибір між простою реєстрацією «Login» та можливістю зіграти в гру «Play the game and get a discount!». На рисунку 2.17. зображено сторінку дякую, після реєстрації

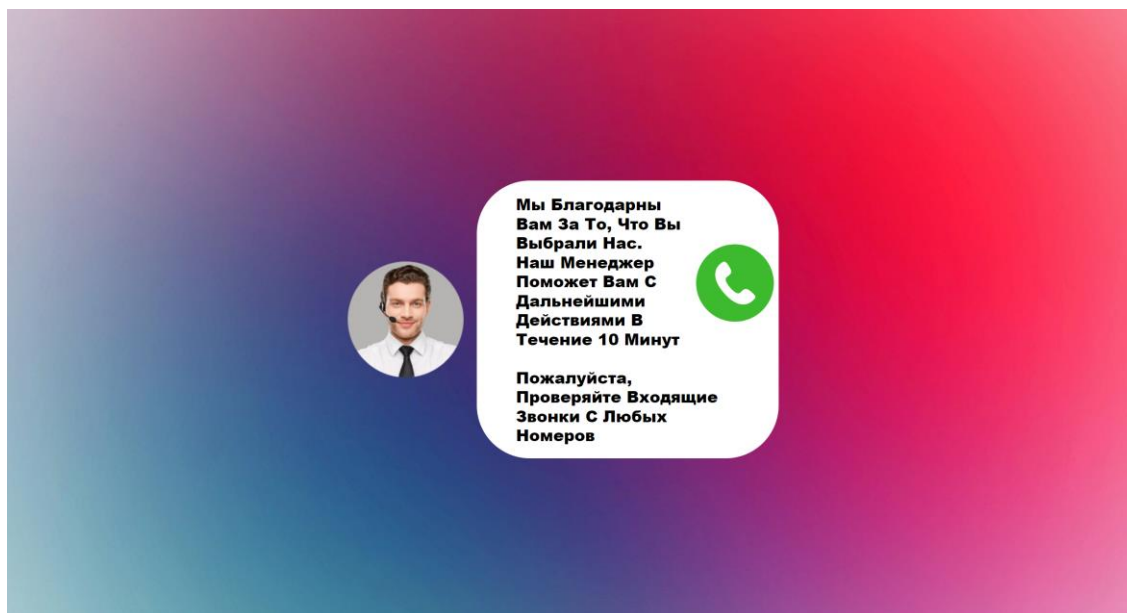


Рис. 2.17. Сторінка дякую

Наступним кроком розглянемо саму гру. На рисунку 2.18 зображено момент гри

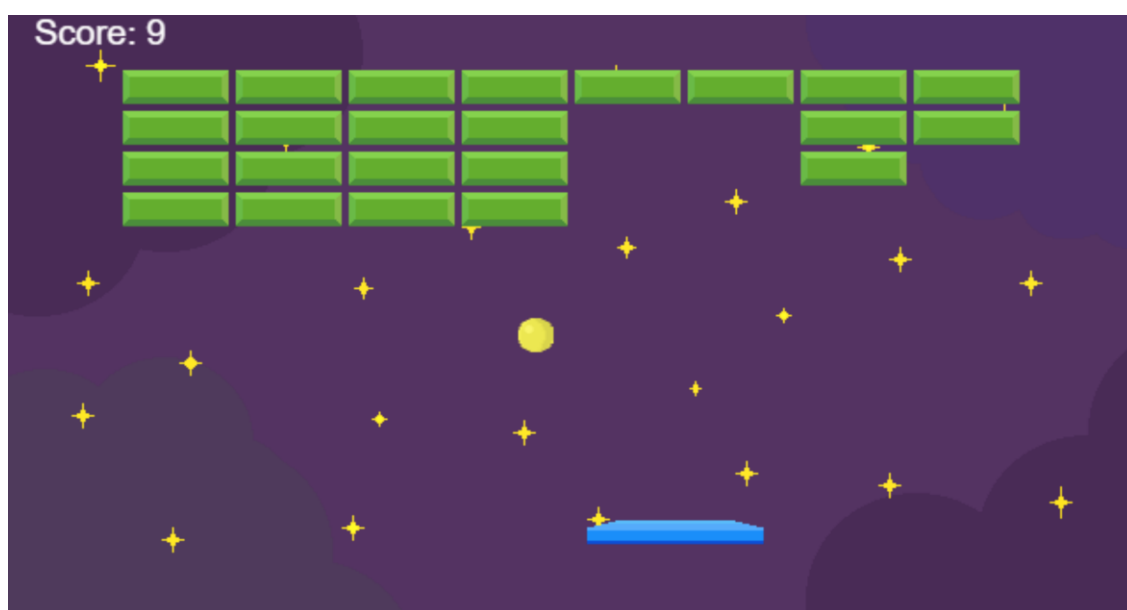


Рис. 2.18. Момент гри

Тут бачимо угорі зліва є лічильник, який рахує скільки балів отримав користувач. Також є певний бекграунд (фон), блоки, м'яч та платформа, яка відбиває цей м'яч. На рисунку 2.19 зображено сторінку реєстрації після гри, та момент передачі кількості балів. Як бачимо, кнопка яка пропонувала гру – зникла.

