

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-наукова програма «Управління проектами»

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
на тему:
“Управління проектом створення NFT маркетплейсу”**

Студента 2-го курсу групи УПз-11

Кім Юлії Юріївни

(підпис студента)

Науковий керівник

Доктор техн.наук
Олексій Єгорченков

(дата)

Попередній захист:

(Висновок: “До захисту в Екзаменаційній комісії”)
(дата)
(підпис)

Завідувач кафедри технологій управління

(підпис) (прізвище, ініціали)

Київ – 2022

ЗАТВЕРДЖУЮ

Завідувач кафедри професор Морозов В.В.

“ ____ ” _____ 20__ року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студентка: Кім Юлія Юріївна

Група: УПз-21

1. Тема кваліфікаційної роботи: “Управління проектом створення NFT маркетплейсу”. Затверджена наказом по від “ ____ ” _____ 20__ р. No ____.
2. Строк подання студентом готової роботи - “ ____ ” _____ 20__ р.
3. Цільова установка та вихідні дані до роботи: дослідження характеристики об’єкту управління планування виконання проекту(календарне планування, зміст, бюджет, ресурси) та планування іншими галузями управління проекту.
4. Зміст роботи: аналіз предметної галузі, проведення маркетингових досліджень, проведення інвестиційних досліджень, макроекономічний аналіз, опис мети, порівняльний аналіз конкурентів, фактори можливостей, фактори загроз, аналіз конкурентів, розроблення ринкової стратегії проекту, розробка життєвого циклу проекту, побудова організаційної структури, планування ресурсів проекту.
5. Перелік графічного матеріалу (слайдів): титульна сторінка, мета дипломної роботи, життєвий цикл проекту, дерево цілей, дерево проблем, економічне обґрунтування, маркетингове дослідження, організаційна структура, висновки.
6. Календарний план виконання роботи:

| № з/п | Назва частин роботи | План виконання роботи |
|-------|--|-----------------------|
| 1. | Вивчення літературних джерел з предмету дослідження | 01.06.2022-01.09.2022 |
| 2. | Збір і вивчення матеріалів досліджуваного підприємства | 01.09.2022-01.09.2022 |
| 3. | Складання розгорнутого плану кваліфікаційної роботи | 01.09.2022-07.09.2022 |
| 4. | Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін. | 07.09.2022 |
| 5. | Підготовка розділу 1 “ ” | 14.09.2022-28.09.2022 |
| 6. | Підготовка розділу 2 “ ” | 28.09.2022-12.10.2022 |
| 7. | Підготовка розділу 3 “ ” | 12.10.2022-26.10.2022 |
| 7. | Підготовка розділу 4 “ ” | 26.10.2022-09.10.2022 |
| 9. | Оформлення кваліфікаційної роботи | 09.11.2022-14.11.2022 |
| 10. | Передача кваліфікаційної роботи рецензенту для рецензування | 14.11.2022 |
| 11. | Передача кваліфікаційної роботи науковому керівникові | 20.11.2022 |
| 12. | Попередній захист кваліфікаційної роботи | 05.12.2022 |
| 13. | Захист роботи | 23.12.2022 |

Дата видачі завдання “ _____ ” _____ 20__ р.

Керівник роботи кандидат техн. Наук, Олексій Єгорченков

(підпис)

Завдання прийняв до виконання студент групи:

студентка групи УПз-21, Кім Юлія Юріївна

(підпис)

ЗМІСТ

| | |
|---|----|
| АНОТАЦІЯ..... | 6 |
| ВСТУП..... | 5 |
| РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК ПРОЕКТУ..... | 8 |
| 1.1. План розробки проекту та масштабування його на ринок..... | 8 |
| 1.2. Опис ідеї проекту..... | 9 |
| 1.3. Макроекономічний аналіз..... | 10 |
| 1.4. Технологічний аудит ідеї проекту..... | 18 |
| 1.5. Аналіз ринкових можливостей запуску проекту..... | 20 |
| 1.6. Розроблення ринкової стратегії проекту..... | 27 |
| 1.7. Розроблення маркетингової програми проекту..... | 30 |
| 1.8. Проведення інвестиційних досліджень..... | 32 |
| 1.9. Висновки..... | 35 |
| РОЗДІЛ 2. ПЛАНУВАННЯ ВИКОНАННЯ ПРОЕКТУ..... | 37 |
| 2.1. Життєвий цикл проекту..... | 37 |
| 2.2. Розробка організаційної структури управління проектом. Формування команди проекту..... | 38 |
| 2.3. Календарне планування проекту..... | 40 |
| 2.4. Планування ресурсів проекту..... | 45 |
| РОЗДІЛ 3. ОГЛЯД ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ..... | 56 |
| 3.1 Огляд мови програмування..... | 56 |
| 3.2 Огляд середовища розробки..... | 63 |
| 3.3 Огляд додаткового інструментарію..... | 64 |
| РОЗДІЛ 4. ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ... | 78 |
| 4.1 Аналіз варіантів діяльності..... | 78 |
| 4.2 Проектування компонентного складу..... | 80 |
| 4.3 Проектування внутрішньої будови..... | 81 |
| 4.4 Розробка графічного інтерфейсу..... | 86 |
| 4.5 Тестування..... | 89 |
| ВИСНОВКИ..... | 93 |

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....94

АНОТАЦІЯ

кваліфікаційної магістерської роботи на тему:

«Управління проектом створення NFT маркетплейсу»

Студентки: Кім Юлії Юрїївни

Науковий керівник: Олексій Єгорченков

Рік захисту - 2022.

Темою даної роботи було обрано «Управління проектом створення NFT маркетплейсу», *предметною областю* є маркетплейси, NFT проекти, криптовалюти та блокчейн-технології.

Мета даної роботи - це розробка та планування маркетплейсу з продажу NFT мистецтва, для отримання прибутку та розвитку мистецтва, а також популяризації криптовалютних платежів.

Об'єктом дослідження є проект розробки NFT-маркетплейсу.

Предметом дослідження є процеси ініціалізації проекту, в тому числі проведення маркетингових та інвестиційних досліджень з використанням інформаційних моделей розрахунку економічних показників доцільності інвестицій в проект.

Новизна отриманих результатів полягає у розробці коду, який буде забороняти копіювати та поширювати NFT роботи, що стане вирішенням проблеми незаконного поширення арт-робіт та унікальним рішенням у світі NFT маркетплейсів.

Робота містить 106 сторінок.

ВСТУП

В сучасному світі тенденція NFT стає все більш популярною з кожним днем і помалу витісняє з ринку таких гігантів, як біткоїн.

Кожного дня з'являється все більша кількість NFT-маркетплейсів, які пропонують купити або продати унікальні NFT.

Виходячи з усього вищесказаного, можна зробити висновок про високу актуальність NFT-маркетплейсів в сучасному світі.

Об'єктом дослідження є проєкт розробки NFT-маркетплейсу.

Предметом дослідження є процеси ініціалізації проєкту, в тому числі проведення маркетингових та інвестиційних досліджень з використанням інформаційних моделей розрахунку економічних показників доцільності інвестицій в проєкт.

Мета роботи полягає в аналізі доцільності створення власного NFT-маркетплейсу.

Для досягнення поставленої мети слід виконати наступні завдання:

1. провести аналіз поняття NFT;
2. провести аналіз поняття маркетплейсу;
3. проаналізувати поняття веб-сайтів та веб-розробки;
4. розглянути існуючі рішення;
5. провести огляд мови програмування;
6. провести огляд середовища розробки;
7. провести огляд додаткового інструментарію;
8. провести аналіз варіантів діяльності;
9. провести проектування компонентного складу системи;
10. провести проектування внутрішньої будови системи;
11. розробити графічний інтерфейс користувача.

РОЗДІЛ 1

АНАЛІЗ ТА ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК ОБ'ЄКТУ

1.1 План розробки проекту та масштабування його на ринок

В першу чергу необхідним є проведення маркетингового аналізу, який включає в себе:

- конкурентний аналіз, для отримання вичерпної інформації про існуючі методи вирішення проблеми;
- формування ідеї проекту та виділення цільової аудиторії;
- розробка стратегії виведення товару на ринок, базуючись на аналізі ринкового середовища.

Наступним кроком являється організація стартапу. На цьому етапі мають бути:

- складений план та побудований таймлайн розробки та запуску продукту;
- запланований обсяг виробництва та оцінений потенційний обсяг ресурсу, необхідний для виконання плану;
- розраховані витрати, необхідні для реалізації проекту, та витрати на запуск проекту.

Далі необхідно виконати фінансово-економічний аналіз та оцінити ризики стартап-проекту, в межах якого:

- визначити обсяг інвестиційних втрат;
- розрахувати основні фінансово-економічні показники проекту (собівартість, ціну продукту/послуги, податковий збір та чистий прибуток) та визначити показники інвестиційної привабливості проекту (рентабельність продажів, період окупності проекту);
- визначити основні ризики проекту та способи для їх запобігання.

Фінальним кроком є розробка заходів з комерціалізації продукту. Цей крок являється важливим для масштабування та збільшення розмірів продукту.

Для того, щоб залучити інвесторів та знайти різні способи фінансування проекту, необхідно:

- провести дослідження на предмет інтересів потенційних інвесторів та бізнесів;
- скласти інвестиційну пропозицію, яка включає в себе як опис самого продукту та його теперішні розміри, так і можливі шляхи розширення та розвитку;
- обрати канали комунікації із потенційно зацікавленими персонами.

Далі буде приведено результати виконання кожного з описаних кроків.

1.2 Опис ідеї проекту

Стартап-проект полягає у вирішенні проблеми пошуку підходящих пропозицій на ринку NFT. Суть продукту полягає у тому, що система являє собою NFT-маркетплейс, здатний виконувати продаж токенів.

У таблиці 1.1 наведена інформаційна карта стартапу.

Таблиця 1.1

Інформаційна карта стартап-проекту

| | |
|---------------------------|--|
| Назва проекту | Управління проектом створення NFT маркетплейсу |
| Коротка анотація | Система NFT-маркетплейсу. |
| Термін реалізації проекту | 12 місяців |

| | |
|-----------------------------------|---|
| Необхідні ресурси | Приміщення з комп'ютерами, доступом до Інтернету, доступ до електромережі Програмне забезпечення для розробки, хмарне сховище для даних, антивірусні програми Фінансові кошти на оплату заробітної плати виконавцям на термін 12 місяців, а також на такі витрати як: оренда приміщення, комунальні послуги, оренда хмарного сховища тощо |
| Опис проблеми, яку вирішує проект | Проект вирішує проблему купівлі-продажу незамінних токенів. |
| Головні цілі та завдання проекту | Метою проекту є створення системи, яка буде створювати можливість купівлі-продажу незамінних токенів. |
| Очікувані результати | NFT-макетплейс |

1.3. Макроекономічний аналіз

Зацікавлені сторони проекту та впливи:

Таблиця 1.3

| № | Зацікавлені сторони | Потреба | Вигода |
|--------------|---------------------|---------|--------|
| Внутрішні ЗС | | | |

| | | | |
|---|-----------------------------------|---|--|
| 1 | Замовник проекту | Створення продукту для задоволення проекту споживача. | Отримання прибутку в 7 млн.грн за 3 роки. |
| 2 | Керівник проекту, команда проекту | Розширення клієнтської бази для збільшення прибутку. | Отримання постійних клієнтів в результаті успішної реалізації проекту та рекомендацій замовника. |
| 3 | Фізичні особи | Забезпечує наповненість проекту. | Задоволення потреб клієнтів у швидкому розміщенні оголошень, реалізації товарів. |
| 4 | Інвестори проекту | Отримання прибутку. | Продаж долі акцій компанії, отримання прибутку. |

Характер та ступінь впливу політичних факторів на проект

Таблиця 1.4

| Фактори впливу політичного середовища | Характер впливу фактору | Оцінка експертів ступеня впливу фактору (max = 3 б.) | | | Середній бал |
|---------------------------------------|-------------------------|--|-----------|-----------|--------------|
| | | Експерт 1 | Експерт 2 | Експерт 3 | |

| | | | | | |
|--|---|---|---|---|--------|
| Кількісні та якісні обмеження на імпорт | — | 3 | 3 | 2 | — 3.0 |
| Бюрократизація і рівень корупції | — | 2 | 2 | 3 | — 2.33 |
| Стійкість політичної влади та існуючого уряду | — | 1 | 2 | 2 | — 1.67 |
| Податкова політика держави | — | 3 | 3 | 3 | — 3.0 |
| Вірогідність розвитку військових дій в країні | — | 3 | 3 | 2 | — 2.67 |
| Тенденції до регулювання або дерегулювання галузі | + | 1 | 3 | 2 | + 2.0 |
| Антимонопольне та трудове законодавство | + | 1 | 2 | 1 | + 1.33 |
| Майбутнє і поточне законодавство, що регулює правила роботи в галузі | + | 3 | 3 | 2 | + 2.67 |

| | | | | | |
|--|---|---|---|---|--------|
| Підтримка інноваційних компаній з боку держави | + | 3 | 3 | 3 | + 3.0 |
| Лібералізація зовнішнього економічного співробітництва | + | 2 | 2 | 1 | + 1.67 |
| Рівень інфляції | — | 2 | 3 | 3 | — 2.67 |
| Курси основних валют | — | 3 | 2 | 2 | — 2.33 |
| Рівень наявних доходів населення | — | 3 | 3 | 3 | — 3.0 |
| Ступінь глобалізації та відкритості економіки | + | 1 | 1 | 1 | + 1.0 |
| Монетарна та фіскальна політика держави | — | 2 | 2 | 1 | -1,67 |
| Рівень розвитку підприємництва та бізнес середовища | + | 3 | 3 | 3 | + 3.0 |

| | | | | | |
|---|---|---|---|---|--------|
| Цінова конкуренція з боку зарубіжних компаній | + | 1 | 2 | 2 | + 1.67 |
| Інвестиційний клімат в галузі | + | 1 | 2 | 1 | + 1.33 |

Характер та ступінь впливу соціально-культурних факторів на проект

Таблиця 1.4

| Фактори впливу соціально-культурних факторів | Характер впливу фактору | Оцінка експертів ступеня впливу фактору (max = 3 б.) | | | Середній бал |
|---|-------------------------|--|-----------|-----------|--------------|
| | | Експерт 1 | Експерт 2 | Експерт 3 | |
| Рівень підготовки молодих спеціалістів в галузі | + | 2 | 2 | 3 | + 2.33 |
| Вимоги до якості продукції та рівня сервісу | + | 2 | 3 | 1 | + 2.0 |
| Рівень міграції та імміграційні настрої | — | 3 | 3 | 3 | — 3 |

| | | | | | |
|---|---|---|---|---|--------|
| Культура формування заощаджень і кредитування суспільства | + | 3 | 2 | 2 | + 2.33 |
| Розвиток релігії та інших вірувань | + | 1 | 1 | 1 | + 1.0 |
| Ставлення до імпортованих товарів і послуг | + | 2 | 2 | 3 | + 2.33 |
| Спосіб життя і звички споживання | + | 2 | 2 | 2 | + 2.0 |
| Темпи росту населення | + | 1 | 2 | 3 | + 2.0 |

Характер та ступінь впливу технологічних факторів на проект

Таблиця 1.5

| Фактори впливу технологічних факторів | Характер впливу фактору | Оцінка експертів ступеня впливу фактору (max = 3 б.) | | | Середній бал |
|---------------------------------------|-------------------------|--|-----------|-----------|--------------|
| | | Експерт 1 | Експерт 2 | Експерт 3 | |

| | | | | | |
|--|---|---|---|---|--------|
| Рівень інновацій та технологічного розвитку галузі | + | 2 | 2 | 3 | + 2.33 |
| Ступінь використання, впровадження та передачі технологій | + | 2 | 2 | 2 | + 2.0 |
| Доступ до новітніх технологій | + | 2 | 2 | 3 | + 2.33 |
| Витрати на дослідження та розробки | + | 1 | 2 | 1 | + 1.33 |
| Виробництво якісної нової продукції (розвиток конкурентних технологій) | + | 2 | 1 | 1 | + 1.33 |
| Законодавство в галузі технологічного оснащення галузі | + | 3 | 2 | 2 | + 2.33 |

Фактори, які спричиняють найбільший вплив на діяльність компанії за результатами оцінювання PEST-аналізу

Таблиця 1.6

| Політичні | | Економічні | |
|---|--------|--|--------|
| Фактор | Вага | Фактор | Вага |
| Податкова політика держави | — 3.0 | Рівень наявних доходів населення | - 3.0 |
| Підтримка інноваційних компаній з боку держави | + 3.0 | Рівень розвитку підприємництва та бізнес середовища | + 3.0 |
| Соціально-культурні | | Технологічні | |
| Фактор | Вага | Фактор | Вага |
| Рівень підготовки молодих спеціалістів в галузі | + 2.33 | Законодавство в галузі технологічного оснащення галузі | + 2.33 |
| Рівень міграції та імміграційні настрої | - 3.0 | Витрати на дослідження та розробки | + 1.33 |

Підсумки здійсненого PEST-аналізу для компанії

Таблиця 1.7

| Фактори | Зміни в галузі | Зміни в організації | Дії |
|-----------|------------------------------------|---------------------|---|
| Політичні | Поява трудових законотворчих змін. | Рух капіталу. | Можливість розвитку e-commerce бізнесу. |

| Фактори | Зміни в галузі | Зміни в організації | Дії |
|---------------------|--|---|--|
| Економічні | Збільшення податкового навантаження. | Збільшення внутрішнього ринку, розвиток капіталу. | Оплата податків, розвиток внутрішнього та зовнішнього податкових ринків. |
| Соціально-культурні | Орієнтація населення на онлайн-шопінг, зміни в стандартах та тенденціях зміни якості продуктів/послуг. | Надання можливостей розвитку бізнесу маленьким та середнім підприємцям на ринку. Надання нових робочих місць. | Залучення нових споживачів. Розвиток онлайн-платформи. |
| Технологічні | Розповсюдження маркет-плейсів з купівлі/продажу товарів. | Розвиток e-commerce ринку. Розвиток бізнесу доставки. | Розвиток персоналу, підвищення рівню кваліфікації внутрішнього ринку. |

1.3 Технологічний аудит ідеї проекту

Даний підрозділ присвячено розбору ідеї та конкурентному аналізу.

У таблиці 1.2 наведений опис ідеї стартапу.

Таблиця 1.3

Опис ідеї стартапу

| Зміст ідеї | Напрямки застосування | Вигоди для користувача |
|------------|-----------------------|------------------------|
| | | |

| | | |
|---|---|---|
| <p>Основна ідея полягає в створенні системи, яка при старті буде отримувати користувацькі фільтри і на їх основі виконувати пошук підходящих пропозицій</p> | <p>Купівля-продаж незамінних токенів.</p> | <p>Користувач може виставити фільтри для пошуку</p> |
| | | <p>Користувач може завантажити власний токен</p> |
| | | <p>Користувач може викупити токен зі списку</p> |

Далі проведемо порівняльний аналіз конкурентів проекту та наведемо результати у таблиці 1.4.

Таблиця 1.4

Порівняльний аналіз конкурентів проекту

| № п/п | Техніко-економічні характеристики ідеї | (потенційні) товари/концепції конкурентів | | W | N | S |
|-------|--|---|----------------|---|---|---|
| | | Власний проект | Квартирант | | | |
| 1 | Швидкість пошуку | Регулюється | Не регулюється | | | + |
| 2 | Зміна фільтрів | Присутня | Присутня | | + | |
| 3 | Кількість фільтрів | Велика | Помірна | | + | |

Далі аналізуємо реальність технічно здійснити ідею проекту (таблиця 1.5).

Таблиця 1.5 – Технологічна здійсненність продукту

| № п/п | Ідея проекту | Технології і реалізації | Наявність технологій | Доступність технологій |
|---|---|---------------------------------------|-----------------------------------|------------------------|
| 1 | Створення комплексної системи NFT-маркетплейсу. | Використання мови програмування Java | Наявні | Доступні |
| 2 | | Використання мови програмування Julia | Не наявні, необхідні допрацювання | Доступні |
| 3 | | Використання мови програмування Rust | Наявні, необхідні допрацювання | Доступні |
| Обрана технологія реалізації ідеї проекту: Java | | | | |

1.4 Аналіз ринкових можливостей запуску проекту

Далі проведемо попередній аналіз ринку для запуску проекту (таблиця 1.6).

Таблиця 1.6

Попередня характеристика потенційного ринку стартап-проекту

| № п/п | Показники ринку (найменування) | Характеристика |
|-------|-----------------------------------|--------------------|
| 1 | Кількість головних гравців, од | 1 |
| 2 | Загальний обсяг продаж, грн/ум.од | 6000 |
| 3 | Динаміка ринку (якісна оцінка) | Позитивна, зростає |

| | | |
|---|--|----------|
| 4 | Наявність обмежень для входу (вказати характер обмежень) | Відсутні |
| 5 | Специфічні вимоги до стандартизації та сертифікації | Відсутні |
| 6 | Середня норма рентабельності в галузі (або по ринку), % | 12% |

Тепер проведемо характеристику потенційних клієнтів, які можуть бути зацікавлені в проекті (таблиця 1.7).

Таблиця 1.7

Характеристика потенційних клієнтів стартап-проекту

| № п/п | Потреби, що формує ринок | Цільова аудиторія (цільові сегменти ринку) | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару |
|-------|------------------------------------|--|---|-----------------------------|
| 1 | Купівля-продаж незамінних токенів. | Особисте використання користувачем | Цікавить стабільність роботи | Простота використання |

Обрахуємо фактори загроз (таблиця 1.8) та можливостей (таблиця 1.9). Проаналізуємо загрози, щоб зрозуміти можливі перешкоди при запуску продукт на ринок. Фактори можливостей же треба обрахувати, щоб знати усі сприятливі умови та по можливості ними скористатися.

Таблиця 1.8

Фактори загроз

| № п/п | Фактор | Зміст загрози | Можлива реакція компанії |
|-------|---------------|---|--|
| 1 | Конкуренція | Хоча ринок є відкритим і неосвоїним, на ньому вже є один великий гравець, який відомий на ринку, який вже має свою цільову групу покупців | Знайти точки додаткової цінності для користувача |
| 2 | Якість пошуку | Через зміни чи оновлення ресурсів, на базі яких ведеться пошук, якість пошуку може бути суттєво знижена. | Проводити періодичні оновлення програмного продукту. |

Таблиця 1.9

Фактори можливостей

| № п/п | Фактор | Зміст можливості | Можлива реакція компанії |
|-------|-------------------------|--|---|
| 1 | Універсальність | Продукт не залежить від апаратної платформи як у більшості конкурентів | Зробити акцент при маркетингу, продовжувати розвиток як окремого продукту |
| 2 | Простота у використанні | Від користувача треба лише задати критерії фільтрації | Реалізувати зручний інтерфейс |
| 3 | Якість та гарантії | Надавати найбільш якісні послуги та сервіси | Пропонувати моделі з найкращими результатами, а також надавати усю необхідну технічну підтримку |

| | | | |
|---|-----------------------------|---|--|
| 4 | Безкоштовний сервіс при MVP | Максимально швидко набрати базу своїх клієнтів та заявити про себе на ринку | Розгорнути широкий маркетинг, а також активно боротися за клієнтів конкурентів |
|---|-----------------------------|---|--|

Далі розглянемо питання конкуренції, а саме визначимо її тип та рівень (таблиця 1.10).

