Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

на тему:

Розробка програмного модуля захисту файлів текстур двовимірних моделей для онлайн-сервісу

Виконав: студент 4 курсу, гр.1КІТС-20б
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

_____ Кудревич В.І.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. МБІС
_____ Карпінець В.В.
(прізвище та ініціали)
«10» Червня 2024 р.

Рецензент: к.т.н., доцент каф. ОТ
_____ Крупельницький Л.В.
(прізвище та ініціали)
«10» Червня 2024 р.

Допущено до захисту
Голова секції УБ кафедри МБІС
д.т.н., проф. Яремчук Ю.Є. _____
“10” Червня 2024р.

Вінниця ВНТУ – 2024

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти І-й (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС

_____ **д.т.н., проф. Яремчук Ю.Є.**
“ 22 ” березня 2024 р.

З А В Д А Н Н Я
на бакалаврську дипломну роботу студенту
Кудревичу Вадиму Ігоровичу
(прізвище, ім`я, по-батькові)

1. Тема роботи Розробка програмного модуля захисту файлів текстур двовимірних моделей для онлайн-сервісу
Керівник роботи к.т.н., доцент кафедри МБІС Карпінець Василь Васильович
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від “11” березня 2024 року № 80
2. Термін подання студентом роботи “5” червня 2024 року
3. Вихідні дані до роботи Метою даної роботи є розробка програмного модуля захисту файлів текстур двовимірних моделей для онлайн сервісів з продаже та зберігання даних текстур
4. Зміст текстової частини:
У першому розділі проаналізовано предметну область використуння текстур двовимірних моделей та оглянуто актуальність розробки захищеного модуля для їх зберігання. Другий розділ присвячений проектуваннюта розробці алгоритмув роботи модуля захисту. Також, в другому розділі спроектовано онлайн-сервіс для інтеграції даного мудуля захисту. У третьому розділі представлено програмну реалізацію модуля захисту та онлайн-сервісу, включаючи серверну та клієнтську частини. Модуль захисту інтегровано в розроблений сервіс та проведенно огляд функціоналу

5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)
У першому розділі бакалаврської дипломної роботи наявно 3 рисунки, у другому - 8 рисунків, у третьому розділі – 9 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Карпінець В.В., к.т.н., доц. Каф. МБІС		
Розділ II	Карпінець В.В., к.т.н., доц. Каф. МБІС		
Розділ III	Карпінець В.В., к.т.н., доц. Каф. МБІС		

7.Дата видачі завдання 22 березня 2024 р.

1 КАЛЕНДАРНИЙ ПЛАН

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Визначення напряму бакалаврської робои та формулювання теми	22.03.2024	25.03.2024	Виконано
2	Аналіз предметної області даної теми	26.0.2024	16.04.2024	Виконано
3	Розробка алгоритму роботи	17.04.2024	26.04.2024	Виконано
4	Написання бакалаврської роботи на основі розробленої теми	27.04.2024	23.05.2024	Виконано
5	Передзахист бакалаврської дипломної роботи	24.05.2024	25.05.2024	Виконано
6	Виправлення, уточнення та корегування бакалаврської дипломної роботи	26.05.2024	11.06.2024	Виконано
7	Захист бакалаврської дипломної роботи	12.06.2024	17.06.2024	Виконано

Студент _____ Кудревич В.І.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Карпінець В.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 83 сторінок формату А4, на яких є 20 рисунків, список використаних джерел містить 31 найменування.

В даній дипломній роботі здійснено програмну реалізацію модуля захисту для інтернет-сервісу для зберігання та продажу текстур двовимірних моделей. Для написання сервісу було обрано мову програмування PHP.

Було розглянуто предметну область, визначено актуальність проблеми та проаналізовано існуючі аналоги на їх переваги та недоліки. Після цього було теоретично розроблено сервіс та модуль захисту для нього.

В результаті було отримано конкурентоспроможний сервіс для продажу та зберігання двовимірних текстур, який вирізняється своєю високою захищеністю, що дуже важливо для зберігання права на інтелектуальну власність.

Ключові слова: цифрові водяні знаки, шифрування, текстура, модуль захисту

ABSTRACT

The bachelor's thesis consists of 83 A4 pages, which have 20 figures, the list of sources used contains 31 items.

In this diploma work, the software implementation of the protection module for the Internet service for storing and distributing the textures of two-dimensional models was carried out. The PHP programming language was chosen to write the service.

The subject area was considered, the relevance of the problem was determined, and the existing analogues were analyzed for their advantages and disadvantages. After that, the service and the protection module for it were theoretically developed.

As a result, a competitive service for the sale and storage of two-dimensional textures was obtained, which is distinguished by its high security, which is very important for the preservation of intellectual property rights.

Keywords: digital watermarks, encryption, texture, protection module.

ЗМІСТ

ВСТУП

Актуальність. У сучасному світі інформаційної безпеки, забезпечення надійного шифрування даних є важливою складовою для захисту конфіденційної інформації в онлайн-середовищі. Особливо це стосується текстур - цінного ресурсу для створення футажів, ігор і 3D-моделей. Проте, існуючі сервіси з продажу текстур залишаються вразливими перед можливими атаками зловмисників через відсутність ефективного шифрування.

З цією проблемою пов'язане і моє дослідження, метою якого є розробка імплементації інтернет-магазину текстур, який використовує шифрування для забезпечення безпеки даних користувачів. Актуальність цього дослідження підкреслюється не лише потребою в захисті інтелектуальної власності, а й загальною необхідністю у підвищенні рівня кібербезпеки в онлайн-середовищі.

В рамках дослідження буде проведено аналіз існуючих аналогів сервісів з продажу текстур, виокремлені їх переваги та недоліки, зокрема, відсутність ефективного шифрування. На основі цього аналізу буде запропоновано концепцію нового сервісу, який поєднує переваги існуючих аналогів та додає новий рівень безпеки завдяки застосуванню шифрування даних користувачів.

Далі в дослідженні розглянуться технічні аспекти реалізації цього сервісу, включаючи вибір мови програмування, розробку архітектури сервісу та теоретичну розробку модулю шифрування текстур та вбудовування водяних знаків в них.

Метою роботи є створення модуля захисту для файлів текстур двовимірних моделей. Задля наглядності буде розроблено спеціалізований онлайн-сервіс для продажу та зберігання двовимірних текстур.

Для досягнення мети необхідно виконати такі завдання:

– провести аналіз предметної області та визначити актуальність створення даного модуля;

– дослідити існуючі аналоги та проаналізувати їх;

– розробити архітектуру модуля захисту для онлайн-сервісу;

– проаналізувати та обрати метод шифрування та вбудовування ЦВЗ;

– розробити архітектуру сервісу для інтеграції модуля захисту;

– практично реалізувати модуль захисту;

– створити сервіс для зберігання та продажу двовимірних текстур;

– інтегрувати модуль захисту в розроблений сервіс.

Предметом дослідження є розробка і реалізація інтернет-магазину текстур, спрямованого на задоволення потреб у цифрових матеріалах для різноманітних творчих та професійних проектів. Основна увага приділяється аспектам забезпечення безпеки даних користувачів шляхом використання криптографічних методів шифрування.

Об'єктом дослідження є процес зберігання та обміну даними в інтернет-магазині текстур з механізмом захисту конфіденційності, цілісності та доступності цих даних під час їх передачі через мережу Інтернет. Важливою частиною дослідження є інтеграція криптографічних методів в процес обробки та зберігання даних, а також аналіз ефективності цих методів у контексті забезпечення безпеки даних користувачів.

Практична цінність роботи полягає у можливості інтеграції розробленого модулю захисту в системи зберігання та продажу текстур. Дана розробка має важливе значення, оскільки дозволяє захистити інтелектуальну власність, таку як текстури двохвимірних моделей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДВОВИМІРНИХ ТЕКСТУР ТА МЕТОДІВ ЇХ ЗАХИСТУ

1.1 Аналіз предметної області використання текстур

Предметною областю даного проекту є створення інтернет-магазину, спеціалізованого на продажі текстур для різноманітних цілей, наприклад для створення, ігор та 3D-моделей. Текстури в цьому контексті використовуються як основний будівельний матеріал для створення візуальних компонентів у цифровому середовищі. У галузі ігрової індустрії, текстури є важливим елементом для створення реалістичних графічних об’єктів та сценаріїв.

Проект спрямований на створення зручного та надійного інтернет-магазину, який надає користувачам доступ до широкого асортименту текстур для їх творчих потреб адже перелік можливих напрямів і клієнтів дуже широкий:

– архітектурне моделювання в архітектурній індустрії текстури використовуються для створення візуалізацій будівельних проектів, інтер’єрів та ландшафтів;

– моделювання одягу - текстури використовуються для створення реалістичних тканин та матеріалів у віртуальних примірочних кімнатах або програмах для моделювання моди;

– у медичній графіці - текстури можуть бути використані для створення тривимірних моделей органів, тканин та біологічних структур для навчальних чи діагностичних цілей;

– інтер'єрний дизайн для створення візуалізацій інтер'єрів, меблів та декору, щоб допомогти клієнтам уявити собі кінцевий результат дизайну;

– індустріальне моделювання у виробничих процесах текстури використовуються для створення тривимірних моделей обладнання, машин та пристроїв для аналізу, дизайну та виробництва [1].

Для нас текстура - це плоске двовимірне зображення, накладене на полігони, що складаються з тривимірних структур, які ми бачимо на екрані. Для комп'ютера ж це просто невеликий блок пам'яті у вигляді двовимірного масиву. Кожен елемент

масиву являє собою значення кольору одного з пікселів текстурного зображення (зазвичай це піксель текстури, званий текселем).

Кожна вершина багатокутника має дві координати (зазвичай позначаються u, v), які повідомляють комп'ютеру, який піксель текстури з нею пов'язаний. Сама вершина має три набори координат (x, y, z), і процес прикріплення текстури до вершини називається накладенням текстури. Проігноруємо z-координати вершин і розглянемо все на площині (рис. 1.1).

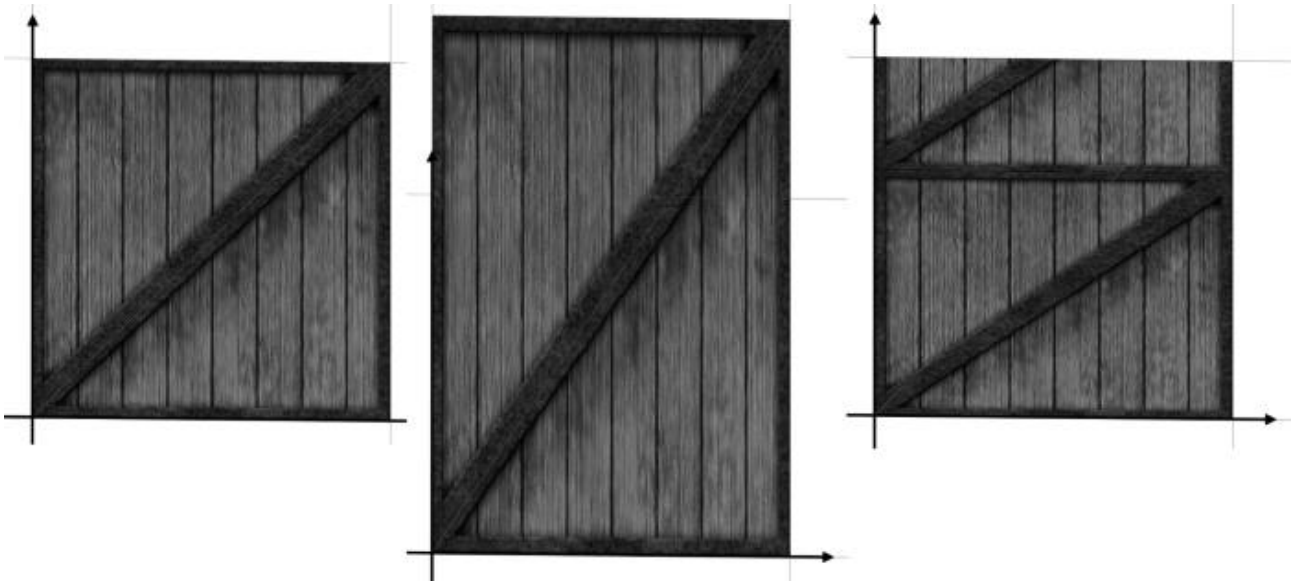


Рисунок 0.1 – Приклад текстури

Зліва направо: u- і v-координати текстури прив'язані безпосередньо до x- і y-координат кутових вершин; на другому зображенні y-координати вершин збільшилися, і текстура залишилася прив'язаною до цих вершин, але витягнулася по вертикалі. На правому зображенні текстура змінилася: значення u збільшилося, але це призвело до стиснення текстури, яке повторюється [2].

Це відбувається тому, що, хоча збільшення u-значення збільшило фактичну висоту текстури, вона все ще повинна вписуватися в межі примітиву, і в результаті деякі частини текстури повторюються. Це один зі способів створення ефекту "повторення текстури", який часто використовується в 3D-іграх. Приклади такого ефекту можна побачити в сцені з кам'янистими або трав'янистими ділянками або в кадрі з цегляною стіною.

Змінивши сцену, збільшимо кількість примітивів і відновимо глибину сцени. Нижче представлений класичний пейзажний знімок, але текстура коробки була скопійована і повторена на всіх примітивах (рис. 1.2).

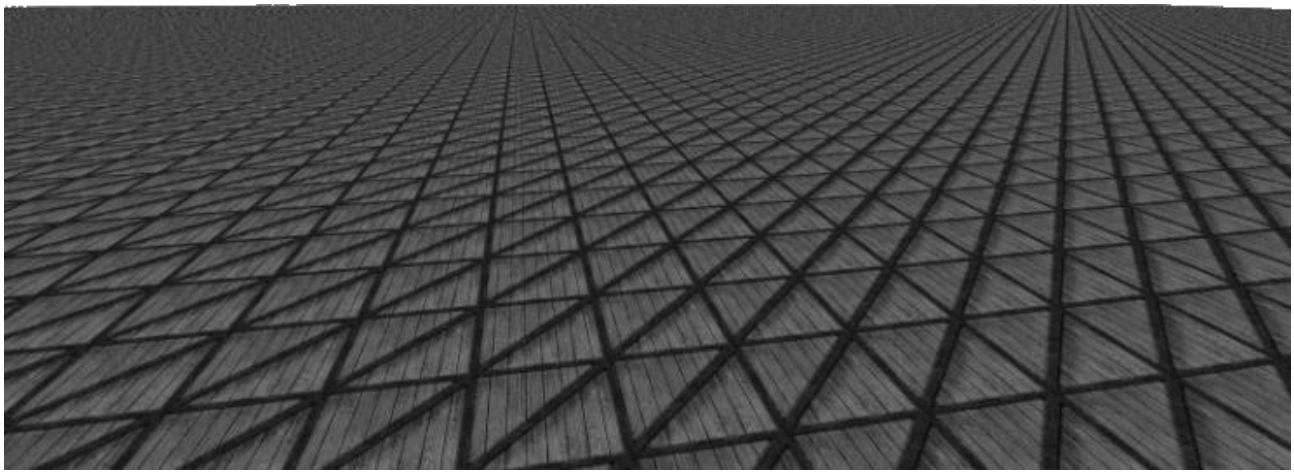


Рисунок 0.2 – Використання текстури

Оригінальна текстура боксу у форматі GIF має розмір 60 КБ і роздільну здатність 257 x 257 пікселів. З огляду на те, що вихідна роздільна здатність області кадру, вкритої текстурою боксу, становить 1800 x 690 пікселів, у цій області має бути всього 20 текстур боксу [3].

Однак, можемо побачити що відображається набагато більше 20 боксів. Це говорить про те, що текстури боксів були успішно зменшені до розміру 257 x 257 пікселів. Цей процес називається "зменшенням текстури".

Таким чином одне зображення невеликого розміру може утворювати моделі значних розмірів без необхідності перевантажувати пам'ять великими зображеннями.

Загалом текстури відіграють важливу роль у комп'ютерній графіці і можуть створювати реалістичні та привабливі візуальні ефекти. Текстури - це зображення, що наносяться на поверхню 3D-моделі для додавання деталізації та реалістичності [4].

Текстурне моделювання - це процес створення текстур у комп'ютерній графіці та застосування їх до об'єктів. У комп'ютерній графіці текстури - це зображення або візерунки, що наносяться на поверхню об'єкта для створення візуальних ефектів і додавання деталей. Текстури використовуються для додавання реалістичності та

деталізації об'єктів у 3D-графіці. Текстури можуть бути створені з фотографій або малюнків, або за допомогою спеціалізованого програмного забезпечення. Текстури можуть бути представлені найрізноманітнішими матеріалами, зокрема деревом, каменем, металом і тканиною. Кожна точка на поверхні об'єкта має координати, які називаються координатами текстури. Координати текстури визначають кількість текстури, застосованої до кожної точки на поверхні об'єкта. Це дозволяє створювати різні ефекти, такі як рельєф, переходи кольорів, пошкодження тощо [5].

Текстури можна застосовувати до найрізноманітніших об'єктів, включно з моделями персонажів, оточенням та архітектурними елементами. У комп'ютерній графіці вони відіграють важливу роль у створенні реалістичних і привабливих візуальних сцен (рис. 1.3).

Рисунок 0.3 – Приклад текстури

Текстури відіграють важливу роль у створенні реалістичних і привабливих візуальних ефектів у комп'ютерній графіці. Текстури додають об'єктам і поверхням деталей, кольору і текстури, роблячи їх більш реалістичними і цікавими. Вони дозволяють створювати реалістичні матеріали, такі як дерево, камінь і метал. Текстури надають об'єктам природнішого та реалістичнішого вигляду, додаючи такі деталі, як текстура деревної кори, шорсткість каменю або блиск металу.

Їх можна використовувати в комп'ютерній графіці для створення настрою та атмосфери. Наприклад, текстури неба, хмар і ландшафту можна використовувати для створення реалістичних фонових зображень або для створення атмосфери певного місця чи часу. Також їх використовують для підвищення деталізації та реалістичності об'єктів і поверхонь. Наприклад, за допомогою текстур можна додати на поверхню подряпини, бруд і потертості, завдяки чому об'єкти виглядатимуть реалістичніше і реалістичніше [6].

Їх ще часто використовують для створення спеціальних ефектів, таких як вогонь, вода і дим. За допомогою текстур можна створювати різноманітні візуальні ефекти, які додають комп'ютерній графіці динамічності та цікавості.

Текстури також можна використовувати для покращення візуального дизайну комп'ютерної графіки. Текстури можуть додати проекту стилю, унікальності та естетичного вигляду, їх можна використовувати для створення фонових зображень, візерунків, малюнків та інших візуальних елементів, які роблять проект більш привабливим і цікавим.

Загалом, текстури відіграють важливу роль у комп'ютерній графіці і можуть створювати реалістичні та привабливі візуальні ефекти, підвищувати деталізацію та реалістичність об'єктів, створювати атмосферу та спецефекти, а також покращувати візуальний дизайн проекту.