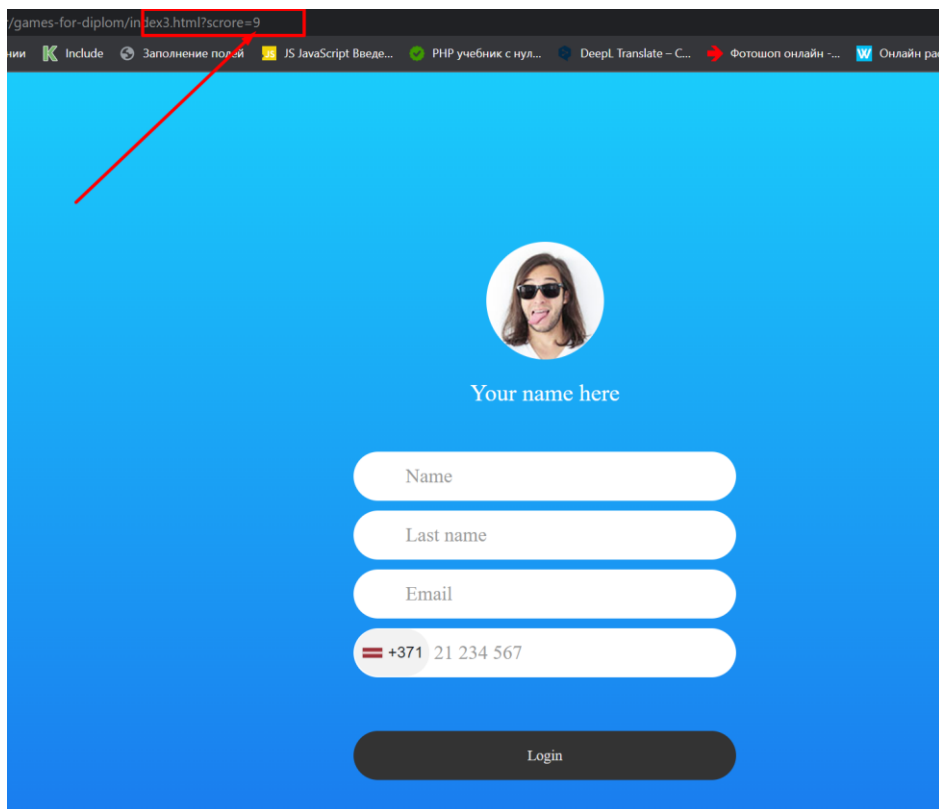


Рис. 2.19. Форма реєстрації після гри та передача балів

На цьому рисунку бачимо, що в адресний рядок додався параметр із кількістю балів та зникла кнопка яка пропонує гру. На рисунку 2.20 зображено як це все зберігається на програмній стороні. «1» відповідає за прийом параметрів, те що ввів користувач, та результат гри. «2» ініціалізація та перевірка на знижку користувача, в залежності скільки балів прийшло – рахується знижка користувача. «3» передача цих всіх параметрів у масив, а масив в свою чергу відправляє два запити, на трекер Keitaro та на сервіс Webhook.

```

33 $email = $_GET['email'];
34 $fname = $_GET['username'];
35 $lname = $_GET['lastname'];
36 $phone = $_GET['tel'];
37 $ip = $_SERVER['REMOTE_ADDR'];
38 $score = $_GET['score'];
39
40 if ($score <= 10) {
41     $sale = 5;
42 }
43 elseif ($score > 10 && $score <= 20) {
44     $sale = 10;
45 }
46 elseif ($score > 20 && $score <= 30) {
47     $sale = 15;
48 }
49 elseif ($score > 30) {
50     $sale = 20;
51 } else {
52     $sale = "Пробач, не визначив знижку користувача";
53 }
54
55 $data = [
56     'firstname' => $fname,
57     'lastname' => $lname,
58     'email' => $email,
59     'phone' => $phone,
60     'iso' => $result,
61     'ip' => $ip,
62     'from_url' => "http://example.com",
63     'test' => "test",
64     'score' => $score,
65     'sale' => $sale,

```

Рис. 2.20. Структура обробки даних

А ось в такому вигляді, як на рисунку 2.21 зображено як це все приходить на сервер.

```

1  firstname=test
2  lastname=test
3  email=test%40gmail.com
4  phone=1231231231231
5  iso=LV
6  ip=195.123.209.224
7  from_url=http%3A%2F%2Fexample.com
8  test=test
9  score=11
10 sale=10

```

Рис. 2.21. Вид, в якому дані приходять на сервер

2.2. Середовище розробки

2.2.1. Visual Studio Code

Visual Studio Code – Редактор коду для Windows, Mac OS та Linux зі спеціальними налаштуваннями Microsoft, випущений під традиційною ліцензією на продукт Microsoft.

Visual Studio Code поєднує в собі простоту редактора коду з тим, що потрібно розробникам для їх основного циклу редагування-складання-налагодження. Він забезпечує повну підтримку редагування коду, навігацію та розуміння, а також легке налагодження, багату модель розширюваності та легку інтеграцію з існуючими інструментами. Програма була представлена вперше в 2015 році, на конференції Build.

Цей редактор допоможе веб-розробникам і верстальникам усіх рівнів. З одного боку, він підходить для новачків, оскільки його інтерфейс інтуїтивно зрозумілий і простий. З іншого боку, VS Code має багато вбудованих функцій, які цікавлять досвідчених розробників..

На рисунку 2.22 зображено інтерфейс VS Code

```
1  <?php
2
3  $client = @$_SERVER['HTTP_CLIENT_IP'];
4  $forward = @$_SERVER['HTTP_X_FORWARDED_FOR'];
5  $remote = @$_SERVER['REMOTE_ADDR'];
6  $result = array('country'=>'', 'city'=> '');
7
8  if(filter_var($client, FILTER_VALIDATE_IP)) $ip = $client;
9  elseif(filter_var($forward, FILTER_VALIDATE_IP)) $ip = $forward;
10 else $ip = $remote;
11
12 $ip_data = @json_decode(file_get_contents("http://www.geoplugin.net/json.gp?ip=".$ip));
13 if($ip_data && $ip_data->geoplugin_countryName != null)
14 {
15     $result = $ip_data->geoplugin_countryCode;
16 }
17
```

Рис. 2.22. Інтерфейс VS Code

2.3. Проектування додатку

Насамперед, потрібно розробити структуру сайту. На рисунку 2.23 зображено структуру сайту.

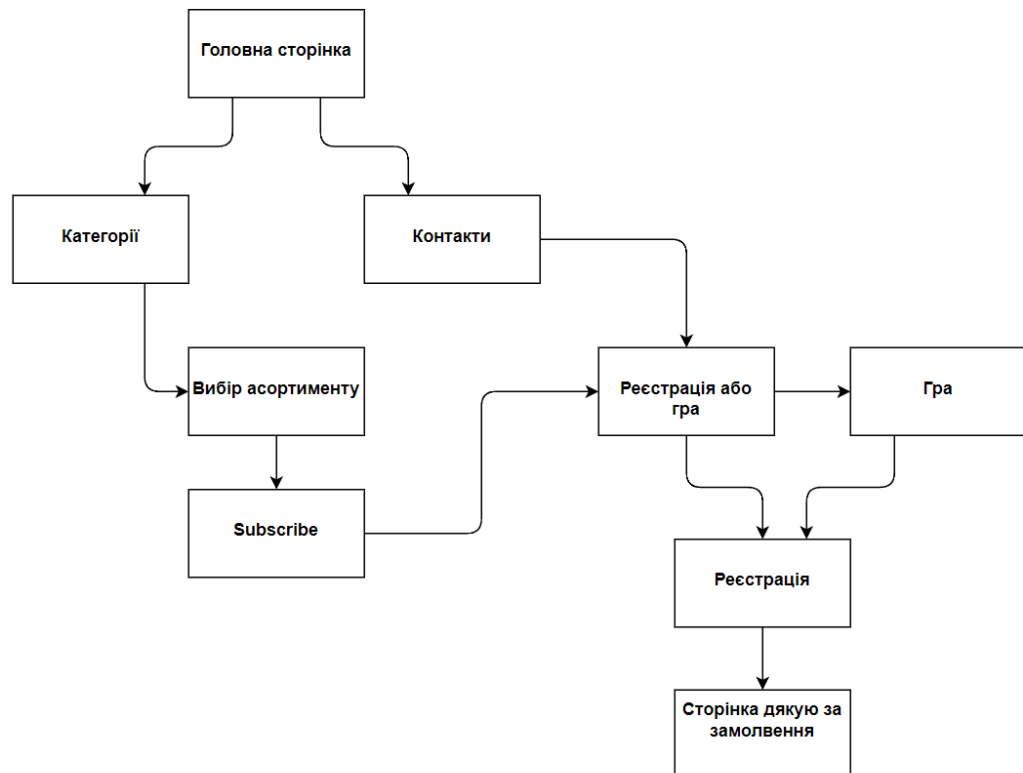


Рис. 2.23. приклад сторінки HTML

Для розробки додатку структура взаємодії буде оптимальною, коли дані обробляються більше на стороні клієнта, а сервер зберігає лише базу даних.

Майже вся обробка даних виконується на стороні клієнта, за винятком процедур забезпечення цілісності даних та іншої логіки, пов'язаної з обслуговуванням бази даних, яка переважно виконується на сервері

На рисунку 2.24 зображено структуру взаємодії.

