

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

ВИПУСКНА РОБОТА

на тему:

«Хмарне сховище для інвентаризації даних»

Завідувач

випускаючої кафедри

А. С. Довбиш

Керівник роботи

О. Б. Проценко

Студентка групи ІІІ-72

М. І. Мисік

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедри Довбиш А.С.

“ _____ ” _____ 2021 р.

Завдання
до випускної роботи

Студента четвертого курсу, групи ІН-72 спеціальності “Інформатика”
денної форми навчання Мисіка Михайла Івановича.

Тема: “ Хмарне кладовище для інвентаризації даних ”

Затверджена наказом по СумДУ

№ _____ від _____ 2021 рік.

Зміст пояснювальної записки: 1) аналітичний огляд методів побудови веб-додатку; 2) постановка завдання й формування завдань дослідження; 3) огляд і опис засобів для розробки; 4) розробка веб-системи; 5)тестування; 6) аналіз результатів.

Дата видачі завдання “ _____ ” _____ 2021 рік.

Керівник випускної роботи _____ Проценко О.Б.

Завдання прийняв до виконання _____ Мисік М.І.

РЕФЕРАТ

Записка: 58 стор., 18 рис., 1 додаток, 8 джерел.

Об'єкт дослідження — хмарне сховище для інвентаризації даних.

Мета роботи — розробити веб-систему, яка повинна бути виконувати операції з належною швидкістю сприйняття при навантаженні великою кількістю інформації, проста в дизайні і зрозуміла для користувача. Сайт повинен бути адаптивним до всіх пристроїв, також повинен працювати у всіх сучасних веб-браузерах.

Результати — виконано вибір методів вирішення поставленої задачі на основі мов програмування PHP, Javascript. Реалізовано та виконане тестування веб-системи для інвентаризації даних.

ВЕБ-СИСТЕМА, ІНВЕНТАРИЗАЦІЯ, ФРЕЙМБОРК, HTML, CSS,
JAVASCRIPT, PHP, MARIA DB, JSON, SQL, BOOTSTRAP, JQUERY

Зміст

ВСТУП	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
1.1 Задача аналізу експериментального спектра.....	7
1.2 Клієнт-серверна технологія для веб-додатку	8
1.3 Огляд хмарних сховищ.....	9
1.4 Постановка задачі.....	12
У цій роботі потрібно створити хмарне сховище для інвентаризації даних, яке повинне містити в собі такий функціонал:	12
2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ	13
2.1 Програмні засоби для реалізації	13
2.2 Вибір засобів програмування.....	16
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	18
3.1 Розробка моделі веб-додатку	18
3.2 Структурно-функціональний аналіз проекту.....	19
3.3 Функціональні вимоги	22
3.4 Реалізація основних компонентів.....	23
ВИСНОВКИ.....	31
СПИСОК ЛІТЕРАТУРИ.....	32
ДОДАТОК ЛІСТИНГ МОДУЛІВ ЕКСПЕРТНОЇ СИСТЕМИ.....	34

ВСТУП

За останні роки можливість доступу до інтернету відкриває багато можливостей та відкриває нові межі та з'являються нові цілі у бізнесі.

З появою інтернету, всі прагнуть зробити діджиталізацію своєї сфери діяльності, так як, це буде наступна ступінь у розвитку та можливістю конкурувати з багаточисленними масштабами.

Деяким, з виходом бізнесу в онлайн сектор, інтернет допоміг залишитися на плаву та навіть досягти нових досягнень у своєму ремеслі.

Він має змогу об'єднати та оптимізувати роботу у деяких відношеннях, аби залишити більше часу на створення нових ідей та стратегій у цій палкій битві.

Раніше для введення інформації, щодо наявності певних товарів використовували зошити, креслили великі таблиці, заповнювали їх. На це витрачалась значна доля часу. Потрібно було їх перекреслювати, іноді дуже важко знайти потрібну інформацію, так як, кількість унікальних найменувань товарів може перебільшувати сотні й навіть тисячі одиниць.

Помилка у веденні такого зошита могла коштувати дуже багато організації, від непланового місця на складі до втрати репутації партнерів й навіть клієнта за такої хиби.