При моделюванні текстур в комп'ютерній графіці можна зіткнутися з різними проблемами і обмеженнями, які ускладнюють процес створення реалістичних і якісних текстур.

Розглянемо деякі з них:

1. Обмеження роздільної здатності текстури. Однією з основних проблем є обмежена роздільна здатність текстур. Кожна текстура має кількість пікселів, яка визначає її деталізацію. Низька роздільна здатність текстури може призвести до того, що великі об'єкти та рендери крупним планом виглядатимуть розмитими та неприродними.

2. Спотворення текстури, коли об'єкти нахилені або зігнуті. Ще однією проблемою є спотворення текстури, коли об'єкти нахиляються або скручуються.

Якщо об'єкт має складну форму або деформується, текстура може спотворюватися і виглядати неприродно. Це може вимагати додаткової роботи з корекції текстури або використання спеціальних технік моделювання.

3. Проблеми з узгодженням текстур і геометрії об'єкта. Ще однією проблемою є відповідність між текстурою і геометрією об'єкта. Якщо текстура не відповідає формі та розміру об'єкта, він може виглядати спотвореним і неприродним. Це може вимагати ретельного позиціонування і масштабування текстури для кращої відповідності геометрії об'єкта.

4. Обмежені можливості моделювання деталей. Ще одним обмеженням є обмежені можливості моделювання деталей. Деякі текстури містять багато деталей, наприклад, дрібні текстури або складні візерунки. Однак, при моделюванні текстур може існувати обмеження на кількість деталей, які можна обробити. Тому, можливо, доведеться йти на компроміси, щоб вибрати найважливіші деталі для включення в текстуру.

5. Проблеми освітлення і тіні. При моделюванні текстур слід також враховувати питання світла і тіні. Світло і тіні впливають на візуальне сприйняття і реалістичність текстури. Якщо текстура не відповідає світлу і тіням об'єкта, вона може виглядати пласкою і неприродною. Тому для досягнення більш реалістичного ефекту може знадобитися використання спеціальних технік освітлення і затінення.

Загалом, текстурне моделювання має проблеми та обмеження, які можуть ускладнити процес створення якісних, реалістичних текстур. Тому доступ до гарних і професійних текстур необхідний [7].

Сучасна індустрія відеоігор класу ААА домоглася великих успіхів у поліпшенні якості та реалістичності текстур. Якість і чіткість текстур залежить від вимог самої гри. Під час розроблення гри важливо враховувати платформу, на якій вона розробляється. Наприклад, на ПК розробникам необхідно створювати текстури для різних налаштувань графіки (високих, середніх і низьких). Якщо гравець використовує застарілий пристрій, гра, найімовірніше, не буде оптимально працювати. Якщо гра розробляється виключно для консолі, наприклад PlayStation або Xbox, розробнику буде трохи легше, оскільки консолі мають більш стабільні

ресурси. Однак деякі технології, що покращують графіку, доступні тільки на комп'ютерах, наприклад HairWorks для реалістичного моделювання волосся і RTX для реалістичного освітлення ігрових сцен, тому якість графіки в будь-якому разі вища на комп'ютерах. Це стосується наступних областей. Сьогодні існує безліч різних методів текстурування 3D-моделей і безліч програм для їх реалізації.

Перші методи текстурування були засновані на ручному опрацюванні даних; їх використовували з самого початку створення ігор WarCraft. Розробники малювали всі подряпини і пил у 2D-редакторі та використовували координати сканування, щоб зрозуміти, де додати ті чи інші елементи в текстуру. Цей метод займає багато часу, і неможливо відстежити історію створення текстури на кожному етапі. У сучасних AAA-відеоіграх цей ручний метод використовується рідко, оскільки існують новіші та досконаліші інструменти для створення красивих і цікавих текстур.

Одним із найпопулярніших методів текстурування сьогодні є створення текстур за допомогою процедурних текстур і генераторів. Цей сучасний метод містить у собі кілька шейдерів, різні інструменти та PBR: за допомогою PBR текстури реалістично відображаються в грі та показують фізичні властивості різних матеріалів. Наприклад, метал має свій власний сонячний відблиск, пластик - інший, і всі матеріали відображаються по-різному. До винаходу PBR такого якісного та реалістичного текстурування не існувало [8].

Використовуючи процедурні текстури, такі як генератори та гранж, художники створюють текстури на моделях. Розробники можуть вибирати різні пензлі, додавати подряпини та іржу, змішувати різні кольори. За допомогою генератора можна швидко додати на модель бруд і пил. Кожен шар текстури має свої налаштування, наприклад, ефекти висоти, щоб подряпини виглядали глибше, і все це на текстурі без додавання полігонів.

Цей метод швидкий і креативний, оскільки текстура малюється не на скані, а на самій моделі. Однак цей метод вимагає більшого технічного досвіду через сучасні вимоги до PBR, шейдерів та інших підпрограм, що працюють на відеоприскорювачах. Щоб текстури мали реалістичний вигляд, важливо грамотно

використовувати генератори і гранж. Наприклад, якщо ви просто накидаєте текстури разом без належної логіки, то в підсумку замість реалістичного вигляду текстур ви отримаєте брудні текстури.

1.2 Аналіз актуальності захищеності двовимірних текстур та методів їх захисту

У сучасному цифровому світі, коли інформація стає однією з найцінніших активів, захист інтелектуальної власності, зокрема двовимірних текстур, набуває вельми важливого значення. Збереження і захист цих текстур від несанкціонованого використання є викликом для багатьох галузей, зокрема для інтернет-магазинів, які використовують їх для презентації товарів.

Актуальність проблеми зберігання двовимірних текстур полягає в тому, що вони стають об'єктом бажання для порушників авторських прав та конкурентів, які можуть використовувати їх без дозволу власника з метою власної користі. Це може призвести до втрати прибутку, порушення репутації компанії та втрати конкурентної переваги.

Окрім того, зростаюча кількість даних та розвиток технологій у сфері комп'ютерного зору та обробки зображень створюють нові можливості для швидкого і ефективного виявлення, копіювання та модифікації двовимірних текстур, що підсилює необхідність ефективних методів їх захисту.

Таким чином, розробка ефективних та надійних методів збереження та захисту двовимірних текстур як наукової власності стає актуальною задачею для бізнесу та наукової спільноти. Вирішення цієї проблеми не лише сприятиме захисту інтелектуальної власності, а й забезпечить стабільність та інноваційний розвиток в цифровому середовищі [9].

Під час збереження двовимірних текстур існує декілька потенційних проблем, які можуть виникнути.

Підробка та несанкціоноване використання. Це одна з основних проблем, з якою зіштовхуються власники двовимірних текстур. Несанкціоновані користувачі

можуть використовувати ці текстури без дозволу власника для власних цілей, порушуючи авторські права.

Також можлива втрата якості під час збереження двовимірних текстур у зашифрованому або водяному знаку форматі може виникнути втрата якості. Це може відбутися через стиснення даних або додавання зайвих артефактів, що може знизити якість зображення.

Існує ризик, що захист, який застосовується до двовимірних текстур, може бути обхідним або вразливим перед атаками хакерів або зловмисників. Це може призвести до неправомірного доступу до текстур і їх незаконного використання.

Сумісність із різними платформами. Деякі методи збереження та захисту двовимірних текстур можуть бути несумісними з певними платформами або програмним забезпеченням, що може ускладнити їх використання або обмін між різними користувачами.

Вартість інфраструктури захисту, розробка та підтримка ефективної інфраструктури для збереження та захисту двовимірних текстур може бути дорогим завданням для компаній та організацій, особливо для малих підприємств або стартапів [10].

Питання захисту авторських прав на зображення сьогодні стоїть дуже гостро. Те, що публікується в інтернеті, може виглядати дуже схожим на чиюсь власність.

Стеганографія - означає приховування однієї цифрової інформації в іншій цифровій інформації. У цьому випадку графічне зображення містить іншу інформацію (наприклад, текст). Зміна колірного тону пікселя трохи змінює зображення, але людське око нічого не помічає. Це і є принцип стеганографії.

Цей метод можна використовувати для захисту даних як в аудіо-, так і в відеофайлах. Така система надзвичайно розумна. Це пов'язано з тим, що навряд чи хтось буде перевіряти, чи має кожне зображення шифр. В інтернеті є стеганографії з відкритим вихідним кодом, які можна завантажити і модифікувати, і ніхто не запідозрить, що на зображенні є підпис або в нього вбудували підпис. Тим більше, що прості зміни в молодший біт важко зрозуміти без знання алгоритму.

Хороші програми, призначені для захисту графічних і звукових файлів, виконують свою роботу настільки добре, що результати майже непомітні. Приблизно одна десята від розміру зображення - це обсяг інформації, який можна зашифрувати. Розшифрувати інформацію, приховану в таких файлах, можна. Єдина проблема полягає в тому, що програма для розшифрування повинна мати ключ.

Якщо продукт, захищений водяним знаком, викрадений зловмисником, проти нього можна порушити справу, а авторське право, нанесене на викрадений продукт, можна представити як доказ. Для цього існують спеціальні програми. Деякі з них є плагінами для графічних пакетів (Photoshop або CorelDraw), інші - окремими продуктами.

Видимі водяні знаки задають спеціальне положення на зображенні. Водяний знак може бути текстовим або графічним. Можна додати знак, що тільки цей файл містить нульовий водяний знак, або додати бінарний логотип (накладання бінарного логотипу) чи текст (багатобітний водяний знак). Для обробки великої кількості файлів необхідно використовувати пакетний режим.

Створити водяний знак - це непросте завдання, і поганий захист зведе нанівець усі зусилля. Наприклад, повідомлення, що потребують захисту, шифруються за допомогою певного ключа (зазвичай встановлюється користувачем). Потім значення пікселів змінюються, зображення розбивається на блоки і виконується перетворення Фур'є. Під час цього процесу інформація, що використовується для ідентифікації матеріалу, маскується зашумленими деталями зображення. Чим більше деталей на зображенні, тим вища щільність вбудованого ідентифікаційного коду. Деякі продукти доступні для пакетного пошуку графічних зображень з водяними знаками в Інтернеті.

Digimarc була першою компанією, яка розробила цю технологію. Компанія розробила MarcSpider, пошукового робота, який шукає в Інтернеті зображення з водяними знаками, щоб виявити крадіїв інтелектуальної власності в Інтернеті. Технологію Digimarc наразі використовує низка компаній, зокрема Getty Images, FPG та Indexstock. Пошуки проводяться в розширеному домені.

Якщо на певному сайті знайдено підозріле посилання, програма починає перевіряти сам сайт. Єдина проблема, полягає в тому, що плагіат зображень можна виявити лише в тому випадку, якщо плагіатор є зареєстрованим учасником системи. Тільки в цьому випадку робот буде шукати необхідні авторські права на зображення і повідомляти про це відповідні органи. Захист від водяних знаків заснований на тому, що такі знаки неможливо видалити. Точніше, їх можна видалити, але якість зображення (або якість звуку у випадку аудіофайлів) значно знижується, і продукт втрачає свою товарну привабливість. Те саме стосується і "шумових" ефектів [11].

Якщо на зображенні з'являються значні зміни або спотворення, знаки залишаються недоторканими, але не повністю. Тут в нагоді стає можливість розбиття зображення на блоки. Якщо один блок спотворюється, інші залишаються.

1.3 Огляд існуючих сервісів для збереження та продажу двовимірних текстур та їх аналіз

Для розуміння конкурентів запропонованого проєкту, розглянемо найпопулярніші аналоги які надають послуги доступу до текстур.

– Texture Haven: Зручний сайт, що пропонує великий вибір якісних текстур безкоштовно;

– Poliigon: Цей сайт пропонує великий вибір текстур високої якості, а також 3D-моделей та інших графічних ресурсів;

– Textures.com: Має широкий асортимент текстур для різних потреб, включаючи текстури для 3D-моделювання, анімації та візуалізації;

– Substance Source: Від Adobe, цей ресурс пропонує текстури та матеріали для використання в програмах для реалістичного моделювання та текстурування, таких як Substance Painter;

– 3DTotal: Цей сайт пропонує не лише текстури, а й навчальні матеріали та ресурси для художників і 3D-моделювальників.

Texture Haven – ресурс зручний тим, що пропонує великий вибір якісних текстур безкоштовно. Його інтерфейс користувача простий та зрозумілий, що

полегшує пошук потрібних ресурсів. Проте, в порівнянні з деякими іншими платформами, Texture Haven обмеженим у функціональності, не має можливості преміум-оплати. В цілому це чудовий ресурс для тих, хто шукає якісні безкоштовні текстури з простим і зручним інтерфейсом, але він не задовольняє всі потреби користувачів, які шукають додаткові можливості платних ресурсів.

Переваги Texture Haven:

– безкоштовні текстури високої якості;

– простий та зрозумілий інтерфейс користувача;

– широкий вибір різноманітних текстур для різних потреб;

– наявність категорій та системи пошуку для швидкого знаходження потрібних ресурсів.

Недоліки Texture Haven:

– обмежений вибір у порівнянні з деякими іншими платформами;

– відсутність додаткових функцій, таких як корзина для покупок або можливість преміум-оплати;

– можливість знайти деякі текстури нижчої якості;

– відсутність розширеної підтримки користувачів або можливості отримання додаткової інформації про ресурси.

Poliigon - пропонує великий арсенал текстур і матеріалів високої якості, які вражають своєю реалістичністю та деталізацією що робить Poliigon особливим. Кожна текстура відточена з неймовірною уважністю до найдрібніших аспектів, що дозволяє втілити будь-яку свою ідею з неймовірною точністю та реалізмом. Будь-то це текстура дерева зі складними відтінками кори або металева поверхня з вражаючими відблисками світла. На сайті багатий вибір ресурсів які забезпечать текстурами найвищої якості [12].

Однак, Poliigon - це не лише сховище текстур, а й інструмент творчого виразу. Завдяки його різноманіттю і реалістичності, можна легко створювати неймовірно реалістичні сцени та образи, що здатні захопити уявлення глядачів та залишити незабутні враження.

Переваги Poliigon:

– великий вибір текстур і матеріалів, він пропонує широкий асортимент текстур і матеріалів для різних потреб графічного дизайну та візуалізації;

– висока якість ресурсів, текстури і матеріали на poliigon відрізняються високою якістю та реалістичністю, що дозволяє створювати вражаючі візуальні ефекти;

– доступність різних форматів і роздільної здатності дозволяє використовувати їх у різних проектах та програмах;

– зручний пошук і навігація, сайт має зручну систему пошуку і навігації, що дозволяє швидко знаходити потрібні ресурси.

Недоліки Poliigon:

– платність деяких ресурсів хоча poliigon пропонує безкоштовні ресурси, деякі з їхніх текстур та матеріалів доступні тільки за плату;

– відсутність безкоштовного плану для користувачів з обмеженим бюджетом для отримання доступу до деяких преміум-ресурсів poliigon потрібно придбати платний план, що може бути не доступним для користувачів з обмеженим бюджетом;

– можливість відчутного завантаження сайту через великий обсяг ресурсів. у зв'язку з великим обсягом ресурсів, сайт poliigon може відчувати завантаження під час пошуку або завантаження ресурсів, особливо на повільних інтернет-з'єднаннях.

Textures.com - це онлайн-ресурс, який надає доступ до великого обсягу текстур високої якості для використання в графічному дизайні, візуалізації та інших цифрових проектах. Сайт пропонує широкий вибір текстур різних типів, включаючи дерево, камінь, метал, тканини, шкіру та багато інших [13].

Однією з ключових переваг Textures.com є його великий асортимент текстур, який охоплює різноманітність матеріалів і поверхонь. Текстури доступні у різних форматах, таких як JPEG, PNG та TIFF, і різних роздільних здатностях, що дозволяє користувачам зручно використовувати їх у різних проектах та програмах.

Крім того, Textures.com має досить розгалужену систему пошуку і фільтрації, що дозволяє користувачам швидко знаходити потрібні ресурси за різними

критеріями, такими як тип матеріалу, розмір, роздільна здатність тощо. Це дозволяє ефективно використовувати час користувача та знайти саме те, що потрібно для їхнього проекту.

Недоліками Textures.com може бути обмежений вибір деяких типів текстур у порівнянні з іншими платформами, а також можливість технічних проблем під час завантаження або використання великих файлів текстур на повільних інтернет-з'єднаннях. Також, деякі текстури можуть мати обмежену роздільну здатність або якість у порівнянні з преміум-ресурсами.

Substance Source - це онлайн-платформа, розроблена Adobe, яка спеціалізується на наданні доступу до високоякісних текстур, матеріалів та ресурсів для використання у програмах для реалістичного моделювання та текстурування, таких як Substance Painter [14].

Однією з ключових переваг Substance Source є його унікальність і інноваційність у наданні текстур та матеріалів. Сюди входять такі функціональності, як параметризовані матеріали, які дозволяють змінювати різні аспекти текстур у реальному часі, а також інтеграція з програмами Adobe Creative Cloud, що полегшує роботу з ресурсами у вашому улюбленому графічному редакторі.

Крім того, Substance Source пропонує широкий вибір різноманітних матеріалів, які охоплюють різноманітність поверхонь, від дерева та каменю до металу та тканин. Це дозволяє графічним дизайнерам та візуалізаторам швидко знаходити потрібні ресурси для своїх проектів і забезпечує високий рівень якості та реалістичності відтворення матеріалів.

Недоліки Substance Source це обмежений вибір деяких типів матеріалів у порівнянні з іншими платформами, а також можливість платних підписок для доступу до деяких преміум-ресурсів. Також, інтеграція з Adobe Creative Cloud може бути не такою ефективною для користувачів, які використовують інші графічні програми.

3DTotal - це онлайн-ресурс, що спеціалізується на графічному дизайні та візуальних ефектах. Він пропонує широкий вибір текстур, моделей, референсів та

навчальних матеріалів для художників, дизайнерів та візуалізаторів. Основною метою 3DTotal є надання якісних ресурсів та навчальних матеріалів, що сприяють розвитку професіоналізму у сфері графічного дизайну та візуальних ефектів.

Переваги 3DTotal:

– широкий вибір ресурсів. 3dtotal пропонує великий асортимент текстур, моделей та навчальних матеріалів для різних потреб графічного дизайну та візуалізації;

– навчальні матеріали. сайт містить багато навчальних матеріалів, включаючи статті, відеоуроки та курси, що допомагають удосконалювати навички користувачів;

– 3DTotal є платформою для спілкування та співпраці художників і дизайнерів з усього світу, що сприяє обміну досвідом і ідеями.

Недоліки 3DTotal:

– деякі матеріали та навчальні ресурси на 3dtotal доступні лише за плату, що може бути обмеженням для користувачів з обмеженим бюджетом;

– обмежений вибір безкоштовних ресурсів. у порівнянні з іншими платформами, кількість безкоштовних ресурсів на 3dtotal може бути обмеженою;

– вимоги до обладнання. деякі високоякісні ресурси на 3dtotal можуть вимагати потужного обладнання для оптимального використання.