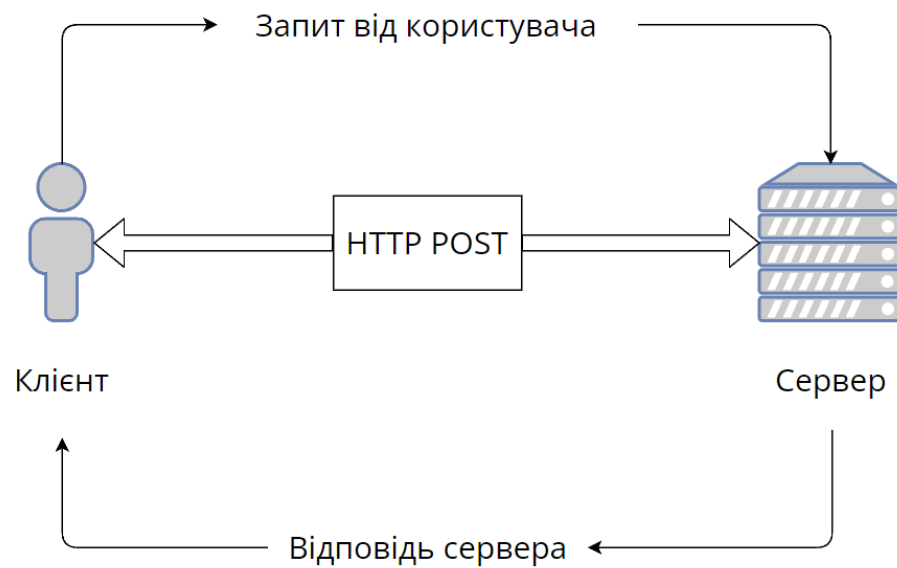


Рис. 2.24. структура взаємодії.

Висновки до другого розділу

У другому розділі було розглянуто питання технологій, які слід обрати скільки для розробки додатку. Було розглянуто мови програмування, середовище розробки додатку та трекер для управління клієнтською базою, налаштування та відслідковування трафіку

Виявлено позитивні та негативні риси:

1. Стеку мов програмування.
2. Середовищу розробки.
3. Трекеру.

Виходячи з розглянутих даних отримано наступні результати:

1. Для розробки веб додатку було обрано мову програмування PHP.
2. Для формування сторінки та її стилів HTML + CSS.
3. Для формування форми та валідації JavaScript.
4. Для налаштування та відслідковування трафіку – трекер Keitaro.
5. Середовище розробки Visual Studio Code

Сьогодні для людини, що працює з комп'ютерами, важливо вивчити хоча б одну мову програмування, тому що всі інновації та технології засновані на глибокому розумінні комп'ютерів, операційної системи, програмного API або якоїсь апаратної периферії. Всі вони створені програмістами, використовують певний спосіб мислення. І щоб набути такий спосіб мислення, потрібно звикнути хоча б до однієї з мов програмування і отримати кваліфікацію в області розробки програмного забезпечення.

Для будь-якої людини, яка починає вивчати програмування, важливо зосередитися на поняттях програмування, а не на специфіці мови, тому що вони можуть бути різними для різних мов програмування, але Python забезпечує найвищий рівень програмування. Таким чином, людині не потрібно думати про управління пам'яттю, яке неминуче в C ++, або про ієрархію класів, яка неминуча в Java, або про типи змінних та оголошеннях, які існують майже в кожній мові програмування.

РОЗДІЛ 3. РОЗРОБКА ПЛАТФОРМИ

3.1. Загальні відмоті

За результатами попереднього розділу для розробки веб додатків були вибрані наступні технології:

1. Клієнтська частина буде написана на HTML – відповідає за розмітку веб сторінки, CSS, JavaScript – відповідають за стилі та валідацію;
2. База даних буде братися із трекеру;
3. Серверна частина буде написана на мові PHP.

Місце розташування веб додатку – сервер від провайдера. На цьому рисунку 3.1 зображена конфігурація сервера.

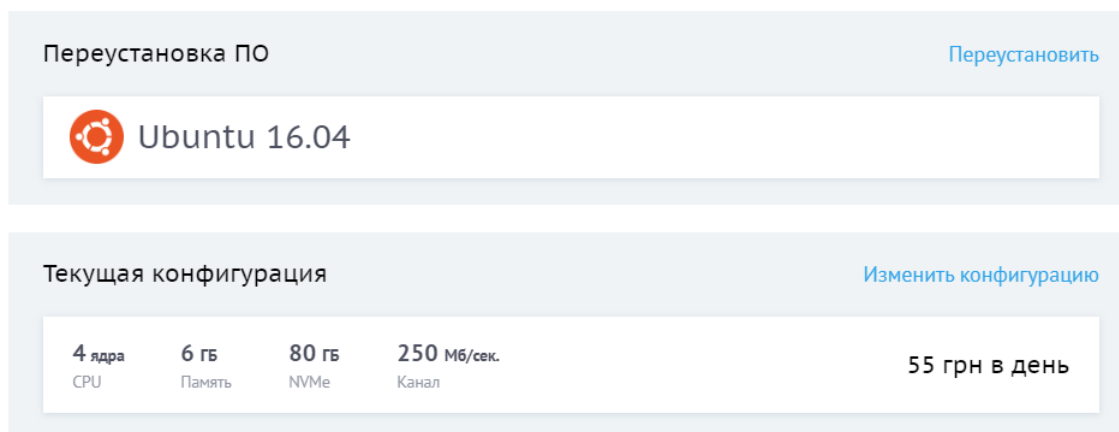


Рис. 3.1. Приклад оренди сервера

Серверна частина додатку відповідає за зв'язок між трекером та інтерфейсом користувача.

Клієнтська частина відповідає за вигляд, який бачить користувач.

3.2. Серверна частина

Серверна частина додатку була виконана за допомогою мови програмування PHP.

Додаток повинен відповідати наступним критеріям:

1. Отримувати та розпізнавати введені користувачем дані (ім'я, фамілію, пошту та телефон)
2. Отримувати та зберігати в трекері всі дані від користувача
3. Розпізнавати дії користувача, якщо він буде грати в гру, та передати ці значення в трекер.

Для виконання всіх критеріїв першочергово потрібно зв'язатися з трекером.

Параметр `subid` з'єднується з трекером, тому потрібно його ініціалізувати.

Параметр `subid` тільки зв'яже користувача з трекером, для того щоб отримувати кліки – нам потрібен `pixel`

На рисунку 3.2 показаний приклад генерування `subid` та `pixel`.

```

<script>
function getUrlParameter(sParam) {
    var sPageURL = window.location.search.substring(1);
    var sURLVariables = sPageURL.split('&');
    for (var i = 0; i < sURLVariables.length; i++) {
        var sParameterName = sURLVariables[i].split('=');
        if (sParameterName[0] == sParam) {
            return sParameterName[1];
        }
    }
}

var elements = document.getElementsByTagName("a");
for (let i = 0; i < elements.length; i++) {
    elements[i].setAttribute("href", "{offer}&sub3=" + getUrlParameter('sub3') + "&sub2={subid}");
}
</script>

```

Рис. 3.2. Приклад генерування `subid` та `pixel`.

Далі нам потрібно помістити цей фрагмент коду на головну сторінку перед закриваючим тегом `</body>`.

Дані передаються на `webhook`, за допомогою PHP кода зображеного на рисунку 3.3.

Далі нам потрібно помістити цей фрагмент коду на головну сторінку перед закриваючим тегом `</body>`.