Таблиця 1.10

Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища | У чому проявляється дана характеристика | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|--|---|---|
| Вказати тип конкуренції: недосконала конкуренція | Представлено мало продуктів та експертів | Зробити максимальним збут застосунку |
| За рівнем конкурентної боротьби: міжнародний | Наявні проекти, розроблені та можуть бути доступні у всьому світі | Розширити цільову аудиторію, розробити інтерфейс на різних мовах |
| За галузевою ознакою: внутрішньогалузева | Можуть працювати з різними галузями | Покращити персоналізацію |
| Конкуренція за видами товарів: товарно-родова | Конкуренція з аналізами інших систем та експертів | Підтримувати та покращувати якість існуючих функцій |
| За характером конкурентних переваг: нецінова | Різні компанії пропонують різну якість | Розробляти якісніші алгоритми і моделі |

| | | |
|------------------------------|--|---|
| За інтенсивністю: марочна | Вже представлені компанії із сильним брендом | Предметно створити комунікаційну стратегію для вибудови свого бренду |
|------------------------------|--|---|

Далі необхідно виконаємо аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера (таблиця 1.11).

Таблиця 1.11

Аналіз конкуренції в галузі за М. Портером

| Складові аналізу | Прямі конкуренти у галузі | Потенційні конкуренти | Постачаль ники | Клієнти | Товарозамі нники |
|---------------------|---|--|---------------------------------------|---|--|
| | Інші існуючі системи та продукти | Якість, ціни, кількість користувачів , капіталовкла дення | Фактори сили постачаль ників | Контроль якості, порівнянн я цін | Сила бренду, якість, ціна, масштаби |
| Висновки | Конкуренці я з невеликою інтенсивніс тю, а також підігрітий ринок | Можливості входження на ринок, нові потенційні конкуренти | Постачаль ники відсутні | Клієнти не диктують умови роботи на ринку | Товарозамі нники відсутні |

Маючи результати аналізу конкуренції (таблиця 1.12), характеристики ідеї стартап-проекту (таблиця 1.6), характеристики потенційних клієнтів і їх

вимоги до продукту (таблиця 1.7) та фактори ринкового середовища (таблиці 1.7 і 1.8) було сформульовано та обґрунтовано перелік факторів конкурентоспроможності (таблиця 1.12).

Таблиця 1.12

Обґрунтування факторів конкурентоспроможності

| № п/п | Фактор конкурентоспроможності | Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим) |
|-------|-------------------------------|---|
| 1 | Універсальність | Продукт не залежить від апаратної платформи як у більшості конкурентів |
| 2 | Простота у використанні | Від користувача треба лише задати пошукові фільтри |
| 3 | Якість та гарантії | Надавати найбільш якісні послуги та сервіси |
| 4 | Безкоштовний сервіс при MVP | Максимально швидко набрати базу своїх клієнтів та заявити про себе на ринку |

Тепер можна провести аналіз сильних та слабких сторін продукту (таблиця 1.13).

Таблиця 1.13

Порівняльний аналіз сильних та слабких сторін системи

| № п/п | Ф а к т о р конкурентоспроможності | Бали 1-20 | Рейтинг товарів-конкурентів | | | | | | |
|-------|------------------------------------|-----------|-----------------------------|----|----|---|---|---|---|
| | | | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
| 1 | Універсальність | 20 | + | | | | | | |
| 2 | Простота у використанні | 16 | | + | | | | | |

| | | | | | | | | | |
|---|-----------------------------|----|--|---|---|--|--|--|--|
| 3 | Якість та гарантії | 10 | | + | | | | | |
| 4 | Безкоштовний сервіс при MVP | 17 | | | + | | | | |

Далі проведемо SWOT-аналіз продукту (таблиця 1.14).

Таблиця 1.14

SWOT-аналіз стартап-проекту

| | |
|--|--|
| <p>Сильні сторони</p> <p>Універсальність</p> <p>Простота у використанні</p> <p>Якість та гарантії</p> <p>Безкоштовний сервіс при MVP</p> | <p>Слабкі сторони</p> <p>Відсутність сильного бренду</p> <p>Не сформована база клієнтів</p> <p>Не підключені альтернативні канали маркетингу</p> |
| <p>Можливості</p> <p>Покращення системи</p> <p>Персоналізація</p> <p>Збільшення кількості пошукових ресурсів</p> | <p>Загрози</p> <p>Нові системи та експерти</p> <p>Збут</p> |

Дякуючи проведенню SWOT-аналізу, було визначено сильні та слабкі сторони, можливості та загрози, пов'язані з конкуренцією та плануванням стартап-проекту. Далі слід спроектувати альтернативну ринкову поведінку для інтеграції стартап-проекту на ринок та приблизний час реалізації системного комплексу, з урахуванням потенційних проектів, що можуть бути виведені на ринок та наведемо результати у таблиці 1.15.

Таблиця 1.15 – Альтернативи ринкового впровадження стартап проекту

| № п/п | Альтернатива (орієнтовний комплекс заходів) ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|-------|--|--------------------------------|-------------------|
| 1 | Вихід на ринок з нижче якістю | 70% | 4 місяці |
| 2 | Пропонувати одразу платне використання | 50% | 6 місяців |
| 3 | Представлення користувачам системи без інтерфейсу | 60% | 5 місяці |

У даному пункті був проведений детальний аналіз ринку та продукту. Також відповідно до результатів проведеного конкурентного аналізу, визначених факторів ринку та його сприятливість, описання ідеї та характеристик стартап-проекту, робимо можна зробити висновок, що існують дуже сприятливі умови для виходу продукту на ринок.

1.5 Розроблення ринкової стратегії проекту

Для розробки ринкової стратегії продукту, у першу чергу, необхідно проаналізувати цільову аудиторію проекту (таблиця 1.16).

Таблиця 1.16

Вибір цільових груп потенційних споживачів

| № п/п | Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит у межах цільової групи (сегменту) | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|-------|--|---|---|--------------------------------------|--------------------------|
| 1 | Персональні користувачі | Висока | 25% | Висока | Середня |
| 2 | Великі бізнеси | Середня | 2% | Середня | Середня |

| | | | | | |
|-----------------------------|----------------------------|---------|----|---------|---------|
| 3 | Малі та середні бізнеси | Середня | 5% | Середня | Середня |
| Які цільові групи обрано: 1 | | | | | |

Маючи аналіз цільових груп, далі визначимо базову стратегію розвитку продукту (таблиця 1.17).

Таблиця 1.17

Визначення базової стратегії розвитку

| № п/п | Обрана альтернатива розвитку проекту | Стратегія охоплення ринку | Ключові конкурентоспромо жні позиції відповідно до обраної альтернативи | Базова стратегія розвитку |
|----------|---|---------------------------------|--|---------------------------------|
| 1 | 1 | Диференційованого маркетингу | Масштабування та максимізація | Оптимальних витрат |

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 1.18, 1.19).

Таблиця 1.18

Визначення базової стратегії конкурентної поведінки

| | | | |
|---|---|--|--|
| Чи є проект «першопрохідцем » на ринку? | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів? | Чи буде компанія копіювати основні характеристики товару конкурента, і які? | Стратегія конкурентної поведінки |
|---|---|--|--|

| | | | |
|----|-----|----|----------------|
| Ні | Так | Ні | Виклику лідера |
|----|-----|----|----------------|

Таблиця 1.19

Визначення стратегії позиціонування

| | | | |
|--|---------------------------|--|--|
| Вимоги до товару цільової аудиторії | Базова стратегія розвитку | Ключові конкурентоспроможні позиції власного стартап-проекту | Вибір асоціацій, які мають сформувану комплексну позицію власного проекту (три ключових) |
| Універсальність Простота у використанні Якість результатів | Оптимальних витрат | Універсальність Простота у використанні Якість та гарантії Безкоштовне використання при MVP | Система, яка краще всіх гарантує купівлю-продаж незамінних токенів. Система з простим інтерфейсом |

1.6 Розроблення маркетингової програми проекту

Після проведеного комплексного аналізу, можемо повноцінно описати ключові переваги концепції потенційного товару (таблиця 1.20) та побудувати концепцію маркетингових комунікацій (таблиця 1.21).

Таблиця 1.20 – Ключові переваги концепції потенційного товару

| № п/п | Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити) |
|----------|---|--|---|
| 1 | Пошук по великій кількості критеріїв | Більш підходящі результати | Постійне покращення алгоритмів пошуку, які можуть охоплювати більше фільтрів |
| 2 | Універсальність | Система не залежить від апаратної платформи цифрової фотографії | Такого виду систему може використовувати будь який користувач |
| 3 | Простий інтерфейс | Система дуже проста у використанні | Система із інтуїтивно зрозумілим інтерфейсом, який вимагає всього лише вибору фільтрів |

Концепція маркетингових комунікацій

| № п/п | Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення | Концепція рекламного звернення |
|-------|---------------------------------------|--|--|--|---|
| 1 | Пошук спеціалізованих систем | b2b продажі Зв'язок через теплі контакти Таргетована реклама у соціальних мережах Публікація в спеціалізованих виданнях, журналах | Точність Якість Універсальність | Поєднати повідомлення про те, що це якісна система, яка є незалежною | Таргетована реклама на цільову аудиторію |
| 2 | Пошук доступного та дешевого продукту | Рекламні банери в Інтернеті, форуми, реклами від інфлюенсерів | Простота Безкоштовне використання MVP | Вселити довіру у бренд та продукт | Реклама у лідерів думок Вивіски в публічних місцях Таргетована реклама на цільову аудиторію |

1.7. Проведення інвестиційних досліджень

Інвестиційні дослідження проводились за допомогою побудови економічної моделі на 8 місяців.

В наведеній нижче таблиці видно, що виручка від розміщення NFT проектів на маркетплейсі почне приносити дохід з 2 місяця.

| ВИРУЧКА ВІД РЕАЛІЗАЦІЇ | ПДВ | Експ.мито. | 01.03.2022 | 1 міс | 2 міс | 3 міс | 4 міс | 5 міс | 6 міс | 7 міс | 8 міс | ВСЬОГО |
|---|-----|------------|------------|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| Местная валюта тис.грн. | | | | | | | | | | | | |
| Створення додатку | 20% | 0% | | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 120 |
| Створення веб-сайту | 20% | 0% | | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 160 |
| Реклама | 20% | 0% | | 0 | 0 | 60 | 120 | 120 | 180 | 240 | 240 | 960 |
| Розміщення на маркет плейсі | 20% | 0% | | 0 | 0 | 2 500 | 2 500 | 5 000 | 5 000 | 10 000 | 10 000 | 35 000 |
| Иностранная валюта тис.долл. | | | | | | | | | | | | |
| Назва продукту проекта | 0% | 15% | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| = Ітого виручка (без ПДВ та акцизів) | | тис.грн. | | 35 | 35 | 2 595 | 2 655 | 5 155 | 5 215 | 10 275 | 10 275 | 36 240 |
| - местная валюта | | тис.грн. | | 35 | 35 | 2 595 | 2 655 | 5 155 | 5 215 | 10 275 | 10 275 | 36 240 |
| - иностранная валюта | | тис.долл. | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| = Ітого акцизи | | тис.грн. | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - местная валюта | | тис.грн. | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - иностранная валюта | | тис.долл. | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| = ПДВ до виручки | | тис.грн. | | 7 | 7 | 519 | 531 | 1 031 | 1 043 | 2 055 | 2 055 | 7 248 |
| = Экспортное мито | | тис.грн. | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - местная валюта | | тис.грн. | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - иностранная валюта | | тис.долл. | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис.1.7.1 Виручка від реалізації

Рисунок 1.7.2 демонструє заробітну плату персоналу проекту, а саме таких професій, як розробники, тестувальник, проектний менеджер, маркетолог.

| ДЖЕРЕЛА ФІНАНСУВАННЯ ПРОЄКТУ | | | 01.03.2022 | 1 міс | 2 міс | 3 міс | 4 міс | 5 міс | 6 міс | 7 міс | 8 міс | ВСЬОГО |
|--|-----------|---------|------------|---------|---------|---------|--------|--------|--------|--------|-------|---------|
| Потреби у фінансуванні постійних активів | тис.грн. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - місцева валюта | тис.грн. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - іноземна валюта | тис.долл. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Потреби у фінансуванні чистого обігового капіталу | тис.грн. | 2 404 | -570 | 794 | 252 | 270 | -234 | -242 | -1 273 | -1 727 | | -326 |
| - місцева валюта | тис.грн. | 2 404 | -570 | 794 | 252 | 270 | -234 | -242 | -1 273 | -1 727 | | -326 |
| - іноземна валюта | тис.долл. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| = Потреби у фінансуванні інвестиційних витрат | тис.грн. | 2 404 | -570 | 794 | 252 | 270 | -234 | -242 | -1 273 | -1 727 | | -326 |
| - місцева валюта | тис.грн. | 2 404 | -570 | 794 | 252 | 270 | -234 | -242 | -1 273 | -1 727 | | -326 |
| - іноземна валюта | тис.долл. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| 1. СТАТУТНИЙ КАПІТАЛ | | | | | | | | | | | | |
| Засновницький капітал (зміни) | тис.грн. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - вноси у місцевій валюті | тис.грн. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - вноси в іноземній валюті | тис.долл. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Акціонерний капітал (зміни) | тис.грн. | 50 000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 000 |
| - прості акції | тис.грн. | 50 000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 000 |
| - привілейовані акції | тис.грн. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2. ЦІЛЬОВЕ ФІНАНСУВАННЯ ТА НАДХОДЖЕННЯ | | | | | | | | | | | | |
| Обсяг фінансування (зміни) | тис.грн. | 30 000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 000 |
| - у місцевій валюті | тис.грн. | 30 000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 000 |
| - у іноземній валюті | тис.долл. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3. ПОЗИКОВИЙ КАПІТАЛ | | | | | | | | | | | | |
| Залучення кредитів | тис.грн. | 100 000 | 0 | 0 | 10 000 | 0 | 0 | 0 | 0 | 0 | 0 | 110 000 |
| - у місцевій валюті | тис.грн. | 100 000 | 0 | 0 | 10 000 | 0 | 0 | 0 | 0 | 0 | 0 | 110 000 |
| - у іноземній валюті | тис.долл. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| = Разом джерела фінансування | тис.грн. | 180 000 | 0 | 0 | 10 000 | 0 | 0 | 0 | 0 | 0 | 0 | 190 000 |
| - у місцевій валюті | тис.грн. | 180 000 | 0 | 0 | 10 000 | 0 | 0 | 0 | 0 | 0 | 0 | 190 000 |
| - у іноземній валюті | тис.долл. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| = Вільні грошові кошти | тис.грн. | 177 596 | 174 762 | 170 563 | 154 138 | 125 219 | 99 305 | 73 459 | 78 265 | 84 677 | 0 | 84 677 |
| - у місцевій валюті | тис.грн. | 177 596 | 174 762 | 170 563 | 154 138 | 125 219 | 99 305 | 73 459 | 78 265 | 84 677 | 0 | 84 677 |
| - у іноземній валюті | тис.долл. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис 1.7.3. Джерела фінансування проекту

На рисунку 1.7.3 зазначені джерела фінансування проекту, з яких статутний капітал 50000, цільове фінансування 30000, а позиковий капітал 100000.

| ЧИСЕЛЬНІСТЬ І ЗАРБІТНА ПЛАТА | | 01.03.2022 | 1 міс | 2 міс | 3 міс | 4 міс | 5 міс | 6 міс | 7 міс | 8 міс | ВСЬОГО | |
|---|-----------|------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----|
| Основний виробничий персонал | | | | | | | | | | | | |
| Розробник Фул Стек | | | | | | | | | | | | |
| - чисельність | чел. | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | - | |
| - місячний оклад | грн./мес. | 20 000 | 20 000 | 20 000 | 20 000 | 20 000 | 20 000 | 20 000 | 20 000 | 20 000 | - | |
| - витрати на заробітну плату | тис.грн. | | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 160 | |
| Розробник Додатку | | | | | | | | | | | | |
| - чисельність | чел. | | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | - | |
| - місячний оклад | грн./мес. | 20 000 | 20 000 | 20 000 | 20 000 | 20 000 | 0 | 0 | 0 | 0 | - | |
| - витрати на заробітну плату | тис.грн. | | 20 | 20 | 20 | 20 | 0 | 0 | 0 | 0 | 80 | |
| = Чисельність основного виробничого персоналу | чел. | | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | - | |
| = Заробітна плата основного виробничого персоналу | тис.грн. | | 40 | 40 | 40 | 40 | 20 | 20 | 20 | 20 | 240 | |
| = Нарахування на заробітну плату (єдиний соціальний податок + страхування) | тис.грн. | | 14 | 14 | 14 | 14 | 7 | 7 | 7 | 7 | 85 | |
| Допоміжний виробничий персонал | | | | | | | | | | | | |
| Тестувальник | | | | | | | | | | | | |
| - чисельність | чел. | | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | - | |
| - місячний оклад | грн./мес. | 15 000 | 15 000 | 15 000 | 15 000 | 15 000 | 15 000 | 15 000 | 15 000 | 15 000 | - | |
| - витрати на заробітну плату | тис.грн. | | 0 | 0 | 15 | 15 | 15 | 15 | 0 | 0 | 60 | |
| = Чисельність допоміжного персоналу | чел. | | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | - | |
| = Заробітна плата допоміжного персоналу | тис.грн. | | 0 | 0 | 15 | 15 | 15 | 15 | 0 | 0 | 60 | |
| = Нарахування на заробітну плату (єдиний соціальний податок + страхування) | тис.грн. | | 0 | 0 | 5 | 5 | 5 | 5 | 0 | 0 | 21 | |
| Адміністративно-управлінський персонал | | | | | | | | | | | | |
| Проектний менеджер | | | | | | | | | | | | |
| - чисельність | чел. | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | - | |
| - місячний оклад | грн./мес. | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | - | |
| - витрати на заробітну плату | тис.грн. | | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 200 | |
| = Чисельність адміністративно-управлінського персоналу | чел. | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | - | |
| = Заробітна плата адміністративно-управлінського персоналу | тис.грн. | | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 200 | |
| = Нарахування на заробітну плату (єдиний соціальний податок + страхування) | тис.грн. | | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 71 | |
| Персонал з продажів | | | | | | | | | | | | |
| Маркетолог | | | | | | | | | | | | |
| - чисельність | чел. | | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | - | |
| - місячний оклад | грн./мес. | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | 25 000 | - | |
| - витрати на заробітну плату | тис.грн. | | 0 | 0 | 25 | 25 | 25 | 25 | 25 | 25 | 150 | |
| = Чисельність персоналу з продажів | чел. | | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | - | |
| = Заробітна плата персоналу з продажів | тис.грн. | | 0 | 0 | 25 | 25 | 25 | 25 | 25 | 25 | 150 | |
| = Нарахування на заробітну плату (єдиний соціальний податок + страхування) | тис.грн. | | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 9 | 53 | |
| = Разом чисельність | чел. | | 0 | 3 | 3 | 5 | 5 | 4 | 4 | 3 | 3 | - |
| = Разом витрати на зарплату | тис.грн. | | 0 | 65 | 65 | 105 | 105 | 85 | 85 | 70 | 70 | 650 |
| = Разом Нарахування на заробітну плату (єдиний соціальний податок + страхування) | тис.грн. | | 0 | 23 | 23 | 37 | 37 | 30 | 30 | 25 | 25 | 231 |

Рис. 1.7.3. Чисельність і заробітна плата

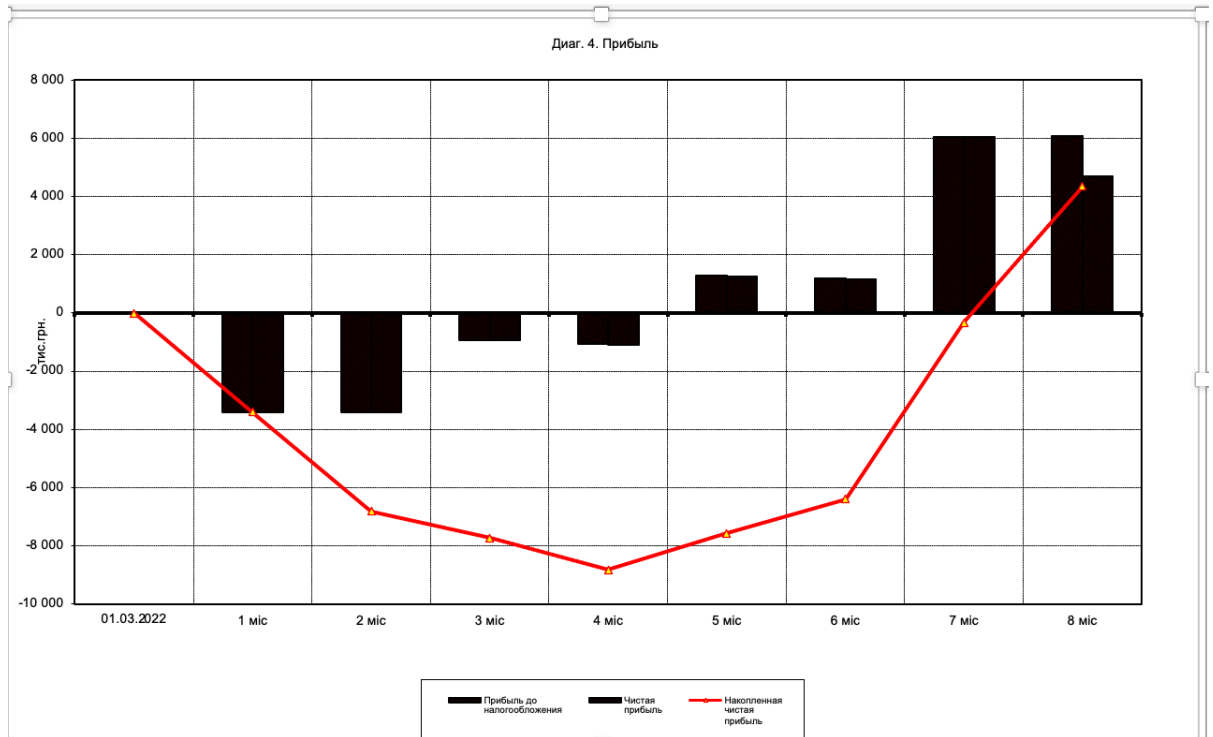


Рис. 1.7.4. Прибуток

Додаткові рисунки для детального ознайомлення можна знайти у додатку В.