Більшість платформ, які пропонують безкоштовні або комерційні текстури, зазвичай не надають інструментів або технічних механізмів для захисту контенту від несанкціонованого використання або копіювання.

Також мають відкритий або публічний доступ до свого контенту, що робить його вразливим до копіювання або незаконного використання [15].

Часто у них можуть відсутні обмеження або заходи захисту, які забезпечують власникам контенту контроль над його використанням. Також деякі платформи можуть надавати доступ до текстур у відкритому форматі, що робить їх доступними для використання будь-яким користувачам без обмежень або захисту.

У випадку розробки сервісу з шифруванням важливо підкреслити присутність захисту текстур як наукової власності, особливо враховуючи що проект передбачає комерційне використання.

Враховуючи аналіз існуючих аналогів, було визначено спільну проблему у всіх сервісах – відсутність захисту. Саме тому, було вирішено реалізувати розробку програмного модуля захисту файлів текстур двовимірних моделей для онлайн-сервісу. Даний модуль захисту повинен реалізувати вбудовування цифрових водяних знаків та шифрування текстур перед зберіганням.

1.4 **Висновки та постановка задачі**

У цьому розділі було здійснено аналіз предметної області і різноманітних сфер застосування текстур, що включають створення, ігор і 3D-моделей. Такий ретельний огляд дозволив нам краще розібратися з важливістю та широким спектром використання текстур у сучасному цифровому світі.

Аналіз існуючих аналогів дозволив зрозуміти актуальність проблеми створення захищеного сервісу для продажу та зберігання двовимірних текстур. Саме тому, після розгляду теоретичної частини, було поставлено наступні задачі:

- розробити архітектуру модуля захисту для онлайн-сервісіу;
- проаналізувати та обрати метод шифрування та вбудовування ЦВЗ;
- розробити архітектуру сервісу для інтеграції модуля захисту;
- практично реалізувати модуль захисту;
- створити сервіс для зберігання та продажу двовимірних текстур;
- інтегрувати модуль захисту в розроблений сервіс.

2 РОЗРОБКА АЛГОРИТМУ МОДУЛЯ ЗАХИСТУ ФАЙЛІВ ТЕКСТУР ДВОВИМІРНИХ МОДЕЛЕЙ ДЛЯ ОНЛАЙН-СЕРВІСУ

Після огляду предметної області та обгрунтування актуальності проблеми створення захищеного сервісу для зберігання та продажу двовимірних текстур почався процес проектування майбутнього сервісу. Спочатку було обрано модель архітектури, розроблено сервіс, розроблено архітектуру захищеного модулю та обгрунтовано вибір мови програмування для реалізації додатку.

2.1 Розробка модуля захисту двовимірних текстур для онлайн-сервісу

Для розробки модуля захисту потрібно визначити, які методи захисту буде включати в себе даний модуль. Першочерговим є захист файлів текстур від компрометації. Дана задача може вирішитись шляхом вбудовування цифрових водяних знаків в текстури.

Цифрові водяні знаки бувають двох видів: приховані та видимі. Для кращого захисту використаємо обидва: на прев’ю будемо накладати видимий ЦВЗ, а на сам файл текстури – прихований. Даний метод допоможе не лише захистити доступні всім користувачам зображення прев’ю текстури, а й накласти авторство на самі файли.

Також, для надійності захисту, варто використати шифрування. Тобто, файл після завантаження буде зашифровано, а лише потім збережено до бази даних. Таким чином, файл текстури буде додатково захищено від несанкціонованого доступу навіть у разі зламу сервера та бази даних.

Отже, модуль захисту буде складатись з таких методів:

1. Вбудовування цифрових водяних знаків. ЦВЗ допоможуть запобігти компрометації файлів, оскільки вони міститимуть дані про власників текстур. По-перше, в файл текстури вбудовується прихований водяний знак, який допомагає визначити власність. Також, потрібно захистити прев’ю текстури від викрадення за допомогою вбудовування видимого цифрового водяного знаку.

2. Шифрування даних. Для захисту текстур від викрадення у разі злому серверу чи бази даних потрібно використати алгоритм шифрування для безпечного зберігання завантажених текстур.

В результаті, було сформовано загальний алгоритм роботи модуля захисту:

Крок 1: Завантаження текстур.

На даному етапі користувач завантажує файли прев'ю та самої текстури на сервер. Також, на даному етапі він заповнює поле, які дані внести в ЦВЗ. Цими даними можуть бути дата та час створення, або завантаження текстури, а також авторство.

Крок 2: Накладання видимого ЦВЗ на прев'ю

На цьому кроці інформація про авторство накладається на прев'ю текстури, яке буде відображено на самому сайті.

Крок 3: Накладання прихованого ЦВЗ на текстуру

Тут прихований ЦВЗ накладається на файл текстури з високим розширенням, який пізніше буде завантажено до бази даних.

Крок 4: Шифрування текстури

Прев'ю текстури надсилається на базу даних в стандартному вигляді після накладання ЦВЗ. Сам ж файл текстури після накладання на нього цифрових водяних знаків проходить етап шифрування, після цього надсилається до бази даних. Файл буде розшифровано, коли покупець купить текстуру та завантажить її з особового кабінету.

Крок 5: Збереження в базі даних

Після роботи етапів шифрування та накладання ЦВЗ, сервер відсилає оброблені дані в базу даних, де вони і будуть зберігатись.

Крок 6: Оновлення списку текстур:

Після завантаження нової текстури в базу даних, список текстур на сайті оновлюється задля відображення нових товарів.

Алгоритм роботи модуля захисту був зображений на окремому рисунку, задля кращого візуального розуміння (рис. 2.1).

Рисунок 2.1 – Алгоритм роботи модуля захисту

Вибір схеми шифрування повинен ґрунтуватися на різних факторах, таких як зручність використання, безпека, продуктивність і підтримка: OpenSSL має наступні особливості в порівнянні з іншими схемами шифрування в PHP:

1. Вбудована підтримка в PHP: OpenSSL є стандартним рішенням для роботи з криптографією в PHP і вбудований в базову конфігурацію PHP. Це означає, що для використання OpenSSL не потрібно встановлювати додаткові розширення або бібліотеки, що робить його простим у використанні та обслуговуванні.

2. Широка функціональність: OpenSSL пропонує широкий спектр функцій

шифрування, включаючи різні алгоритми шифрування (RSA, AES і т.д.), хеш-функції і цифрові підписи, що робить його потужним інструментом для різних потреб шифрування і безпеки даних [16].

3. Довіра та безпека: OpenSSL - одна з найпоширеніших криптографічних бібліотек з відкритим вихідним кодом. Вона пройшла численні аудити безпеки та часові тести, і їй довіряють за її безпеку та надійність.

4. Підтримка різних алгоритмів: OpenSSL підтримує різні алгоритми шифрування і хешування, що дозволяє вибрати найкращий метод для конкретних потреб вашого проекту або програми.

5. Продуктивність: OpenSSL відомий своєю високою продуктивністю. Це особливо важливо для додатків, які обробляють великі обсяги даних або вимагають негайної реакції.

6. Широка спільнота та документація: Оскільки OpenSSL є широко використовуваним рішенням, він має велику спільноту користувачів і хорошу документацію.

7. В цілому, вибір OpenSSL для шифрування в PHP виправданий його надійністю, широким спектром можливостей, швидкістю і простотою використання.

Можливе використання інших методів шифрування, але вони не мають таких значних переваг перед обраним методом.

1. Mcrypt:

– переваги: широко використовувана бібліотека шифрування в PHP до версії PHP 7.1. Має простий інтерфейс і підтримує різні алгоритми шифрування;

– недоліки: Mcrypt застаріла і не підтримується останніми версіями PHP. Це може призвести до проблем з безпекою та сумісністю.

2. Sodium (libsodium):

– переваги: Sodium - це сучасна бібліотека криптографії, яка надає високорівневий інтерфейс для безпечного шифрування та інших криптографічних операцій. Входить до складу PHP, починаючи з версії 7.2;

– недоліки: Sodium може бути складним у використанні для користувачів, які

звикли до інтерфейсу OpenSSL; деякі функції, доступні в OpenSSL, можуть бути складними або недоступними в Sodium [17].

3. Розширення шифрування PHP (PECL):

– перевага: розширення PHP, яке додає підтримку шифрування до PHP. Може включати деякі додаткові можливості та оптимізації, недоступні у стандартній версії PHP або інших бібліотеках;

– недоліки: у деяких середовищах розширення PECL може встановлюватися важко або нестабільно. Крім того, може знадобитися додаткова конфігурація та адміністрування з боку адміністратора сервера.

OpenSSL може працювати в декількох режимах, включаючи RSA (Rivest-Shamir-Adleman) RSA - одна з найвідоміших і широко використовуваних схем асиметричного шифрування, заснована на принципі використання ключа для шифрування і дешифрування даних.

RSA був винайдений в 1977 році дослідниками Роном Рівестом, Аді Шаміром і Леонардом Адлеманом в Массачусетському технологічному інституті (MIT) в США.

Вони опублікували статтю "Метод отримання цифрових підписів і криптосистем з відкритим ключем", в якій представили криптографічний алгоритм і робочий протокол для отримання цифрових підписів і генерації асиметричних ключів.

Після вибору схеми шифрування, потрібно обрати методи вбудовування ЦВЗ та метод шифрування файлу текстури.

RSA базується на математичних принципах теорії чисел, зокрема на складності факторизації великих простих чисел. Основна ідея полягає в тому, що відкритий ключ використовується для шифрування повідомлення, а закритий ключ - для його розшифрування.

Процес генерації ключа включає в себе генерацію двох великих простих чисел, їх перемноження для отримання відкритого ключа, а потім обчислення приватного ключа на його основі. RSA широко використовується для забезпечення безпеки таких веб-протоколів, як HTTPS, SSH і SSL/TLS. Він також використовується для

створення цифрових підписів, автентифікації користувачів і шифрування даних електронної пошти.

Основна особливість безпеки RSA полягає в тому, що йому складно обчислювати великі числа. Чим більше число, тим складніше розкласти його на прості множники.

Розвиток квантових комп'ютерів, які можуть вирішити проблему факторизації великих чисел швидше, може принести нові виклики для безпеки RSA [18].

RSA є одним з найбільш важливих і ефективних криптографічних методів в сучасному інформаційному світі, і його винахідники повинні бути визнані за їх внесок в криптографію та інформаційну безпеку.

2.2 Розробка архітектури сервісу для інтеграції модуля захисту

Задля практичного застосування розробленого модуля захисту, було вирішено розробити онлайн сервіс для продажу та зберігання двовимірних текстур.

Під час проектування для сервісу було обрано архітектуру MVC: Принцип MVC (Model-View-Controller) - це архітектурна модель, яка використовується при проектуванні та розробці програмного забезпечення. Ця модель розділяє систему на три взаємопов'язані частини: модель даних, представлення (інтерфейс користувача) і модуль управління (рис. 2.2).

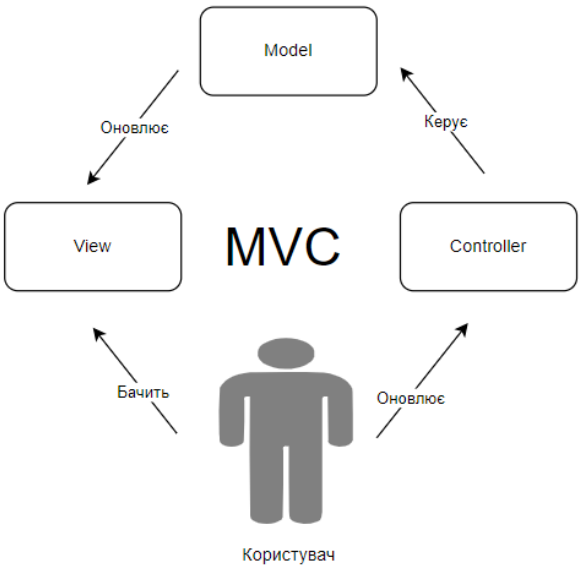


Рисунок 2.2 - Шаблон проектування MVC

Вона складається з таких компонентів:

1. Model - набір низькорівневих функцій для маніпулювання базою даних.

2. Views - файли представлень. На практиці це набір HTML файлів.

3. Controller - сполучна ланка між моделлю та представленням. Він приймає запити, обробляє їх і повертає користувачеві сформовану сторінку.

Шаблони використовуються для відокремлення даних (моделі) від користувацького інтерфейсу (подання). Це гарантує, що зміни в інтерфейсі користувача мають мінімальний вплив на дані і що зміни можуть бути внесені в модель даних без зміни інтерфейсу користувача [19].

Основні елементи архітектури системи включають

– структура бази даних;

– взаємодія між клієнтом і сервером;

– методи зберігання моделей/даних/результатів;

– спосіб аналізу та обробки результатів.

Взаємодія між клієнтом і сервером базується на використанні mySql запитів для передачі даних. Необхідно було узгодити, яке поле відповідає за кожен тип переданого значення. В результаті вийшов зрозумілий і надійний інтерфейс додатку, який дозволяє користувачеві отримувати бажані результати.

Розроблений веб-ресурс має можливість навчання нейронної мережі на вхідних даних. Я використовував формат JSON - текстовий формат для обміну даними між комп'ютерами; JSON базується на тексті і є зручним для читання людиною. Формат дозволяє описувати об'єкти та інші структури даних. Формат в основному використовується для передачі структурованої інформації через мережі (за допомогою процесу, який називається серіалізацією).

Крім того, враховуючи вимогу зменшення навантаження на обчислювальні ресурси сервера, необхідно було розробити модель для зберігання навчених моделей. Для цього ми запропонували підхід, який використовує відбитки моделей (хеш-суми даних і текстовий код алгоритму).

Важливим аспектом є створення діаграми, яка описує алгоритм використання сервісу з точки зору користувача.

Діаграма варіантів використання описує функціональне призначення системи або те, що система повинна робити (рис. 2.3). Її розробка має наступні цілі:

- розробити початкову концептуальну модель системи, яка згодом буде розроблена у вигляді логічної та фізичної моделі;

- сформулювати загальні вимоги до функціональної поведінки проектованої системи;

- визначити загальні межі та контекст предметної області, що підлягає моделюванню на ранній стадії проектування системи;

- розробити вихідну документацію для взаємодії між розробником системи та замовником або користувачем.

На діаграмі показані користувачі системи та інші компоненти, тобто розроблене програмне забезпечення, сервери даних, бази даних і сама система аналізу. Взаємодія з сервісом відбувається через виконання акторами (користувачами) певних операцій. Так, користувач в першу чергу виконує такі операції, як звернення до системи з проханням обробити запит, задання початкових параметрів або запуск процесу обробки чи аналізу даних.

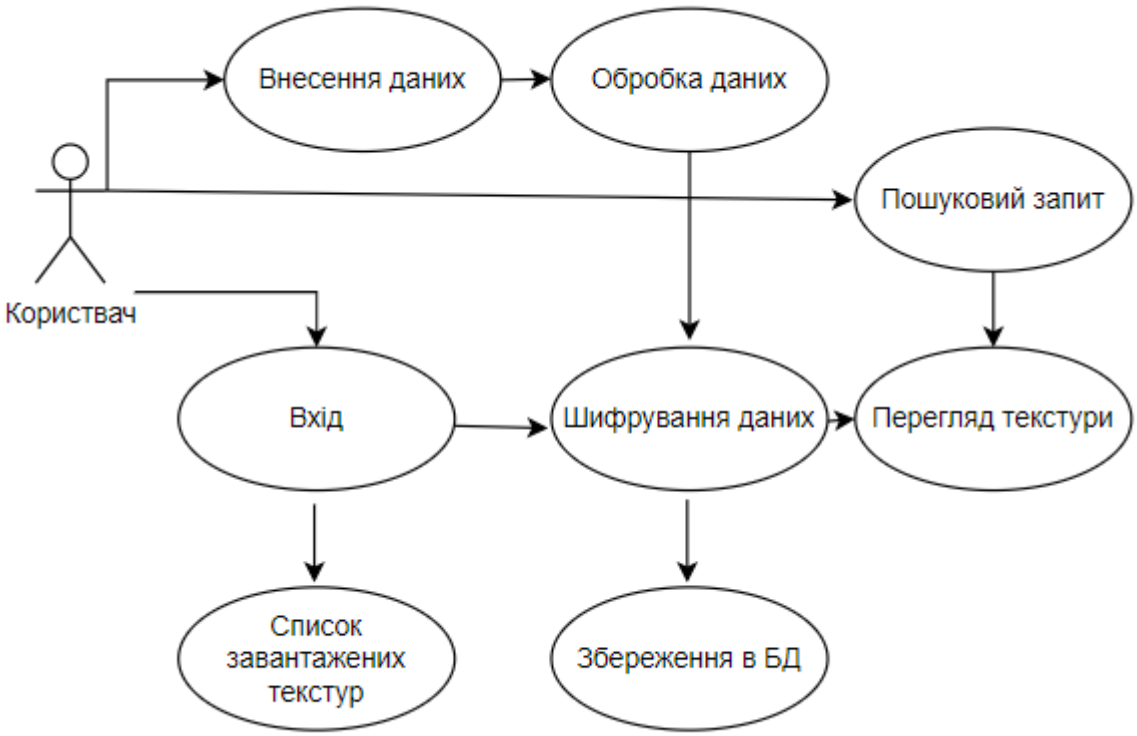


Рисунок 2.3 – Use-case діаграма запропонованого застосунку

Дана діаграма дозволяє покроково зрозуміти алгоритм взаємодії користувача з сервісом. Користувач взаємодіє із системою. Далі описані кроки, які користувач може виконувати.

Користувач. Взаємодіє із системою. Далі описані кроки, які користувач може виконувати:

1. Внесення даних. Користувач вводить дані в систему. До них відносяться файли текстур, прев'ю та відомості авторства.
2. Обробка даних. Введені дані проходять етап обробки на сервері, де вбудовуються цифрові водяні знаки.
3. Пошуковий запит. Після обробки даних користувач може виконати пошуковий запит.
4. Вхід. Користувач проходить автентифікацію.
5. Шифрування даних. Після завантаження дані шифруються для забезпечення безпеки.
6. Збереження в БД. Зашифровані дані зберігаються в базі даних.
7. Перегляд текстури. Користувач може переглянути текстури після виконання пошукового запиту.
8. Доступ до списку завантажених текстур. Користувач має доступ до списку завантажених текстур.

Ці етапи пов'язані між собою:

– після внесення даних відбувається обробка даних;

– після обробки даних можна перейти до створення текстури і шифрування даних;

– вхід до системи дозволяє отримати список завантажених текстур;

– після шифрування даних дані можуть бути збережені в БД;

– пошуковий запит веде до перегляду текстур.

Після створення загальної архітектури сервісу було розроблено план розробки проекту. Для цього потрібно було розбити проект на етапи, використовуючи декомпозицію (рис. 2.4). Декомпозиція - це науковий метод, який використовує структуру завдання, щоб дозволити замінити рішення великої задачі рішенням серії

менших, взаємопов'язаних, але простіших завдань.