Дані передаються на `webhook`, за допомогою PHP кода зображеного на рисунку 3.3.

```
20 $url = "https://webhook.site/5cb790b0-c4e1-4b1e-ac80-29f6de6c1310";
21
22 $curl = curl_init($url);
23 curl_setopt($curl, CURLOPT_URL, $url);
24 curl_setopt($curl, CURLOPT_POST, true);
25 curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
26
27 $headers = array(
28     | "Content-Type: application/json",
29 );
30 curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
31
32 $email = $_GET['email'];
33 $fname = $_GET['username'];
34 $lname = $_GET['lastname'];
35 $phone = $_GET['tel'];
36 $ip = $_SERVER['REMOTE_ADDR'];
37 $score = $_GET['score'];
```

Рис. 3.3. передача даних за допомогою PHP.

Webhook – це такий сервіс, куди можна відправляти запити, і вони там також будуть збережені у JSON форматі.

JSON – текстовий формат даних, який використовується практично у всіх скриптових мовах програмування

На рисунку 3.4. зображено структуру JSON

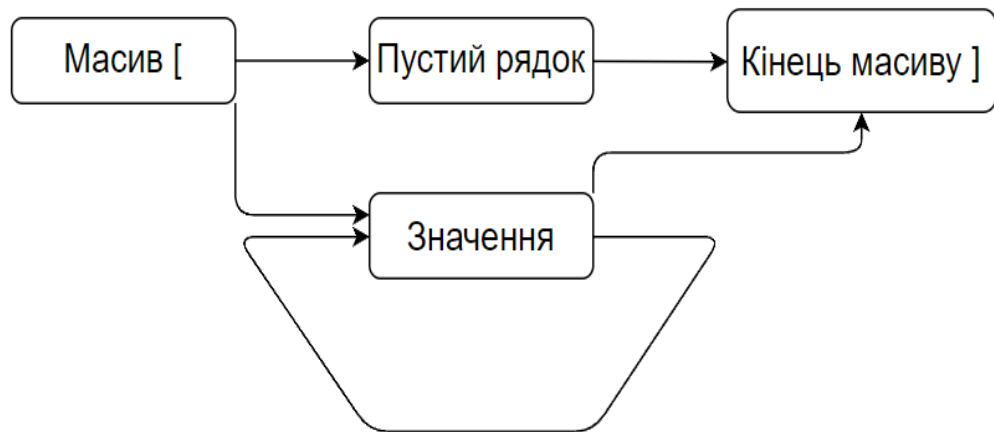


Рис. 3.4. структура JSON.

Також потрібно дізнаватися з якої країни прийшов користувач.

Для цього використовуємо geoIPplugin звідки будемо діставати код країни, для того щоб правильно відобразити сторінку дякую

На рисунку 3.5. забражено структуру geoIPplugin

```

<?php

$client = @$_SERVER['HTTP_CLIENT_IP'];
$forward = @$_SERVER['HTTP_X_FORWARDED_FOR'];
$remote = @$_SERVER['REMOTE_ADDR'];
$result = array('country'=>'', 'city'=> '');

if(filter_var($client, FILTER_VALIDATE_IP)) $ip = $client;
elseif(filter_var($forward, FILTER_VALIDATE_IP)) $ip = $forward;
else $ip = $remote;

$ip_data = @json_decode(file_get_contents("http://www.geoplugin.net/json.gp?ip=".$ip));
if($ip_data && $ip_data->geoplugin_countryName != null)
{
    $result = $ip_data->geoplugin_countryCode;
}

echo $result;

?>

```

Рис. 3.5. фрагмент коду структури geoIPplugin

Сама сторінка дякую реалізована як універсальна, по коду вище буде зчитуватися код країни, наприклад (PL – Польща). Виходячи з цього, сторінка відобразиться на польській мові.

На рисунку 3.6. зображено сторінку (дякую за реєстрацію) на польській мові.

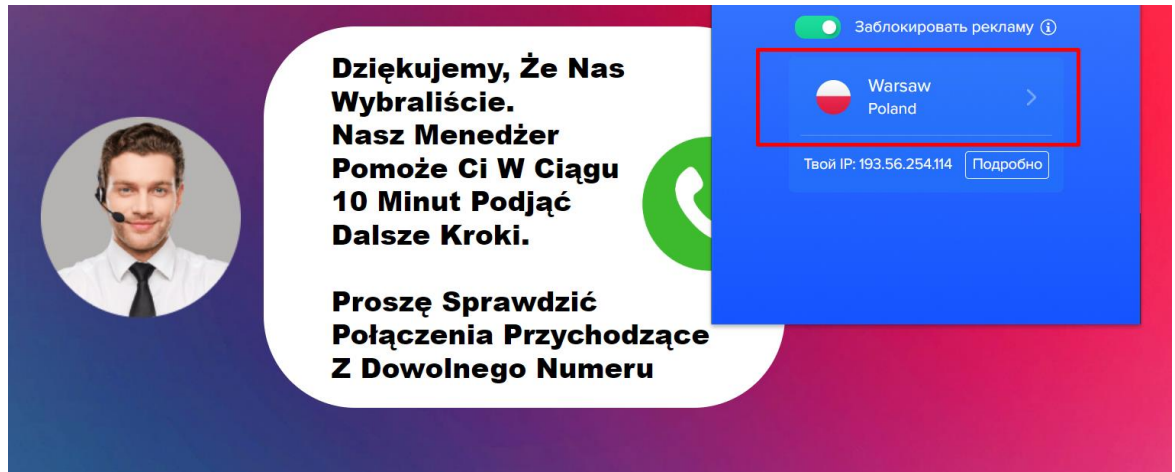


Рис. 3.6. сторінка дякую на польській мові.

Також потрібно відправити запит на трекер, щоб в ньому все відобразилося вірно. Для цього застосовуємо PHP метод `file_get_contents()`;

На рисунку 3.7. зображено відправку на трекер всіх даних.

```

83 $postback = "https://punchagency.space/ab0fa69/postback";
84 $sendpostback = $postback . "?payout=0&status=lead&subid="
85 . urlencode($_GET['subid'])
86 . "&sub_id_10=" . urlencode($email)
87 . "&sub_id_11=" . urlencode($phone)
88 . "&sub_id_12=" . urlencode($fname)
89 . "&sub_id_13=" . urlencode($lname);
90
91 file_get_contents($sendpostback);

```

Рис. 3.7. відправка даних на трекер.

3.3. Canvas гра

Початкова сторінка гри зображена на на рисунку 3.8.