За допомогою нових відкриті та досягненню нових знань, прийнято рішення об'єднати цю тяжку ношу. Для цього потрібно створити сховище, в якому легко орієнтуватися, як керівнику, так постачальникам і клієнтам. З можливістю удосконалення та об'єднання всіх проблем минулого часу.

Для вирішення даної проблеми існують загальнодоступні інструменти, які були створені спеціалізованими компаніями, але вони мають певні

недоліки та ціна на послуги завжди тримає в смуті, так як, легко може варіативно змінюватись.

Задача, поставлена на цій роботі, зробити зрозумілий ресурс з даними які можна оновлювати, створювати, видаляти та мати можливість швидкого пошуку, також ресурс повинен мати змогу показати інформацію товарів у сховищі клієнтам які мають дані для авторизації в ресурсі.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Задача аналізу експериментального спектра

Веб-сайт - це сукупність програмних, інформаційних, а також аналітичних засобів, пов'язаних між собою маючи певну логіку. На мові бізнесу ж, веб-сайт - це відображення успішності фірми, її обличчя.

Веб сторінки й різноманітні веб-файли обслужують веб-серверами - спеціальним програмним забезпеченням, який має змогу доставляти веб-контент клієнтам за допомогою веб-протоколів. Різноманітність їх дуже велика, але на сьогоднішній найбільш поширеними є такі веб-протоколи[1]:

- **HTTP** (найстаріший, найбільш відомий і використовуваний)
- **HTTPS** (схожий на HTTP, але з удосконаленою можливістю безпечної відправлення даних)
- **SPDY** (новий, з гарним рівнем безпеки, сумісний з HTTP протокол, створений компанією Google та має змогу застосовуватись певними сьогоденними браузером)
- **WebSocket** (один з нових протоколів двосторонньої взаємодії, який доступний тільки певними браузером)

В якості клієнта, якому веб-сервер зазвичай приносить вміст веб-сайту, є, як правило, веб-браузер - спеціальне програмне забезпечення, яке здатне відображати веб-сторінки і вміє взаємодіяти з іншими типами веб-вмісту.

Веб-браузер, що є спеціальним розробленим програмним забезпеченням, яке має змогу показувати ці сторінки та може взаємодіяти з різноманітними типами веб-вмісту, виступає клієнтом, якому веб-сервер приносить певний вміст веб-сайтів

Веб-контент як і веб-сайти можна створити динамічним та статичним[2]:

- **Статичний веб-сайт**, у вигляді певних файлів лежить на сервері, і надходять веб-сервером у вигляді якими були створені певною особою. За статистикою, такі сторінки створюють виключно для перегляду інформації яку хоче донести засновник та не містять здатності до інтерактивної взаємодії з їхніми користувачами.
- **Динамічні**, генеруються певною створеною програмою, яка працює на веб-сервері, і може складатися з різних компонентів веб-сервером в режимі реального часу. За статистикою, ці сторінки мають можливість інтерактивної взаємодії з користувачем, даючи змогу не тільки подивитись, а й редагування й змінення інформації на ній.

1.2 Клієнт-серверна технологія для веб-додатку

Додаток, де клієнт має змогу кооперувати з певним сервером за підтримки браузера, в якості сервера використовується — веб-сервер.

Усі основні операції, такі як зберігання, редагування, в основному , виконуються сервером, розмін інформації здійснюється по мережі.

Веб-додатки мають змогу виконувати операції на різноманітних платформах, це є одною з переваг, тому що, клієнти не залежать від певної операційної системи встановленої у користувача додатку.

Архітектура клієнт-сервера передбачена такими основними компонентами:

- набори серверів, які містять різноманітну інформацію;
- клієнти, які є користувачами сервісами, що присвоюється серверами;

При взаємодії між клієнтом та сервером застосовують API(Application Programming Interface), він визначає функціональність, надана програмою в

основі якої лежать певні модулі та бібліотеки, має змогу абстрагуватися від реалізованого функціоналу.

API використовується для реалізації певних функцій, його використання можна відслідкувати для виведення певних інформацій, авторизації, створення, редагування користувачів, отримання інформації с серверів.