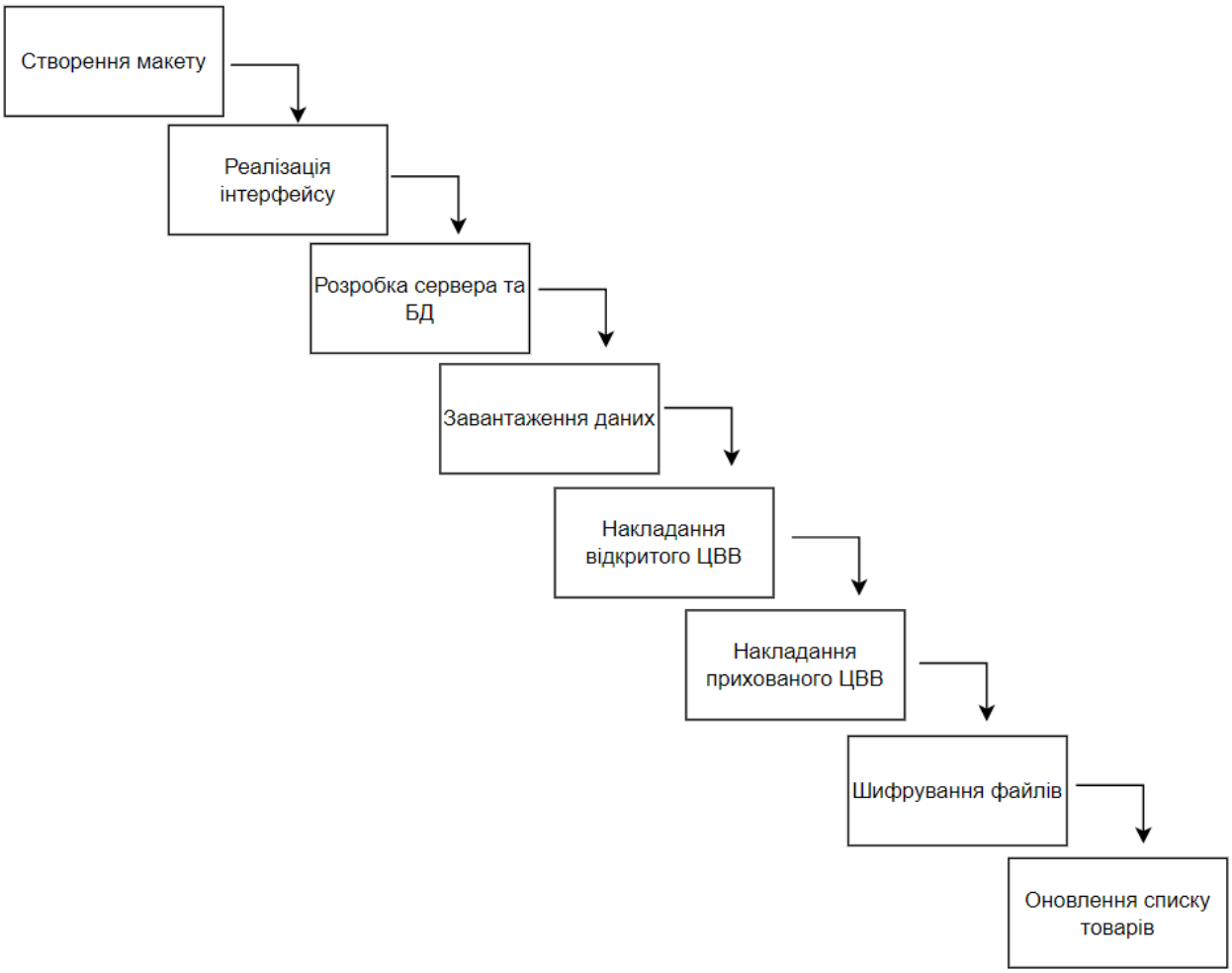


Рисунок 2.4 - Декомпозиція процесу розробки даного сервісу

Оскільки декомпозиція - це процес розкладання, то будь-яку досліджувану систему можна розглядати як складну систему, що складається з окремих взаємопов'язаних підсистем. В якості систем можуть виступати не тільки матеріальні об'єкти, а й процеси, явища і поняття [20].

Дана блок-схема зображує всі можливі процеси сервісу, які потрібно реалізувати:

1. Створення макету. Першим етапом є створення макету дизайну сервісу.
2. Реалізація інтерфейсу. Другим етапом є формування інтерфейсу відповідно до шаблону та внесення його в шаблонізатор.
3. Розробка сервера та БД. Після цього варто створити сервер, який буде передавати дані з сервісу на БД та саму базу даних, яка зберігатиме всю необхідну інформацію.

4. Завантаження. Потрібно реалізувати функцію, яка буде слугувати завантаженням водяного знаку, прев'ю та файлу текстури.

5. Накладання відкритого ЦВЗ. Наступним є реалізація накладання ЦВЗ на прев'ю.

6. Накладання прихованого ЦВЗ. Після цього варто реалізувати накладання ЦВЗ на сам файл текстури.

7. Шифрування. Важливим є реалізація шифрування файлів текстур перед зберіганням.

8. Оновлення списку товарів. Після внесення нового товару потрібно створити функцію, яка буде відображати його на вкладкі «Нові».

Діаграми діяльності використовуються для моделювання поведінки проектної системи, де необхідно детально описати алгоритми та логіку виконання операцій. Основною метою є використання блок-схем та діаграм алгоритмів. Кожна діаграма фокусується на конкретному процесі або послідовності процесів, які приводять користувача до бажаного результату. При моделюванні процесів за допомогою мови UML також використовуються діаграми діяльності. Діаграми діяльності представляються у вигляді графів, де вершини представляють стани, а дуги - переходи з одного стану або типу в інший (рис. 2.5).

Іншими словами, під роботою з діаграмами діяльності розуміють показ, візуалізацію та експонування особливостей реалізації операцій класу, коли необхідно зобразити роботу алгоритму та перебіг його процесів. При цьому кожна дія та її стан виконується операцією певного класу, а її тип є процесом послідовності дій [21].

Елементи діаграми діяльності:

1. Вхід до сервісу:
   - Це початкова точка, з якої починається процес.
2. Користувач авторизований:
   - Вирішується, чи користувач вже авторизований на сайті.
   - Якщо так (Yes), процес переходить до перегляду товарів.
   - Якщо ні (No), користувач перенаправляється до блоку реєстрації.

3. Користувач зареєстрований:

– Якщо так (Yes), процес переходить до форми авторизації.

– Якщо ні (No), користувач перенаправляється до форми реєстрації

4. Дані внесено вірно:

– Перевіряється правильність введених даних.

– Якщо так (Yes), відбувається авторизація користувача на сайті.

– Якщо ні (No), користувач повертається до форми авторизації для виправлення даних.

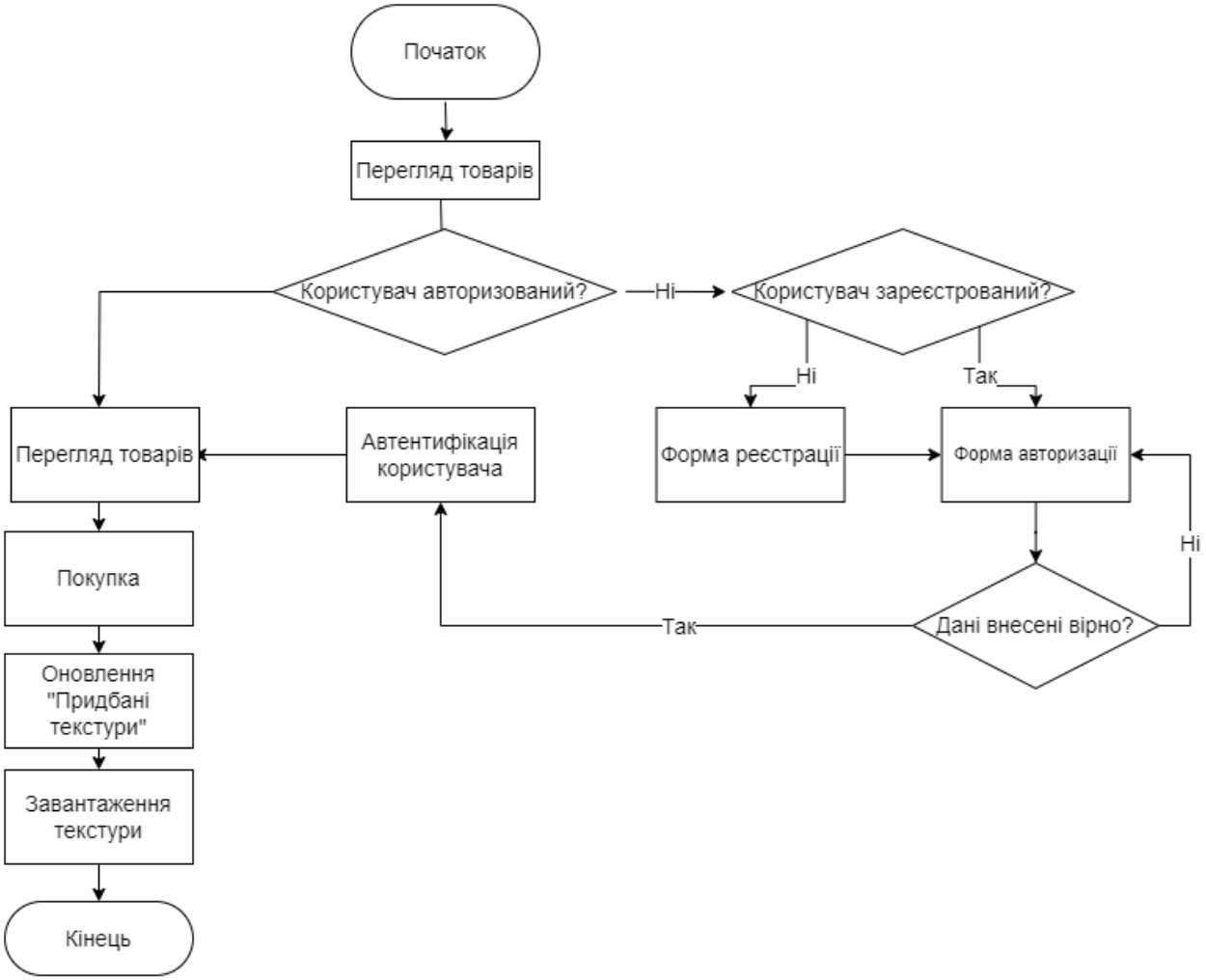


Рисунок 2.5 – Структурна схема функціонування розробленого сервісу

5. Авторизація користувача на сайті:

– Після успішної авторизації користувач отримує доступ до можливостей сайту і переходить до перегляду товарів.

6. Перегляд товарів:
   – Користувач переглядає доступні товари на сайті.
7. Покупка:
   – Користувач обирає товар і здійснює покупку.
8. Оновлення «Придбані текстури»:
   – Після покупки товар з’являється в вкладкі «Придбані текстури».
9. Завантаження текстури
   – Після завантаження текстури, вона розшифровується та зберігається на комп’ютер покупця.

Наступним етапом є створення діаграми, яка показує зв’язки інтерфейсу з сервером та базою даних.

Діаграма, що показує обчислювальні вузли, називається діаграмою розгортання. Компоненти на цій діаграмі представляють екземпляри одиниць коду. Діаграми розгортання показують запущені екземпляри компонентів, а діаграми компонентів показують зв'язки між типами компонентів (рис. 2.6).

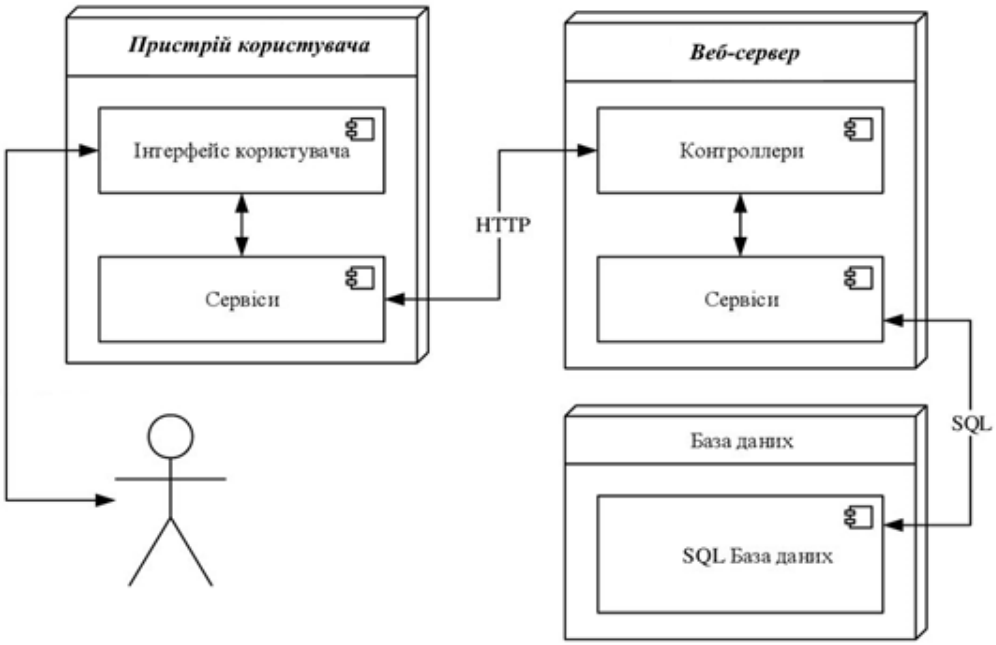


Рисунок 2.6 – Діаграма розгортання розглянутого сервісу

Останнім етапом є створення архітектури бази даних, оскільки це дуже важлива складова проекту. Для реалізації поставлених задач нам буде достатньо бази даних, яка складається з трьох таблиць:

1. Таблиця «Користувачі».
2. Таблиця «Текстури».
3. Таблиця «Покупки».

Таблиця, яка містить дані про користувачів буде відповідальною за авторизацію користувачів та зберігання їх даних. Вона повинна містити наступні атрибути: ID користувача, його ім’я, логін, нікнейм для авторства та пароль, який буде зберігатись у вигляді геш-суми.

Таблиця «Текстури» буде зберігати відомості про товари. Саме з неї будуть відображатись текстури на сайті. Вона міститиме наступні атрибути: ID текстури, назва текстури, опис, ціна, категорія, ID автора.

Останньою таблицею є «Покупки». Це невеличка таблиця, яка об’єднує користувачів та текстури. Вона містить лише 3 атрибути: ID покупки, ID користувача, ID текстури.

Також, було створено UML-діаграму для відображення зв’язків між таблицями (рис. 2.7).

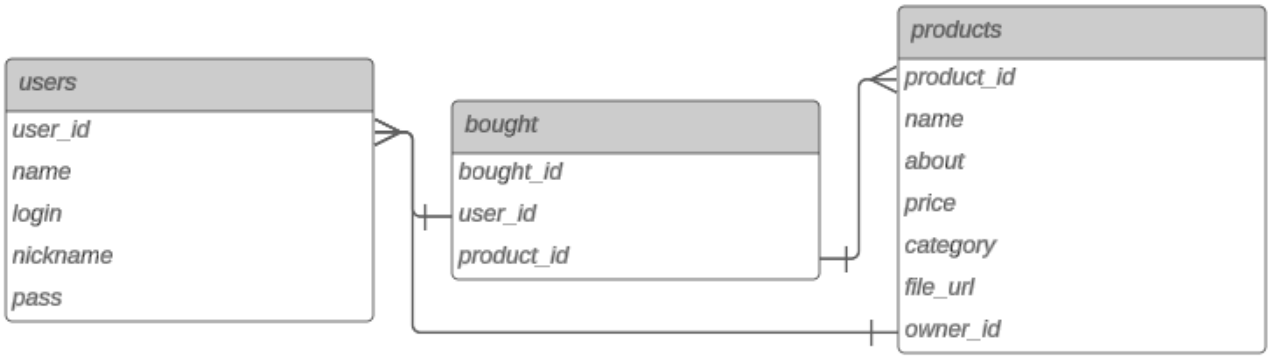


Рисунок 2.7 – UML-діаграма бази даних даного сервісу

В результаті було отримано повністю сформовану архітектуру сервісу та його компонентів. Важливим є розбиття процесу розробки на менші етапи, оскільки це дуже допомагає під час практичної реалізації розуміти послідовність дій. Також, важливим є розуміння побудови сервісу для обрання мови програмування, яка найкраще підходить під даний проект.

2.3 Обґрунтування вибору середовища розробки для реалізації модуля захисту

Оскільки розроблювані сервіси вимагають можливості обробки великих обсягів даних, а вибір технології, яка буде використовуватися для програми, має вирішальне значення для створення безперебійно функціонуючого інтернет-ресурсу, перед початком розробки було досліджено доступні інструменти та порівняно їхні плюси та мінуси, щоб переконатися, що в проекті використовується найефективніша технологія.

Тому, залежно від завдання аналізу, слід вибрати мову програмування, яка зможе виконати поставлене завдання.

Наприклад, скриптова мова PHP була створена для створення HTML-сторінок на стороні веб-сервера; PHP є однією з найпоширеніших мов для веб-розробки; PHP підтримується більшістю хостинг-провайдерів; PHP є програмним проектом з відкритим вихідним кодом PHP є програмним проектом з відкритим вихідним кодом [22].

PHP інтерпретується веб-сервером і перетворюється на HTML-код, який надсилається клієнтській стороні; на відміну від скриптових мов JavaScript, браузер отримує готовий HTML-код, і користувач ніколи не бачить PHP-коду. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінки.

PHP має вбудовані бібліотеки для роботи з MySQL і PostgreSQL; завдяки стандарту Open Database Interface він може підключатися до будь-якої бази даних, для якої доступний драйвер.

Традиційність - PHP знайома програмістам, які працюють у різних галузях. Багато мовних конструкцій запозичено з C та Perl; код PHP дуже схожий на код типових програм на C та Pascal. Сирий код доступний і безкоштовний. Ефективність - більшість PHP-скриптів (особливо невеликих за розміром) обробляються швидше, ніж аналогічні програми, написані на Perl.

Високорівнева динамічна мова програмування загального призначення. Perl - це мова програмування загального призначення, яка спочатку була розроблена як

програмне забезпечення для обробки текстів, а зараз використовується для вирішення широкого спектру завдань, включаючи системне адміністрування, веб-розробку, розробку мережевого програмного забезпечення та програмного забезпечення з графічним інтерфейсом користувача.

Мова робить акцент на практичності (простота використання, ефективність, повнота), а не на естетиці (маленький, елегантний, мінімалістичний). Мова має багато можливостей, таких як підтримка різних парадигм програмування (процедурне програмування, об'єктно-орієнтоване програмування, функціональне програмування), управління пам'яттю, вбудована підтримка систем обробки текстів, підтримка великого набору сторонніх модулів [23].

Загальна структура мови Perl, як правило, походить від мови C. Perl є процедурною, зі змінними, операторами присвоювання, блоками коду, розділеними круглими дужками, керуючими структурами та функціями.