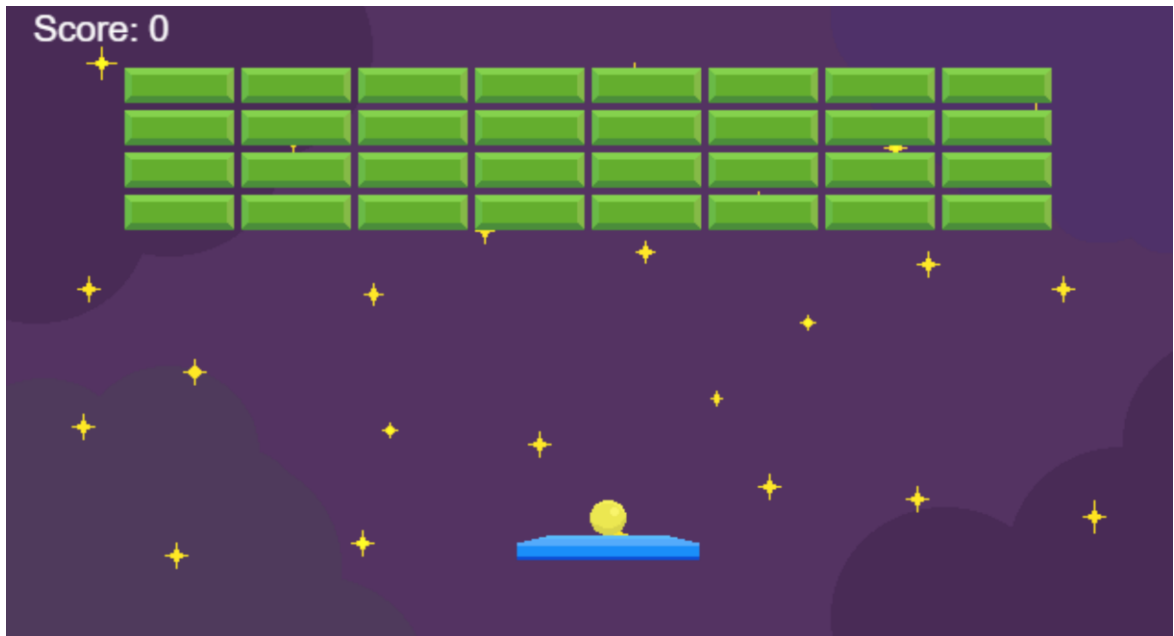


Рис. 3.8. початкова сторінка гри.

Гра була реалізована за допомогою Canvas на мові JavaScript.

Для початку треба було намалювати всі елементи, задній фон, м'яч, платформу яка рухається, та блоки які будемо збивати. Це я зробив у Adobe Photoshop

Далі потрібно було створити всі необхідні константи та параметри, на яких буде конструюватися гра. На рисунку 3.9 зображено процес створення констант та параметрів

```

1  const KEYS = {
2      LEFT: 37,
3      RIGHT: 39,
4      SPACE: 32
5  };
6
7  let game = {
8      running: true,
9      ctx: null,
10     platform: null,
11     ball: null,
12     blocks: [],
13     score: 0,
14     rows: 4,
15     cols: 8,
16     width: 640,
17     height: 360,
18     sprites: {
19         background: null,
20         ball: null,
21         platform: null,
22         block: null
23     },
24     sounds: {
25         bump: null,
26     },

```

Рис. 3.9. процес створення констант та параметрів.

За допомогою `render` розташовуємо та виводимо на екран всі блоки, платформу та м'яч

На рисунку 3.10 зображено процес розташування та виводу елементів гри

```

128     Debrating B |
129     render() {
130         this.ctx.clearRect(0, 0, this.width, this.height);
131         this.ctx.drawImage(this.sprites.background, 0, 0);
132         this.ctx.drawImage(this.sprites.ball, this.ball.frame * this.ball.width, 0, this.ball.width, this.ball.height, this.ball.x, this.ball.y, this.ball.width);
133         this.ctx.drawImage(this.sprites.platform, this.platform.x, this.platform.y);
134         this.renderBlocks();
135         this.ctx.fillText("Score: " + this.score, 15, 20);
136     },
137     Debrating B |
138     renderBlocks() {
139         for (let block of this.blocks) {
140             if (block.active) {
141                 this.ctx.drawImage(this.sprites.block, block.x, block.y);
142             }
143         }
144     },

```

Рис. 3.10. процес розташування та виводу елементів гри.

Далі нам потрібно все між собою зв'язати, для цього створюємо метод `update()` та реалізуємо метод `collide()` який відповідає за зіштовхування блоків та м'яча.

На рисунку 3.11 зображено процес створення логіки для оновлення гри та взаємодію предметів між собою

```

89     update() {
90         this.collideBlocks();
91         this.collidePlatform();
92         this.ball.collideWorldBounds();
93         this.platform.collideWorldBounds();
94         this.platform.move();
95         this.ball.move();
96     },
97     addScore() {
98         ++this.score;
99
100         if (this.score >= this.blocks.length) {
101             this.end("Ви перемогли");
102         }
103     },
104     Debt rating B |
105     collideBlocks() {
106         for (let block of this.blocks) {
107             if (block.active && this.ball.collide(block)) {
108                 this.ball.bumpBlock(block);
109                 this.addScore();
110                 this.sounds.bump.play();
111             }
112         }
113     },
114     collidePlatform() {
115         if (this.ball.collide(this.platform)) {
116             this.ball.bumpPlatform(this.platform);
117             this.sounds.bump.play();
118         }
119     },
120     run() {
121         if (this.running) {
122             window.requestAnimationFrame(() => {
123                 this.update();
124                 this.render();
125                 this.run();
126             });
127         }
128     },

```

Рис. 3.11. процес створення логіки гри.

В кінці гри вам виведеться результат – скільки балів ви набрали та який відсоток знижки ви отримали.

На рисунку 3.12 зображено кінець гри та вивід результатів

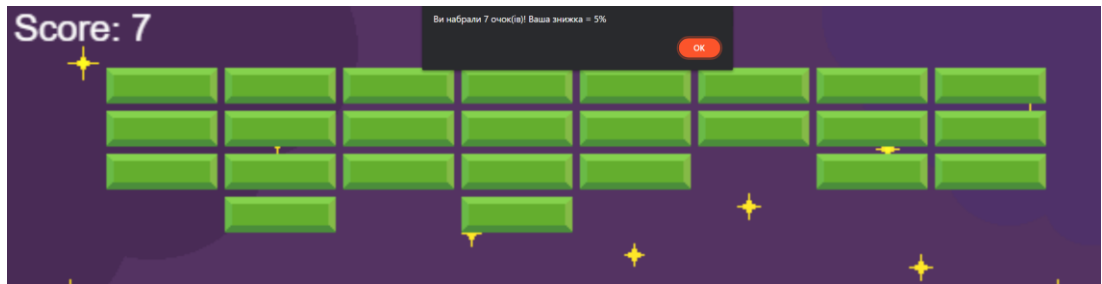


Рис. 3.12. кінець гри та вивід результатів

3.4. Реєстрація клієнтів

Була виконана за допомогою мови програмування JavaScript та її бібліотеки International Telephone Input.

Форма виглядає так, як на рисунку 3.13.

A registration form on a blue gradient background. At the top center is a circular profile picture of a woman with sunglasses. Below it is the text "Your name here". The form consists of five white rounded rectangular input fields: "Name", "Last name", "Email", and a phone number field containing "+48 512 345 678" with a red flag icon on the left. Below the input fields are two dark grey rounded rectangular buttons: "Login" and "Play the game and get a discount!".

Рис. 3.13. форми реєстрації

Найголовніша можливість це відслідковування по IP, та в залежності яке IP у користувача – той код країни і буде показаний.