1.3 Огляд хмарних сховищ

Нині існує багато прикладів хмарних сховищ, залежно від типу, на них можна зберігати різноманітну інформацію, файли, програми, зображення.

Облегшує користування та не прив'язує його до конкретного пристрою, так як, для перегляду необхідно мати дані для авторизації та гарний доступ до мережі інтернет. Для реалізації наших задач, потрібно виводити дані табличною, також сховище має змогу інтегруватися для можливості перегляду наявності певного товару в інтернет магазині.

Для успішної роботи, необхідними елементами будуть виступати:

- забезпечення безпеки(шифрування даних авторизації, робота з даними) ;
- швидка та інтуїтивна платформа (для швидкого орієнтування в разі надання доступу для певних осіб) ;
- взаємодія з даними (можливість оновлювати, добавляти, видаляти дані) ;
- багатфункціональність (можливість перегляду наявності певного товару в інтернет магазині) ;
- створення користувачів(користувач з інтерфейсом адміністратора, має змогу створювати користувачів з обмеженим доступом по інформації, а саме: перегляд та пошук її) .

Для аналізу схожої ідеї будемо розглядати print-express.

Помітні плюси:

- Велика кількість фільтрів;
- Простий пошук певних категорій;
- Сторінка додання елементів (рис 1.2);
- Різноманітний сторонній функціонал(можливість створення звітів (рис 1.3), відправлення повідомлень).

Недоліки:

- платне користування;
- велика кількість непотрібного для нас функціоналу;
- не зручний у використанні;
- швидкість роботи ресурсу.

The screenshot shows the main interface of the PRINT-EXPRESS application. At the top, there is a navigation bar with the application name and user options. Below it is a menu with various categories like 'Начало работы', 'Компании', 'Документы', etc. The main content area displays a list of invoices under the heading 'Счета'. The table below contains the following data:

НОМЕР	ДАТА	НА КОГО	СУММА	СУММА С НДС	ОПЛАЧЕНО	ОСТАТОК	КАТЕГОРИЯ	ОПИСАНИЕ
1	25.07.2011	ООО "Аргумент"	25 000,00	25 000,00	17 500,00	7 500,00	Шелкография	Дизайн сайта, переделка сайта
2	26.07.2011	ООО "Нефтесервис"	3 000,00	3 000,00	3 540,00	-540,00	Шелкография	Создание информационного сайта
4	26.07.2011	ООО "ИнтелКом"	28 765,00	28 765,00	28 765,00	0,00	Другое	Печать диссертации
3	27.07.2011	ООО "Водовозов"	6 000,03	6 000,03	5 000,04	999,99	Другое	Создание сайта-визитки
5	28.07.2011	ЗАО "Тонус"	645,00	645,00	700,00	-55,00	Дизайн	Печать визиток
6	28.07.2011	ООО "Водовозов"	5 070,00	5 070,00	5 070,00	0,00	Дизайн	Разработка дизайна сайта
7	28.07.2011	ООО "ИнтелКом"	8 474,58	8 474,58	10 000,00	-1 525,42	Дизайн	Уборка
8	01.08.2011	ООО "Крун"	4 234,00	4 234,00	0,00	4 234,00		Чай, Кофе
			81 188,61	81 188,61	70 575,04	10 613,57		

Рисунок 1.1 — Основна сторінка

Copyright © ClientBase.ru
Версия: 1.9.5

Рисунок 1.2 — Сторінка додавання елементів

ОПИСАНИЕ	ЕД. ИЗМ	ЦЕНА	КОЛ-ВО	СУММА
Визитки	шт.	200,00	1 000	200 000,00
Плакаты	шт.	100,00	1 000	100 000,00
Календари		300,00	1 000	300 000,00

Рисунок 1.3 — Сторінка створення звітів

1.4 Постановка задачі

У цій роботі потрібно створити хмарне сховище для інвентаризації даних, яке повинне містити в собі такий функціонал:

- Авторизацію;
- Різні типи користувачів(адміністратор та відвідувач) ;
- Простий пошук;
- Можливість додавання даних, оновлення їх;
- Кабінет адміністратора, з можливістю редагування інформації;
- Можливість створювати, оновлювати, видаляти товарообіг;
- Можливість з правами рівня адміністратор створювати та видаляти відвідувачів;
- Пошук з використанням фільтрів.