Також нижче наведені схематичні графіки, з яких виходить, що на підставі розрахованих показників з економічної точки зору проєкт є прийнятним для реалізації.

Згідно даних на графіку 1.7.4. можемо зазначити, що проєкт виручки в перші 4 місяці проєкт приносити не буде, проте з 5 місяця проєкт NFT маркетплейсу почне отримувати прибуток.

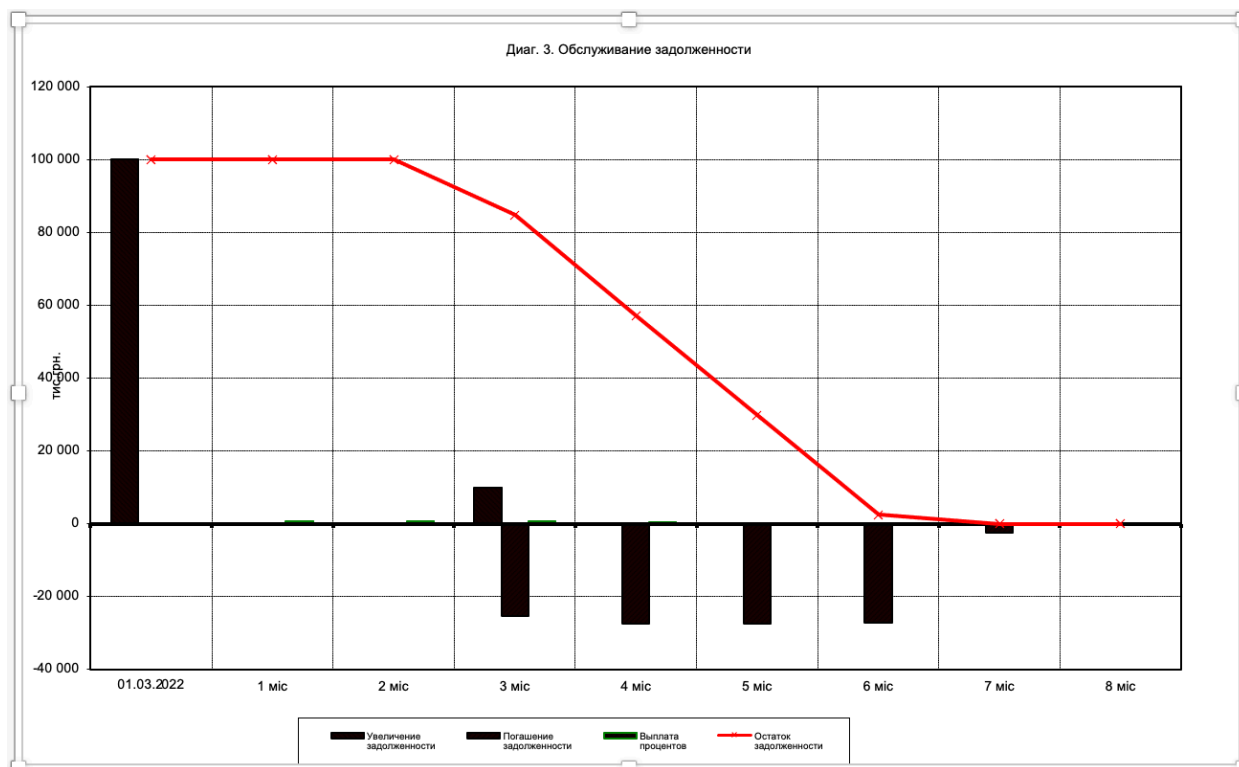


Рис. 1.7.5. Обслуговування заборгованості

На рисунку 1.7.5. Показан рівень заборгованості на початок проекту в 100 000 грн, починаючи з 2 місяця можемо спостерігати зниження і вже на 7 відсутність заборгованості.

1.8. Висновки

Даний розділ був присвячений дослідженню стартап-проекту. В якості такого була представлена система NFT-маркетплейсу.

У рамках розділу було досліджено розробку стратегій виходу на ринок та маркетинг-стратегії для цього. Зокрема, даний ринок являється сприятливим з невеликою кількістю представлених компаній конкурентів. Оскільки вони дають лише частину функцій, а запропонована система є універсальною та доступною, то у стартап-проекту є всі шанси стати монополістами на ринку.

Також були опрацьовані сильні та слабкі сторони проекту, SWOT аналіз, аналіз конкурентів та цільової аудиторії, інвестиційне дослідження. На

основі всіх досліджень був сформований концепт маркетингової стратегії для обраних цільових аудиторій.

РОЗДІЛ 2. ПЛАНУВАННЯ ВИКОНАННЯ ПРОЕКТУ

2.1. Життєвий цикл проекту

Життєвий цикл проекту — це період часу від задуму проекту до його закінчення, який може характеризуватися моментом здійснення перших витрат за проектом (поява проекту) і отриманням останньої вигоди (ліквідація проекту).

Життєвий цикл проекту складається з 6 етапів, наведених в таблиці 2.1.

Табл.2.1. Життєвий цикл проекту

| Назва фази життєвого циклу проекту | Опис |
|------------------------------------|--|
| Виявлення вимог | <p>Виявлення вимог є критичним для успішної реалізації проекту. На цій стадії потрібно визначити, зібрати, класифікувати потреби та обмеження від замовників та зацікавлених сторін.</p> <p>Результатом цієї роботи є аналіз предметної області, результати якої дадуть змогу достовірно описати нагальну проблему та шляхи її вирішення.</p> <p>Традиційними методами збору та виявлення вимог є інтерв'ю, консультації, анкетування.</p> |
| Специфікація вимог | <p>Мета цієї стадії полягає в тому, щоб створити документ та специфікацію вимог для проекту, тобто це процес документації результату етапу виявлення вимог.</p> |

| | |
|----------------------|--|
| Підготовка до роботи | Цей етап включає в забезпечення персоналу усіма необхідними ресурсами, як матеріальними, так і нематеріальними, підготовку юридичної документації. |
| Розробка платформи | Це процес протипування, програмування, дизайну, та наповнення контентом платформи. Цей етап є циклічним та повторюваним, з тестуванням та переробленням помилок за необхідністю. |
| Запуск платформи | Етап, коли платформа вже повністю готова і доступна до користувачів. Випуск маркетплейса на ринок для вільного доступу інтернет-користувачів. |
| Закриття проекту | Етап закриття проекту, під час якої відбувається передача проекту та результатів замовнику. |

Проект довжиною у 8 місяців.

Початок проекту - 01.03.2022.

Кінець проекту - 01.11.2022.

2.2. Розробка організаційної структури управління проектом.

Формування команди проекту

Необхідно визначити склад команди та спеціалістів, відповідальних за реалізацію проекту.

На чолі компанії знаходиться CEO.

В компанії присутні 6 відділів, а саме:

- 1) Розробники - відповідальні за створення онлайн-сервісу та цифрові бізнес-рішення. Команда проекту.
- 2) Support відділ - відповідальні за швидку та якісну підтримку користувачів, систем та інфраструктур.
- 3) Маркетинговий відділ - відповідальні за аналіз ринку, прогнозування обсягу продажів, вивчення цільової аудиторії проекту та вивчають тенденції розвитку ринку.
- 4) Бухгалтерія - призначені для ведення бухгалтерського обліку, забезпечення контролю за наявністю та рухом майна.
- 5) HR відділ - відповідальні за участь у процесах рекрутингу, HR-адміністрування, адаптації нового персоналу.
- 6) Юридичний відділ - консультація з правових питань.

До складу команди проекту входять:

СТО - несе загальну відповідальність за управління фізичною та кадровою технологічною інфраструктурою, включаючи розгортання технології, керування мережею та системою, інтеграційне тестування та розвиток персоналу з технічних операцій.

Керівник проекту - несе відповідальність за планування, закупівлі та реалізацію проекту в будь-якому підприємстві, яке має визначений обсяг, визначений початок і визначений кінець.

Backend розробник - код створення, який дозволяє базі даних і програмі спілкуватися один з одним. Розробники серверної частини піклуються та обслуговують серверну частину веб-сайту, включаючи бази даних, сервери та програми, і вони контролюють те, що ви не бачите.

Frontend розробник - це розробка графічного інтерфейсу користувача веб-сайту за допомогою HTML, CSS і JavaScript.

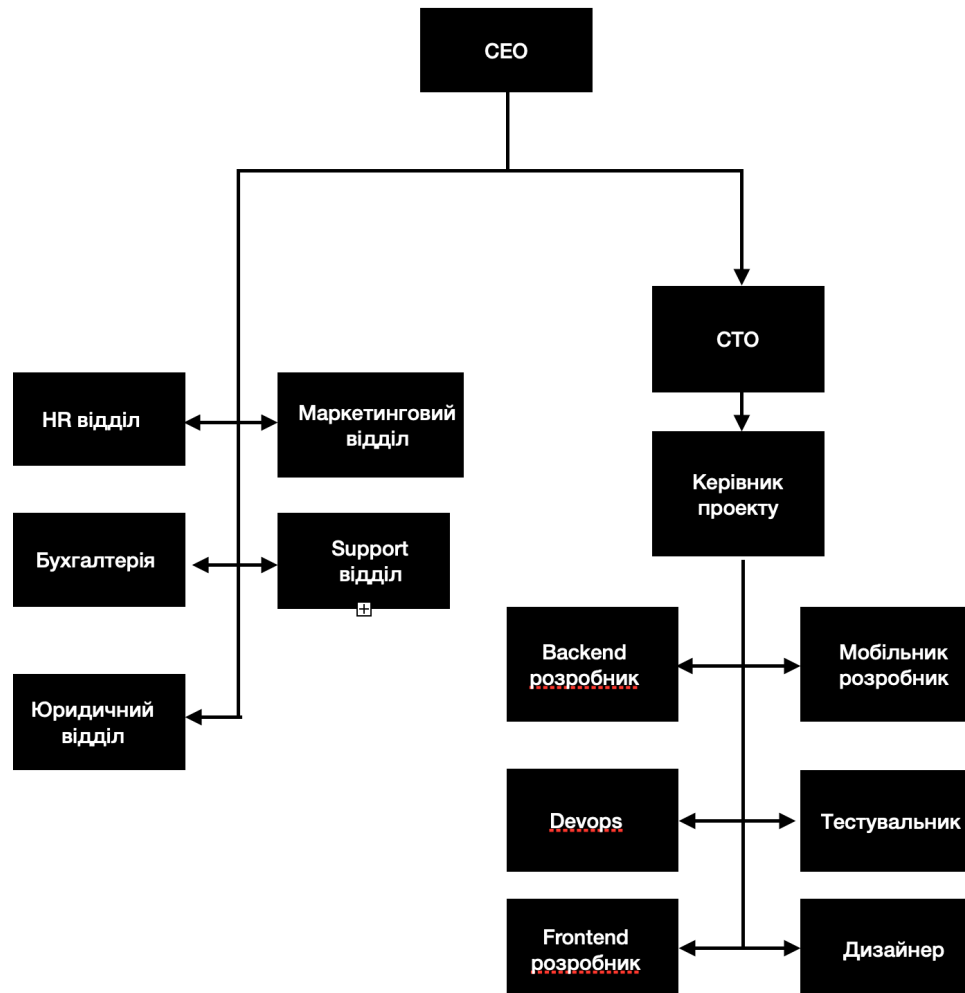
Мобільний розробник - спеціалізуються на мобільних технологіях, таких як створення додатків для платформ Android від Google, iOS від Apple і Windows Phone від Microsoft.

Devops - він визначається як розробник програмного забезпечення.

Тестувальник - займається тестуванням програмного забезпечення, виявленням та усуненням помилок у програмному забезпеченні.

Дизайнер - створення графічного дизайну для проекту.

Схематично продемонструємо організаційну структуру компанії:



2.3. Календарне планування проекту

Календарне планування проекту проведемо в додатку «MSProject».

Старт проекту було заплановано на 01.03.2022, всього 6 фаз життєвого циклу проекту.

| | Ім'я ▾ | Тривалість ▾ | Початок ▾ | Завершення ▾ |
|---|---|--------------|-----------|--------------|
| 1 | <input checked="" type="radio"/> Проект розробки NFT маркетплейсу | 1 день | 1.3.2022 | 1.3.2022 |
| 2 | <input type="radio"/> Початок проекту | 1 день | 1.3.2022 | 1.3.2022 |
| 3 | <input type="radio"/> 1. Виявлення вимог | 11 дн. | 1.3.2022 | 15.3.2022 |
| 4 | <input type="radio"/> 1.1.2. Інтерв'ю | 1 день | 1.3.2022 | 1.3.2022 |
| 5 | <input type="radio"/> 1.1.3. Анкетування | 1 день | 2.3.2022 | 2.3.2022 |

Рис.2.3. Фрагмент календарного плану. Перший етап проекту

На першому етапі відбувається процес збору вимог. Керівник проекту з самого початку має зробити процес збору, перегляду, документування, зрозуміти потреби і обмеження зацікавлених сторін і користувачів, а також замовника проекту.

| | Ім'я ▾ | Тривалість ▾ | Початок ▾ | Завершення ▾ |
|---|--|--------------|------------|--------------|
| 6 | <input type="radio"/> 2. Специфікація вимог | 4 дн. | 10.3.2022 | 15.3.2022 |
| 7 | <input type="radio"/> 2.1. Написання вимог для розробників | 1 день | 16.12.2022 | 16.12.2022 |
| 8 | <input type="radio"/> 2.2. Написання вимог для дизайнера | 1 день | 17.3.2022 | 17.3.2022 |

Рис.2.4. Фрагмент календарного плану. Другий етап проекту

Другий етап складається з документації конкретних потреб проекту та вимог, які були зазначені після аналізу першого етапу проекту. Цей етап є вкрай важливим, оскільки є основою подальшої роботи над проектом.

| | Ім'я ▾ | Тривалість ▾ | Початок ▾ | Завершення ▾ |
|----|--|--------------|-----------|--------------|
| 9 | <input type="radio"/> 3. Підготовка до роботи | 5 дн. | 14.3.2022 | 18.3.2022 |
| 10 | <input type="radio"/> 3.1. Закупівля ноутбуків | 6 дн. | 14.3.2022 | 21.3.2022 |
| 11 | <input type="radio"/> 3.2. Закупівля серверів | 5 дн. | 14.3.2022 | 18.3.2022 |
| 12 | <input type="radio"/> 3.3. Оренда приміщення | 11 дн. | 11.3.2022 | 25.3.2022 |

Рис.2.3. Фрагмент календарного плану. Третій етап проекту

Третій етап проекту - це підготовка до роботи, яка складається з забезпечення команди усім необхідним, підписанням юридичної документації тощо.

Четвертий та основний етап - це розробка маркетплейсу. Фреймворк управління для цього проекту - SCRUM. Для проекту зі створення NFT маркетплейсу методологія підходить, тому що, в SCRUM

- Роботи діляться на невеликі підзадачі;
- Підзадачі виконуються командою у невеликі терміни(1-4 тижні), що називається спринтом;
- Кожен учасник - командний гравець, який задіяний в процесі створення проекту.

Саме в цьому проекті буде 5 спринтів. Також кожен спринт поділен на три підгрупи, згідно задач, які мають бути виконані: дизайн, frontend, backend.

Рис 2.4. Фрагмент календарного плану. Перший спринт

В першому спринті основними завданнями є створення дизайну шаблону маркетплейсу для дизайнера, створення шаблону та домашньої сторінки для фронтенд розробника, а також верстка рейтингу NFT, карточки NFT та розділу відгуків про автора для бекенд розробника.

| | Ім'я | Тривалість | Початок | Завершення |
|----|---|------------|-----------|------------|
| 24 | ○ 4.2. Спринт 2 | 3 дн. | 11.4.2022 | 13.4.2022 |
| 25 | ○ 4.2.1. Корзина дизайн | 3 дн. | 11.4.2022 | 13.4.2022 |
| 26 | ○ 4.2.2. Список NFT frontend | 3 дн. | 11.4.2022 | 13.4.2022 |
| 27 | ○ 4.2.4. Список товарів frontend | 3 дн. | 11.4.2022 | 13.4.2022 |
| 28 | ○ 4.2.5. Верстка домашньої сторінки backend | 3 дн. | 11.4.2022 | 13.4.2022 |

Рис 2.5. Фрагмент календарного плану. Другий спринт

Під час другого спринта для дизайну основним завданням є створення корзини, для фронтенду - список NFT та каталог товарів, для бекенд - верстка домашньої сторінки.

| | Ім'я | Тривалість | Початок | Завершення |
|----|---|------------|-----------|------------|
| 29 | 4.3. Спринт 3 | 10 дн. | 14.4.2022 | 27.4.2022 |
| 30 | 4.3.1. Оформлення замовлення дизайн | 10 дн. | 14.4.2022 | 27.4.2022 |
| 31 | 4.3.2. Реєстрація дизайн | 10 дн. | 14.4.2022 | 27.4.2022 |
| 32 | 4.3.3. Список розділів дизайн | 10 дн. | 14.4.2022 | 27.4.2022 |
| 33 | 4.3.4. Список NFT frontend | 10 дн. | 14.4.2022 | 27.4.2022 |
| 34 | 4.3.5. Деталізація NFT frontend | 10 дн. | 14.4.2022 | 27.4.2022 |
| 35 | 4.3.6. Реєстрація frontend | 10 дн. | 14.4.2022 | 27.4.2022 |
| 36 | 4.3.7. Корзина frontend | 10 дн. | 14.4.2022 | 27.4.2022 |
| 37 | 4.3.8. Авторизація backend | 10 дн. | 14.4.2022 | 27.4.2022 |
| 38 | 4.3.9. Верстка список NFT і деталізація backend | 10 дн. | 14.4.2022 | 27.4.2022 |

| | Ім'я | Тривалість | Початок | Завершення |
|----|-------------------------------------|------------|-----------|------------|
| 13 | 4. Розробка маркетплейсу | 1 день | 25.3.2022 | 25.3.2022 |
| 14 | Початок розробки | 1 день | 25.3.2022 | 25.3.2022 |
| 15 | 4.1. Спринт 1 | 10 дн. | 28.3.2022 | 8.4.2022 |
| 16 | 4.1.1. Домашня сторінка дизайн | 10 дн. | 28.3.2022 | 8.4.2022 |
| 17 | 4.1.2. Шаблон маркетплейсу дизайн | 10 дн. | 28.3.2022 | 8.4.2022 |
| 18 | 4.1.3. Список товарів дизайн | 10 дн. | 28.3.2022 | 8.4.2022 |
| 19 | 4.1.4. Домашня сторінка frontend | 10 дн. | 28.3.2022 | 8.4.2022 |
| 20 | 4.1.5. Шаблон маркетплейсу frontend | 10 дн. | 28.3.2022 | 8.4.2022 |
| 21 | 4.1.6. Рейтинг NFT backend | 10 дн. | 28.3.2022 | 8.4.2022 |
| 22 | 4.1.7. Карточка товару backend | 10 дн. | 28.3.2022 | 8.4.2022 |
| 23 | 4.1.6. Відгуки про автора backend | 10 дн. | 28.3.2022 | 8.4.2022 |

Рис 2.6. Фрагмент календарного плану. Третій спринт

Під час третього спринту основним завданням для дизайну є оформлення замовлення, реєстрація, список розділів, для фронтенду список

NFT, деталізація NFT, реєстрація, корзина, для бекенду авторизація, верстка

| | Ім'я | Тривалість | Початок | Завершення |
|----|--|------------|-----------|------------|
| 39 | ○ 4.4. Спринт 4 | 6 дн. | 28.4.2022 | 5.5.2022 |
| 40 | ○ 4.4.1. Особистий кабінет дизайн | 6 дн. | 28.4.2022 | 5.5.2022 |
| 41 | ○ 4.4.2. Мобільний сайт дизайн | 6 дн. | 28.4.2022 | 5.5.2022 |
| 42 | ○ 4.4.3. Оформлення замовлення дизайн | 6 дн. | 28.4.2022 | 5.5.2022 |
| 43 | ○ 4.4.4. Список розділів frontend | 6 дн. | 28.4.2022 | 5.5.2022 |
| 44 | ○ 4.4.5. Порівняння NFT frontend | 6 дн. | 28.4.2022 | 5.5.2022 |
| 45 | ○ 4.4.6. Особистий кабінет frontend | 6 дн. | 28.4.2022 | 5.5.2022 |
| 46 | ○ 4.4.7. Верстка корзини backend | 6 дн. | 28.4.2022 | 5.5.2022 |
| 47 | ○ 4.4.8. Оформлення замовлення backend | 6 дн. | 28.4.2022 | 5.5.2022 |
| 48 | ○ 4.4.9. Каталог авторів backend | 6 дн. | 28.4.2022 | 5.5.2022 |

списка та деталізацію NFT.