На рисунку 3.14 зображено код який відповідає за відслідковування по IP, та ініціалізацію плагіну International Telephone Input. Ось так це виглядає «під капотом».

```
100 - <script>
101     var input = document.querySelector("#phone");
102 -     window.intlTelInput(input, {
103         initialCountry: "auto",
104         placeholderNumberType: "MOBILE",
105         autoPlaceholder: "polite",
106         customPlaceholder: null,
107 -         geoIpLookup: function(callback) {
108             $.get('https://ipinfo.io', function() {}, "json").always(function(resp) {
109                 var countryCode = (resp && resp.country) ? resp.country : "us";
110                 callback(countryCode);
111             })
112         },
113         utilsScript: "https://cdnjs.cloudflare.com/ajax/libs/intl-tel-input/17.0.8/js/utils.js",
114     });
115 </script>
```

Рис. 3.14. ініціалізація плагіну International Telephone Input

3.5. Тестування

3.5.1. Інтерфейсу користувача

Додаток повністю пройшов тестування інтерфейсу користувача, в ньому повністю відсутні всі баги. Також присутня адаптація до екрану в різних мобільних та планшетних пристроїв.

Також були знайдені та виправлені:

1. Помилки в інтерфейсі.

На рисунку 3.14 показано тестування адаптації веб додатку на формі реєстрації та на основному сайті

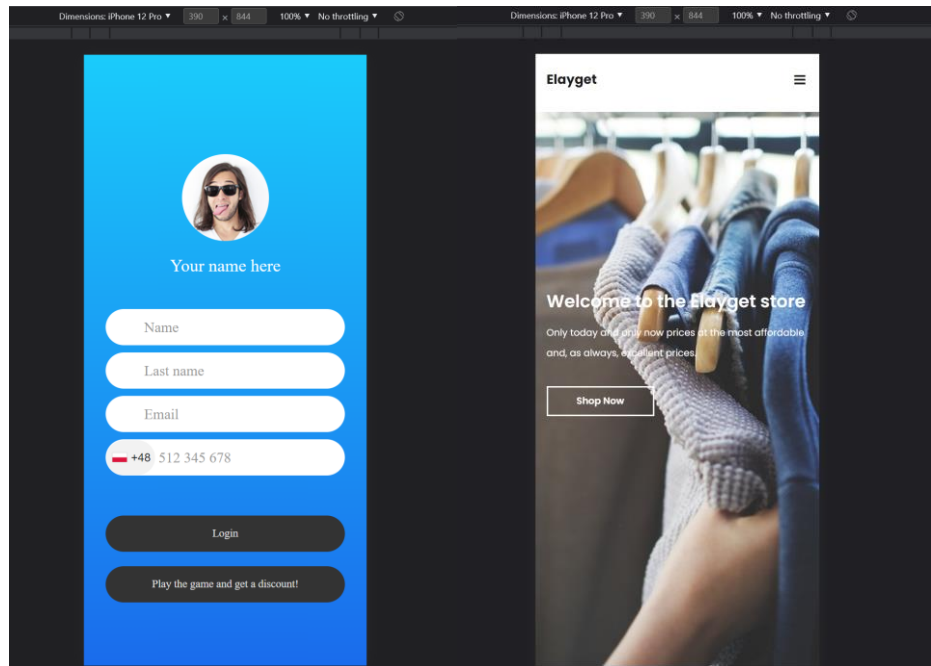


Рис. 3.14. вигляд мобільної версії додатку на формі реєстрації та на основному сайті

3.5.2. Функціональне тестування

Функціональні тести повинні охоплювати всі реалізації додатку.

Підчас цього виду тестування було виявлено наступні баги:

1. Не надходила кількість набраних балів після гри.
2. Не відпрацьовував плагін International Telephone Input;
3. Невірний підрахунок балів, лічильник нараховував менше менше балів, ніж насправді

3.6. Висновки до третього розділу

У третій частині тема обговорення: повноцінна розробка веб-додатку.

Після заповнення цього розділу розробка та тестування програми було завершено. Всі сайти зв'язані з трекером. Трекер та серверна частина були розроблені на мові програмування PHP. Частина додатку для користувачів була розроблена на HTML, CSS, JavaScript та його плагіні International Telephone Input.

ВИСНОВКИ

У ході роботи над кваліфікаційною роботою було виконано усі поставлені завдання.

Був проведений аналіз усіх переваг та недоліків аналогів.

У процесі аналізу були вибрані наступні технології для того, щоб відтворити всі завдання: HTML, CSS, JavaScript, PHP, Keitaro, International Telephone Input, та середовище розробки Visual Studio Code.

Результатом роботи, є розробка веб-магазину зі знижковою системою у вигляді гри.

ПЕРЕЛІК ПОСИЛАНЬ

- 1) <https://www.investopedia.com/web-20-web-30-5208698#toc-the-bottom-line>
- 2) <https://affhub.net/traffic-sources-for-affiliate-marketing/>
- 3) https://artradarjournal.com/paintings/canvas/what-is-canvas-js/#qa_2
- 4) <https://www.geeksforgeeks.org/html-canvas-basics/>
- 5) <https://www.thoughtco.com/how-hard-is-javascript-to-learn-2037676>
- 6) <https://keitaro.io/en/>
- 7) https://gdetraffic.com/Obzory/Obzor_Keitaro_funktsionalnogo_treker_a_dlya_arbitrazhnikov
- 8) https://www.w3schools.com/html/html_intro.asp

ДОДАТОК А

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра Інженерії програмного забезпечення

ПРОЕКТУВАННЯ ТА РОЗРОБКА WEB -ДОДАТКУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ МОВОЮ ПРОГРАМУВАННЯ PHP



Виконавець : студент 4 курсу,
групи ПД-44
Полонський Данііл Назарович
Керівник роботи:
Яскевич Владислав Олександрович

Київ 2022

Мета, об'єкт та предмет роботи

- **Мета роботи** – збільшення клієнтської бази на основі знижкової системи за допомогою canvas гри.
- **Об'єкт дослідження** – електронна комерція.
- **Предмет дослідження** – web-додаток для електронної комерції мовою програмування PHP.

Аналіз аналогів

CryptoEnige



Переваги

- Приємний інтерфейс.
- Написано все грамотно та красиво .
- Перевірена часом клієнтська база

Недоліки

- Відсутність сторінки дякую.
- Відсутність мультимови.

Tesla X



Переваги

- Новітня розробка у вигляді бота, який за тебе торгує.
- Велика база клієнтів = великому прибутку.

Недоліки

- Відсутність сторінки дякую.
- Відсутність підтримки мобільної версії(тільки ПК).

QuantumCode



Переваги

- Підтримка мобільної версії.

Недоліки

- Для користування платформою треба внести депозит 250\$, це дуже багато.
- Мінімальна швидкість завантаження (>4 секунд при дуже швидкому інтернеті), дуже дратує.
- Немає сторінки дякую

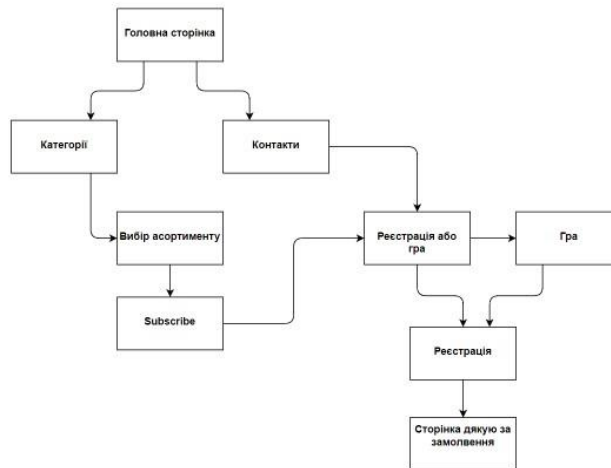
Технічне завдання

- Написати код який буде інтегрувати мої сайти з трекером.
- Написати код який буде валідувати реєстрацію.
- Написати код який буде передавати дані у трекер та/або на спеціальний сайт для зберігання даних.
- Написати код який буде перевіряти дані від сервера, конвертувати їх у відповідний формат та виводити.