Perl також запозичив багато особливостей з мови програмування оболонки UNIX. Всі змінні позначаються провідним символом, який однозначно вказує на тип даних змінної (наприклад, скаляр, масив, хеш). Що ще важливіше, ці символи дозволяють перетворювати змінні в рядки, і Perl має ряд вбудованих функцій, які надають інструменти, що зазвичай використовуються в програмуванні в командній оболонці, такі як сортування і виклик системних служб.

У Perl 5 додано підтримку складних типів даних, функцій першого класу та об'єктної моделі. Остання включає в себе посилання, пакети, виконання методів класу, змінні з оголошенням лексичної області видимості, директиви компілятора тощо. Найважливішим покращенням, запровадженим у Perl 5, є можливість вбудовувати код у "пакети" як модулі для повторного використання. У всіх версіях Perl є автоматичне введення даних і автоматична перевірка пам'яті. Інтерпретатор знає тип і вимоги до пам'яті кожного об'єкта в програмі і виділяє та звільняє пам'ять шляхом підрахунку посилань. Перетворення з одного типу даних в інший (наприклад, з числа в рядок) відбувається автоматично під час виконання, а неможливі перетворення типів даних є фатальними помилками.

Ruby - повністю об'єктно-орієнтована мова програмування, інтерпретована з відкритою динамічною типізацією, що поєднує Perl-подібний синтаксис з об'єктно-орієнтованим підходом мови програмування Smalltalk. Вона також запозичує деякі функції з таких мов програмування, як Python, Lisp, Dylan та CLU.

Багатоплатформна реалізація інтерпретатора мови Ruby поширюється як вільне програмне забезпечення. Вихідний код проекту поширюється під ліцензією BSD ("2 BSDL") та ліцензією Ruby, яка посилається на останню версію ліцензії GPL і повністю сумісна з v3 [24].

При виборі мови програмування ключовим параметром залишалася ефективність. Це дуже важливий фактор при програмуванні в багатокористувацьких середовищах, в тому числі в Інтернеті.

Дуже важливою перевагою PHP є його "двигун": PHP не має ні компілятора, ні інтерпретатора. Мова є широкомовним інтерпретатором, і цей апарат двигуна PHP дозволяє йому обробляти скрипти на дуже високих швидкостях.

За деякими оцінками, більшість PHP-скриптів (особливо невеликих) обробляються швидше, ніж аналогічні програми, написані на Perl.

PHP також надає розробникам і адміністраторам гнучкі та ефективні інструменти безпеки. Вони поділяються на дві категорії: інструменти системного рівня та інструменти рівня додатків.

PHP реалізує механізми безпеки, які під управлінням адміністратора і при правильному налаштуванні забезпечують максимальну свободу і безпеку. Безпечний режим дозволяє користувачам накладати ряд важливих обмежень при використанні PHP. Наприклад, можна обмежити максимальний час виконання і використання пам'яті (неконтрольоване споживання пам'яті негативно впливає на продуктивність сервера). Як і у випадку з cgi-bin, адміністратори можуть встановити обмеження на директорії, в яких користувачі можуть переглядати і виконувати PHP-скрипти, а також використовувати PHP-скрипти для перегляду конфіденційної інформації (наприклад, файлів passwd) на сервері.

Стандартний набір функцій PHP включає ряд надійних механізмів шифрування. PHP також сумісний з багатьма сторонніми додатками і легко

інтегрується з безпечними технологіями електронної комерції. Ще однією перевагою є те, що вихідний текст PHP-скриптів може відображатися в браузері. Реалізація ГВЧ на стороні сервера запобігає перехопленню нетривіальних скриптів користувачами, які знають достатньо, щоб виконати команду View Source.

Оскільки PHP є вбудованою мовою, вона є дуже гнучкою відповідно до потреб розробників. Хоча PHP часто рекомендують використовувати разом з HTML, її можна однаково добре інтегрувати з JavaScript, WML, XML та іншими мовами. Крім того, добре структурований PHP-додаток можна легко розширювати за потреби (але це справедливо для всіх основних мов програмування).

PHP-скрипти повністю компілюються на стороні сервера перед відправкою клієнту, тому немає проблеми залежності від браузера. Фактично, PHP-скрипти можна надсилати на будь-який пристрій з браузером, включаючи традиційні ПК, а також мобільні телефони, електронні ноутбуки, пейджери та ноутбуки. Програмісти, які працюють з допоміжними програмами, можуть запускати PHP в режимі командного рядка.

PHP не містить специфічного для веб-сервера коду, тому користувачі не обмежені певним сервером (з яким вони можуть бути незнайомі): Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold, Zeus - всі вони підтримуються. PHP працює на всіх цих серверах. Оскільки ці сервери працюють на різних платформах, PHP зазвичай є портативною, автономною мовою і працює на таких платформах, як UNIX, Solaris, FreeBSD, Windows 95/98/NT/2000/XP/2003.

Враховуючи всі переваги PHP, було обрано саме цю мову програмування дл янаписання застосунку. Наступним етапом є аналіз та обрання бази даних для реалізації сервісу.

PostgreSQL - це безкоштовна об'єктно-реляційна система управління базами даних (СКБД), яка працює на AIX, різних системах BSD, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, Microsoft. Програми доступні для різних UNIX-подібних платформ, включаючи AIX, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX і Microsoft Windows [25].

Перевагами PostgreSQL є:

– висока продуктивність та надійні механізми транзакцій та реплікації;

– спадковість;

– легка розширюваність.

На відміну від інших проектів з відкритим вихідним кодом, таких як Apache, FreeBSD і MySQL, PostgreSQL не управляється конкретною компанією, а може розвиватися у співпраці з багатьма людьми і компаніями, які хочуть використовувати цю СУБД і впроваджувати останні досягнення. Сервер PostgreSQL - це база даних на мові C++.

Сервер PostgreSQL написаний на мові C. Зазвичай він розповсюджується у вигляді набору текстових файлів, що містять сирий код. Щоб встановити його, файли потрібно скомпілювати на комп'ютері і скопіювати в певний каталог. Весь процес детально описаний в документації.

PostgreSQL - це широко використовувана система управління базами даних з відкритим вихідним кодом. Третій реліз додає підтримку одночасної роботи декількох менеджерів зберігання даних, покращений механізм запитів і розширену внутрішню систему правил. В даний час POSTGRES використовується для реалізації великих систем, таких як системи аналізу фінансових даних, пакети моніторингу потокових функцій, бази даних для відстеження астероїдів, медичні інформаційні системи та численні географічні системи. POSTGRES також використовується як навчальний інструмент в різних університетах.

MySQL - це рішення для малих і середніх додатків. Вона пропонує можливість роботи як з таблицями MyISAM, які підтримують повнотекстовий пошук, так і з таблицями InnoDB, які підтримують операції на рівні окремих записів [26].

MySQL має API і роз'єми для Delphi, C, Java, Lisp, Perl, Python, Ruby, Smalltalk, Component Pascal і Tcl, а також бібліотеку для мов платформи .NET з підтримкою ODBC-драйвера MyODBC.

MongoDB підтримує зберігання документів у JSON-подібному форматі, має гнучку мову для запитів, може створювати індекси для різних атрибутів зберігання, забезпечує ефективне зберігання великих двійкових об'єктів, підтримує журнали

транзакцій для модифікації та вставки даних в базу даних, підтримує журнали транзакцій.

Розподілене зберігання пар ключ-значення в оперативній пам'яті з можливістю забезпечення довговічності зберігання відповідно до вимог користувача. Програмне забезпечення з відкритим вихідним кодом написано мовою ANSI C. Розробка Redis фінансується компанією VMware. Вихідний код проекту поширюється під ліцензією BSD.

Redis надає подібну до Memcached функціональність для зберігання даних ключ/значення, розширену підтримкою структурованих даних, таких як списки, хеші, кластери тощо. На відміну від Memcached, Redis зберігає дані постійно на диску і гарантує безпеку бази даних у разі збою. Клієнтські бібліотеки доступні для найпопулярніших мов, включаючи Perl, Python і Tcl.

Серед можливих баз даних я обрав MySQL через її швидкість та інші параметри, описані нижче.

MySQL дозволяє отримувати доступ до даних, що зберігаються в базі даних, та маніпулювати ними, захищає дані від пошкодження та порушення цілісності, а також надає метадані, необхідні для опису даних Основна відмінність між СУБД та СКБД полягає в тому, що остання має справу лише з реляційними базами даних, а не з таблицями чи зв'язками між таблицями.

MySQL - це безкоштовна система управління реляційними базами даних. Ця система управління базами даних (СУБД) з відкритим вихідним кодом була створена як альтернатива комерційним системам.

В даний час MySQL є однією з найбільш широко використовуваних систем управління базами даних. Сумісна з широким спектром мов програмування, MySQL в першу чергу використовується для створення динамічних веб-сторінок. MySQL надає багатий набір функцій, які підтримують безпечне середовище для зберігання, управління та пошуку даних.

Вона характеризується швидкістю, гнучкістю та простотою використання і була розроблена для підвищення продуктивності транзакцій у великих базах даних. Вихідний код сервера скомпільовано на багатьох платформах. Можливості сервера

найкраще реалізуються на UNIX-системах, де багатоканальна підтримка покращує загальну продуктивність системи (рис. 2.1).

MySQL можна безкоштовно використовувати в некомерційних цілях.

Переваги сервера MySQL:

– його легко встановити та використовувати;

– необмежена кількість користувачів може одночасно працювати з базою даних;

– таблиці можуть містити до 50 мільйонів рядків;

– висока швидкість виконання команд;

– проста та ефективна система безпеки.

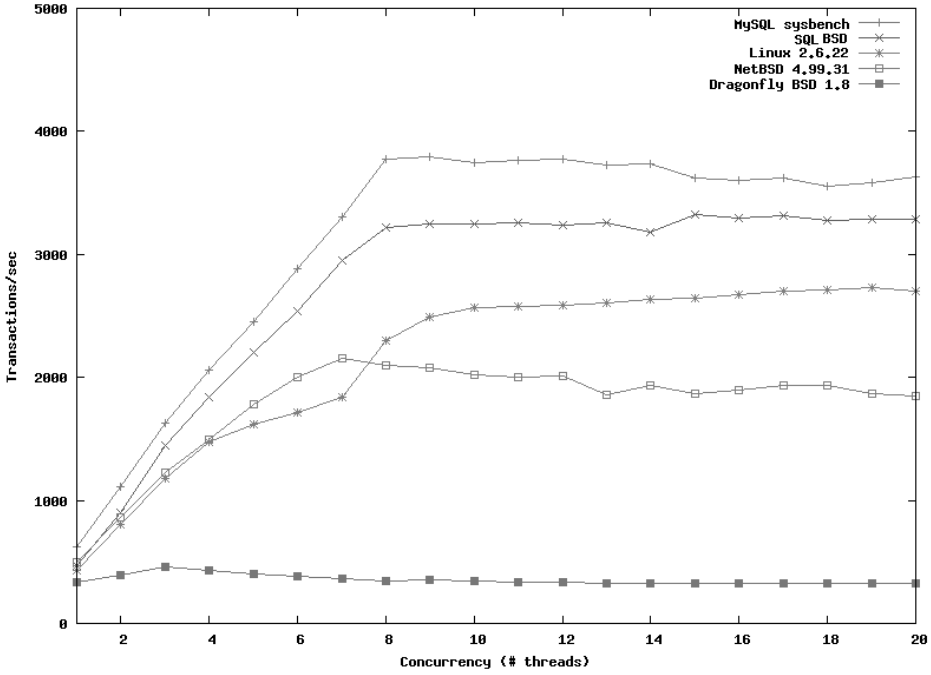


Рисунок 2.1 - Швидкодія MySQL

MySQL може підтримувати роботу баз даних значного розміру. Згідно з документацією, що супроводжує MySQL, деякі бази даних, що використовуються компанією MySQL AB (розробником MySQL), зберігають до 50 мільйонів записів.

Переносимість MySQL працює на різних платформах, включаючи Unix, Linux, Windows, OS/2, Solaris і Mac OC. Крім того, MySQL працює на широкому спектрі платформ [27].

Безпека MySQL має систему контролю доступу до даних, яка шифрує дані під час передачі.

Дуже високий рівень безпеки забезпечує база даних mysql, яка створюється під час встановлення пакета і містить п'ять таблиць. За допомогою цих таблиць можна визначити, які користувачі з якого домену можуть працювати над якими таблицями і які команди можуть використовувати. Паролі, що зберігаються в базі даних, можуть бути зашифровані за допомогою функції паролів, вбудованої в MySQL.

Оскільки код має відкритий вихідний код, необхідну функціональність можна додавати до пакету, а функціональність можна розширювати за потреби.

Завдяки відкритому коду, безкоштовній, стабільній та надійній роботі, сформувалася спільнота, яка не тільки лояльно ставиться до MySQL, але й бере участь у розвитку пакету та навчає недосвідчених людей, як ним користуватися. Існують численні списки розсилки та конференції, де можна отримати безкоштовну допомогу в будь-який час доби [28].

Враховуючи всі переваги MySQL та проаналізувавши аналоги, було обрано саме цю базу даних для реалізації проекту.

Наразі також існують версії програми для більшості поширених комп'ютерних платформ. Це означає, що вам не обов'язково використовувати певну операційну систему. Ви можете вибрати, з якою операційною системою ви хочете працювати - наприклад, Linux або Windows - але якщо ви зміните операційну систему, ви не втратите жодних даних і вам не знадобляться додаткові інструменти для їх перенесення.

2.4 **Висновки до розділу 2**

Для розробки сервісу обрано мову програмування PHP. Цей вибір зумовлений його широким застосуванням у веб-розробці, високою швидкістю виконання коду та великою кількістю готових рішень та фреймворків, що спрощують розробку та підтримку проекту.

Архітектура сервісу буде побудована за принципом клієнт-серверної взаємодії методом MVC. Клієнтська частина буде реалізована з використанням HTML, CSS

та JavaScript для взаємодії з користувачем у браузері. Серверна частина буде написана на мові програмування PHP і відповідатиме за обробку запитів користувачів, зберігання та відправлення даних до бази даних.

Вибір методів шифрування текстур та вбудовування водяних знаків в них обґрунтований їхньою ефективністю та стандартами безпеки. Ми вирішили використовувати алгоритми шифрування RSA для забезпечення конфіденційності та цілісності даних, а також водяні знаки для захисту від несанкціонованого використання текстур. Ці методи будуть застосовані на етапі завантаження та зберігання текстур в нашій базі даних.

Розглядаючи використання RSA засобами бібліотеки OpenSSL, ми провели ретельний аналіз і виявили його високу ефективність та надійність у забезпеченні криптографічного захисту нашого сервісу. Цей метод виявився не лише ефективним, але й надійним, надаючи нам засоби для захисту конфіденційної інформації на всіх етапах обміну даними. Використання RSA засобами OpenSSL виявилося ключовим аспектом у підвищенні безпеки та довіри до нашого інтернет-магазину.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОГО МОДУЛЮ ЗАХИСТУ ДВОВИМІРНИХ ТЕКСТУР

В даному розділі буде оглянуто процес реалізації розробленого сервісу. Він включає в себе розробку дизайну, реалізацію інтерфейсу, створення серверу та налаштування бази даних. Також, в даному розділі буде розглянуто процес реалізації модулю захисту та інтеграцію його в сервіс.

3.1 Розробка модулю захисту двовимірних текстур для онлайн-сервісу

Розроблений модуль захисту включає в себе дві складові: вбудовування цифрових водяних знаків та шифрування файлів текстур. Першим було реалізовано шифрування текстур за допомогою алгоритму RSA перед зберіганням їх в базу даних.

Загалом модуль шифрування текстур повинен включати два етапи. Перший це шифрування даних, другий це розшифрування при завантаженні користувачем. Ця функція виконує шифрування та розшифрування зображення за допомогою асиметричного шифрування з використанням ключів RSA. Для цього було реалізовано наступні етапи роботи криптоалгоритму:

1. Генерація ключів. Спочатку функція генерує пару ключів - приватний і публічний ключі. Розмір приватного ключа 2048 біт, тип ключа - RSA.
2. Отримання публічного ключа. Публічний ключ отримується з приватного ключа.
3. Зчитування оригінального зображення. Функція зчитує вміст файлу зображення у бінарному форматі.
4. Оригінальні дані зображення шифруються за допомогою публічного ключа, використовуючи функцію openssl_public_encrypt. Зашифровані дані зображення зберігаються у змінній $encryptedImageData.
5. Запис зашифрованого зображення.
6. Розшифрування зашифрованого зображення при завантаженні
7. Функція читає зашифровані дані з файлу.
8. Зашифровані дані розшифровуються за допомогою приватного ключа,

використовуючи функцію openssl_private_decrypt.

9. Розшифровані дані зображення записуються у файл.

У підсумку, ця функція дозволяє шифрувати зображення за допомогою публічного ключа та розшифровувати його за допомогою приватного ключа.

```
// Генеруємо ключі (публічний і приватний)
$privateKey = openssl_pkey_new(array(
   "private_key_bits" => 2048,
   "private_key_type" => OPENSSL_KEYTYPE_RSA,
));
// Отримуємо публічний ключ з приватного
openssl_pkey_export($privateKey, $privateKeyPem);
$publicKey = openssl_pkey_get_details($privateKey)["key"];
// Шлях до оригінального зображення
$imagePath = $this->img';
// Зчитуємо вміст файлу зображення у бінарному форматі
$imageData = file_get_contents($imagePath);
// Шифруємо дані зображення за допомогою публічного ключа
openssl_public_encrypt($imageData, $encryptedImageData, $publicKey);
// Шлях для збереження зашифрованого зображення
$encryptedImagePath = '$this->url;
// Записуємо зашифровані дані у файл
file_put_contents($encryptedImagePath, $encryptedImageData);
// Читаємо зашифровані дані з файлу
$encryptedImageData = file_get_contents($encryptedImagePath);
```

Після завершення шифрування даних, зашифрований файл зберігається в базі даних, а після цього виконується оновлення сторінки задля відображення нового елементу. Після покупки текстури іншим користувачем, файл розшифровується на сервері, після чого завантажується на комп’ютер власника [29]. Після покупки користувач отримує розшифрований файл. Функція, відповідальна за розшифрування даних наведена нижче:

```
// Розшифровуємо зашифровані дані зображення за допомогою приватного ключа
openssl_private_decrypt($encryptedImageData, $decryptedImageData, $privateKey);
// Шлях для збереження розшифрованого зображення
$decryptedImagePath = '$this->sabeUrl';
// Записуємо розшифровані дані у файл
file_put_contents($decryptedImagePath, $decryptedImageData);
```

Після реалізації шифрування даних, потрібно реалізувати функцію вбудовування ЦВЗ, яка в свою чергу поділяється на два етапи: вбудовування видимого цифрового водяного знаку в прев'ю текстури на сайті та вбудовування прихованого цифрового водяного знаку в файл текстури.