Взаємодія з редагування інформації повинна виконуватись у режимі реального часу, продукт повинен мати змогу зберігати в собі інформації про сотні одиниць екземплярів.

2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ

2.1 Програмні засоби для реалізації

Для вирішення даних задач, потрібні допоміжні програмні засоби, більш зручні у використанні та які задовольняють розробку за всіма її особливостями.

У створенні серверної частини, вибір пав на OS(Open Server Panel), локальна серверна платформа, для розробки програми. Не маючи власних серверів, можна розроблювати на власному локальному сервері, що робить її зручною у створенні різних веб-додатків[3].

Швидкий у вивченні та зрозуміло інтуїтивний інтерфейс, великий набір допоміжних функцій, значні можливості для налаштування компонентів. Її використання поліпшує розробку, надає змогу працювати та тестувати веб-сервіси в локальній середі. Широко використовується у розробці, так як, програма є зручною і простою у освоєнні, безкоштовна, а її надійність у реалізації можна порівняти з рівнями серверів Linux.

Нині існує велика кількість різноманітних програмування, але не існує найкращої. Основна мета мови програмування це вирішення проблеми, яка поставлена перед програмістом. В різних мовах є переваги, але їх можливо побачити тільки в окремих задачах.

На даний час, ми маємо змогу класифікувати мови веб-програмування на клієнтські та серверні. Програми, які роблять на стороні клієнта, так званого веб-браузера, носять назву клієнтські. Програми які роблять операції на сервері, називаються серверні мови веб-програмування.

Для серверної частини використовується мова програмування PHP із застосуванням баз даних наданих можливостями OS. Клієнтська частина буде розроблятися за допомогою мови програмування JavaScript, допоміжною

бібліотеки JQuery, HTML, CSS зокрема її фреймворк bootstrap, який допоможе більш швидко адаптувати сайт під різні розміри екранів та прискорити процес розробки.

PHP — мова програмування для розробки веб-додатків різноманітної складності. Підтримується великою кількістю сервісів, що надаються хостинги, є одною з лідируючих мов для створення швидких та динамічних веб-сайтів.

За рахунок легкого розуміння, швидкої обробки даних, простоті використання, практичності та поширенню стартових написаних кодів під своєю ліцензією. Для створення веб-сайтів є велика кількість вбудованих

Мова PHP отримала широку популярність завдяки своїй швидкості обробки, простоті, функціональності і розповсюдженню початкових кодів під власною ліцензією. Для розробки сайтів використовується широкий набір вбудованих можливостей:

- мимовільне відокремлювання змінних оточення сервера, характеристики GET і POST в обумовлений масиви;
- зручна та доступна для розуміння робота з локальними файлами, сесіями, сокетами, HTTP авторизацією, cookies, XForms;
- робота з різноманітними СКБД(системами управління баз даних), такі як MySQL, Maria DB та інші;
- відправка HTTP заголовків в автоматичному режимі (полегшує пошукову оптимізацію сайту);
- швидка та якісна обробка різноманітних завантажених на файлів.

HTML (Hypertext Markup Language) — це код, який дає змогу структурувати і показувати різноманітний контент вашої сторінки. HTML не є мовою програмування; це мова розмітки, і використовується, для допомоги в сприйнятті інформації вашому браузеру, як показувати веб-сторінки, де ставити акценти, до чого цей текст відноситься та з чим він пов'язується.

Він повинен бути зрозумілим для сприйняття, іноді трапляється так, що стилі не можуть завантажитися на ваш комп'ютер, або ж у вас поганий інтернет зв'язок і ви не зможете їх завантажити, тому правильна розмітка дає змогу безперешкодно зрозуміти інформацію, яка дана на сайті і до чого вона відноситься.