Рис 2.7. Фрагмент календарного плану. Четвертий спринт

На четвертому спринті відбувається для дизайну створення особистого кабінету, мобільного сайту, оформлення замовлення, для фронтенду список розділів, порівняння NFT, особистий кабінет, для бекенду верстка корзини, оформлення замовлення, каталог авторів.

| | Ім'я | Тривалість | Початок | Завершення |
|----|--|------------|----------|------------|
| 49 | ○ 4.5. Спринт 5 | 11 дн. | 6.5.2022 | 20.5.2022 |
| 50 | ○ 4.5.1. Завершення mobile версії дизайн | 11 дн. | 6.5.2022 | 20.5.2022 |
| 51 | ○ 4.5.1. Правки дизайн | 11 дн. | 6.5.2022 | 20.5.2022 |
| 52 | ○ 4.5.3. Mobile версія frontend | 11 дн. | 6.5.2022 | 20.5.2022 |
| 53 | ○ 4.5.4. Правки frontend | 11 дн. | 6.5.2022 | 20.5.2022 |
| 54 | ○ 4.5.5. Завершення верстки backend | 11 дн. | 6.5.2022 | 20.5.2022 |
| 55 | ○ 4.5.6. Правки backend | 11 дн. | 6.5.2022 | 20.5.2022 |

Рис 2.8. Фрагмент календарного плану. П'ятий спринт

На п'ятому спринті у дизайна основним завданням є завершення мобільного версії та правки, для фронтенду мобільна версія та правки, а для бекенду завершення верстки та правки.

| | | | | | |
|----|-----------------------|---------------------------|--------|----------|-----------|
| 56 | <input type="radio"/> | 5. Запуск платформи | 11 дн. | 6.5.2022 | 20.5.2022 |
| 57 | <input type="radio"/> | 5.1. Запуск вебсайта | 11 дн. | 6.5.2022 | 20.5.2022 |
| 58 | <input type="radio"/> | 5.2. Запуск mobile версії | 11 дн. | 6.5.2022 | 20.5.2022 |

Рис 2.8. Фрагмент календарного плану. П'ятий етап проекту

П'ятий етап проекту - це предзавершальний етап, а саме - запуск платформи, до якого входить запуск вебсайту та мобільної версії.

| | | | | | |
|----|-----------------------|---------------------------------|--------|----------|-----------|
| 59 | <input type="radio"/> | 6.Закриття проекту | 11 дн. | 6.5.2022 | 20.5.2022 |
| 60 | <input type="radio"/> | 6.1. Проведення аудиту | 11 дн. | 6.5.2022 | 20.5.2022 |
| 61 | <input type="radio"/> | 6.2. Документування результатів | 11 дн. | 6.5.2022 | 20.5.2022 |
| 62 | <input type="radio"/> | 6.3. Проект завершено | 11 дн. | 6.5.2022 | 20.5.2022 |

Рис 2.9. Фрагмент календарного плану. Шостий етап проекту

Шостий і останній етап проекту складається з

- Проведення аудиту
- Документування результатів

Якщо результати проекту задовольняють, то проект можна вважати завершеним.

2.4. Планування ресурсів проекту

Кожен проект потребує ресурсів, щоб сприяти його реалізації та, зрештою, успіху. Успішне управління проектом означає ефективне управління різними ресурсами проекту. Вони повинні бути визначені якомога раніше, в ідеалі ще до запуску проекту.

1. Людські ресурси

Також відомі як «робочі ресурси», вони є найважливішими, оскільки саме вони виконують роботу. Вони сприяють виконанню завдань, необхідних для безперебійної роботи та завершення проекту.

- Керівник проекту
- Backend розробник
- Frontend розробник
- Мобільний розробник
- Devops
- Тестувальник
- Дизайнер

2. Матеріальні ресурси

Матеріальні ресурси включають

- Сервери
- Ноутбук
- Приміщення

Вони включають як ресурси, якими компанія вже володіє, так і ті, які вона купує або орендує для виконання проекту.

3. Фінансові ресурси

Фінансові ресурси відповідають бюджету проекту, який буде визначено перед запуском спонсором проекту.

Вони використовуються для фінансування:

- Людські та матеріальні ресурси проекту, загалом охоплюючи винагороду учасників проекту
- Придбання матеріальних ресурсів або їх оренда,
- Інші витрати, наприклад витрати на відрядження.

4. Ресурси часу

Ресурси часу – це періоди часу, доступні та використані для виконання кожного завдання. Тривалість завдання залежатиме від запланованих і наявних людських ресурсів.

Ці ресурси не є невичерпними: вони мають межі, тому важливо знати, як ними керувати для досягнення своїх цілей.

Ресурси часу:

- План проекту
- Графік проекту

- Вкладений час

| | Ім'я | Тип | Група | Стандартна ставка | Витрати на використання |
|----|---|--------------|------------|-------------------|-------------------------|
| 1 | <input type="radio"/> Керівник проекту | Трудовий | Люд | 156₺/год | 0₺ |
| 2 | <input type="radio"/> Backend розробник | Трудовий | Люд | 125₺/год | 0₺ |
| 3 | <input type="radio"/> Frontend розробник | Трудовий | Люд | 125₺/год | 0₺ |
| 4 | <input type="radio"/> Мобільний розробник | Трудовий | Люд | 125₺/год | 0₺ |
| 5 | <input type="radio"/> Devops | Трудовий | Люд | 125₺/год | 0₺ |
| 6 | <input type="radio"/> Тестувальник | Трудовий | Люд | 94₺/год | 0₺ |
| 7 | <input type="radio"/> Дизайнер | Трудовий | Люд | 100₺/год | 0₺ |
| 8 | <input type="radio"/> Сервери | Матеріальний | Прилад | 0₺/год | 13 000₺ |
| 9 | <input type="radio"/> Обладнання | Матеріальний | Прилад | 0₺/год | 25 000₺ |
| 10 | <input type="radio"/> Офіс | Матеріальний | Приміще... | 0₺/год | 15 000₺ |

Рис.2.4 Лист ресурсів проекту

2.4. Планування вартості проекту

Після планування термінів та ресурсів по проекту необхідно розрахувати вартість проекту.

Для розрахунку вартості проекту враховуються:

- Витрати на трудові ресурси;
- Витрати на матеріальні ресурси;
- Адміністративно-господарські витрати;
- Витрати на забезпечення команди необхідними ресурсами;
- Витрати на послуги;
- Витрати на корпоративну культуру.

Розподіл вартості проекту представляє всі категорії витрат конкретного проекту. Для аналізу витрат потрібно визначити необхідну кількість людських та матеріальних ресурсів, а також вартість кожного ресурсу за годину.

Витрати на трудові ресурси

| Професія | Кількість людей | Вартість, грн/год | Годин | Загальна вартість, грн |
|--------------------------|-----------------|-------------------|-------|------------------------|
| Керівник проекту | 1 | 1562/год | 750 | 117 000 |
| Backend розробник | 1 | 1252/год | 1120 | 140 000 |
| Frontend розробник | 1 | 1252/год | 1120 | 140 000 |
| Мобільний розробник | 1 | 1252/год | 1120 | 140 000 |
| Devops | 1 | 1252/год | 1120 | 140 000 |
| Тестувальник | 1 | 942/год | 750 | 70 500 |
| Дизайнер | 1 | 1002/год | 650 | 65 000 |
| Загальна вартість | | | | 812 500 |

Розрахунок витрат на трудові ресурси - це загальна вартість кожного ресурсу залежно від кількості людей та годин на проекті. Загальна вартість суми, яку потрібно витратити на трудові ресурси складає 812 500 грн.

Також, необхідно зробити розрахунок витрат на матеріальні ресурси. Матеріальними ресурсами в проекті є - ноутбуки для кожного члена команди, а також 2 сервери.

Витрати на матеріальні ресурси

| Назва | Кількість, шт. | Вартість/шт., грн | Вартість, грн |
|--------------------------|----------------|----------------------|----------------|
| Ноутбук | 7 | 25 000 | 175 000 |
| Сервер | 2 | 13 000 | 26 000 |
| Загальна вартість | | | 201 000 |

За результатами підрахунків, витрати на матеріальні ресурси складають 201 000 грн.

Одним з пунктів для розрахунку витрат є також адміністративно-господарські витрати, серед яких, витрати на оренду офісу, водопостачання, водовідведення, електроенергію. Виплати мають відбуватися щомісячно, 8 місяців поспіль.

Адміністративно-господарські витрати

| Назва | Кількість міс. | Грн/міс. | Вартість, грн. |
|--------------------------|----------------|----------|----------------|
| Оренда офісу | 8 | 15 000 | 120 000 |
| Водопостачання | 8 | 2 000 | 16 000 |
| Водовідведення | 8 | 200 | 1 600 |
| Електроенергія | 8 | 3 000 | 24 000 |
| Загальна вартість | | | 161 600 |

Загальна вартість адміністративно-господарські витрат становить 161 600 грн.

Наступним етапом аналізу є аналіз витрат на забезпечення команди необхідними ресурсами для комфортної роботи команди в офісі. Це такі ресурси, як:

- Канцелярське приладдя;
- Забезпечення харчових продуктів, напоїв;
- Забезпечення місця для комфортного відпочинку.

Табл.2.7

| Назва | Од.вимір. | Кількість од./міс. | Вартість/ од., грн | Кількість міс. | Вартість, грн. |
|--------------------------|-----------|-----------------------|-----------------------|-------------------|-------------------|
| Ручки | шт | 10 | 10 | 8 | 800 |
| Блокноти | шт | 10 | 20 | 8 | 1 600 |
| Фрукти | ящ | 15 | 30 | 8 | 3 600 |
| Вода | л | 600 | 50 | 8 | 240 000 |
| Кава | упак | 10 | 100 | 8 | 8 000 |
| Чай | упак | 10 | 50 | 8 | 4 000 |
| Молоко | упак | 10 | 50 | 8 | 4 000 |
| Цукор | упак | 5 | 40 | 8 | 1 600 |
| Печиво | упак | 10 | 40 | 8 | 3 200 |
| Цукерки | упак | 10 | 70 | 8 | 5 600 |
| Пуфіки | шт | 4 | 200 | 1 | 800 |
| Газовані напої | Л | 200 | 30 | 8 | 48 000 |
| Загальна вартість | | | | | 321 100 |

Витрати на забезпечення команди необхідними ресурсами

Загальна вартість витрат на забезпечення команди необхідними ресурсами складає 321 100 грн.

Наступним етапом є аналіз витрат на послуги, а саме

- Інтернет;
- Прибирання в офісі;

- Підписка на додаток з англійської мови.

Ці витрати розраховуються строком на 8 місяців, щомісячно.

Табл.2.8

| Назва | Кількість міс. | Грн/міс. | Вартість, грн. |
|--|----------------|----------|----------------|
| Інтернет | 8 | 200 | 1 600 |
| Прибирання | 8 | 2000 | 16 000 |
| Підписка на онлайн додаток Grammarly | 8 | 1000 | 8 000 |
| Загальна вартість | | | 25 600 |

Витрати на послуги

Загальна вартість послуг становить 25 600 грн.

Витрати на корпоративну культуру - ще один пункт витрат проекту, який включає в себе кошти на:

- Проведення тимблдингу;
- Участь в конференції;
- Курси англійської мови.

Вони необхідні для мотивації співробітників, підтримці зацікавленості в проекті та компанії, підтримки репутації компанії. Тімблдинг та курси будуть проходити 8 місяців, конференція 1 раз за 8 місяців.

Табл.2.9

Витрати на корпоративну культуру

| Назва | Кількість міс. | Грн/міс. | Вартість, грн. |
|---------------------------|----------------|----------|----------------|
| Тімблдинг | 8 | 5000 | 40 000 |
| Конференція | 1 | 10000 | 10 000 |
| Курси англійської мови | 8 | 5000 | 40 000 |
| Загальна вартість | | | 90 000 |

Згідно з розрахунками, загальна вартість витрат на корпоративну культуру складає 90 000 грн.

Для врахування ризиків та закладення коштів в бюджет на вирішення проблемних ситуацій, потрібно врахувати непередбачувані витрати, а саме:

- Техніка;
- Послуги;
- Адміністративно-господарські витрати;
- Тімблдінг.

Закладається загальна сума в бюджет, без прив'язки до місяців.

Табл.2.10

Непередбачувані витрати

| Назва | Сума на весь період, грн |
|---|--------------------------|
| Техніка | 20 000 |
| Послуги | 10 000 |
| Адміністративно-господарські витрати | 10 000 |
| Тімблдінг | 5000 |
| Загальна вартість | 45 000 |

Загальна вартість на непередбачувані витрати, яка закладена в бюджет, складає 45 000 грн.

Після окремого розрахунку витрат проекту, потрібно розрахувати зведені витрати для формування загального бюджету.

Таким чином сформуємо бюджет проекту, наведено в таблиці 2.11.

Зведені витрати по проекту

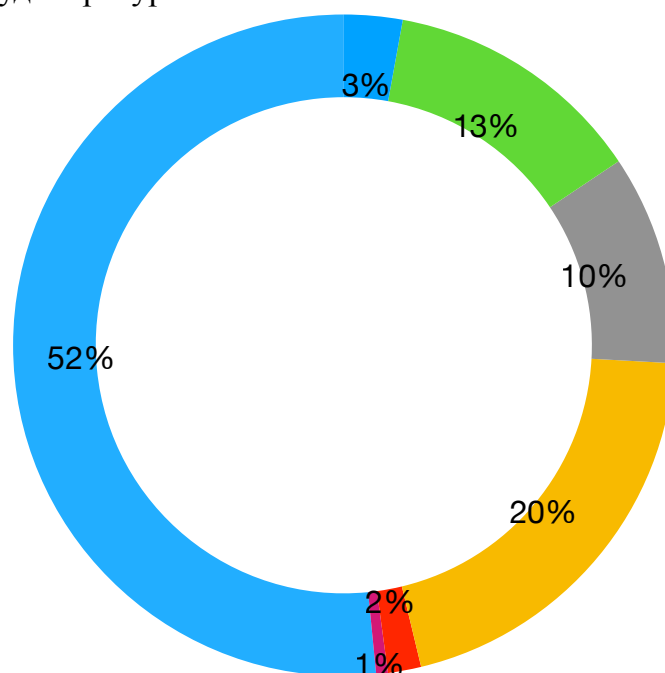
| Вид витрат | Сума, грн |
|--|------------------|
| Витрати на трудові ресурси | 812 500 |
| Витрати на матеріальні ресурси | 201 000 |
| Адміністративно-господарські витрати | 161 600 |
| Витрати на забезпечення команди необхідними ресурсами | 321 100 |
| Витрати на послуги | 25 600 |
| Витрати на корпоративну культуру | 90 000 |
| Непередбачувані витрати | 45 000 |
| Загальна сума | 1 656 800 |

Таким чином, в результаті розрахунку всіх витрат по проекту, вартість проекту 1 656 800 грн. Розглянемо витрати по проекту у відсотковому співвідношенні.

Згідно відсоткового співвідношення, 52% бюджету складають трудові ресурси, 20% витрати на забезпечення команди необхідними ресурси, 13% витрати на матеріальні ресурси, 10% адміністративно-господарські витрати, 3% закладено на непередбачуванні виплати, 2% витрати на послуги, 1% витрати на корпоративну культуру.

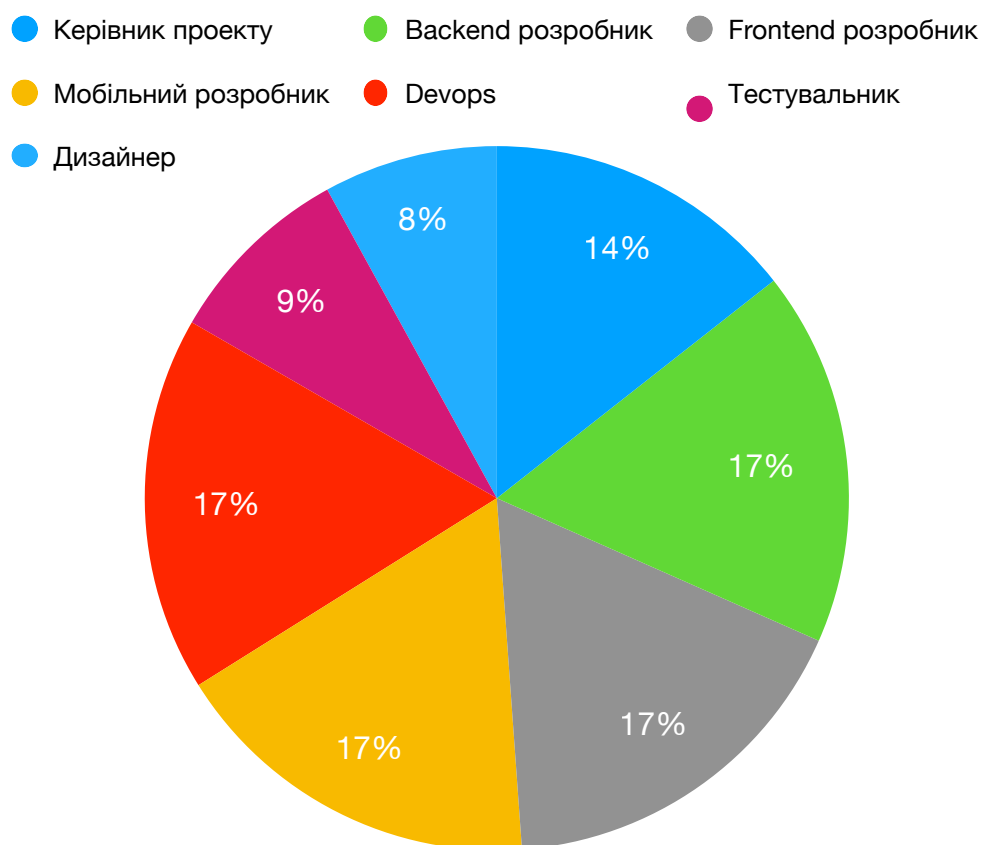
Рис.2.5

- Непередбачувані витрати
- Витрати на матеріальні ресурси
- Адміністративно-господарські витрати
- Витрати на забезпечення команди необхідними ресурсами
- Витрати на послуги
- Витрати на корпоративну культуру
- Витрати на трудові ресурси



Так як, витрати на трудові ресурси є найбільшим пунктом витрат, також наведемо відсоткове відношення нижче.

Рис.2.6.



РОЗДІЛ 3

ОГЛЯД ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ

3.1 Огляд мови програмування

Java — це високорівнева, заснована на класах, об'єктно-орієнтована мова програмування, яка розроблена так, щоб мати якомога менше залежностей реалізації. Це мова програмування загального призначення, призначена для того, щоб програмісти могли писати один раз, запускати в будь-якому місці (WORA), [17] що означає, що скомпільований код Java може працювати на всіх платформах, які підтримують Java, без необхідності перекомпіляції. [18] Програми Java зазвичай компілюються у байт-код, який може працювати на будь-якій віртуальній машині Java (JVM) незалежно від базової архітектури комп'ютера. Синтаксис Java подібний до C і C++, але має менше засобів низького рівня, ніж будь-який з них. Середовище виконання Java надає динамічні можливості (такі як відображення та модифікація коду під час виконання), які зазвичай недоступні в традиційних скомпільованих мовах. Станом на 2019 рік Java була однією з найпопулярніших мов програмування, що використовуються відповідно до GitHub, [19][20] особливо для веб-додатків клієнт-сервер, із 9 мільйонами розробників. [21]

Спочатку Java була розроблена Джеймсом Гослінгом у Sun Microsystems і випущена в травні 1995 року як основний компонент Java-платформи Sun Microsystems. Оригінальні та довідкові компілятори Java, віртуальні машини та бібліотеки класів спочатку були випущені Sun за власними ліцензіями. Станом на травень 2007 року, відповідно до специфікацій Java Community Process, Sun переліцензувала більшість своїх технологій Java за ліцензією лише GPL-2.0. Oracle пропонує власну віртуальну машину HotSpot Java, однак офіційною довідковою реалізацією є OpenJDK JVM, яка є безкоштовним програмним забезпеченням з відкритим

вихідним кодом і використовується більшістю розробників і є JVM за замовчуванням для майже всіх дистрибутивів Linux.

Станом на березень 2022 року Java 18 є останньою версією, тоді як Java 17, 11 і 8 є поточними версіями довгострокової підтримки (LTS). Oracle випустила останнє загальнодоступне оновлення за нульовою вартістю для застарілої версії Java 8 LTS у січні 2019 року для комерційного використання, хоча в іншому випадку вона все ще підтримуватиме Java 8 з загальнодоступними оновленнями для особистого використання на невизначений термін. Інші виробники почали пропонувати безкоштовні збірки OpenJDK 8 і 11, які все ще отримують безпеку та інші оновлення.

Oracle (та інші) настійно рекомендують видаляти застарілі та непідтримувані версії Java через невирішені проблеми безпеки в старих версіях.[22] Oracle радить своїм користувачам негайно перейти на підтримувану версію, наприклад одну з версій LTS (8, 11, 17).

Java JVM і байт-код

Однією з цілей дизайну Java є переносимість, що означає, що програми, написані для платформи Java, повинні працювати подібним чином на будь-якій комбінації апаратного забезпечення та операційної системи з належною підтримкою часу виконання. Це досягається шляхом компіляції коду мови Java в проміжне представлення, яке називається байт-кодом Java, замість безпосереднього машинного коду, специфічного для архітектури. Інструкції байт-коду Java аналогічні машинному коду, але вони призначені для виконання віртуальною машиною (VM), написаною спеціально для апаратного забезпечення хоста. Кінцеві користувачі зазвичай використовують Java Runtime Environment (JRE), встановлену на своєму пристрої для автономних програм Java або веб-браузер для Java-апплетів.

Стандартні бібліотеки надають загальний спосіб доступу до специфічних для хоста функцій, таких як графіка, потоки та мережа.

Використання універсального байт-коду спрощує перенесення. Однак накладні витрати на інтерпретацію байт-коду в машинні інструкції змушували інтерпретовані програми майже завжди працювати повільніше,

ніж рідні виконувані файли. Компілятори Just-in-time (JIT), які компілюють байт-код у машинний код під час виконання, були представлені на ранньому етапі. Компілятор Hotspot Java насправді є двома компіляторами в одному; і з GraalVM (включений, наприклад, у Java 11, але вилучений з Java 16), що дозволяє багаторівневу компіляцію.[47] Сама Java не залежить від платформи і адаптована до конкретної платформи, на якій вона має працювати, за допомогою віртуальної машини Java (JVM), яка перекладає байт-код Java на машинну мову платформи.[48]

Продуктивність

Програми, написані на Java, мають репутацію повільніших і вимагають більше пам'яті, ніж програми, написані на C++.[49][50] Однак швидкість виконання програм Java значно покращилася із запровадженням компіляції «точно вчасно» в 1997/1998 роках для програм на Java. Java 1.1,[51] додавання мовних функцій, які підтримують кращий аналіз коду (наприклад, внутрішні класи, клас `StringBuilder`, необов'язкові твердження тощо), а також оптимізації у віртуальній машині Java, наприклад HotSpot, ставши JVM Sun за замовчуванням у 2000 році. З Java 1.5 продуктивність була покращена за допомогою додавання пакета `java.util.concurrent`, включаючи безблоковані реалізації `ConcurrentMaps` та інших багатоядерних колекцій, а також покращено з Java 1.6.