Функція генерує випадкові значення для кольору тексту, розміру та розташування вотермарку на зображенні, що робить кожен вотермарк унікальним і важким для видалення.

Вотермарк додається до зображення без зміни його основного змісту або якості. Застосування вотермарку ускладнює незаконне копіювання або використання зображень без дозволу.

Першим було реалізовано вбудовування ЦВЗ в прев'ю:

```
$img = ImageCreateFromJPEG($existFile); // Відкриття існуючого зображення
$font = dirname(__FILE__) . '/fonts/Arial.ttf'; // Визначення шрифту
$text = $mark; // Текст вотермарку
// Додавання вотермарку з випадковими значеннями для кольору та розташування тексту
$color = imagecolorallocatealpha($img, random_int(0, 250), random_int(0, 250), random_int(0, 250), random_int(50, 120));
imagettftext($img, random_int(14, 30), 0, 25, 25, $color, $font, $text);
imagettftext($img, random_int(14, 30), 0, 100, 100, $color, $font, $text);
$color = imagecolorallocatealpha($img, random_int(0, 250), random_int(0, 250), random_int(0, 250), random_int(50, 120));
imagettftext($img, random_int(14, 30), 0, 75, 175, $color, $font, $text);
imagettftext($img, random_int(14, 30), 0, 50, 250, $color, $font, $text);
imagejpeg($img, $existFile); // Збереження зображення з вотермарком
```

Після цього було реалізовано вбудовування цифрового водяного знаку в текстуру:

```
public function embedLSB($imagePath, $text) {
    // Завантажити зображення
    $image = imagecreatefromjpeg($imagePath)
    // Конвертувати текст в бінарний формат
    $binaryText = '';
    for ($i = 0; $i < strlen($text); $i++) {
            $binaryText .= sprintf("%08d", decbin(ord($text[$i])));
    // Вбудовувати бінарний текст у LSB кожного пікселя
    $textIndex = 0;
```

```
    for ($x = 0; $x < imagesx($image); $x++) {
            for ($y = 0; $y < imagesy($image); $y++) {
    // Отримати RGB колір пікселя
    $rgb = imagecolorat($image, $x, $y);
    $r = ($rgb >> 16) & 0xFF;
    $g = ($rgb >> 8) & 0xFF;
    $b = $rgb & 0xFF;
// Замінити останній біт RGB значення на біт з тексту
    $r = ($r & 0xFE) | ($binaryText[$textIndex++] ?? 0);
   $g = ($g & 0xFE) | ($binaryText[$textIndex++] ?? 0);
    $b = ($b & 0xFE) | ($binaryText[$textIndex++] ?? 0);
    // Змінити колір пікселя на оновлене RGB значення
    imagesetpixel($image, $x, $y, imagecolorallocate($image, $r, $g, $b));
    // Завершити, якщо всі біти тексту вже вбудовані
    if ($textIndex >= strlen($binaryText)) {
                                    break 2;
$uploaddir = 'img-logo/';
```

Ця функція не лише забезпечує захист цифрових зображень, але й надає ефективний механізм для ідентифікації та захисту авторських прав. Застосовуючи цей метод, можна зберегти контроль над використанням контенту в мережі.

В результаті ми отримали реалізований ефективний модуль захисту та інтегрували його в сервіс.

3.2 **Розробка онлайн-сервісу для збереження та продажу двовимірних текстур**

Задля практичної інтеграції модулю захисту в сервіс, було розроблено інтернет-платформу для продажу та зберігання двовимірних текстур.

Першим етапом розробки онлайн-сервісу є створення макету дизайну майбутнього застосунку. Цей етап є дуже важливим у розробці будь-якого сервісу, оскільки користувач буде взаємодіяти саме з інтерфейсом сервісу, а не напряму з сервером чи базою даних. Для розробки дизайну було обрано сервіс Figma.

Figma – це найпопулярніший безкоштовний застосунок для створення дизайну з різних напрямів. В даному застосунку дизайнери створюють макети не лише веб-

сайтів, а й роблять презентації, створюють прев'ю для відео, малюють різні візуали для комерційної діяльності [30].

Розробка дизайну полягала у створенні зручного та інтуїтивно зрозумілого застосунку для користувача. Тому, було розроблено макет веб-сервісу, який містить всі потрібні сторінки (рис. 3.1).

Рисунок 3.1 – Розроблений дизайн

Наступним етапом після розробки дизайну є реалізація візуальної частини застосунку. Для цього використовується мова гіпертекстової розмітки HTML 5 для побудови загальної структури сервісу та каскадна таблиця стилів CSS 3 для надання дизайну побудованій структурі.

Розроблений HTML шаблон було інтегровано в проект за допомогою PHP шаблонізатора що надає можливість працювати з базою даних. Інформація про товари передається у вид де кожен елемент даних перетворюється у структурований гіпертекстовий документ. Фрагмент коду, який відповідає за обмін даними між PHP та HTML:

```
<div class="page__main-block">
    <div class="main-block__container">
            <p class="main-block__title">
              Головна
            </p>
```

```
            <div class="main-block__list">
                    <?php if ( isset($_GET['id']) && $_GET['id'] == 1 ): ?>
                            <a href="/">Всі</a>
                            <a href="?id=1" class="main-block__list_active">Найпопулярніші</a>
                            <a href="?id=2">Нові</a>
                            <a href="?id=3">Найдешевші</a>                                  <?php else: ?>
                            <a href="/" class="main-block__list_active">Всі</a>
                            <a href="?id=1">Найпопулярніші</a>
                            <a href="?id=2">Нові</a>
                            <a href="?id=3">Найдешевші</a>
                    <?php endif; ?>
            </div>
    </div>
</div>
```

Найважливішим елементом сервісу є текстура. Саме дані про текстури будуть зберігатися, оновлюватись та шифруватися. Тому, важливим є створення загального класу для змінної «products», яка виступає текстурою.

Відповідно в змінній «products» міститься інформація про кожен товар. Для виводу всіх товарів потрібно почергово перебрати всі такі змінні в базі даних, відсортувати їх та вивести на інтерфейс:

```
 <div class="page__list">
<?php if (!empty($products)): ?>
    <div class="list__container">
    <?php foreach ($products as $prod): ?>
<a href=in/product?id=<?=$prod->id;?>" class="list__item">
img src="<?=$prod->img;?>" alt="" class="item__img"><p class="item__title"><?=$prod-  >name;?></p>
<p class="item__price"><?=$prod->price;?>$</p>
</a>
                            <?php endforeach; ?>
                    </div>
            <?php endif; ?>
 </div>
```

Після створення інтерфейсу та підключення його до шаблонізатора наступає етап створення серверу. Це частина сервісу, яка слугує посередником між користувачем та базою даних. Саме тут на прев’ю та текстуру накладатимуться

цифрові знаки. Також, саме сервер слугує шифрувальником текстури перед внесенням її до БД. Тому, сервер є головною частиною захисного модуля проекту [31].

На етапі розробки сервісу потрібно просто запустити сервер та реалізувати передачу даних на БД через нього:

```
class Router {
    @var array
   protected static $routes = [];
    @var array
   protected static $route = [];
    @param string $regexp регулярне вираження маршруту
    @param array $route маршрут ([controller, action, params]
   public static function add($regexp, $route = []) {
      self::$routes[$regexp] = $route;
   }
   @return array
   public static function getRoutes() {
      return self::$routes;
   }
    @return array    public static function getRoute() {
      return self::$route;
   }    @param string $url вхідний URL
    @return boolean
```

Після запуску серверу потрібно створити базу даних. Вона міститиме відомості про текстури та потрібні дані для авторизації користувача. Під час створення бази даних потрібно враховувати всі дані, які потрібні для роботи сервісу: назва зображення, ціна, опис, маршрут до файлу, категорія текстури та ідентифікаційний номер користувача.

Для сервісу потрібно створити три таблиці: таблицю текстур, користувачів та таблицю покупок. Їх взаємодія була описана в теоретичній частині розробки. Першою для реалізації було обрано таблицю текстур (рис. 3.2). Вона містить всі потрібні атрибути для роботи сервісу.

id	img	name	about	price	cat	file_url	owner_id
1	/img-logo/1.png	Хліб	Текстура для використання. Висока якість.	180	1	/img-logo/1.png	0
2	/img-logo/2.png	Місяць	Текстура поверхні. Висока якість.	140	2	/img-logo/2.png	0
3	/img-logo/3.png	Корозія	Текстура для використання. Висока якість.	100	3	/img-logo/3.png	0
4	/img-logo/4.png	Метал	Текстура металевої конструкції. Висока якість.	60	3	/img-logo/4.png	0
5	/img-logo/5.png	Сніг	Текстура для використання. Висока якість.	170	2	/img-logo/5.png	0
6	/img-logo/6.png	Лід	Текстура для використання. Висока якість.	150	1	/img-logo/6.png	0
7	/img-logo/7.png	Тканина	Текстура поверхні. Висока якість.	110	2	/img-logo/7.png	0
8	/img-logo/8.png	Метал	Текстура для використання. Висока якість.	70	1	/img-logo/8.png	0
27	/img-logo/171442523526ArtStone1.jpg	Сніг	Опис	100	2	/img-logo/171442523583ArtStone1.jpg	1
28	/img-logo/171442533597ArtStone2.jpg	Сонце	Опис	250	2	/img-logo/171442533592ArtStone2.jpg	1

Рисунок 3.2 – таблиця «products»

Наступним етапом було створення таблиць покупок та користувачів (рис.3.3). Варто зазначити, що таблиця користувачів містить пароль для авторизації, проте задля підвищення захищеності, він зберігається не в стандартному вигляді, а у вигляді геш-суми. Таким чином, навіть у разі злому сервісу, зловмисники не отримають доступ до акаунтів користувачів.

←T→	id	name	login	nicname	pass
☐ ✎ ⊞ ⊖	1	User	admin	User	c4ca4238a0b923820dcc509a6f75849b
☐ ✎ ⊞ ⊖	2	Вадим	kudron	7kudron7	59fa240479013d36ac67775fd6d819cf

Рисунок 3.3 – таблиця «users»

В результаті ми отримали повністю робочий сервіс, готовий для роботи. Розроблений дизайн враховував комфорт користувача при взаємодії з сервісом, сервер забезпечує надійне з’єднання графічної частини з базою даних, яка в свою чергу містить всі необхідні атрибути задля роботи сервісу.

3.3 Опис функціоналу сервісу

Важливим є описати функціонал сервісу з точки зору користувача.

Перш ніж завантажувати свої текстури, користувач повинен увійти в свій акаунт. Це потрібно для збереження його налаштувань, що до обробки зображення (рис. 3.4).

Після входу в акаунт користувач може вибрати опцію завантаження текстури. Це кнопка «Завантажити нову текстуру». Після вибору текстури, користувач повинен завантажити файл із власною текстурою. Це виконується шляхом вибору файлу на комп'ютері та його завантаження на сервер. Так він повинен вказати авторство і текст watermark.

Після завантаження текстури користувач може переглянути її на вкладці «Профіль».

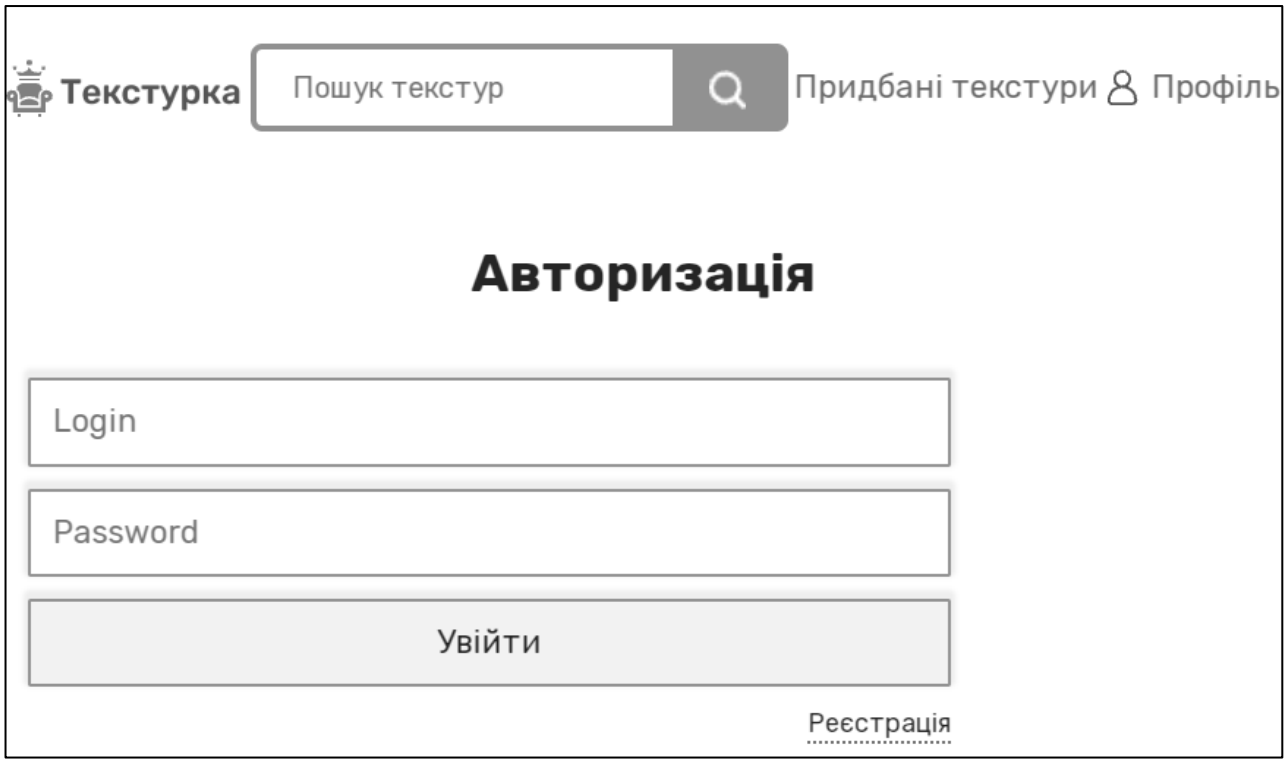


Рисунок 3.4 – Авторизація користувача

Для нового користувача пропонується форма Реєстрації (рис. 3.5). Після внесення своїх даних в неї, вони зберігаються на сервері. Пароль зберігається у вигляді геш-суми задля додаткового захисту.

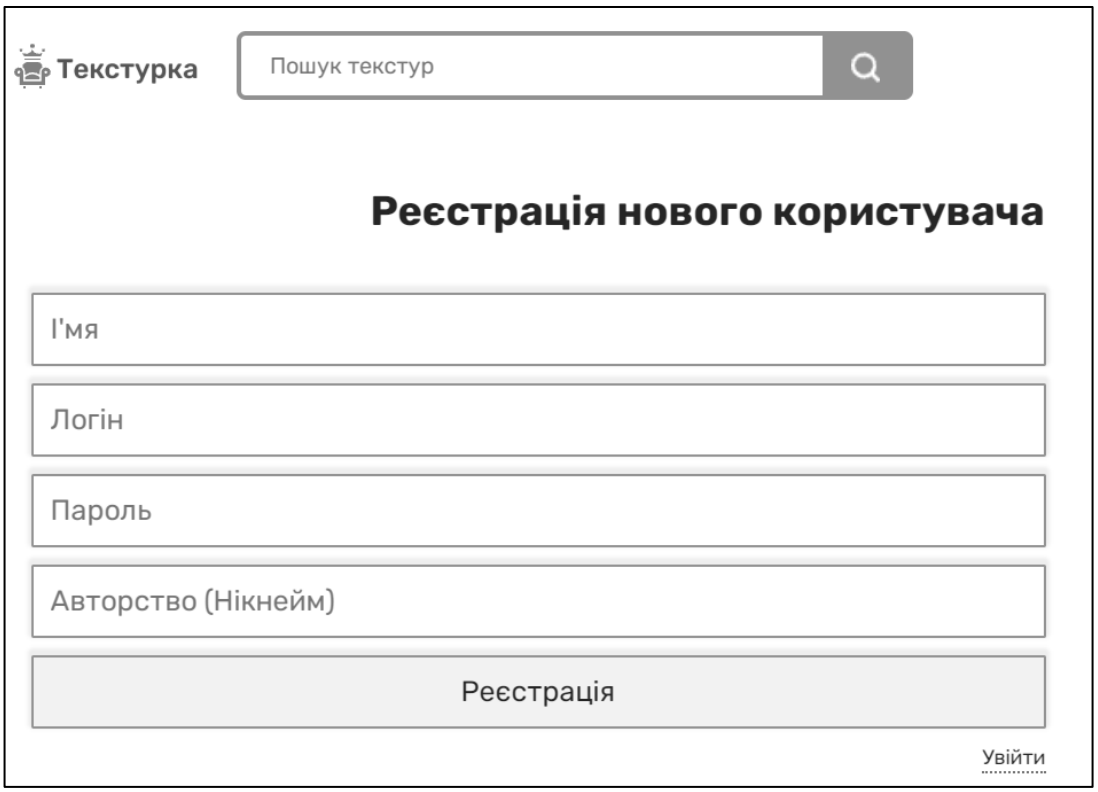


Рисунок 3.5 – Реєстрація користувача

На сторінці профілю відображаються всі завантажені текстури. Також присутня кнопка для створення нового товару (рис. 3.6).