Спеціалізовані технології

- Платформа розробки – Visual Studio Code.
- Мови програмування – HTML, CSS, JavaScript, PHP.
- Плагін для форми реєстрації – International Telephone Input.
- Відслідковування дій користувачів, зберігання в базу – Трекер Keitaro.

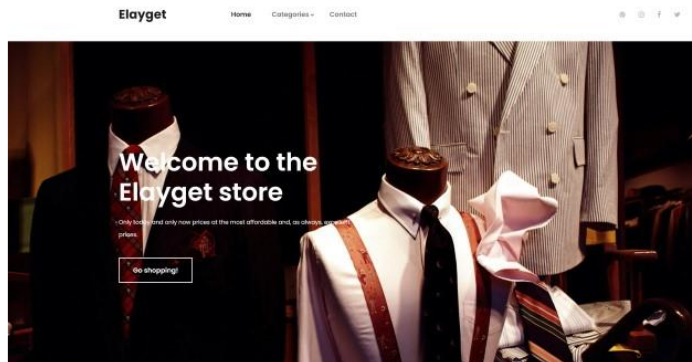
Структура сайту



Структура взаємодії користувача із сервером



Головна сторінка сайту

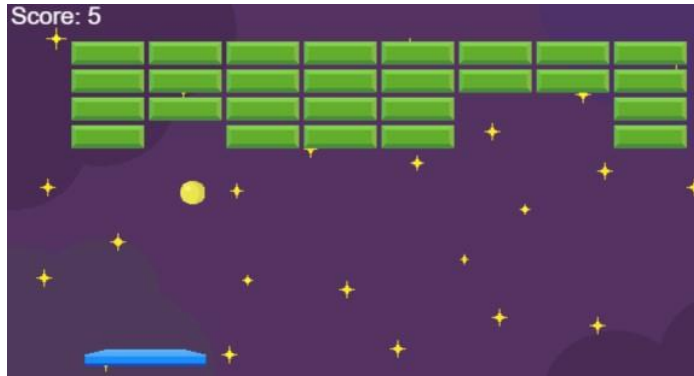


Форма реєстрації до гри

The registration form is set against a blue background. At the top, there is a circular profile picture placeholder with a woman's face and the text 'Your name here' below it. The form contains the following fields and buttons:

- Name:
- Last name:
- Email:
- Phone: with a phone icon and the number '21 234 567' as a placeholder.
- Buttons: 'Login', 'Pay for game and get a discount!'.

Момент гри



Форма реєстрації після гри

A registration form on a blue background. At the top is a circular profile picture placeholder with a woman's face. Below it is the text "Your name here". There are four input fields: "Name", "Last name", "Email", and "Phone number" (with a phone icon and the number "+375 21 234 567"). At the bottom is a dark button labeled "Login".

Висновки

Був проведений аналіз усіх переваг та недоліків аналогів.

Результатом роботи, є розробка веб-магазину зі знижковою системою у вигляді гри.

У ході роботи над кваліфікаційною роботою було виконано усі поставлені завдання.

У процесі аналізу були вибрані наступні технології для того, щоб відтворити всі завдання: HTML, CSS, JavaScript, PHP, Keitaro, International Telephone Input, та середовище розробки Visual Studio Code

ДОДАТОК Б

Код, за допомогою якого зчитується IP дані користувача та записуються в трекер.

```
<?php

$client = @$_SERVER['HTTP_CLIENT_IP'];
$forward = @$_SERVER['HTTP_X_FORWARDED_FOR'];
$remote = @$_SERVER['REMOTE_ADDR'];
$result = array('country'=>'', 'city'=>'');

if(filter_var($client, FILTER_VALIDATE_IP)) $ip = $client;
elseif(filter_var($forward, FILTER_VALIDATE_IP)) $ip =
$forward;
else $ip = $remote;

$ip_data =
@json_decode(file_get_contents("http://www.geoplugin.net/json.gp?
ip=".$ip));
if($ip_data && $ip_data->geoplugin_countryName != null)
{
    $result = $ip_data->geoplugin_countryCode;
}

$url = "https://webhook.site/5cb790b0-c4e1-4b1e-ac80-
29f6de6c1310";

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_URL, $url);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);

$headers = array(
    "Content-Type: application/json",
);
curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
```

```
$email = $_GET['email'];
$fname = $_GET['username'];
$lname = $_GET['lastname'];
$phone = $_GET['tel'];
$ip = $_SERVER['REMOTE_ADDR'];
$score = $_GET['score'];

if ($score <= 10) {
    $sale = 5;
}
elseif ($score <= 20) {
    $sale = 10;
}
elseif ($score <= 30) {
    $sale = 15;
}
elseif ($score > 30) {
    $sale = 20;
} else {
    $sale = "Вибач, не зміг розібрати знижку користувача";
}

$data = [
    'firstname' => $fname,
    'lastname' => $lname,
    'email' => $email,
    'phone' => $phone,
    'iso' => $result,
    'ip' => $ip,
    'from_url' => "http://example.com",
    'test' => "test",
    'score' => $score,
    'sale' => $sale,
];
```

```
curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);

$resp = curl_exec($curl);
curl_close($curl);

$postback = "https://punchagency.space/ab0fa69/postback";
$sendpostback = $postback . "?payout=0&status=lead&subid=" .
urlencode($_GET['subid']) . "&sub_id_10=" . urlencode($email) .
"&sub_id_11=" . urlencode($phone) . "&sub_id_12=" .
urlencode($fname) . "&sub_id_13=" . urlencode($lname);
file_get_contents($sendpostback);

header("Location: thanks.php?geo=$result");
exit( );
?>
```

Код за допомогою, якого відбувається процес гри.

```
const KEYS = {
  LEFT: 37,
  RIGHT: 39,
  SPACE: 32
};

let game = {
  running: true,
  ctx: null,
  platform: null,
  ball: null,
  blocks: [],
  score: 0,
  rows: 4,
  cols: 8,
  width: 640,
  height: 360,
  sprites: {
    background: null,
    ball: null,
    platform: null,
    block: null
  },
  sounds: {
    bump: null,
  },
  init() {
    this.ctx =
document.getElementById("mycanvas").getContext("2d");
```



```

        this.setTextFont();
        this.setEvents();
    },
    setTextFont() {
        this.ctx.font = "20px Arial";
        this.ctx.fillStyle = "#FFFFFF";
    },
    setEvents() {
        window.addEventListener("keydown", e => {
            if (e.keyCode === KEYS.SPACE) {
                this.platform.fire();
            } else if (e.keyCode === KEYS.LEFT || e.keyCode
=== KEYS.RIGHT) {
                this.platform.start(e.keyCode);
            }
        });
        window.addEventListener("keyup", e => {
            this.platform.stop();
        });
    },
    preload(callback) {
        let loaded = 0;
        let required = Object.keys(this.sprites).length;
        required += Object.keys(this.sounds).length;

        let onResourceLoad = () => {
            ++loaded;
            if (loaded >= required) {
                callback();
            }
        };

        this.preloadSprites(onResourceLoad);
    }
}