Автоматичне управління пам'яттю

Java використовує автоматичний збірник сміття для управління пам'яттю в життєвому циклі об'єкта. Програміст визначає час створення об'єктів, а середовище виконання Java відповідає за відновлення пам'яті, коли об'єкти більше не використовуються. Як тільки посилання на об'єкт не залишаються, недоступна пам'ять стає придатною для автоматичного звільнення збирачем сміття. Щось подібне до витоку пам'яті все ще може статися, якщо код програміста містить посилання на об'єкт, який більше не потрібен, як правило, коли об'єкти, які більше не потрібні, зберігаються в контейнерах, які все ще використовуються. Якщо викликаються методи для неіснуючого об'єкта, створюється виняток нульового покажчика.[53][54]

Одна з ідей, що лежить в основі моделі автоматичного управління пам'яттю Java, полягає в тому, що програмісти можуть позбутися від тягара виконання ручного керування пам'яттю. У деяких мовах пам'ять для створення об'єктів неявно виділяється в стеку або явно виділяється і звільняється з купи. В останньому випадку відповідальність за управління пам'яттю покладається на програміста. Якщо програма не звільняє об'єкт, відбувається витік пам'яті. Якщо програма намагається отримати доступ або звільнити пам'ять, яка вже була вивільнена, результат буде невизначеним і важко передбачити, і програма, швидше за все, стане нестабільною або аварійно завершить роботу. Це можна частково виправити за допомогою розумних покажчиків, але вони додають накладних витрат і ускладнюють. Зауважте, що збір сміття не запобігає витoku логічної пам'яті, тобто тих, де пам'ять все ще посилається, але ніколи не використовується.

Збір сміття може статися в будь-який момент. В ідеалі це відбуватиметься, коли програма неактивна. Він гарантовано спрацює, якщо в купі недостатньо вільної пам'яті для виділення нового об'єкта; це може призвести до миттєвої зупинки програми. Явне керування пам'яттю неможливе в Java.

Java не підтримує арифметику вказівників у стилі C/C++, де адресами об'єктів можна арифметично маніпулювати (наприклад, додаючи або віднімаючи зміщення). Це дозволяє збірнику сміття переміщувати об'єкти, на які посилаються, і гарантує безпеку та безпеку типу.

Як і в C++ та деяких інших об'єктно-орієнтованих мовах, змінні примітивних типів даних Java зберігаються або безпосередньо в полях (для об'єктів), або в стеку (для методів), а не в купі, як це зазвичай вірно для непримітивних даних. типи (але див. escape-аналіз). Це було свідоме рішення дизайнерів Java з міркувань продуктивності.

Java містить кілька типів збирачів сміття. Починаючи з Java 9, HotSpot використовує Garbage First Garbage Collector (G1GC) за замовчуванням.[55] Однак є також кілька інших збирачів сміття, які можна використовувати для

керування купою. Для більшості програм на Java достатньо G1GC. Раніше в Java 8 використовувався Parallel Garbage Collector.

Вирішення проблеми управління пам'яттю не звільняє програміста від тягара належної обробки інших видів ресурсів, таких як мережеві з'єднання або підключення до бази даних, дескриптори файлів тощо, особливо за наявності винятків.

Jakarta Servlet (раніше Java Servlet) — це програмний компонент Java, який розширює можливості сервера. Хоча сервлети можуть відповідати на багато типів запитів, вони найчастіше реалізують веб-контейнери для розміщення веб-додатків на веб-серверах і, таким чином, кваліфікуються як веб-API сервлетів на стороні сервера. Такі веб-сервлети є аналогом Java для інших технологій динамічного веб-контенту, таких як PHP і ASP.NET.

Вступ

Сервлет Jakarta обробляє або зберігає клас Java в Jakarta EE, який відповідає API сервлетів Jakarta [1], стандарту для реалізації класів Java, які відповідають на запити. Сервлети в принципі можуть спілкуватися за будь-яким протоколом клієнт-сервер, але найчастіше вони використовуються з HTTP. Таким чином, «сервлет» часто використовується як скорочення «сервлет HTTP».[2] Таким чином, розробник програмного забезпечення може використовувати сервлет для додавання динамічного вмісту на веб-сервер за допомогою платформи Java. Згенерований вміст зазвичай являє собою HTML, але можуть бути й інші дані, такі як XML і частіше JSON. Сервлети можуть підтримувати стан змінних сеансу в багатьох транзакціях сервера за допомогою файлів cookie HTTP або зіставлення URL-адрес.

API Jakarta Servlet певною мірою було замінено двома стандартними технологіями Java для веб-сервісів:

1. веб-сервіси Jakarta RESTful Web Services (JAX-RS 2.0), корисні для служб AJAX, JSON і REST, а також
2. Веб-служби Джакарти XML (JAX-WS), корисні для веб-служб SOAP.

Для розгортання та запуску сервлета необхідно використовувати веб-контейнер. Веб-контейнер (також відомий як контейнер сервлетів) по суті є компонентом веб-сервера, який взаємодіє із сервлетами. Веб-контейнер відповідає за керування життєвим циклом сервлетів, зіставлення URL-адреси з певним сервлетом і забезпечення того, що запитувач URL-адрес має правильні права доступу.

API сервлетів, що міститься в ієрархії пакетів Java `javax.servlet`, визначає очікувані взаємодії веб-контейнера та сервлета.[2]

Сервлет - це об'єкт, який отримує запит і генерує відповідь на основі цього запиту. Базовий пакет сервлетів визначає об'єкти Java для представлення запитів і відповідей сервлетів, а також об'єкти, що відображають параметри конфігурації сервлета та середовище виконання. Пакет `javax.servlet.http` визначає HTTP-специфічні підкласи загальних елементів сервлетів, включаючи об'єкти керування сеансом, які відстежують численні запити та відповіді між веб-сервером і клієнтом. Сервлети можуть бути запаковані у файл WAR як веб-додаток.

Сервлети можуть бути автоматично створені з Jakarta Server Pages (JSP) компілятором Jakarta Server Pages. Різниця між сервлетами і JSP полягає в тому, що сервлети зазвичай вбудовують HTML в код Java, тоді як JSP вбудовують код Java в HTML. Хоча пряме використання сервлетів для генерації HTML (як показано в прикладі нижче) стало рідкістю, веб-фреймворк MVC вищого рівня в Jakarta EE (JSF) все ще явно використовує технологію сервлетів для низькорівневої обробки запитів/відповідей через `FacesServlet`. Дещо давнішим є використання сервлетів у поєднанні з JSP у шаблоні, який називається «Модель 2», який є різновидом контролера `model-view-controller`.

Поточна версія Servlet 5.0.[3]

Jakarta Server Pages (JSP; раніше JavaServer Pages) — це набір технологій, які допомагають розробникам програмного забезпечення створювати динамічно створювані веб-сторінки на основі HTML, XML, SOAP або інших типів документів. Випущений у 1999 році компанією Sun

Microsystems[1], JSP подібний до PHP і ASP, але використовує мову програмування Java.

Для розгортання та запуску Jakarta Server Pages потрібен сумісний веб-сервер із контейнером сервлетів, наприклад Apache Tomcat або Jetty.

Архітектурно JSP можна розглядати як високорівневу абстракцію сервлетів Java. JSP транслюються в сервлети під час виконання, тому JSP є сервлетом; кожен сервлет JSP кешується і повторно використовується, доки оригінальний JSP не буде змінено.[2]

Серверні сторінки Jakarta можна використовувати окремо або як компонент перегляду серверної моделі-вигляду-контролера, як правило, з JavaBeans як моделлю і сервлетами Java (або структурою, як-от Apache Struts) як контролером. Це тип архітектури Model 2.[3]

JSP дозволяє мережувати код Java і певні заздалегідь визначені дії зі статичним вмістом веб-розмітки, таким як HTML. Отримана сторінка компілюється і виконується на сервері для доставки документа. Скомпільовані сторінки, а також будь-які залежні бібліотеки Java містять байт-код Java, а не машинний код. Як і будь-яка інша програма .jar або Java, код має виконуватися на віртуальній машині Java (JVM), яка взаємодіє з операційною системою хоста сервера, щоб забезпечити абстрактне, нейтральне для платформи середовище.

JSP зазвичай використовуються для доставки документів HTML і XML, але за допомогою OutputStream вони також можуть доставляти інші типи даних.[4]

Веб-контейнер створює неявні об'єкти JSP, такі як запит, відповідь, сесія, програма, конфігурація, сторінка, pageContext, вихід і виняток. JSP Engine створює ці об'єкти на етапі перекладу.

Компілятор

Компілятор сторінок JavaServer — це програма, яка аналізує JSP і перетворює їх у виконуваний сервлет Java. Програма такого типу зазвичай вбудовується в сервер додатків і запускається автоматично під час першого доступу до JSP, але сторінки також можуть бути попередньо скомпільовані

для кращої продуктивності або скомпільовані як частина процесу збірки для перевірки на наявність помилок.[9]

Деякі контейнери JSP підтримують налаштування того, як часто контейнер перевіряє часові позначки файлу JSP, щоб побачити, чи змінилася сторінка. Зазвичай ця позначка часу встановлюється на короткий інтервал (можливо, секунди) під час розробки програмного забезпечення та довший інтервал (можливо, хвилини або навіть ніколи) для розгорнутого веб-додатка. [10]

3.2 Огляд середовища розробки

IntelliJ IDEA — це інтегроване середовище розробки (IDE), написане на Java для розробки комп'ютерного програмного забезпечення. Він розроблений компанією JetBrains (раніше відомий як IntelliJ) і доступний у вигляді ліцензованої спільноти Apache 2[2] та у приватній комерційній версії. Обидва можна використовувати для комерційного розвитку.[3][4]

Історія

Перша версія IntelliJ IDEA була випущена в січні 2001 року і була однією з перших доступних середовищ IDE Java з інтегрованими можливостями розширеної навігації та рефакторингу коду.[5][6].

У звіті InfoWorld за 2010 рік IntelliJ отримав найвищу оцінку центру тестування з чотирьох найкращих інструментів програмування Java: Eclipse, IntelliJ IDEA, NetBeans і JDeveloper.[7]

У грудні 2014 року Google оголосила про версію 1.0 Android Studio, IDE з відкритим кодом для програм Android, засновану на версії IntelliJ IDEA для спільноти з відкритим кодом.[8] Інші середовища розробки, засновані на фреймворку IntelliJ, включають AppCode, CLion, DataGrip, GoLand, PhpStorm, PyCharm, Rider, RubyMine, WebStorm і MPS.[9]

Особливості

Допомога в кодуванні

IDE надає певні функції[12], як-от завершення коду шляхом аналізу контексту, навігацію по коду, яка дозволяє безпосередньо переходити до класу або оголошення в кодї, рефакторинг коду, налагодження коду[13] , літінг і параметри для виправлення невідповідностей за допомогою пропозицій.

Вбудовані інструменти та інтеграція

IDE забезпечує[12] інтеграцію з інструментами збірки/пакування, такими як grunt, bower, gradle і SBT. Він підтримує системи контролю версій, такі як Git, Mercurial, Perforce і SVN. До таких баз даних, як Microsoft SQL Server, Oracle, PostgreSQL, SQLite і MySQL, можна отримати доступ безпосередньо з IDE у версії Ultimate через вбудовану версію DataGrip.

Екосистема плагінів

IntelliJ підтримує плагіни, за допомогою яких можна додати додаткові функції до IDE. Плагіни можна завантажити та встановити або з веб-сайту сховища плагінів IntelliJ, або через вбудовану функцію пошуку та встановлення плагінів IDE. Кожне видання має окремі сховища плагінів, і станом на 2019 рік у виданнях Community та Ultimate є понад 3000 плагінів кожен.[14]

3.3 Огляд додаткового інструментарію

Мова розмітки гіпертексту або HTML є стандартною мовою розмітки для документів, що відображаються у веб-браузері. У цьому можуть допомогти такі технології, як каскадні таблиці стилів (CSS) і мови сценаріїв, такі як JavaScript.

Веб-браузер отримує документи HTML з веб-сервера або локального сховища та перетворює документи на мультимедійні веб-сторінки. HTML семантично описує структуру веб-сторінки і спочатку містить підказки про те, як повинен виглядати документ.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою структури HTML зображення та інші об'єкти (наприклад,

інтерактивні форми) можуть бути вбудовані в скопійовані сторінки. HTML надає інструменти для створення структурованих документів, які представляють структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, посилання та інші елементи. Елементи HTML розділені тегами, записаними в кутових дужках. Такі теги, як `` і `<input />` вносять вміст безпосередньо на сторінку. Інші теги, такі як `<p>`, оточують та надають інформацію про текст документа, а також можуть включати інші теги як дочірні елементи. Замість відображення тегів HTML браузері використовують їх для інтерпретації вмісту сторінки.

HTML може вбудовувати скрипти, такі як JavaScript, які впливають на поведінку та вміст веб-сторінок. Увімкнення CSS визначає зовнішній вигляд і макет вмісту. Консорціум World Wide Web Consortium (W3C), колишній розпорядник HTML і поточний розпорядник стандартів CSS, заохочує використання CSS замість явного представлення HTML з 1997 року. [2] Форми HTML, відомі як HTML5, використовуються для відображення відео та аудіо, в основному за допомогою елемента `<canvas>` у поєднанні з javascript.

Розмітка HTML складається з кількох ключових компонентів, включаючи ті, які називаються тегами (та їх атрибутами), типи даних на основі символів, посилання на символи та посилання на сутність. Теги HTML найчастіше зустрічаються парами, наприклад `<h1>` і `</h1>`, хоча деякі теги представляють порожні елементи і тому є дивними, наприклад ``. Першим тегом у такій парі є початковий тег, а другий — кінцевий (також його називають початковим і кінцевим тегом).

Іншим важливим компонентом є оголошення типу документа HTML, яке запускає відтворення стандартного режиму.

Нижче наведено приклад класичного «Hello, World!». програма:

```
<!DOCTYPE html>
<html>
  <header>
    <title> Це ім'я</title>
```

```
</head>
<текст>
  <div>
    <p>Привіт, світ! </p>
  </div>
</text>
</html>
```

Текст між `<html>` і `</html>` описує веб-сторінку, а текст між `<body>` і `</body>` є видимим вмістом сторінки. Текст тегу `<title>` `<title>` визначає назву сторінки браузера, яка з'являється на вкладках і заголовках вікна браузера, а тег `<div>` визначає поділ сторінки, що використовується для спрощення дизайну. Ви можете використовувати елемент `<meta>` між `<head>` і `</head>`, щоб визначити метадані сторінки.

Оголошення `doctype <!DOCTYPE html>` призначено для HTML5. Різні браузери повертатимуться до «чудового» режиму, якщо рекламу не ввімкнено. [70]

Елемент

Документи HTML мають на увазі структуру вкладених елементів HTML. Вони представлені в документі тегами HTML, укладеними в кутові дужки наступним чином: `<p>`. [71] У простому загальному випадку область дії елемента визначається парою тегів: «відкриваючий тег» `<p>` і «завершальний тег» `</p>`. Текстовий вміст елемента (якщо є) розміщується між цими тегами.

Мітки також можуть містити додаткові теги міток між початком і кінцем, включаючи поєднання міток і тексту. Це вказує наступні (вкладені) елементи як дочірні елементи батьківського елемента.

Початковий тег також може включати атрибути елементів всередині тегу. Вони вказують додаткову інформацію, таку як ідентифікатор розділу в документі, ідентифікатор, який використовується для прив'язки інформації про стиль до представлення документа, а для деяких тегів, наприклад `` для вбудовування зображень, посилання на ресурси зображень у таких документах. : ``

Деякі елементи, такі як розриви рядків або `
`, не дозволяють вставляти будь-який вміст, текст чи інші теги. Вони вимагають лише порожнього тега (як початкового тегу) і не використовують закриваючий тег.

Багато тегів, включаючи дуже поширений закриваючий тег елемента абзацу `<p>`, є необов'язковими. Браузер HTML або інший проксі-сервер може визначити кінець елемента відповідно до контекстних і структурних правил, визначених стандартом HTML. Ці правила складні, і більшість програмістів HTML не розуміють.

Тому загальна форма елемента HTML: `<tag attribute1="value1" attribute2="value2"> " content " </tag>`. Деякі елементи HTML визначаються як порожні елементи і мають форму `<tag attribute1 = "value1" attribute2 = "value2">`. Порожні елементи можуть не містити вмісту, наприклад теги `
` або вбудовані теги ``. Назви елементів HTML – це імена, які використовуються в тегах. Зверніть увагу, що назви закриваючих тегів починаються з косої риски / , а в порожніх елементах закриваючі теги не є ані обов'язковими, ані дозволеними. Якщо атрибут не вказано, у кожному випадку використовується значення за замовчуванням.

Структурні маркери вказують на призначення тексту

Наприклад, `<h2> Golf </h2>` встановлює "Гольф" як заголовок другого рівня. Структурна розмітка не передбачає певного відтворення, але більшість веб-браузерів мають стилі за замовчуванням для форматування елементів. Вміст можна додатково стилізувати за допомогою каскадних таблиць стилів (CSS). [72]

Теги презентації вказують, як повинен виглядати текст незалежно від його призначення

Наприклад, `жирний текст` вказує на те, що пристрої візуального виводу мають відображати жирний текст, але мало вказує на те, які пристрої не повинні (наприклад, аудіопристрої, які читають текст вголос). У випадку `жирного тексту` і `<i>курсивного тексту</i>`, існують інші елементи, які можуть мати еквівалентні візуальні, але більш семантичні символи, наприклад `сильний текст</i> сильний >>` і `підкреслений`

текст відповідно. Легше зрозуміти, як слухові користувацькі агенти повинні інтерпретувати останні два елементи. Однак вони не еквівалентні своїм аналогам у презентації: наприклад, програми зчитування з екрана не очікують підкреслювати назви книг, але такі назви виділені на екрані курсивом. Відповідно до специфікації HTML 4.0 більшість елементів розмітки презентації застаріли на користь стилів за допомогою CSS.

Гіпертекстова розмітка перетворює частини документа на посилання на інші документи

Елемент прив'язки створює гіперпосилання в документі, а його атрибут href встановлює цільову URL-адресу посилання. Наприклад, тег HTML ` Вікіпедія ` відображатиме слово "Вікіпедія" як гіперпосилання. Щоб відобразити зображення як гіперпосилання, вставте елемент `img` як вміст всередині елемента `a`. Як і `br`, `img` є порожнім елементом з атрибутами, але без вмісту чи закриваючого тега. ` ` .

Атрибути

Більшість атрибутів елемента – це пари ім'я-значення, розділені знаком `=`, а початкова розмітка елемента записується після імені елемента. Значення можуть бути взяті в одинарні або подвійні лапки, хоча в HTML вони можуть бути не в лапках (але не в XHTML). [73][74] Значення атрибутів без лапок вважаються небезпечними. [75] На відміну від атрибутів пари ім'я-значення, є атрибути, які впливають на елементи лише через їх присутність у початковій розмітці елемента [6], наприклад, атрибут `ismap` елемента `img`. [76]

Існує кілька загальних атрибутів, які можуть з'являтися в багатьох елементах:

1. Атрибут `id` забезпечує унікальний ідентифікатор елемента для всього документа. Це використовується для ідентифікації елемента, щоб таблиці стилів могли змінювати його презентаційні властивості, а сценарії могли змінювати, анімувати або видаляти його вміст або презентацію. Доданий до URL-адреси сторінки, він надає глобальний

унікальний ідентифікатор для елемента, як правило, підрозділу сторінки. Наприклад, ідентифікатор «Атрибути» в <https://en.wikipedia.org/wiki/HTML#Attributes>.

2. Атрибут `class` забезпечує спосіб класифікації подібних елементів. Це можна використовувати для семантичних або презентаційних цілей. Наприклад, документ HTML може семантично використовувати позначення `<class="notation">`, щоб вказати, що всі елементи з цим значенням класу підпорядковані основному тексту документа. У презентації такі елементи можуть бути зібрані разом і представлені у вигляді виносок на сторінці замість того, щоб з'являтися там, де вони зустрічаються у джерелі HTML. Атрибути класу використовуються семантично в мікроформатах. Можна вказати кілька значень класів; наприклад, `<class="notation important">` поміщає елемент як до нотації, так і до важливих класів.

3. Автор може використовувати атрибут `style` для призначення властивостей презентації певному елементу. Вважається кращою практикою використовувати ідентифікатор елемента або атрибути класу для вибору елемента з таблиці стилів, хоча іноді це може бути занадто громіздким для простого, конкретного або спеціального стилю.

4. Атрибут `title` використовується для додавання підтекстового пояснення до елемента. У більшості браузерів цей атрибут відображається як підказка.

5. Атрибут `lang` визначає природну мову вмісту елемента, яка може відрізнитися від мови решти документа. Наприклад, в англomовному документі:

```
<p>Ну що ж, <span lang="fr">c'est la vie</span>, як кажуть у Франції.</p>
```

Семантичний HTML – це спосіб написання HTML, який підкреслює важливість закодованої інформації щодо її представлення (зовнішнього вигляду). HTML із самого початку включав семантичну розмітку[83], а також презентаційну розмітку, таку як теги ``, `<i>` і `<center>`. Існують також

семантично нейтральні теги span і div. З кінця 1990-х років, коли каскадні таблиці стилів почали працювати в більшості браузерів, веб-авторам було рекомендовано уникати використання HTML-розмітки презентації для розділення презентації та вмісту. [84]

У дискусії про семантичну мережу 2001 року Тім Бернерс-Лі та інші навели приклад того, як інтелектуальні програмні агенти можуть одного дня автоматично сканувати Інтернет і знаходити, фільтрувати та співвідносити раніше нерелевантні опубліковані факти з користувачами. [85] Такі проксі не є рідкістю навіть зараз, але деякі ідеї Web 2.0, інциденти та сайти порівняння цін можуть бути близькими. Основна відмінність цих гібридів веб-додатків від семантичних проксі Бернерса-Лі полягає в тому, що поточні агрегації та гібриди інформації часто розробляються веб-розробниками, які вже знають веб-розташування та семантику API потрібних даних. Чеканка, порівняння та відповідність.

Веб-сканер або пошукова система є важливим типом веб-проксі, який автоматично сканує та читає веб-сторінки, не знаючи, що він може знайти. Ці програмні агенти покладаються на семантичну чіткість веб-сторінок, які вони знаходять, оскільки вони використовують різні методи та алгоритми для читання та індексації мільйонів веб-сторінок щодня, а також надають користувачам Інтернету інструменти пошуку, які значно знизять корисність у всьому світі. Інтернет.

Щоб павуки пошукових систем розуміли важливість фрагментів тексту, які вони знаходять у HTML-документах, а також для тих, хто створює діаграми сітки та інші гібриди, а також у більш автоматизованих проксі, які вони розробляють, існуюча семантична структура в HTML повинна використовуватися широко і рівномірно. Використовується для визначення значення опублікованого тексту. [86]

Поточні вказівки щодо HTML і XHTML не підтримують теги презентації. Більшість функцій презентації з попередніх версій HTML більше не дозволені, оскільки вони призводять до зниження доступності, збільшення витрат на обслуговування сайту та збільшення розміру документа. [87]

Хороший семантичний HTML також покращує доступність веб-документів (див. також Інструкції щодо доступності веб-вмісту). Наприклад, коли програми зчитування з екрана або аудіобраузери можуть правильно структурувати документ, вони не витратять час на читання повторюваної або невідповідної інформації для користувачів із вадами зору, якщо вони правильно розмічені.