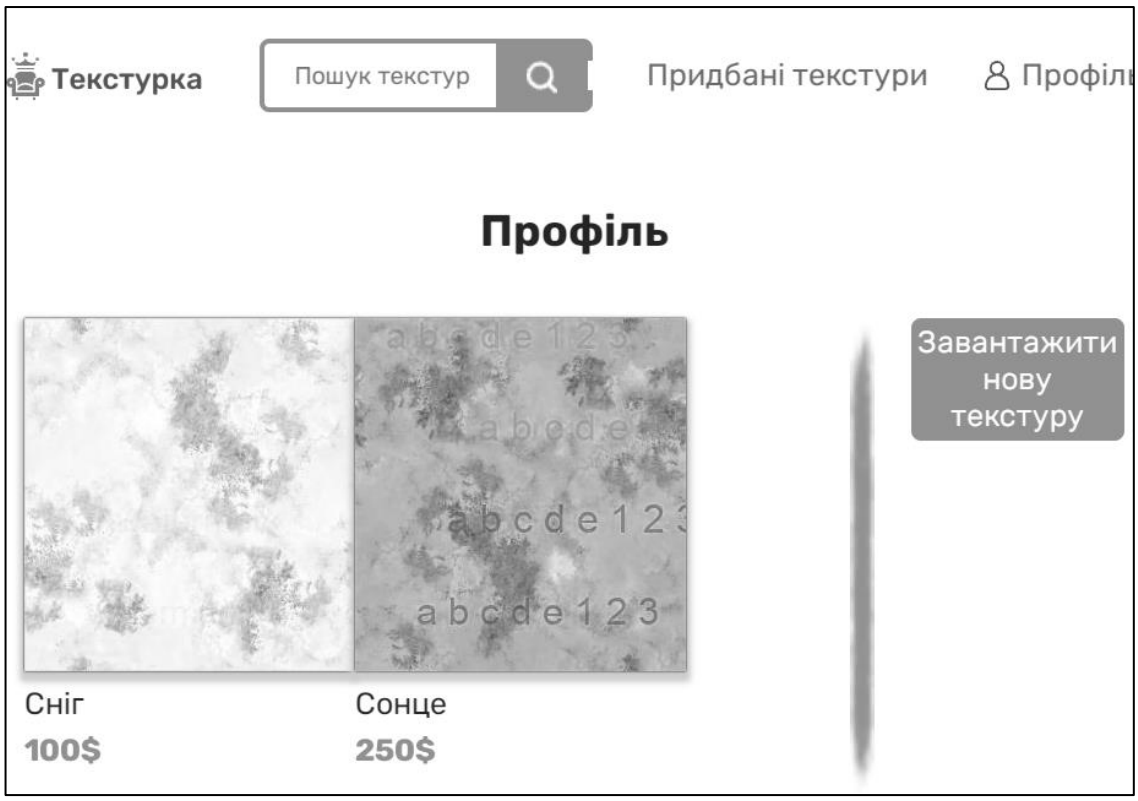


Рисунок 3.6 – Сторінка профілю користувача

Для завантаження нової текстури використовується форма веб-сторінки. Вона приймає певний перелік параметрів необхідних для шифрування і відображення товару (рис. 3.7). Лістинг функції наведено нижче:

```
<form class="load__form" enctype="multipart/form-data" action="/profile/saveData" method="POST">
<p>Завантажте текстуру</p>
<input type="hidden" name="MAX_FILE_SIZE" value="5120000" />
<input class="load" name="images" type="file">
<p>Завантажте прев'ю</p>
<input type="hidden" name="MAX_FILE_SIZE" value="5120000" />
<input class="load" name="more" type="file">
<input name="name" type="text" required placeholder="Назва">
<textarea name="about" required>Опис</textarea>
</form>
```

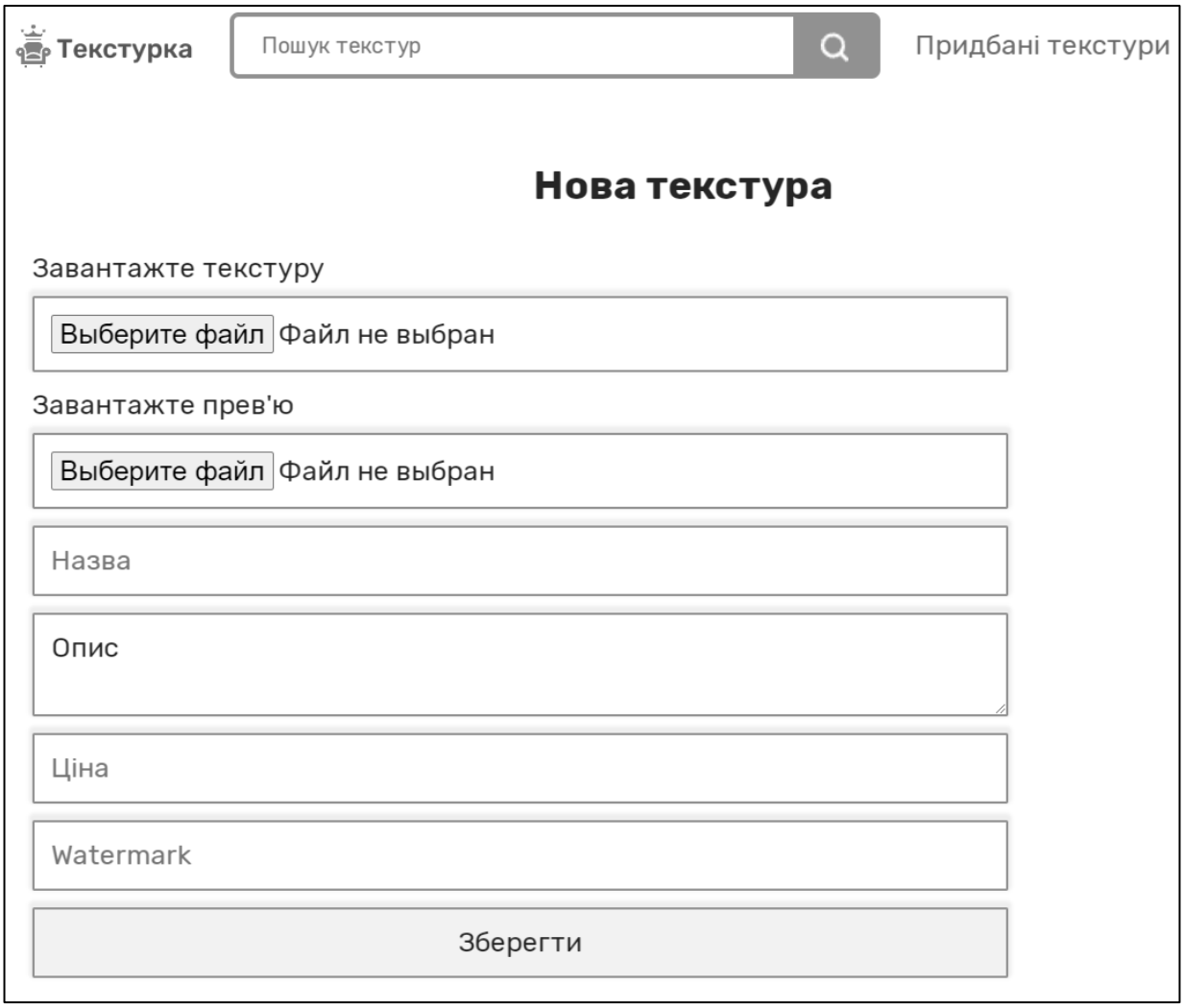


Рисунок 3.7 – Сторінка профілю користувача

Після завантаження товар завантажується на сервер і стає доступний для покупки на головній сторінці. Відповідно певний час він знаходиться в категорії «Нові» (рис. 3.8). Це допомагає покупцям знайти найактуальніші текстури та викупити права на найкращі варіанти першими.

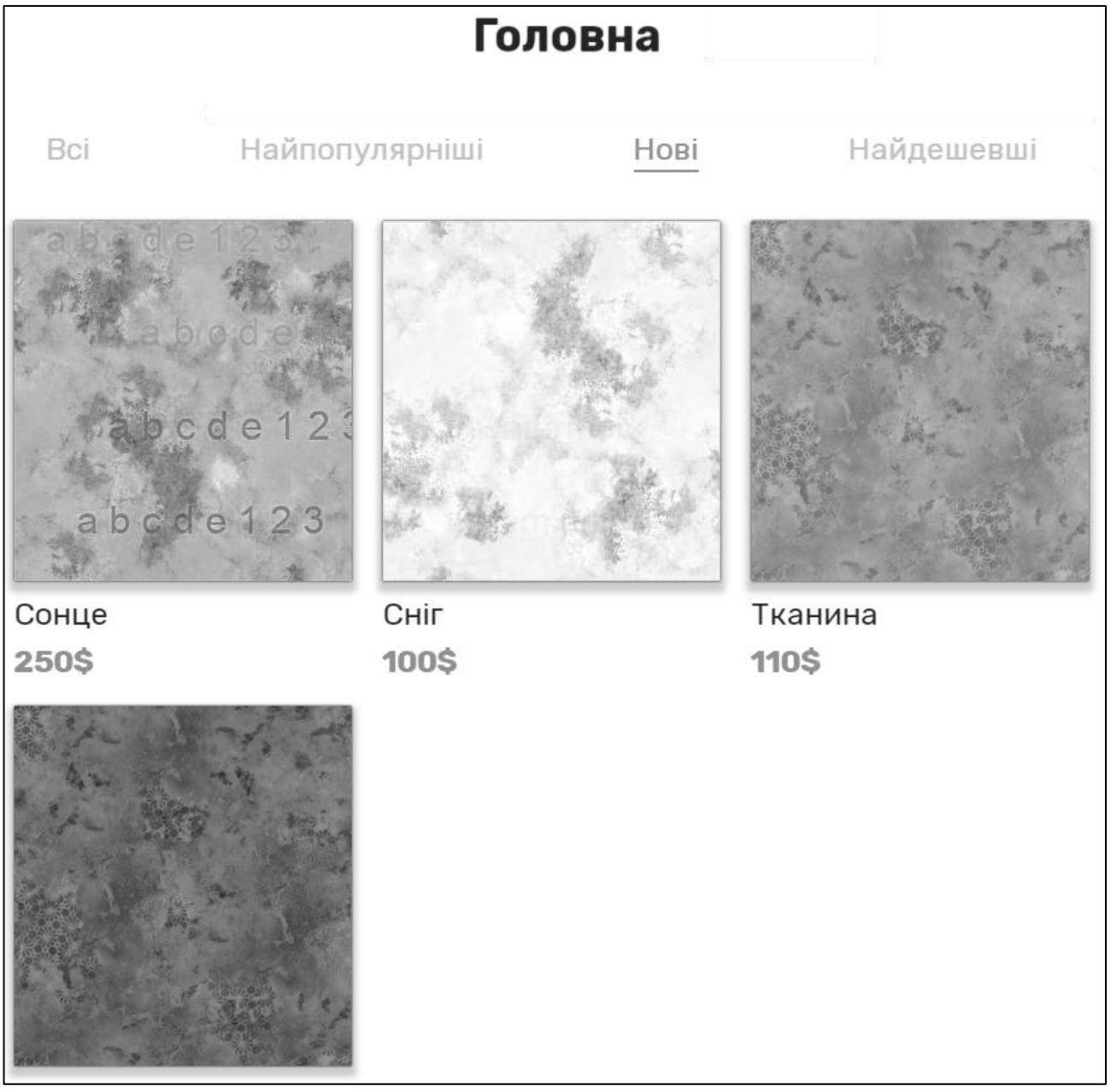


Рисунок 3.8 – Перегляд нових товарів

В результаті, був отримай повністю функціональний додаток, який може конкурувати з існуючими аналогами. Основною перевагою даного додатку є висока захищеність, в порівнянні з іншими застосунками. Це може стати вирішальним у

виборі користувачів, оскільки проблема захисту права на інтелектуальну власність все частіше постає в сучасному світі.

3.4 **Висновки до розділу 3**

Під час аналізу функціоналу сервісу було з'ясовано, що процес завантаження текстур та користування сервісом в цілому є інтуїтивно зрозумілим та зручним для кінцевого користувача. Описано всі етапи взаємодії з платформою, від вибору та завантаження текстур до їхнього збереження та використання.

У розділі "Розробка онлайн-сервісу" представлено детальний опис процесу розробки самого сервісу. Це включає розробку інтерфейсу, серверної частини та бази даних. Надано скріншоти та приклади коду для кращого розуміння та візуалізації проекту.

В розділі "Розробка модулю захисту та його інтеграція" були представлені кроки написання коду для шифрування та вбудовування водяних знаків. Ці модулі були інтегровані в основний функціонал сервісу з метою забезпечення безпеки та захисту власності користувачів.

У підсумку, на основі аналізу та розробки функціоналу, а також інтеграції модулів захисту, можна вважати, що розроблений сервіс є ефективним та забезпечує високий рівень зручності та безпеки для користувачів.

ВИСНОВОК

Було успішно розроблено та реалізовано модуль захисту файлів текстур, який забезпечує високий рівень безпеки для даних користувачів. За допомогою бібліотеки OpenSSL та методу RSA шифрування, забезпечено захищену передачу та зберігання файлів двовимірних текстур, зменшуючи ризик її незаконного доступу чи витоку.

Крім того, кожне зображення, що розміщується в магазині, забезпечене водяним знаком, що дозволяє відстежувати та ідентифікувати автора чи правовласника цих текстур. Це не лише підвищує безпеку продукту, а й додає додатковий шар захисту від піратства та незаконного використання.

Загалом, розроблений модуль став результатом вдалих досліджень та творчого підходу до розв'язання проблеми зберігання та передачі даних в онлайн-середовищі. Його застосування в реальних умовах може значно поліпшити безпеку та захист інтелектуальної власності в інтернеті.

В першому розділі було проведено порівняльний аналіз існуючих інтернет-магазинів текстур, їх переваги та недоліки, що дало змогу виокремити недоліки, пов'язані з відсутністю ефективного шифрування, та обґрунтувати необхідність створення власного модулю захисту з вдосконаленим криптографічним захистом. В результаті, було проведено аналіз предметної області та існуючих аналогів. В результаті було визначено актуальність створення модуля захисту в зв’язку з незахищеністю існуючих сервісів.

У другому розділі було розглянуто принцип роботи модулю, включаючи перенаправлення даних на сервер, їх опрацювання та передачу до бази даних. Також, було обґрунтовано вибір середовища програмування та описана архітектура сервісу. В результаті роботи, після даного розділу було отримано сформовану архітектуру модуля захисту, спроектовано роботу сервісу та обрано середовище розробки для реалізації.

У третьому розділі було детально описано процес практичної реалізації модулю захисту. Було наведено лістинг програми та описано функції. Також, було реалізовано онлайн-сервіс для продажу та зберігання двовимірних текстур. Даний

сервіс було інтегровано в модуль захисту. В результаті роботи було отримано повністю працюючий онлайн-сервіс з високою захищеністю даних завдяки використанню розробленого модулю захисту.

В результаті роботи було:

– проведено аналіз предметної області та визначено актуальність створення даного модуля;

– досліджено існуючі аналоги та проаналізовано їх;

– розроблено архітектуру модуля захисту для онлайн-сервісу;

– проаналізувано та обрано метод шифрування та вбудовування ЦВЗ;

– розроблено архітектуру сервісу для інтеграції модуля захисту;

– практично реалізано модуль захисту;

– створено сервіс для зберігання та продажу двовимірних текстур;

– інтегрувано модуль захисту в розроблений сервіс.

В результаті виконання дипломної роботи було досягнуто поставлених цілей та отримано повністю функціонуючий модуль захисту, який можна інтегрувати в існуючі онлайн-сервіси.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дмитро Івашкевич, Ігор Сімдянов. PHP 7: Підручник. Одеса: БХВ 2014. 311с
2. Девід Сідерхолм. Інтегрований PHP в системі управління: Підручник. Київ: Вільямс 2018. 124с
3. Sklar, D. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5: Навчальний посібник. O'Reilly Media, 2018. 163 с.
4. Duckett, J. HTML and CSS: Design and Build Websites.: Навчальний посібник. John Wiley & Sons, 2014. 89 с.
5. Freeman, E., & Robson, E. Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages: Посібник. O'Reilly Media, 2018. 189 с.
6. Powell, J. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML: Посібник. O'Reilly Media, 2019. 356 с.
7. Williams, C. Pro PHP and jQuery. Apress.: Навчальний посібник. O'Reilly Media, 2017. 265 с
8. Williams, C. Pro HTML5 and CSS3 Design Patterns. Apress. : Навчальний посібник. O'Reilly Media, 2016. 243 с
9. Schmitt, C. CSS: The Definitive Guide. : Навчальний посібник. O'Reilly Media, 2016. 185 с
10. Rosenwasser, R., & Powers, A. Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript. : Навчальний посібник. O'Reilly Media, 2019. 242 с
11. Pazzani, M.J., & Billsus, D. Content-based recommendation systems. In The adaptive web : Підручник Springer, Berlin, Heidelberg, 2020.
12. Граничин О.Н., Поляк Б.Т. Рандомізовані алгоритми оцінювання та оптимізації при майже довільних поміхах: Навчальний посібник. Київ: Освіта, 2003. 191 с.
13. Головін С. Д. PHP 7: Підручник / ДМК Пресс, 2016. — 352 с.
14. Стайлз Д., Джордж Ф., Декерт Э., Гудхарт Д. HTML и CSS. — Київ, 2019. — 1032 с.

15. Макфарланд Д. CSS. Детальне керівництво: Посібник. Символ-Плюс, 2020. — 774 с.

16. Буч Д. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites. O'Reilly Media, 2018. 832 с.

17. Хлебникова Л., Петриченко В. Створення сайтів на HTML та CSS. — М.: Есмо, 2019. — 288 с.

18. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. Handbook of Applied Cryptography: Посібник. CRC Press, 2014. 780 с.

19. Katz, J., & Lindell, Y. Introduction to Modern Cryptography. Chapman and Hall/CRC: Посібник. CRC Press, 2014. 810 с.

20. Paar, C., & Pelzl, J. Understanding Cryptography: A Textbook for Students and Practitioners: Підручник. Springer Science & Business Media, 2015 1026 с.

21. Stallings, W. Cryptography and Network Security: Principles and Practice / Pearson Education Limited, 2017. 256 с.

22. Ferguson, N., Schneier, B., & Kohno, T. Cryptography Engineering: Design Principles and Practical Applications / John Wiley & Sons, 2010, 256 с.

23. Singh, S. The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. / Anchor Books, 2012. 432 с.

24. Boneh, D., & Shoup, V. A Graduate Course in Applied Cryptography / Cryptography & Network Security, 2014. 102 с.

25. Ferguson, N., & Schneier, B. Practical Cryptography / John Wiley & Sons, 2013. 403 с.

26. Katz, J., & Lindell, Y. Introduction to Modern Cryptography: Principles and Protocols / CRC Press, 2015. 256 с.

27. Stinson, D. R. Cryptography: Theory and Practice: Навчальний посібник CRC Press, 2009. 680 с.

28. Тарасов І. Вбудовування та шифрування метаданих в файлах.Електроніка. 2011. № 3. С. 70–74.

29. Тесленко О. К., Кісільчук Б. Я. Формування ключів алгоритму шифрування на базі лінійних структур. Прикладна математика та комп’ютинг. Київ: НТУУ «КПІ», 14–16 листопада 2018 р.

30. Christopher M. Bishop Pattern Recognition and Machine Learning/M.B.Christopher. Springer, 2006. 758 p

31. Clarke B. Principles and Theory for Data Mining and MachineLearning/B.Clarke, E. Fokoue, H. Zhang. Dordrecht Heidelberg LondonNewYork:Springer, 2009. 793 c

ДОДАТКИ

ДОДАТОК А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
_____Юрій ЯРЕМЧУК
" 22 " березня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до бакалаврської дипломної роботи на тему:
Розробка програмного модуля захисту файлів текстур двовимірних моделей для онлайн-сервісу

08-72.БДР.013.00.081.ТЗ

Керівник бакалаврської дипломної роботи
к.т.н., доцент
Карпінець В.В.
" 22 " Березня 2024р.

Вінниця – 2024 р.

1. Найменування та область застосування

Програмний модуль захисту файлів текстур двовимірних моделей для онлайн-сервісів.

Область застосування: зберігання та продаж файлів текстур двовимірних моделей за допомогою онлайн-сервісів.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 80 від 11.03.2024 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка ефективного модуля захисту файлів текстур двовимірних моделей для онлайн-сервісів.

3.2 Призначення: розроблений програмний модуль дозволяє підвищити захищеність онлайн-сервісів продажу та зберігання файлів текстур двовимірних моделей.