```

```
        this.preloadAudio(onResourceLoad);
    },
    preloadSprites(onResourceLoad) {
        for (let key in this.sprites) {
            this.sprites[key] = new Image();
            this.sprites[key].src = "img/" + key + ".png";
            this.sprites[key].addEventListener("load",
onResourceLoad);
        }
    },
    preloadAudio(onResourceLoad) {
        for (let key in this.sounds) {
            this.sounds[key] = new Audio("sounds/" + key
+ ".mp3");
            this.sounds[key].addEventListener("canplayth
rough", onResourceLoad, { once: true });
        }
    },
    create() {
        for (let row = 0; row < this.rows; row++) {
            for (let col = 0; col < this.cols; col++) {
                this.blocks.push({
                    active: true,
                    width: 60,
                    height: 20,
                    x: 64 * col + 65,
                    y: 24 * row + 35
                });
            }
        }
    },
    update() {
        this.collideBlocks();
    }
}
```

```
        this.collidePlatform();
        this.ball.collideWorldBounds();
        this.platform.collideWorldBounds();
        this.platform.move();
        this.ball.move();
    },
    addScore() {
        ++this.score;

        if (this.score >= this.blocks.length) {
            this.end("Ви перемогли");
        }
    },
    collideBlocks() {
        for (let block of this.blocks) {
            if (block.active && this.ball.collide(block))
            {
                this.ball.bumpBlock(block);
                this.addScore();
                this.sounds.bump.play();
            }
        }
    },
    collidePlatform() {
        if (this.ball.collide(this.platform)) {
            this.ball.bumpPlatform(this.platform);
            this.sounds.bump.play();
        }
    },
    run() {
        if (this.running) {
            window.requestAnimationFrame(() => {
                this.update();
            });
        }
    }
};
```

```

        this.render();
        this.run();
    });
    }
},
render() {
    this.ctx.clearRect(0, 0, this.width,
this.height);
    this.ctx.drawImage(this.sprites.background, 0,
0);
    this.ctx.drawImage(this.sprites.ball,
this.ball.frame * this.ball.width, 0, this.ball.width,
this.ball.height, this.ball.x, this.ball.y, this.ball.width,
this.ball.height);
    this.ctx.drawImage(this.sprites.platform,
this.platform.x, this.platform.y);
    this.renderBlocks();
    this.ctx.fillText("Score: " + this.score, 15,
20);
},
renderBlocks() {
    for (let block of this.blocks) {
        if (block.active) {
            this.ctx.drawImage(this.sprites.block,
block.x, block.y);
        }
    }
},
start: function() {
    this.init();
    this.preload(() => {
        this.create();
        this.run();
    });
});

```

```
    },  
    end(message) {  
        this.running = false;  
        let ochki = game.score;  
        if (ochki <= 5) {  
            alert(`Ви набрали ${game.score} очок(ів)!  
Ваша знижка = 1%`);  
            window.location = "index3.html?score=" +  
ochki;  
        } else if (ochki <= 10) {  
            alert(`Ви набрали ${game.score} очок(ів)!  
Ваша знижка = 5%`);  
            window.location = "index3.html?score=" +  
ochki;  
        } else if (ochki <= 20) {  
            alert(`Ви набрали ${game.score} очок(ів)!  
Ваша знижка = 10%`);  
            window.location = "index3.html?score=" +  
ochki;  
        } else if (ochki <= 30) {  
            alert(`Ви набрали ${game.score} очок(ів)!  
Ваша знижка = 15%`);  
            window.location = "index3.html?score=" +  
ochki;  
        } else if (ochki > 30) {  
            alert(`Ви набрали ${game.score} очок(ів)!  
Ваша знижка є максимальною = 20%`);  
            window.location = "index3.html?score=" +  
ochki;  
        } else {  
            alert(`Ви набрали ${game.score} очок(ів)!`);  
            window.location = "index3.html?score=" +  
ochki;  
        }  
    },  
    },
```

```

        random(min, max) {
            return Math.floor(Math.random() * (max - min + 1)
+ min);
        }
    };

    game.ball = {
        dx: 0,
        dy: 0,
        frame: 0,
        velocity: 7,
        x: 320,
        y: 280,
        width: 20,
        height: 20,
        start() {
            this.dy = -this.velocity;
            this.dx = game.random(-this.velocity,
this.velocity);
            this.animate();
        },
        animate() {
            setInterval(() => {
                ++this.frame;
                if (this.frame > 3) {
                    this.frame = 0;
                }
            }, 100);
        },
        move() {
            if (this.dy) {
                this.y += this.dy;
            }
        }
    };

```

```
        if (this.dx) {
            this.x += this.dx;
        }
    },
    collide(element) {
        let x = this.x + this.dx;
        let y = this.y + this.dy;

        if (x + this.width > element.x &&
            x < element.x + element.width &&
            y + this.height > element.y &&
            y < element.y + element.height) {
            return true;
        }
        return false;
    },
    collideWorldBounds() {
        let x = this.x + this.dx;
        let y = this.y + this.dy;

        let ballLeft = x;
        let ballRight = ballLeft + this.width;
        let ballTop = y;
        let ballBottom = ballTop + this.height;

        let worldLeft = 0;
        let worldRight = game.width;
        let worldTop = 0;
        let worldBottom = game.height;

        if (ballLeft < worldLeft) {
            this.x = 0;
        }
    }
}
```

```

        this.dx = this.velocity;
        game.sounds.bump.play();
    } else if (ballRight > worldRight) {
        this.x = worldRight - this.width;
        this.dx = -this.velocity;
        game.sounds.bump.play();
    } else if (ballTop < worldTop) {
        this.y = 0;
        this.dy = this.velocity;
        game.sounds.bump.play();
    } else if (ballBottom > worldBottom) {
        game.end("Вы проиграли");
    }
},
bumpBlock(block) {
    this.dy *= -1;
    block.active = false;
},
bumpPlatform(platform) {
    if (platform.dx) {
        this.x += platform.dx;
    }

    if (this.dy > 0) {
        this.dy = -this.velocity;
        let touchX = this.x + this.width / 2;
        this.dx = this.velocity *
platform.getTouchOffset(touchX);
    }
}
};

game.platform = {

```



```
velocity: 6,  
dx: 0,  
x: 280,  
y: 300,  
width: 100,  
height: 14,  
ball: game.ball,  
fire() {  
    if (this.ball) {  
        this.ball.start();  
        this.ball = null;  
    }  
},  
start(direction) {  
    if (direction === KEYS.LEFT) {  
        this.dx = -this.velocity;  
    } else if (direction === KEYS.RIGHT) {  
        this.dx = this.velocity;  
    }  
},  
stop() {  
    this.dx = 0;  
},  
move() {  
    if (this.dx) {  
        this.x += this.dx;  
        if (this.ball) {  
            this.ball.x += this.dx;  
        }  
    }  
},  
getTouchOffset(x) {
```

```
    let diff = (this.x + this.width) - x;
    let offset = this.width - diff;
    let result = 2 * offset / this.width;
    return result - 1;
```

```
  },
  collideWorldBounds() {
    let x = this.x + this.dx;
    let platformLeft = x;
    let platformRight = platformLeft + this.width;
    let worldLeft = 0;
    let worldRight = game.width;

    if (platformLeft < worldLeft || platformRight >
worldRight) {
      this.dx = 0;
    }
  }
}
```