Каскадні таблиці стилів (CSS) — це мова таблиць стилів, яка використовується для опису подання документів, написаних мовами розмітки, такими як HTML. [1] CSS є наріжним каменем всесвітньої павутини разом з HTML і JavaScript. [2] CSS призначений для розділення презентації та вмісту, включаючи макет, кольори та шрифти. [3] Це поділ покращує зручність використання вмісту; забезпечує більшу гнучкість та контроль у специфікації функцій презентації; дозволяє кільком веб-сторінкам спільно використовувати форматування, вказуючи відповідний CSS в окремих файлах .css, зменшуючи складність і повторюваність структурованого вмісту. А також увімкніть кешування файлів .css, щоб покращити швидкість завантаження сторінки та спільний доступ до файлів.

Розділення формату та вмісту також дозволяє відображати ту саму сторінку розмітки в різних стилях для різних методів візуалізації, наприклад, на екрані, у друкованому вигляді, мовленні (через мовний браузер або програму зчитування з екрана) і шрифт Брайля. Тактильні пристрої. CSS також має альтернативні правила форматування під час доступу до вмісту на мобільних пристроях. [4]

Конкатенація імен походить від заданої схеми пріоритету, щоб визначити, яке правило стилю застосовується, якщо декілька правил відповідають певному елементу. Ця каскадна схема пріоритетів є передбачуваною.

Консорціум World Wide Web Consortium (W3C) підтримує специфікацію CSS. Текст/css типу Інтернет-медіа (тип MIME) зареєстровано для використання з CSS відповідно до RFC 2318 (березень 1998 р.). W3C надає безкоштовну службу перевірки CSS для документації CSS. [5]

Крім HTML, інші мови розмітки підтримують використання CSS, включаючи XHTML, звичайний XML, SVG і XUL.

блок декларацій

Рекламний блок складається зі списку декларацій у дужках. Кожне оголошення містить атрибут, двокрапку (:) і значення. Якщо в блоці є кілька декларацій, для розділення кожної декларації потрібно вставити крапку з комою (;). Після останнього (або окремого) оголошення можна використовувати додаткову крапку з комою. [9]

Властивості вказані в стандарті CSS. Кожна властивість має набір можливих значень. Деякі властивості можуть впливати на будь-який тип елемента, тоді як інші застосовуються лише до певних груп елементів. [10]
[11]

Значенням може бути ключове слово, як-от "center" або "inherit", або батьківський елемент із числовим значенням, як-от 200px (200px), 50vw (50% ширини області перегляду) або 80% (80% ширини). Значення кольору можуть використовувати ключові слова (наприклад, «червоний»), шістнадцяткові значення (наприклад, #FF0000, також скорочено як #F00), значення RGB від 0 до 255 (наприклад, rgb(255, 0, 0)), значення RGBA, що представляють і колір, і альфа-прозорість (наприклад, rgba(255, 0, 0, 0.8)), або значення HSL або HSLA (наприклад, hsl(000, 100%, 50%), hsla(000, 100%, 50%, 80%)). [12]

одиниця довжини

Ненульове значення, що представляє лінійну міру, має містити одиницю довжини, або алфавітний код, або абревіатуру, наприклад 200px або 50vw, або знак відсотка, наприклад 80%. Деякі одиниці - сантиметри (см); дюйми (дюйми); міліметри (мм); pc (піца); і pt (точка) - абсолютні, тобто розмір копії не залежить від структури сторінки; інші - em (em); ex (ex) і px (піксель) [потрібне уточнення] - є відносними, що означає, що такі фактори, як розмір шрифту батьківського елемента, можуть впливати на скопійовані вимірювання. Ці вісім одиниць були властивістю CSS 1 [13] і були збережені у всіх наступних версіях. Запропонований модуль значень і одиниць CSS

рівня 3, якщо прийняти його як рекомендацію W3C, надасть сім додаткових одиниць довжини: ch; ask; rem; vh; max; min; і маса [14]

використання

До CSS майже всі атрибути, представлені документами HTML, містилися в тегах HTML. Усі кольори шрифту, стилі фону, вирівнювання елементів, межі та розміри мають бути чітко описані в HTML і часто повторювані. CSS дозволяє авторам перемістити більшу частину цієї інформації в інший файл, таблицю стилів, що призводить до спрощення HTML.

Наприклад, заголовки (елемент h1), підзаголовки (h2), підзаголовки (h3) тощо визначаються структурно за допомогою HTML. Вибір шрифту, розміру, кольору та акценту для цих елементів є репрезентативним у друку та на екрані.

До CSS автори документів, які хотіли призначити такі типографічні характеристики всім заголовкам h2, повинні були копіювати розмітку представлення HTML для кожного входження такого типу заголовка. Це робить документи складнішими, більшими, більш схильними до помилок і їх складніше обслуговувати. CSS дозволяє відокремити презентацію від структури. CSS може визначати кольори, шрифти, вирівнювання тексту, розмір, межі, інтервали, макет та багато інших типографічних характеристик, і може робити це незалежно для макетів екрана та друку. CSS також визначає невізуальні стилі, такі як швидкість читання та акцент на прослуховуванні. Тепер W3C припинив використання всіх тегів представлення HTML. [15]

Наприклад, у попередньому CSS HTML елемент заголовка, визначений червоним кольором, буде записаний так:

```
<h1> <font color="red"> Розділ 1. </font></h1>
```

За допомогою CSS той самий елемент можна закодувати за допомогою атрибутів стилю замість атрибутів презентації HTML:

```
<h1 style="color:red;">Розділ 1. </h1>
```

Переваги цього можуть бути не відразу очевидні, але потужність CSS стає більш очевидною, коли атрибути стилю розташовані у внутрішніх

елементах стилю, а ще краще, у зовнішньому файлі CSS. Наприклад, припустимо, що документ містить елемент стилю:

```
<стиль>
  h1 {
    червоний;
  }
</style>
```

Усі елементи h1 в документі автоматично стануть червоними без будь-якого явного коду. Якщо пізніше автор хоче, щоб елемент h1 був синім, він може зробити це, змінивши елемент стилю на:

```
<стиль>
  h1 {
    колір: синій;
  }
</style>
```

Замість того, щоб ретельно переглядати документацію та змінювати колір кожного окремого елемента h1. Стили також можна помістити у зовнішні файли CSS, як описано нижче, і завантажити за допомогою синтаксису, подібного до наведеного нижче:

```
<link href="path/to/file.css" rel="stylesheet" type="text/css">
```

Це додатково відокремлює стилі від документів HTML і дозволяє змінювати стилі в кількох документах, просто редагуючи спільні зовнішні файли CSS.

MySQL [5] — це система управління реляційними базами даних з відкритим вихідним кодом (RDBMS). [5] [6] Її назва є комбінацією «Му» (дочка співзасновника Майкла Віденіуса [7]) і «SQL» (скорочення від Structured Query Language). Реляційна база даних організовує дані в одну або кілька таблиць даних, у яких можна зв'язати типи даних; ці відносини допомагають структурувати дані. SQL — це мова, яку програмісти

використовують для створення, модифікації та вилучення даних з реляційних баз даних і для контролю доступу користувачів до баз даних. На додаток до реляційних баз даних і SQL, такі бази даних, як MySQL, працюють з операційними системами, щоб інтегрувати реляційні бази даних у сховище комп'ютера, керувати користувачами, забезпечувати доступ до мережі та полегшувати цілісність баз даних і створювати резервні копії.

MySQL є безкоштовним програмним забезпеченням з відкритим вихідним кодом відповідно до умов GNU General Public License і доступне за різними ліцензіями власності. MySQL належить і спонсорується шведською компанією MySQL AB, яку придбала Sun Microsystems (тепер Oracle Corporation). [8] У 2010 році, коли Oracle придбала Sun, Widenius розширив проєкт MySQL з відкритим кодом, щоб створити MariaDB. [9]

MySQL має окремий клієнт, який дозволяє користувачам безпосередньо взаємодіяти з базою даних MySQL за допомогою SQL, але частіше MySQL використовується разом з іншими програмами для реалізації програм, які потребують функціональності реляційної бази даних. MySQL є компонентом стеку веб-додатків LAMP (та інших), що є скороченням від Linux, Apache, MySQL, Perl/PHP/Python. MySQL використовується багатьма веб-додатками баз даних, включаючи Drupal, Joomla, phpBB і WordPress. MySQL також використовується багатьма популярними веб-сайтами, включаючи Facebook, [10] [11] Flickr, [12] MediaWiki, [13] Twitter [14] та YouTube. [15]

Огляд

MySQL написаний на C і C++. Його синтаксичний аналізатор SQL написаний на yacc, але він використовує саморобний lexer. [16] MySQL працює на багатьох системних платформах, включаючи AIX, BSDi, FreeBSD, HP-UX, ArcaOS, eComStation, IBM i, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp , QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos і Tru64. OpenVMS також має порт MySQL. [17]

Серверне програмне забезпечення MySQL і клієнтські бібліотеки використовують дистрибутив з подвійною ліцензією. Вони надаються відповідно до GPL версії 2 або вашої власної ліцензії. [18]

Підтримку можна знайти в офіційному посібнику. [19] Крім того, різноманітні канали та форуми IRC надають безкоштовну підтримку. Oracle пропонує платну підтримку через свій продукт MySQL Enterprise. Вони відрізняються за обсягом і ціною. Крім того, існує багато сторонніх організацій, які надають підтримку та послуги.

MySQL отримав позитивні відгуки, рецензенти відзначили, що він «в середньому працює дуже добре», і що інтерфейс розробника є, а документація (не кажучи вже про реальний відгук через веб-сайт тощо) дуже і дуже гарна. «[20] Він також був протестований як «швидкий, стабільний і справжній багатокористувацький, багатопоточний сервер баз даних SQL».[21]

розгорнути

MySQL можна вручну зібрати та встановити з джерела, але частіше він встановлюється з бінарних пакетів, якщо не потрібні спеціальні налаштування. У більшості дистрибутивів Linux система керування пакетами полегшує завантаження та встановлення MySQL, хоча для налаштування параметрів безпеки та оптимізації зазвичай потрібна додаткова конфігурація.

Хоча MySQL починався як заміна потужнішим власним базам даних, він розвинувся для підтримки більших потреб. Він як і раніше найчастіше використовується в малих і середніх розгортаннях з одним сервером, або як компонент веб-додатка на основі LAMP, або як окремий сервер бази даних. Значна частина привабливості MySQL пов'язана з його відносною простотою та зручністю використання, які надаються екосистемами з відкритим кодом, такими як phpMyAdmin. MySQL середнього класу можна масштабувати, розгортаючи його на більш потужному обладнанні, такому як багатопроцесорні сервери з гігабайтами пам'яті.

Однак існує обмеження щодо того, наскільки продуктивність («масштабування») може бути масштабована на одному сервері, тому більші масштаби вимагають багатосерверного розгортання MySQL

(«масштабування») для підвищення продуктивності та надійності. Типова конфігурація високого класу може включати потужну первинну базу даних, яка обробляє запис даних і реплікує їх у декілька підлеглих баз даних, які обробляють усі читання. [96] Головний пристрій безперервно надсилає події binlog на підключені підпорядковані пристрої, тому в разі збою підпорядкованого пристрою його можна оновити до нового головного, мінімізуючи час простою. Продуктивність можна додатково підвищити, використовуючи memcached для кешування результатів запитів до бази даних у пам'яті або розділяючи базу даних на менші частини. Вони називаються сегментами і можуть бути розподілені між кількома розподіленими кластерами серверів. [97]

РОЗДІЛ 4

ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ

4.1 Аналіз варіантів діяльності

Найпростіша діаграма використання — це уявлення про взаємодію користувача з системою, яка показує взаємозв'язок між користувачем і різними видами використання, в яких бере участь користувач. Діаграми варіантів використання визначають різні типи користувачів системи та різні варіанти використання, і часто супроводжуються діаграмами інших типів. Мета зображується колом або овалом.

У той час як самі варіанти використання можуть детально досліджувати кожну можливість, діаграми варіантів використання можуть допомогти надати огляд системи більш високого рівня. Раніше було сказано, що «схема використання - це принцип вашої системи».

Завдяки своїй спрощеній природі сценарії використання можуть бути хорошим інструментом комунікації для зацікавлених сторін. Ці малюнки намагаються імітувати реальний світ, даючи зацікавленим сторонам уявлення про те, як буде розвиватися система. Сіау і Лі провели дослідження, щоб визначити, чи існують реальні ситуації, в яких використовуються програми, чи вони непотрібні. Було виявлено, що діаграми варіантів використання більш спрощено повідомляють про наміри системи зацікавленим сторонам, і вони були «пояснені більш повно, ніж діаграми класів».

Метою використання діаграм є показати динамічні аспекти системи. Для повного функціонального та технічного представлення системи доступні додаткові схеми та документація. Вони забезпечують спрощене та графічне представлення того, що насправді має робити система.

Елементи:

1. Межа системи - прямокутник з назвою вгорі та еліпсом (прецедентом) всередині. Його часто можна опустити без корисної інформації про корисну інформацію,
2. актор (англ. actor) - стилізований людський персонаж, що представляє набір ролей користувача (широко зрозумілі: люди, зовнішні сутності, класи, інші системи), що взаємодіють з сутністю (системою, підсистемою, класом). Учасників не можна співвідносити один з одним (за винятком відносин обробки/дослідження),
3. Прецедент - еліпс з написом, що вказує на дію системи (можливо, включаючи можливі варіанти), яка привела до результату, який спостерігав учасник. Назва може бути назвою або описом (з точки зору актора) того, що система «робить» (а не «як»). Цей прецедент пов'язаний з безперервною (атомною) сценою - певною послідовністю дій, які ілюструють поведінку. У сценарії актори обмінюються інформацією про систему. Сценарій можна відобразити на схемі прецедентів у відео огляду UML. Кілька різних сценаріїв можуть бути пов'язані з прецедентом

Для аналізу варіантів використання системи на рисунку 3.1 зображено діаграму варіантів використання.

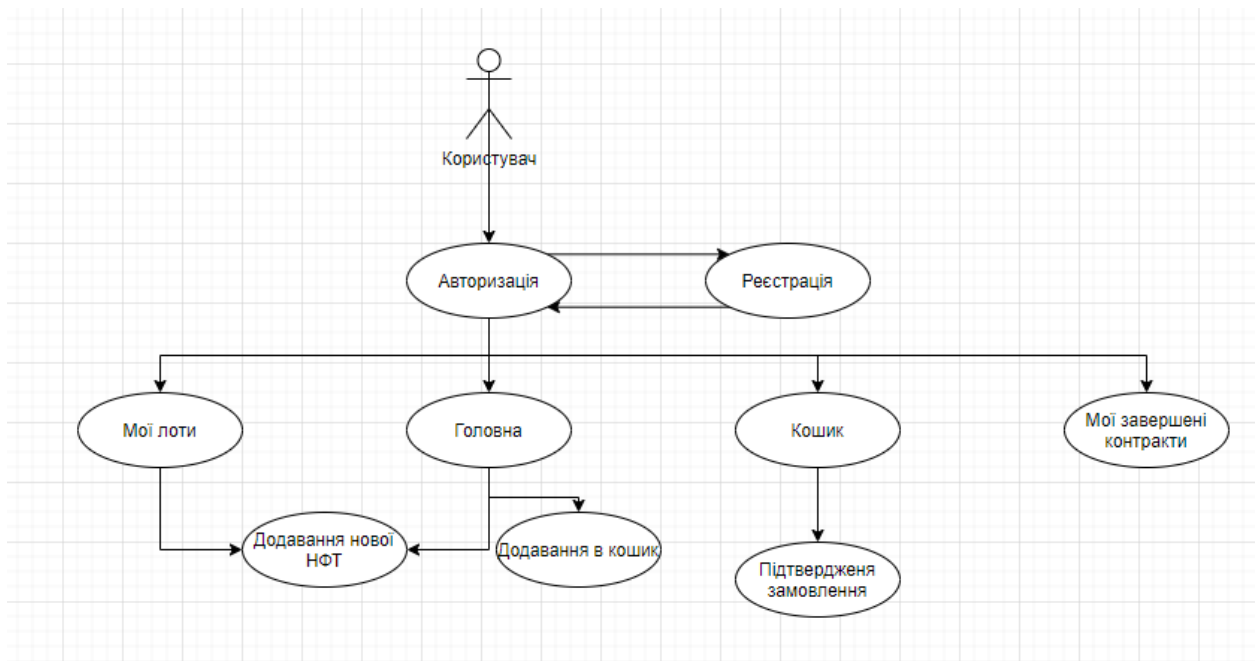


Рис. 3.1. Діаграма варіантів використання

4.2 Проектування компонентного складу

В ході проектування компонентного складу було створено діаграму пакетів, яка відображає внутрішню будову системи на високому рівні, без декомпозиції. Діаграма пакетів зображена на рисунку 3.2.

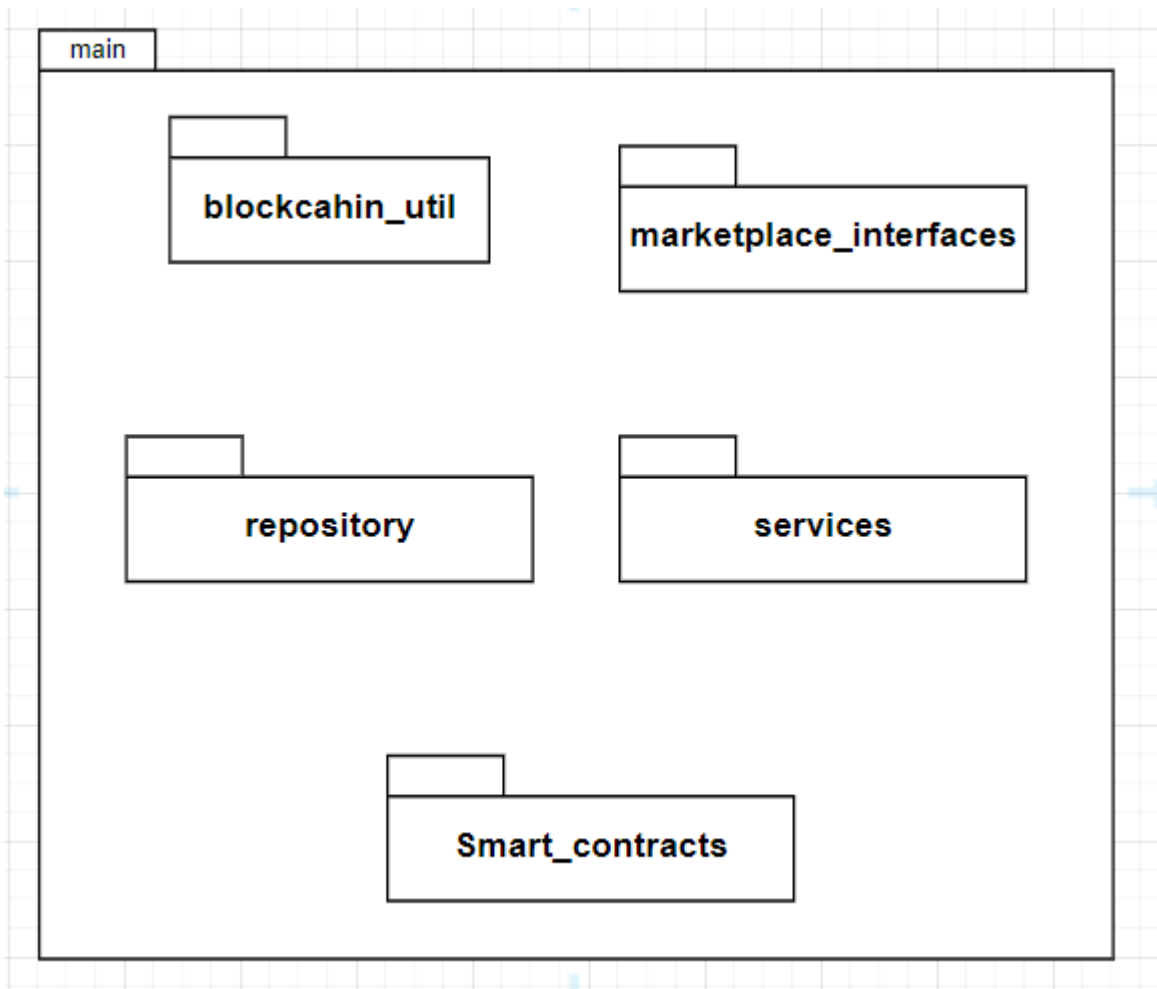


Рис. 3.2. Діаграма пакетів

4.3 Проектування внутрішньої будови

У програмній інженерії діаграма класів уніфікованої мови моделювання (UML) — це статична структурна діаграма, яка описує структуру системи, показуючи класи системи, їх властивості, операції (або методи) і зв'язки між об'єктами.

Діаграми класів є основним будівельним блоком об'єктно-орієнтованого моделювання. Використовується для загального концептуального моделювання структури програми та детального моделювання для перетворення моделей у програмний код. Діаграми класів також можна використовувати для моделювання даних. Класи на діаграмі класів

представляють основні елементи програми, взаємодії та класи, що програмуються.

На схемі класи представлені вікнами, які містять три відділення:

1. Відділення вище містить назву класу. Він друкується жирним шрифтом по центру, а перша літера пишеться великими літерами.
2. Середнє відділення містить властивості класу. Вони вирівняні по лівому краю, а перша літера маленька.
3. Нижнє відділення містить операції, які може виконувати клас. Вони також вирівняні по лівому краю, а перша літера є малою.

При проектуванні системи багато класів ідентифікуються і групуються в схему класів, що допомагає визначити статичні відносини між ними. При детальному моделюванні класи концептуального проектування часто поділяють на підкласи.

Залежності — це семантичні відношення між залежними та незалежними елементами моделі. Він існує між двома елементами, якщо зміна, визначена одним елементом (сервером або цільовим), може спричинити зміну іншого елемента (клієнта чи джерела). Це об'єднання одностороннє. Залежності показані пунктирними лініями з порожніми стрілками, що вказують від клієнтів до постачальників.

Для подальшого опису поведінки системи ці діаграми класів можуть бути доповнені діаграмами станів або автоматами станів UML.

Асоціація являє собою серію посилань. Бінарні асоціації (з двома кінцями) зазвичай представляють у вигляді рядків. Асоціація може пов'язувати будь-яку кількість класів. Асоціація з трьома ланками називається потрійною асоціацією. Асоціації можна назвати, а кінці асоціації можна прикрасити іменами ролей, індикаторами власності, множинністю, видимістю та іншими атрибутами.