4. Джерела розробки

4.1. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. Handbook of Applied Cryptography: Посібник. CRC Press, 2014. 780 с.

4.2. Boneh, D., & Shoup, V. A Graduate Course in Applied Cryptography / Cryptography & Network Security, 2014. 102 с.

4.3. Тесленко О. К., Кісільчук Б. Я. Формування ключів алгоритму шифрування на базі лінійних структур. Прикладна математика та комп’ютинг. Київ: НТУУ «КПІ», 14–16 листопада 2018 р.

4.4. Katz, J., & Lindell, Y. Introduction to Modern Cryptography: Principles and Protocols / CRC Press, 2015. 256 с.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний та легкий у використанні інтерфейс, який буде інтуїтивно зрозумілий новому користувачу;

5.1.2 Реалізація методу захисту транспортування даних не повинна потребувати використання спеціальних ліцензійних програмних додатків, забезпечуючи доступність і використання програмного засобу;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний модуль захисту повинен працювати без помилок, та передбачати виведення відповідних повідомлень у випадку виникнення критичних ситуацій;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції з метою ефективного захисту файлів.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов’язковий огляд фукіоналу для майбутніх користувачів, наведений у пункті 3.3.

7. Вимоги до технічного захисту інформації

7.1 Використання механізмів для запобігання витоку конфіденційної інформації, контроль доступу до даних і обмеження привілеїв користувачів.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Програмний модуль повинен бути розроблений з урахуванням потреб широкого спектру користувачів, забезпечуючи його використання в різних умовах.

9. Стадії та етапи розробки

№ з/п	Назва етапів бакалаврської дипломної роботи	Початок	Закінчення
1.	Визначення напрямку бакалаврської роботи, формулювання теми	22.03.2024	25.03.2024
2.	Аналіз предметної області обраної теми	26.0.2024	16.04.2024
3.	Розробка алгоритму роботи	17.04.2024	26.04.2024
4.	Написання бакалаврської роботи на основі розробленої теми	27.04.2024	23.05.2024
5.	Передзахист бакалаврської дипломної роботи	24.05.2024	25.05.2024
6.	Виправлення, уточнення, корегування бакалаврської дипломної роботи	26.05.2024	11.06.2024
7.	Захист бакалаврської дипломної роботи	12.06.2024	17.06.2024

10. Порядок контролю та прийому

10.1 До приймання бакалаврської дипломної роботи надається:

– ПЗ до бакалаврської дипломної роботи;

- демонстрація результату бакалаврської дипломної роботи;
- презентація;
- відзив керівника роботи;
- відзив рецензента

Технічне завдання до виконання прийняв _____Кудревич В.І.

ДОДАТОК Б. Лістинг програми

```
function embedTextWatermark($imagePath, $watermarkText) {
   // Завантажити зображення
   $image = imagecreatefromjpeg($imagePath);

   // Конвертувати текст в бінарний формат
   $binaryWatermark = '';
   for ($i = 0; $i < strlen($watermarkText); $i++) {
      $binaryWatermark .= sprintf("%08d", decbin(ord($watermarkText[$i])));
   }

   // Вбудовувати бінарний текст у LSB кожного пікселя
   $watermarkIndex = 0;
   for ($x = 0; $x < imagesx($image); $x++) {
      for ($y = 0; $y < imagesy($image); $y++) {
         // Отримати RGB кольор пікселя
         $rgb = imagecolorat($image, $x, $y);
         $r = ($rgb >> 16) & 0xFF;
         $g = ($rgb >> 8) & 0xFF;
         $b = $rgb & 0xFF;

         // Замінити останній біт RGB значення на біт з тексту
         $r = ($r & 0xFE) | ($binaryWatermark[$watermarkIndex++] ?? 0);
         $g = ($g & 0xFE) | ($binaryWatermark[$watermarkIndex++] ?? 0);
         $b = ($b & 0xFE) | ($binaryWatermark[$watermarkIndex++] ?? 0);

         // Змінити колір пікселя на оновлене RGB значення
         imagesetpixel($image, $x, $y, imagecolorallocate($image, $r, $g, $b));

         // Завершити, якщо всі біти тексту вже вбудовані
         if ($watermarkIndex >= strlen($binaryWatermark)) {
            break 2;
         }
      }
   }

   // Зберегти зображення з вбудованим текстовим водяним знаком
   imagejpeg($image, 'watermarked_image.jpg');
```

```
    // Звільнити пам'ять
    imagedestroy($image);
}

// Застосувати функцію для вбудування текстового водяного знака
embedTextWatermark('original_image.jpg', '$this->text');
```

```
<header>
  <div class="header__container">
    <div>
        <a href="/">
          <img class="header__logo" src="/img/logo.png " alt="">
        </a>
    </div>
     <form class="header__search" action="/main/search" method="GET">
       <input type="text"  name="search" placeholder="Пошук текстур">
       <div class="search__wrapper">
         <button type="submit"><img src="/img/search.png" alt="#"></button>
       </div>
     </form>
    <a href="/cart" class="header__cart">
        Придбані текстури
    </a>
    <a href="/profile" class="header__profile">
        <img src="/img/profile.png" alt="#">
        Профіль
    </a>
  </div>
</header>
<main>
  <div class="page__block block">
    <div class="block__container">
        <div class="block__text">
          <p class="block__subtitle">
              Текстурка
          </p>
          <p class="block__title">
              Інтернет-магазин текстур для 3d моделювання
```

```
                    <img src="/img/background-tetx.png" alt="" class="block__green-text">
                </p>
                <a href="" class="block__button">Купити</a>
            </div>

    </div>
</div>
<div class="page__main-block">
    <div class="main-block__container">
            <p class="main-block__title">
                Головна
            </p>
            <div class="main-block__list">
                    <?php if ( isset($_GET['id']) && $_GET['id'] == 1 ): ?>
                            <a href="/">Всі</a>
                            <a href="?id=1" class="main-block__list_active">Найпопулярніші</a>
                            <a href="?id=2">Нові</a>
                            <a href="?id=3">Найдешевші</a>
                    <?php elseif( isset($_GET['id']) && $_GET['id'] == 2 ): ?>
                            <a href="/">Всі</a>
                            <a href="?id=1">Найпопулярніші</a>
                            <a href="?id=2" class="main-block__list_active">Нові</a>
                            <a href="?id=3">Найдешевші</a>
                    <?php elseif( isset($_GET['id']) && $_GET['id'] == 3 ): ?>
                            <a href="/">Всі</a>
                            <a href="?id=1">Найпопулярніші</a>
                            <a href="?id=2">Нові</a>
                            <a href="?id=3" class="main-block__list_active">Найдешевші</a>
                    <?php else: ?>
                            <a href="/" class="main-block__list_active">Всі</a>
                            <a href="?id=1">Найпопулярніші</a>
                            <a href="?id=2">Нові</a>
                            <a href="?id=3">Найдешевші</a>
                    <?php endif; ?>


            </div>
    </div>
</div>
<div class="page__list">
```

```
            <?php if (!empty($products)): ?>
                <div class="list__container">
                    <?php foreach ($products as $prod): ?>
                        <a href="/main/product?id=<?=$prod->id;?>" class="list__item">
                          <img src="<?=$prod->img;?>" alt="" class="item__img">
                          <p class="item__title"><?=$prod->name;?></p>
                          <p class="item__price"><?=$prod->price;?>$</p>
                        </a>
                    <?php endforeach; ?>
                </div>
            <?php endif; ?>
  </div>
</main>
<footer></footer>
```

ДОДАТОК В. Ілюстративний матеріал

Розробка програмного модуля захисту файлів текстур двовимірних моделей для онлайн-сервісу

Виконав: студент групи 1КІТС-20Б Кудревич В.І.
Керівник: к.т.н., доцент каф. МБІС Карпінець В.В.

Актуальність теми

Розширення сфер використання моделей

Збільшення спеціалістів в даній сфері

Незахищеність сервісів для зберігання

Можливі загрози

1) Підробка та несанкціоноване використання

2) Несанкіонований доступ

3) Сумісність із різними платформами

Мета роботи

Метою роботи є створення модуля захисту файлів текстур двовимірних моделей для онлайн-сервісів їх продажу та зберігання

Предметом дослідження є розробка модуля захисту файлів двовимірних текстур.

Об'єктом дослідження є процес зберігання та обміну файлами текстур в онлайн-сервісах

Розробка модуля захисту двовимірних текстур

Крок 1: Завантаження текстур

Крок 2: Накладання видимого ЦВВ на прев’ю

Крок 3: Накладання прихованого ЦВВ на текстуру

Крок 4: Шифрування текстури

Крок 5: Збереження в базі даних

Крок 6: Оновлення списку текстур

Вибір методів захисту

Вбудовування ЦВЗ в прев'ю

Least Significant Bit та нанесення видимого ЦВЗ

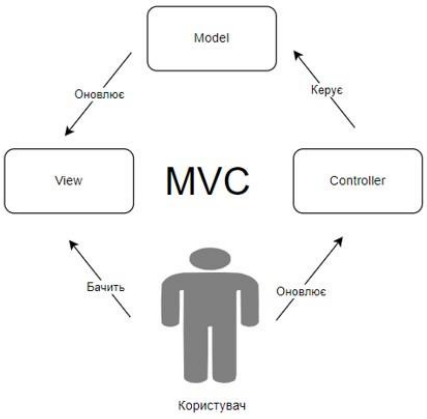
Вбудовування ЦВЗ в текстуру

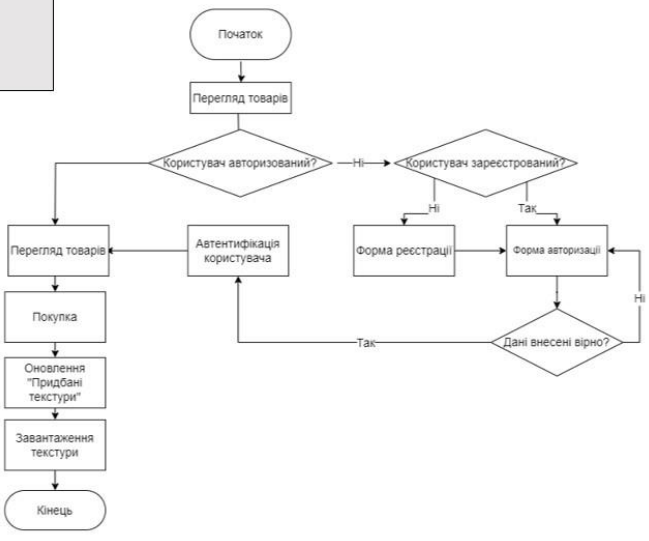
Дискретне косинусне перетворення

Шифрування файлу

RSA

Розробка онлайн-сервісу





Середовище розробки

скриптова мова програмування, була створена для генерації HTML-сторінок на стороні вебсервера

Visual Studio Code

засіб для створення, редагування та зневадження сучасних вебзастосунків

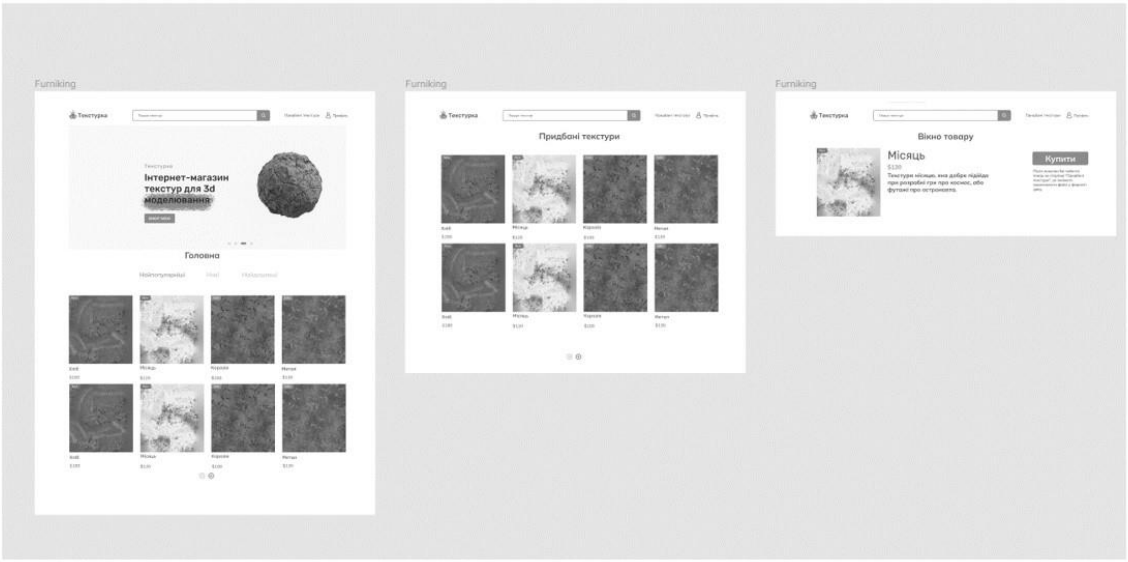
MySQL

вільна система керування реляційними базами даних

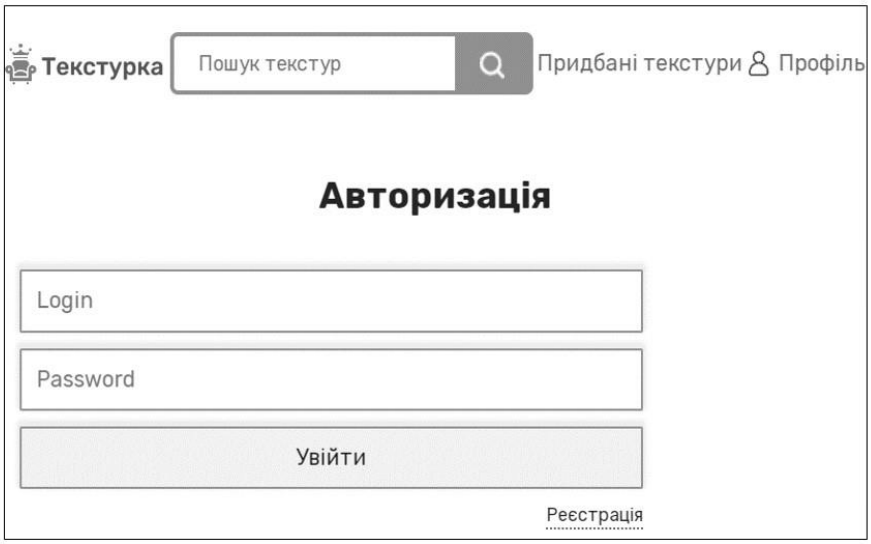
OpenServer
Petroleum Experts Communication Protocol

портативна програмна платформа, створена спеціально для веб-розробників

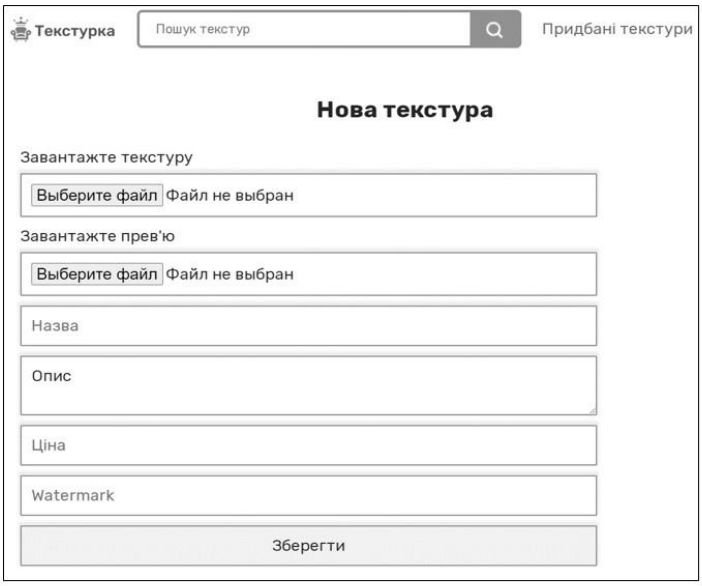
Огляд інтерфейсу



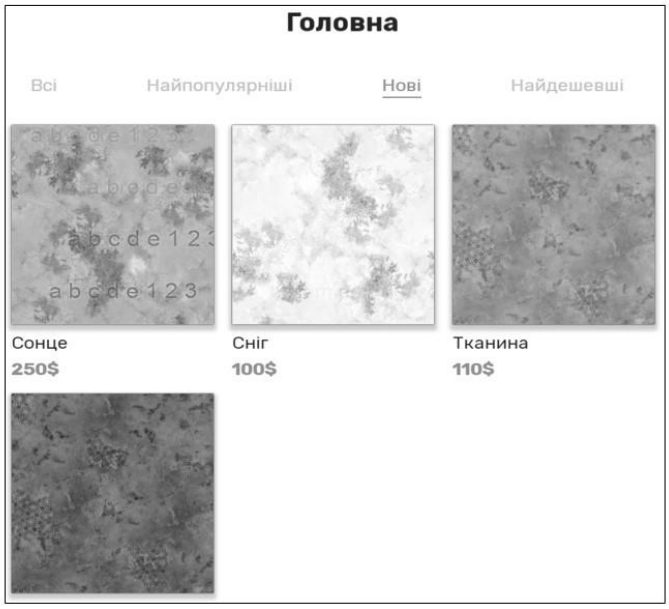
Процес авторизації



Процес завантаження файлу



Оновлення бази даних



Зберігання файлів

id	img	name	about	price	cat	file_url	owner_id
27	/img-logo/171442523526ArtStone1.jpg	Сніг	Опис	100	2	/img-logo/171442523583ArtStone1.jpg	1
28	/img-logo/171442533597ArtStone2.jpg	Сонце	Опис	250	2	/img-logo/171442533592ArtStone2.jpg	1

Висновки

В результаті роботи було:

- проведено аналіз предметної області та визначено актуальність створення даного модуля;
- досліджено існуючі аналоги та проаналізовано їх;
- розроблено архітектуру модуля захисту для онлайн-сервісу;
- проаналізувано та обрано метод шифрування та вбудовування ЦВЗ;
- розроблено архітектуру сервісу для інтеграції модуля захисту;
- практично реалізано модуль захисту;
- створено сервіс для зберігання та продажу двовимірних текстур;
- інтегрувано модуль захисту в розроблений сервіс.

В результаті виконання дипломної роботи було досягнуто поставлених цілей та отримано повністю функціонуючий модуль захисту, який можна інтегрувати в існуючі онлайн-сервіси.

Дякую за увагу!

ДОДАТОК Г. Протокол перевірки на антиплагіат

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка програмного модуля захисту файлів текстур двовимірних моделей для онлайн сервісу

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
(кафедра, факультет)

1 ПОКАЗНИКИ ЗВІТУ ПОДІБНОСТІ UNICHECK

Оригінальність 96 % Схожість 4 %

Аналіз звіту подібності (відмітити потрібне):
1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Кудревич В.І.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Карпінець В.В.
(підпис) (прізвище, ініціали)