Є чотири різні типи асоціацій: двонаправлені, односпрямовані, агрегатні (включаючи складені агрегати) і рефлексивні. Найбільш поширеними є двонаправлені та односпрямовані асоціації.

Наприклад, клас польоту асоціюється з класом літака двонаправлено. Асоціація представляє статичне відношення, яке ділиться між об'єктами двох класів.

Агрегація є варіантом взаємозв'язку "має"; агрегація є більш конкретною, ніж асоціація. Це асоціація, яка представляє частково цілі або часткові стосунки. Як показано на зображенні, професор "має" клас для викладання. Як тип асоціації, агрегація може бути названа та мати ті самі прикраси, що і асоціація. Однак агрегація не може включати більше двох класів; це має бути бінарна асоціація. Крім того, навряд чи існує різниця між агрегаціями та асоціаціями під час реалізації, і діаграма може взагалі пропустити відносини агрегування. [7]

Агрегація може відбуватися, коли клас є колекцією або контейнером інших класів, але вміщені класи не мають сильної залежності життєвого циклу від контейнера. Вміст контейнера все ще існує, коли контейнер знищений.

В UML він графічно представлений у вигляді порожнистої форми ромба на вміщуючому класі одним рядком, що зв'язує його із вміщеним класом. Сукупність - це семантично розширений об'єкт, який у багатьох операціях трактується як одиниця, хоча фізично він складається з декількох менших об'єктів.

Приклад: Бібліотека та студенти. Тут студент може існувати без бібліотеки, зв'язок між студентом і бібліотекою є агрегацією.

Це вказує на те, що один із двох пов'язаних класів (підклас) вважається спеціалізованою формою іншого (супер тип), а суперклас - узагальненням підкласу. На практиці це означає, що будь-який екземпляр підтипу є також екземпляром суперкласу. Зразкове дерево узагальнень цієї форми зустрічається в біологічній класифікації: людина - це підклас маймуни, який є підкласом ссавців тощо. Зв'язок найлегше зрозуміти за допомогою фрази „А - це В” (людина - це ссавець, ссавець - тварина).

Графічне представлення UML узагальнення - це форма порожнистого трикутника на кінці суперкласу рядка (або дерева рядків), що зв'язує його з одним або кількома підтипами.

Відносини узагальнення також відомі як спадщина або відносини "є".

Суперклас (базовий клас) у відносинах узагальнення також відомий як "батьківський", суперклас, базовий клас або базовий тип.

Підтип у відносинах спеціалізації також відомий як "дочірній", підклас, похідний клас, похідний тип, клас успадкування або тип успадкування.

Зверніть увагу, що ці стосунки нічим не схожі на біологічні стосунки батьків та дітей: використання цих термінів надзвичайно поширене, але може ввести в оману.

A - це тип B

Наприклад, "дуб - це тип дерева", "автомобіль - це тип транспортного засобу"

Узагальнення може бути показано лише на діаграмах класів та на діаграмах використання.

При моделюванні UML взаємозв'язок реалізації - це взаємозв'язок між двома елементами моделі, в яких один елемент моделі (клієнт) реалізує (реалізує або виконує) поведінку, яку вказує інший елемент моделі (постачальник).

Графічне представлення UML реалізації - це порожниста форма трикутника на кінці інтерфейсу штрихової лінії (або дерева рядків), яка з'єднує її з одним або кількома реалізаторами. Проста головка стрілки використовується на кінці інтерфейсу штрихової лінії, що з'єднує її з користувачами. У діаграмах компонентів використовується графічна умова «м'яч і сокет» (реалізатори виставляють кульку або льодяник, тоді як користувачі показують сокет). Реалізації можна показати лише на діаграмах класів або компонентів. Реалізація - це взаємозв'язок між класами, інтерфейсами, компонентами та пакетами, що з'єднує елемент клієнта з елементом постачальника. Зв'язок реалізації між класами / компонентами та

інтерфейсами показує, що клас / компонент реалізує операції, пропоновані інтерфейсом.

Залежність - це слабша форма зв'язку, яка вказує на те, що один клас залежить від іншого, оскільки він використовує його в певний момент часу. Один клас залежить від іншого, якщо незалежний клас є змінною параметра або локальною змінною методу залежного класу. Це відрізняється від асоціації, де атрибут залежного класу є екземпляром незалежного класу. Іноді відносини між двома класами дуже слабкі. Вони взагалі не реалізовані зі змінними-членами. Швидше вони можуть бути реалізовані як аргументи функції-члена.

інший. Ці відносини зазвичай описуються як "А має В" (у матері-кота є кошенята, у кошенят - мати-кішка).

Представлення UML асоціації - це лінія, що з'єднує два пов'язані класи. На кожному кінці рядка є додаткові позначення. Наприклад, ми можемо вказати, використовуючи наконечник стрілки, що загострений кінець видно з хвоста стрілки. Ми можемо вказати власність шляхом розміщення кульки, ролі, яку відіграють елементи цього кінця, вказавши ім'я ролі та множинність екземплярів цієї сутності (діапазон кількості об'єктів, які беруть участь в асоціації з точки зору іншого кінця).

Класи сутності моделюють довгоживучу інформацію, якою обробляє система, а іноді і поведінку, пов'язану з цією інформацією. Їх не слід ідентифікувати як таблиці баз даних чи інших сховищ даних.

Вони намальовані як кола з короткою лінією, прикріпленою до нижньої частини кола. Як варіант, їх можна намалювати як звичайні класи із позначенням стереотипу «сутність» над назвою класу.

В якості проектування внутрішньої будови було побудовано діаграму класів, яка зображена на рисунку 3.3.

При завантаженні NFT вони відображаються на головній сторінці, з умовою того що відображаються всі NFT, окрім тих, які додав поточний користувач, оскільки головна сторінка — це сторінка каталогу. (рис. 3.5.)

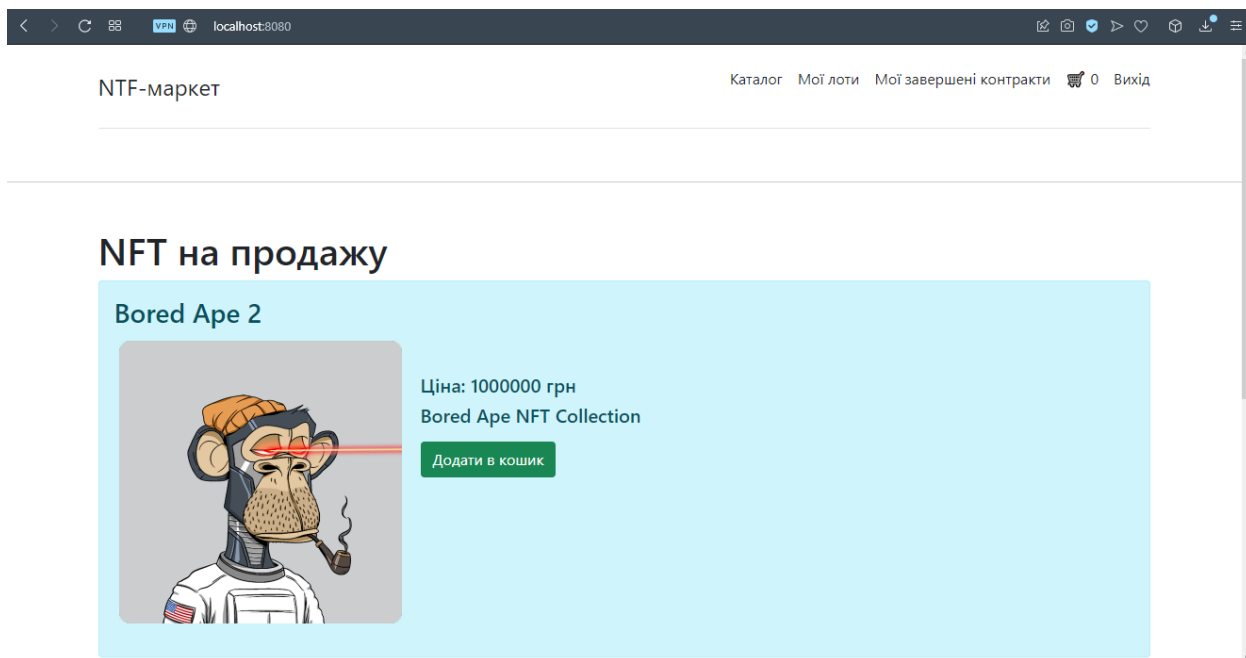


Рис. 3.5. Головна сторінка з NFT

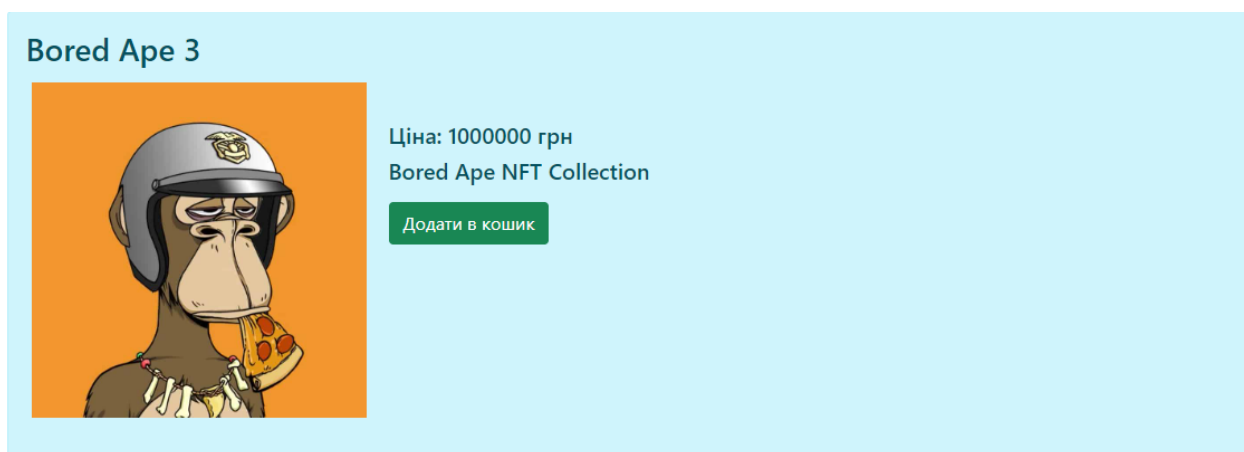


Рис. 3.6. Лот з NFT

На головній сторінці і сторінці власних лотів є кнопка переходу то форми створення нової NFT. (рис. 3.7)

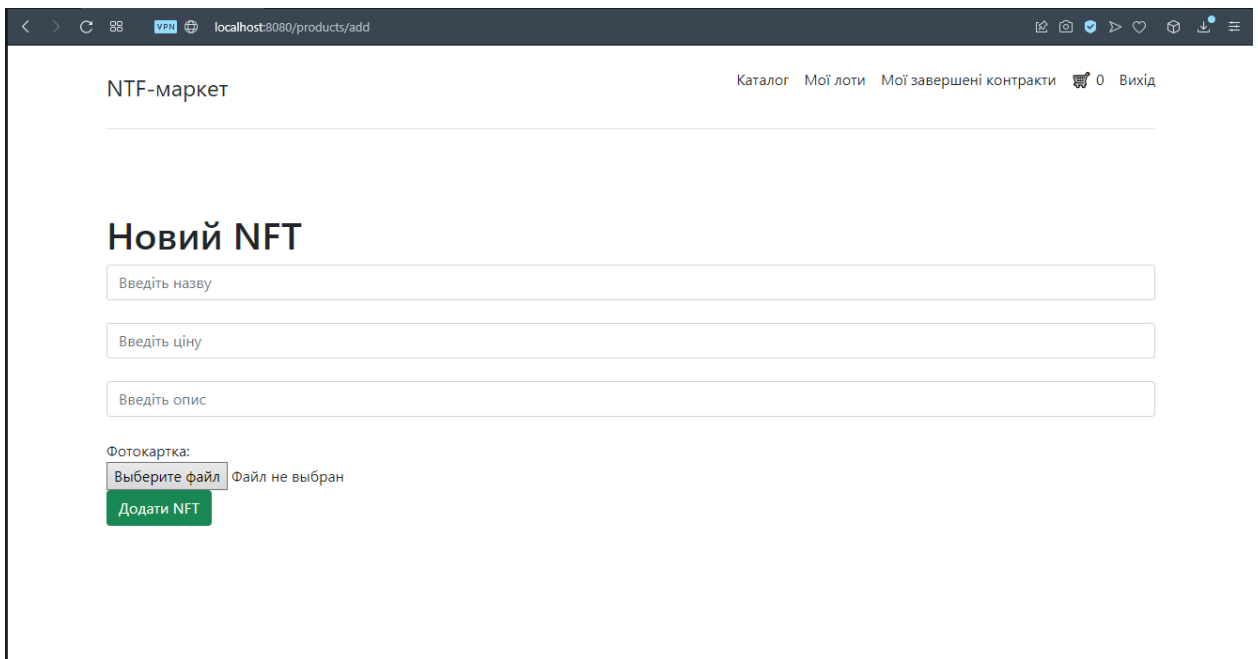


Рис. 3.7. Виставлення NFT на продаж

Окрім цього, користувач має можливість переглянути ті NFT, які на продаж виставив саме він (рис. 3.8).

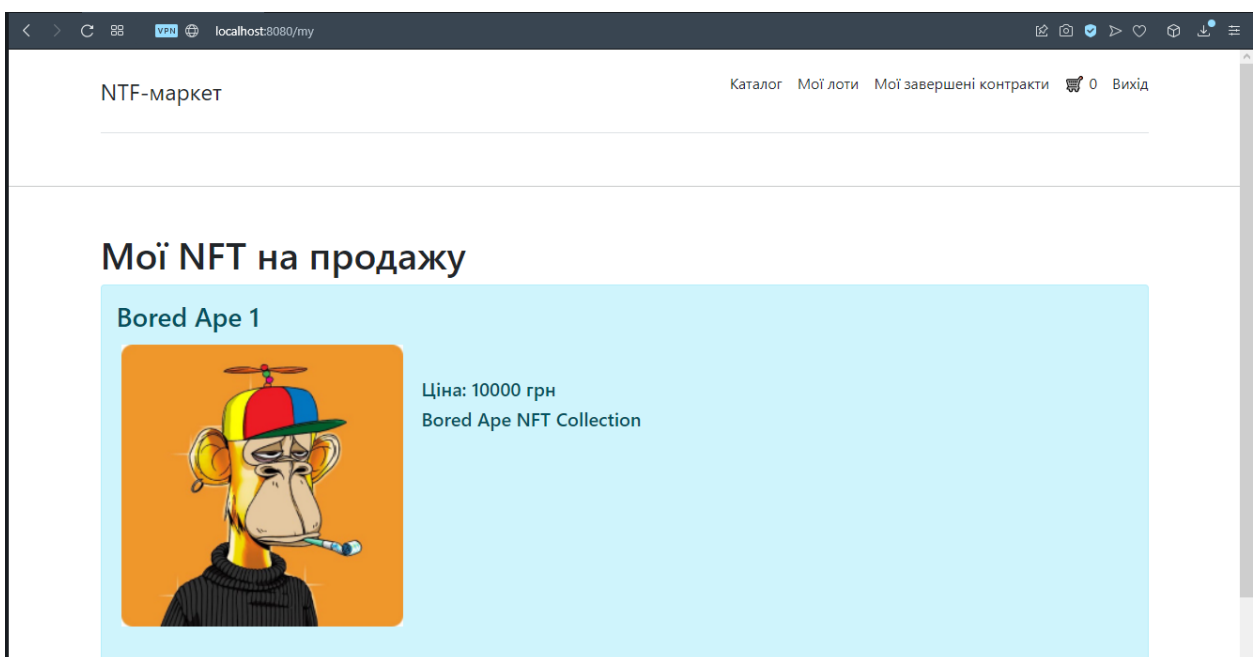


Рис. 3.8. Сторінка власних лотів

4.5 Тестування

Тестування програмного забезпечення – це акт перевірки артефактів і поведінки програмного забезпечення, що тестується, шляхом перевірки та верифікації. Тестування програмного забезпечення також може забезпечити об'єктивне, незалежне уявлення про програмне забезпечення, щоб дозволити бізнесу оцінити та зрозуміти ризики впровадження програмного забезпечення. Методи тестування включають, але не обов'язково обмежуються:

- аналіз вимог до продукту щодо повноти та правильності в різних контекстах, таких як галузева перспектива, бізнес-перспектива, доцільність та життєздатність впровадження, зручність використання, продуктивність, безпека, міркування інфраструктури тощо.
- перегляд архітектури продукту та загального дизайну продукту
- робота з розробниками продуктів над удосконаленням техніки кодування, шаблонів проектування, тестів, які можна написати як частину коду на основі різних методів, таких як граничні умови тощо.
- виконання програми або програми з метою перевірки поведінки
- перевірка інфраструктури розгортання та пов'язаних скриптів та автоматизації
- брати участь у виробничій діяльності, використовуючи методи моніторингу та спостереження

Тестування програмного забезпечення може надати користувачам або спонсорам об'єктивну, незалежну інформацію про якість програмного забезпечення та ризик його збою.

Несправності та збої

Помилки програмного забезпечення виникають через наступний процес: Програміст робить помилку (помилку), що призводить до помилки (дефекту, помилки) у вихідному коді програмного забезпечення. Якщо ця помилка виконана, у певних ситуаціях система дасть неправильні результати, що призведе до збою.

Не всі несправності обов'язково призведуть до збоїв. Наприклад, помилки в мертвому коді ніколи не призведуть до збоїв. Несправність, яка не виявила збоїв, може призвести до збою при зміні середовища. Приклади цих змін у середовищі включають програмне забезпечення, яке запускається на новій апаратній платформі комп'ютера, зміни вихідних даних або взаємодію з іншим програмним забезпеченням. Одна несправність може призвести до широкого спектру симптомів відмови.

Не всі помилки програмного забезпечення викликані помилками кодування. Одним із поширених джерел дорогих дефектів є прогалини у вимогах, тобто нерозпізнані вимоги, які призводять до помилок, пропущених розробником програми. Прогалини вимог часто можуть бути нефункціональними вимогами, такими як тестованість, масштабованість, ремонтпридатність, продуктивність та інші вимоги. безпеки.

Статичне, динамічне та пасивне тестування

У тестуванні програмного забезпечення існує багато підходів. Огляди, покрокові інструкції або перевірки називаються статичним тестуванням, тоді як виконання запрограмованого коду з заданим набором тестових випадків називається динамічним тестуванням.

Статичне тестування часто є неявним, як-от коректура, а також коли інструменти програмування/текстові редактори перевіряють структуру вихідного коду, а компілятори (попередні компілятори) перевіряють синтаксис і потік даних як статичний аналіз програми. Динамічне тестування відбувається під час запуску самої програми. Динамічне тестування може розпочатися до того, як програма буде на 100% завершена, щоб перевірити окремі розділи коду та застосувати до дискретних функцій або модулів. Типовими методами для них є або використання заглушок/драйверів, або виконання з середовища налагодження.

Статичне тестування передбачає перевірку, тоді як динамічне також передбачає перевірку.

Пасивне тестування означає перевірку поведінки системи без будь-якої взаємодії з програмним продуктом. На відміну від активного тестування, тестувальники не надають жодних тестових даних, а переглядають системні журнали та трасування. Вони шукають моделі та специфічну поведінку, щоб приймати якісь рішення. Це пов'язано з перевіркою часу виконання в автономному режимі та аналізом журналу.

Дослідницький підхід

Дослідницьке тестування — це підхід до тестування програмного забезпечення, який коротко описується як одночасне навчання, розробка тесту та виконання тесту. Джем Канер, який ввів цей термін у 1984 році,² визначає дослідницьке тестування як «стиль тестування програмного забезпечення, який підкреслює особисту свободу та відповідальність окремого тестувальника за постійну оптимізацію якості своєї роботи, розглядаючи тест- пов'язане навчання, дизайн тесту, виконання тесту та інтерпретація результатів тесту як взаємодопоміжні дії, які виконуються паралельно протягом усього проекту».

«Коробковий» підхід

Методи тестування програмного забезпечення традиційно поділяються на тестування білого та чорного ящиків. Ці два підходи використовуються для опису точки зору, яку дотримується тестувальник під час розробки тестових випадків. До методології тестування програмного забезпечення також можна застосувати гібридний підхід, який називається тестуванням сірого ящика. Оскільки концепція тестування сірого ящика, яка розробляє тести на основі конкретних елементів дизайну, стає все більш популярною, це «довільне розходження» між тестуванням чорного та білого ящиків дещо зникло

Тестування додатку

| № | Тест-кейс | Очікуваний результат | Отриманий результат |
|---|--|--|--|
| 1 | Створення запиту з пустими полями | При спробі створення запиту з пустими полями система повідомляє користувача про те, що ці поля необхідно заповнити. | При спробі створення запиту з пустими полями система повідомляє користувача про те, що ці поля необхідно заповнити. |
| 2 | Спроба переглянути існуючі ставки при відсутності ставок | При спробі переглянути існуючі ставки при відсутності ставок система повідомляє користувача про те, що в нього відсутні ставки | При спробі переглянути існуючі ставки при відсутності ставок система повідомляє користувача про те, що в нього відсутні ставки |
| 3 | Спроба підтвердження замовлення при порожньому кошику | При спробі підтвердження замовлення при порожньому кошику система повідомляє, що кошик порожній | При спробі підтвердження замовлення при порожньому кошику система повідомляє, що кошик порожній |

Результати тестування показують, що система повністю функціональна і готова до використання в реальних умовах.

ВИСНОВКИ

Мета роботи полягала в створенні власної програмної реалізації NFT-маркетплейсу.

Для досягнення поставленої мети було виконано наступні завдання:

1. провести аналіз поняття NFT;
2. провести аналіз поняття маркетплейсу;
3. проаналізувати поняття веб-сайтів та веб-розробки;
4. розглянути існуючі рішення;
5. провести огляд мови програмування;
6. провести огляд середовища розробки;
7. провести огляд додаткового інструментарію;
8. провести аналіз варіантів діяльності;
9. провести проектування компонентного складу системи;
10. провести проектування внутрішньої будови системи;
11. розробити графічний інтерфейс користувача.

Завдяки чіткому виконанню завдання в результаті було отримано повноцінну функціонуючу систему, придатну до використання в реальних умовах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "NFTs Hit \$17B In Trading In 2021, Up 21,000%". www.pymnts.com. March 10, 2022. Retrieved May 5, 2022.
2. Genç, Ekin (October 5, 2021). "Investors Spent Millions on 'Evolved Apes' NFTs. Then They Got Scammed". Vice Media. Retrieved November 9, 2021.
3. Hawkins, John (January 13, 2022). "NFTs, an overblown speculative bubble inflated by pop culture and crypto mania". The Conversation. Retrieved May 7, 2022.
4. Wilson, Kathleen Bridget; Karg, Adam; Ghaderi, Hadi (October 2021). "Prospecting non-fungible tokens in the digital economy: Stakeholders and ecosystem, risk and opportunity". Business Horizons: S0007681321002019. doi:10.1016/j.bushor.2021.10.007. S2CID 240241342.
5. Mayor, Daniel (April 4, 2022). "NFTs and the Legitimizing Power of Copyright".
6. Dean, Sam (March 11, 2021). "\$69 million for digital art? The NFT craze, explained". Los Angeles Times. Retrieved March 12, 2021.
7. Kastrenakes, Jacob (March 11, 2021). "Beeple sold an NFT for \$69 million". The Verge. Archived from the original on March 21, 2021. Retrieved March 21, 2021.
8. Mendis, Dinusha (August 24, 2021). "When you buy an NFT, you don't completely own it – here's why". The Conversation. Retrieved November 23, 2021.
9. Gault, Matthew (November 3, 2021). "What the Hell Is 'Right-Clicker Mentality'?". Vice. Retrieved November 3, 2021.
10. "WTF Is an NFT, Anyway? And Should I Care?". Wired. ISSN 1059-1028. Retrieved March 13, 2021.

11. Boscovic, Dragan. "How nonfungible tokens work and where they get their value – a cryptocurrency expert explains NFTs". The Conversation. Retrieved April 8, 2021.
12. Kastrenakes, Jacob (March 25, 2021). "Your Million-Dollar NFT Can Break Tomorrow If You're Not Careful". The Verge. Retrieved March 31, 2021.
13. Gallagher, Jacob (March 15, 2021). "NFTs Are the Biggest Internet Craze. Do They Work for Sneakers?". The Wall Street Journal. ISSN 0099-9660. Retrieved June 26, 2021.
14. Thaddeus-Johns, Josie (March 11, 2021). "What Are NFTs, Anyway? One Just Sold for \$69 Million". The New York Times. ISSN 0362-4331. Retrieved June 26, 2021.
15. Salmon, Felix (March 12, 2021). "How to exhibit your very own \$69 million Beeple". Axios. Retrieved March 13, 2021.
16. Clark, Mitchell (March 11, 2021). "NFTs, explained". The Verge. Retrieved March 11, 2021.
17. "NFT blockchain drives surge in digital art auctions". BBC. March 3, 2021. Retrieved March 12, 2021.
18. Majocha, Courtney. "Memes for Sale? Making sense of NFTs". Harvard Law Today. Retrieved June 26, 2021.
19. Cascone, Sarah (May 7, 2021). "Sotheby's Is Selling the First NFT Ever Minted – and Bidding Starts at \$100". Artnet News. Retrieved November 12, 2021.
20. Dash, Anil (April 2, 2021). "NFTs Weren't Supposed to End Like This". The Atlantic. Retrieved May 5, 2021.
21. Ostroff, Caitlin (May 8, 2021). "The NFT Origin Story, Starring Digital Cats". The Wall Street Journal. ISSN 0099-9660. Retrieved December 12, 2021.
22. "The Cult of CryptoPunks". TechCrunch. Retrieved May 5, 2021.
23. Entriken, William; Shirley, Dieter; Evans, Jacob; Natassia, Sachs (January 24, 2018). "EIP-721: Non-Fungible Token Standard". Ethereum Improvement Proposals. Retrieved October 28, 2021.

24. Urbach, Nils (December 13, 2019). "NFTs in Practice – Non-Fungible Tokens as Core Component of a Blockchain-based Event Ticketing Application" (PDF). Fraunhofer Research Center, Finance and Information Management. Retrieved October 28, 2021.
25. "Should You Buy a Bitcoin-Inspired Image of Lindsay Lohan?". Bloomberg.com. March 8, 2021. Retrieved April 7, 2021.
26. Abbruzzese, Jason (June 16, 2017). "This ethereum-based project could change how we think about digital art". Mashable. Retrieved April 6, 2021.
27. "CryptoKitties craze slows down transactions on Ethereum". BBC News. December 5, 2017. Retrieved May 8, 2022.
28. "The NFT Market Tripled Last Year, and It's Gaining Even More Momentum in 2021". Morning Brew. Retrieved April 8, 2021.
29. "NFTs Are Shaking Up the Art World – But They Could Change So Much More". Time. Retrieved April 6, 2021.
30. "U.S. NFT trademarks applications skyrocketed 400x in 2021 with 15 registrations daily in 2022". February 16, 2022.
31. "U.S. NFT trademarks applications skyrocketed 400x in 2021 with 15 registrations daily in 2022". February 16, 2022.
32. "NFT + Metaverse Trademark Tote Board".
33. Howcroft, Elizabeth (March 17, 2021). "Explainer: NFTs are hot. So what are they?". Reuters. Retrieved April 6, 2021.
34. Vigna, Paul (May 3, 2022). "NFT Sales Are Flatlining". The Wall Street Journal. Retrieved May 5, 2022.
35. "NFTs' Nifty Copyright Issues - Intellectual Property - Canada". www.mondaq.com.
36. Patterson, Dan (March 4, 2021). "Blockchain company buys and burns Banksy artwork to turn it into a digital original". CBS News. Retrieved March 17, 2021.
37. "Pak Breaks Record for Most Expensive NFT Sale". HYPEBEAST. December 7, 2021. Retrieved January 16, 2022.

38. Thaddeus-Johns, Josie (March 11, 2021). "What Are NFTs, Anyway? One Just Sold for \$69 Million". The New York Times. ISSN 0362-4331. Retrieved April 14, 2021.
39. Sugiura, Eri (October 13, 2021). "NFTs turn Japan's manga and anime into genuine art". Financial Times. Retrieved November 4, 2021.
40. "Banksy art burned, destroyed and sold as token in 'money-making stunt'". BBC News. March 9, 2021. Retrieved October 13, 2021.
41. Iscoe, Adam (May 8, 2021). "Burnt Banksy's Inflammatory N.F.T. Not-Art". The New Yorker. Retrieved November 18, 2021.
42. Lu, Fei (January 6, 2022). "Does NFT Art Have A Place In The Museum In 2022?". Jing Culture and Commerce. Retrieved January 6, 2022.
43. "Why many art collectors are staying away from the NFT gold rush". The Independent. April 30, 2021. Retrieved January 27, 2022.
44. Valeonti, Foteini; Bikakis, Antonis; Terras, Melissa; Speed, Chris; Hudson-Smith, Andrew; Chalkias, Konstantinos (January 2021). "Crypto Collectibles, Museum Funding and OpenGLAM: Challenges, Opportunities and the Potential of Non-Fungible Tokens (NFTs)". Applied Sciences. 11 (21): 9931. doi:10.3390/app11219931.
45. "Natively Digital: A Curated NFT Sale". sothebys.com.
46. "Beeple sold an NFT for \$69 million". theverge.com. March 11, 2021.
47. "Web3's early promise for artists tainted by rampant stolen works and likenesses". TechCrunch. Retrieved January 30, 2022.
48. Quiroz-Gutierrez, Marco (March 22, 2021). "NFTs Are Spurring a Digital Land Grab – in Videogame Worlds". The Wall Street Journal. Archived from the original on March 22, 2021. Retrieved March 23, 2021.
49. Alexander, Cristina (October 15, 2021). "Is Heroes & Empires free to play?". Gamepur. Gamurs. Archived from the original on November 11, 2021.
50. Tepper, Fitz (March 21, 2018). "CryptoKitties raises \$12M from Andreessen Horowitz and Union Square Ventures". TechCrunch. Retrieved April 7, 2021.

51. Cheng, Evelyn (December 6, 2017). "Meet CryptoKitties, the \$100,000 digital beanie babies epitomizing the cryptocurrency mania". CNBC. Retrieved April 7, 2021.
52. Entriken, William (June 22, 2018). "Move EIP 721 to Final (#1170) ethereum/EIPs@b015a86". GitHub. Retrieved November 7, 2021.
53. Jiang, Sisi (October 15, 2021). "Good Riddance: Steam Bans Games That Feature Crypto And NFTs". Kotaku. Archived from the original on October 15, 2021.
54. "Ubisoft's NFT Announcement Has Been Intensely Disliked". Kotaku. Retrieved December 23, 2021.
55. GameCentral (December 8, 2021). "Ubisoft unlist Quartz NFT announcement video as it gets 16K dislikes". Metro. Retrieved December 23, 2021.
56. "Ubisoft Developers Confused, Upset Over NFT Plans". Game Rant. December 15, 2021. Retrieved December 23, 2021.
57. "Ubisoft Quartz: Some developers of the company didn't like entering the world of NFTs". Pledge Times. December 15, 2021. Retrieved December 23, 2021.
58. Nightingale, Ed (December 16, 2021). "French trade union criticises Ubisoft Quartz as "a useless, costly, ecologically mortifying tech"". Eurogamer. Retrieved December 23, 2021.
59. Jeffrey, Cal. "Game Developers Conference report: most developers frown on blockchain games". TechSpot. Retrieved January 21, 2022.
60. Peters, Jay (January 20, 2022). "Many game developers hate NFTs, too". The Verge. Retrieved January 26, 2022.
61. Thomas, Dana (October 4, 2021). "Dolce & Gabbana Just Set a \$6 Million Record for Fashion NFTs". The New York Times. ISSN 0362-4331. Retrieved January 20, 2022.
62. Lee, Isabelle (November 17, 2021). "Luxury NFTs could become a \$56 billion market by 2030 and could see 'dramatically' increased demand thanks to the metaverse, Morgan Stanley says". Insider. Retrieved January 20, 2022.

63. Stassen, Murray (March 12, 2021). "Music-related NFT sales have topped \$25m in the past month". Music Business Worldwide. Retrieved May 7, 2021.
64. Brown, Abram. "Largest NFT Sale Ever Came From A Business School Dropout Turned Star DJ". Forbes. Retrieved May 20, 2021.
65. Barcelin, Jason (May 1, 2021). "Las Vegas DJ-producer makes millions selling NFTs". Las Vegas Review-Journal. Retrieved October 13, 2021.
66. Hissong, Samantha (March 3, 2021). "Kings of Leon Will Be the First Band to Release an Album as an NFT". Rolling Stone. Retrieved May 19, 2021.
67. Steele, Anne (March 23, 2021). "Musicians Turn to NFTs to Make Up for Lost Revenue". The Wall Street Journal. ISSN 0099-9660. Retrieved May 7, 2021.
68. Hissong, Samantha (March 9, 2021). "Music NFTs Have Gone Mainstream. Who's In?". Rolling Stone. Retrieved May 20, 2021.
69. "CEO of Sweet Talks NFT Partnership with Rapper Lil Pump". Cheddar. March 23, 2021. Retrieved March 29, 2021.
70. Curto, Justin (March 18, 2021). "Musician NFT Projects, Ranked by How Many F's I Can Give". Vulture. Retrieved March 29, 2021.
71. "Rappers and NFTs – How Hip-Hop Is Cashing In on Non-Fungible Tokens". XXL Mag. March 23, 2021. Retrieved March 29, 2021.
72. Kastrenakes, Jacob (March 1, 2021). "Grimes sold \$6 million worth of digital art as NFTs". The Verge. Retrieved April 7, 2021.
73. Halperin, Shirley (April 21, 2021). "Mike Dean and Shepard Fairey Team for NFT Offering 'OBEY 4:22'". Variety. Retrieved May 7, 2021.
74. Kaufman, Gil. "Eminem's First NFT Drop, 'Shady Con,' Includes One-of-a-Kind Slim Shady-Produced Beats". Billboard. Retrieved May 19, 2021.
75. Heal, Jordan. "Deadpool posters can now be bought as NFTs". Coin Rivet. Retrieved May 19, 2021 – via Yahoo!.
76. Chmielewski, Dawn C. (August 3, 2018). "'Deadpool 2' Jumps On The Digital Collectibles Bandwagon". Deadline. Retrieved May 19, 2021.

77. Ravindran, Manori (March 15, 2021). "NFT Craze Enters Film World: 'Claude Lanzmann' Documentary is First Oscar Nominee to Be Released as Digital Token". Variety. Retrieved April 14, 2021.
78. Bosselman, Haley (March 31, 2021). "'Godzilla vs. Kong' to Have First Major Motion Picture NFT Art Release". Variety. Retrieved April 14, 2021.
79. D'Alessandro, Anthony (April 13, 2021). "Kevin Smith To Sell Horror Movie 'Killroy Was Here' As NFT, Launches Jay And Silent Bob's Crypto Studio". Deadline. Retrieved April 14, 2021.
80. Goldsmith, Jill (September 8, 2021). "Enderby Entertainment's Pandemic Film 'Zero Contact' To Premiere On New NFT Platform Vuele; Watch The Trailer – Update". Deadline Hollywood. Retrieved September 15, 2021.
81. Verhoeven, Beatrice (July 7, 2021). "Anthony Hopkins Film 'Zero Contact' to Premiere as NFT on New Platform Vuele". TheWrap. Retrieved September 15, 2021.
82. Finn, John (April 30, 2021). "World's First Movie Score & Soundtrack For Sale As An NFT". ScreenRant. Retrieved September 8, 2021.
83. Diaz, Johnny (November 17, 2021). "Miramax Sues Quentin Tarantino Over Planned 'Pulp Fiction' NFTs". The New York Times. ISSN 0362-4331. Archived from the original on December 28, 2021. Retrieved November 17, 2021.
84. "NFTs and me: meet the people trying to sell their memes for millions". The Guardian. June 23, 2021. Retrieved July 17, 2021.
85. "Iconic 'Doge' meme NFT breaks record, selling for \$4 million". NBC News. Retrieved July 17, 2021.
86. "Charlie Bit Me NFT sale: Brothers to pay for university with auction money". BBC News. June 3, 2021. Retrieved July 17, 2021.
87. Griffith, Erin (February 22, 2021). "Why an Animated Flying Cat With a Pop-Tart Body Sold for Almost \$600,000". The New York Times. ISSN 0362-4331. Retrieved July 17, 2021.

88. Griffith, Erin (February 22, 2021). "Why an Animated Flying Cat With a Pop-Tart Body Sold for Almost \$600,000". The New York Times. ISSN 0362-4331. Retrieved April 7, 2021.
89. "Zoë Roth sells 'Disaster Girl' meme as NFT for \$500,000". BBC News. April 30, 2021. Retrieved July 17, 2021.
90. "Virtual real estate plot sells for close to \$1 mln". Reuters. June 18, 2021.
91. Dickson, EJ (March 16, 2021). "Porn Creators Are Getting In on the NFT Craze". Rolling Stone. Retrieved March 30, 2021.
92. Cole, Samantha (March 19, 2021). "'Building the Cockchain:' How NSFW Artists Are Shaping the Future of NFTs". Vice. Retrieved March 30, 2021.
93. Whitford, Emma (May 28, 2021). "UC Berkeley Will Auction NFTs for 2 Nobel Prize Patents". Inside Higher Ed. Retrieved May 29, 2021.
94. Starr, Michael (April 1, 2021). "Activist turns anti-Nazi act into commodified digital art". The Jerusalem Post. Retrieved October 13, 2021.
95. "Stanislovas Tomas im Interview: "NFTs können unsere Gesellschaft verändern"". BeInCrypto (in German). April 18, 2021. Retrieved October 13, 2021.
96. "CryptoKitties developer launches NBA TopShot, a new blockchain-based collectible collab with the NBA". TechCrunch. Retrieved April 9, 2021.
97. Cooper, Daniel (March 11, 2021). "NFTs are both priceless and worthless". Engadget. Retrieved April 9, 2021.
98. "Auction of Jack Dorsey Tweet NFT Comes in Millions Below Target". Bloomberg News. Retrieved April 14, 2022. (Subscription required.)
99. Howcroft, Elizabeth (August 25, 2021). "NFT sales surge as speculators pile in, sceptics see bubble". Reuters. Retrieved January 17, 2022.
100. Reyburn, Scott (March 30, 2021). "Art's NFT Question: Next Frontier in Trading, or a New Form of Tulip?". The New York Times. ISSN 0362-4331. Archived from the original on December 28, 2021. Retrieved May 3, 2021.
101. Small, Zachary (April 28, 2021). "As Auctioneers and Artists Rush Into NFTs, Many Collectors Stay Away". The New York Times. ISSN

- 0362-4331. Archived from the original on December 28, 2021. Retrieved May 3, 2021.
102. Cuthbertson, Anthony (March 24, 2021). "NFT MILLIONAIRE BEEPLE SAYS CRYPTO ART IS BUBBLE AND WILL 'ABSOLUTELY GO TO ZERO'". The Independent. Retrieved January 5, 2022.(subscription required)
 103. Tarmy, James; Kharif, Olga (April 15, 2021). "These Crypto Bros Want to Be the Guggenheims of NFT Art". Bloomberg Businessweek. Retrieved April 29, 2021.
 104. Vanek Smithj, Stacey; Woods, Darian (August 4, 2021). "The Origin of Value: The Greater Fools Theory: The Indicator from Planet Money". NPR. Retrieved January 26, 2022.
 105. Owen, Allison; Chase, Isabella (December 2, 2021). "NFTs: A New Frontier for Money Laundering?". Royal United Services Institute. Retrieved January 16, 2022.
 106. Coco, Feng (December 2, 2021). "China's market for NFTs, metaverse may drive money laundering". South China Morning Post. Retrieved January 16, 2022.
 107. "Treasury Releases Study on Illicit Finance in the High-Value Art Market". U.S. Department of the Treasury. Retrieved February 4, 2022.
 108. "Study of the Facilitation of Money Laundering and Terror Finance Through the Trade in Works of Art" (PDF). treasury.gov. United States Department of the Treasury. 2022. p. 26. Retrieved May 11, 2022.
 109. "Two Defendants Charged In Non-Fungible Token ("NFT") Fraud And Money Laundering Scheme". March 24, 2022.
 110. Gallagher, Jacob (March 15, 2021). "NFTs Are the Biggest Internet Craze. Do They Work for Sneakers?". The Wall Street Journal. ISSN 0099-9660. Retrieved April 7, 2021.
 111. "NFTs in Practice – Non-Fungible Tokens as Core Component of a Blockchain-based Event Ticketing Application". Researchgate. Retrieved September 8, 2021.

112. "Golden Ticket: How NFTs Can Help Artists Profit From Ticket Resales". MiamiLaw. April 12, 2021. Retrieved September 8, 2021.
113. "NFTs: The future of ticketing?". IQ Mag. May 6, 2021. Retrieved September 8, 2021.
114. "How NFTs Are Set to Disrupt the Music Industry". Entrepreneur. Retrieved September 8, 2021.
115. "I Joined a Penguin NFT Club Because Apparently That's What We Do Now". The New York Times. August 12, 2021. Archived from the original on December 28, 2021.
116. Chayka, Kyle (July 30, 2021). "Why Bored Ape Avatars Are Taking Over Twitter". The New Yorker.
117. "EIP-721: ERC-721 Non-Fungible Token Standard". Ethereum Improvement Proposals. Retrieved April 5, 2021.
118. Volpicelli, Gian (February 24, 2021). "The bitcoin elite are spending millions on collectable memes". Wired UK.
119. "EIP-1155: ERC-1155 Multi Token Standard". Ethereum Improvement Proposals. Retrieved April 5, 2021.
120. Gault, Matthew (November 18, 2021). "Someone Made a Pirate Bay for NFTs". Vice. Retrieved November 18, 2021.
121. Calma, Justine (March 15, 2021). "The climate controversy swirling around NFTs". The Verge. Retrieved April 15, 2021.
122. Marro, Samuele; Donno, Luca (January 29, 2022). "Green NFTs: A Study on the Environmental Impact of Cryptoart Technologies". arXiv:2202.00003 [cs].
123. Krause, Max; Tolaymat, Thabet (2018). "Quantification of energy and carbon costs for mining cryptocurrencies". *Nature Sustainability*. 1: 814. doi:10.1038/s41893-018-0188-8.
124. Gallersdorfer, Ulrich; Klassen, Lena; Stoll, Christian (2020). "Energy Consumption of Cryptocurrencies Beyond Bitcoin". *Joule*. 4 (9): 1843–1846. doi:10.1016/j.joule.2020.07.013. PMC 7402366. PMID 32838201.
125. deVries, Alex (May 16, 2018). "Bitcoin's Growing Energy Problem". *Joule*.

126. Cuen, Leigh (March 21, 2021). "The debate about cryptocurrency and energy consumption". TechCrunch.
127. De-Mattei, Shanti Escalante (April 14, 2021). "Should You Worry About the Environmental Impact of Your NFTs?". Art News. Retrieved June 2, 2021.
128. Matney, Lucas (March 30, 2021). "ConsenSys launches a more energy-efficient NFT ecosystem with a project from artist Damien Hirst as its first drop". Techcrunch. Retrieved April 21, 2021.
129. Di Liscia, Valentina (April 5, 2021). "Does Carbon Offsetting Really Address the NFT Ecological Dilemma?". Hypoallergic. Retrieved April 21, 2021.
130. Howson, Peter. "NFTs: why digital art has such a massive carbon footprint". The Conversation. Retrieved April 6, 2021.
131. "Blockchains compete for the NFT market, but Ethereum remains the dominant player — Report". Retrieved April 18, 2022.
132. Kinsella, Eileen (April 29, 2021). "Think Everyone Is Getting Rich Off NFTs? Most Sales Are Actually \$200 or Less, According to One Report". Artnet News. Retrieved November 19, 2021.
133. Williams, Rhiannon (April 2, 2021). "NFT digital art: Would you pay millions of pounds for art you can't touch?". inews Technology. Retrieved August 28, 2021.
134. Kwan, Jacklin (July 28, 2021). "An artist died. Then thieves made NFTs of her work". Wired. Retrieved August 28, 2021.
135. "Fake Banksy NFT sold through artist's website for £244k". BBC News. August 31, 2021.
136. Beckett, Lois (January 29, 2022). "'Huge mess of theft and fraud:' artists sound alarm as NFT crime proliferates". The Guardian. Retrieved January 31, 2022.
137. Schneider, Tim (April 21, 2021). "The Gray Market: How a Brazen Hack of That \$69 Million Beeple Revealed the True Vulnerability of the NFT Market (and Other Insights)". artnet news. Retrieved August 28, 2021.

138. "OpenSea admits insider trading of NFTs it promoted". BBC News. September 16, 2021. Retrieved November 22, 2021.
139. Clark, Mitchell (October 26, 2021). "Photoshop's new NFT button could prove you're the real digital artist". The Verge. Retrieved November 24, 2021.