В HTML існують різні типи так званих обгортки, які допоможуть семантично зрозуміти до чого цей контент відноситься та як з ним розпоряджатися в майбутньому. Теги, допомагають акцентувати на важливості деякого тексту, мають змогу його відображатися певним чином, формувати його, або робити посилання між сторінками.

Cascading Stylesheets (CSS, каскадні таблиці стилів) — код, який дає змогу стилізувати ваш контент, зробити більш красивим для сприйняття оком.

Як і HTML, CSS не зовсім є мовою програмування, так, він містить в собі схожий стиль з описуванням процесів як є у мовах програмування. Вона є мовою таблиць стилів, яка дозволяє стилізувати ваші елементи(контекст), які ви написали раніше, за допомогою HTML. CSS робить ваш сайт по дійсному унікальним, маж змогу стилізувати його, додати анімацій, інтерактивності, зробити вашу сторінку більш цікавою та унікальною с проміж інших.

JavaScript — одна з найпоширеніших мов програмування, використовується для створення та підтримки різноманітних сценаріїв, що виконують свої процеси на стороні користувача та взаємодіють із сервером. Браузер, встановлений на комп'ютері користувача, дає можливість створити середовище, для взаємодії об'єктів, якими є різноманітні вікна, заготовлені меню, все можливі текстові області, а також куки, демонстрація веб-сторінки.

Чистий JavaScript має універсальність та спромогу удосконалитись під любий новий фреймворк якщо це буде доречно. Одна з найстаріших та найрозповсюджених бібліотек цієї мови є JQuery.

JQuery — зручна у використанні і проста в розумінні бібліотека, за роки користування удосконалювалась та стала універсальним інструментом,

особливо при створенні одно сторінкових сайтів, пришвидшує роботу у створенні різноманітних функцій, також є велика кількість зручних методів.

JSON — стандартний формат який використовується для обміну даних, структурований на основі мови програмування JavaScript.

Є самостійним форматом виконує дії з багатьма іншими мовами програмування. Взагалі, JSON, є структурованим текстом у якому лежать дві структури:

- Упорядкований набір елементів, різних значень під конкретну задачу. Найчастіше у мовах програмування виступає масивом, списком;
- Набір елементів які є пов'язаними між собою як пара ключ: значення. У багатьох мовах існує як об'єкт, хеш-таблиця, списком з ключами, інколи асоціативним масивом.

2.2 Вибір засобів програмування

Нині, дуже багатої популярності є використання різних CMS[4], вони облегшують та прискорюють розробку проектів для тих, у кого не має багато часу для вивчення більш адаптивних для себе реалізацій, великий термін розробки та розуміння логіки деяких процесів. Для того, щоб отримати найякісніші фундаментальні знання був обраний підхід самописного рішення.[5]

Плюси такого підходу:

- Отримання та вдосконалення знань, якими є базовими для більш кращого розуміння роботи усіх процесів;
- Розробка будь-якої логіки та бізнес-процесів, все що буде в планах реалізації та більш великий спектр для розширення, легко масштабувати проект, або ж переписати за допомогою сучасних фреймворків;
- Швидкісна робота у порівнянні з типовими системами управління вмісту;
- Більш швидке розуміння програми для нових програмістів, не

складе труднощів для нових учасників команди при потребі для реалізації певних функцій та швидка адаптація до роботи;

- Велика кількість інформації у відкритих джерелах для поглиблення знань та допомозі при виникненні вирішень певних проблем;

Мінуси самописної системи:

- Розуміння логіки та процесів роботи програми, взаємодію компонентів та їх реалізацію, запровадження стабільної роботи системи та усунення виникнення проблем під час тестування;

- Довша за термінами розробка деяких компонентів, ніж з використанням Content Management System[4], у них, є існуючі типові компоненти, які складно розширити для створення своєї системи, зазвичай базові елементи, виникає необхідність для реалізації деяких функцій власноруч.

Самописним проектом можна розробити свій каркас, який в майбутньому можна адаптувати до будь-яких сучасних тенденцій, розширення та будь-яка логіка лежить на відповідальності автора, такий підхід більш складний, але задає гарних знань у вдосконаленні та розумінні при вивченні чогось нового, маємо змогу розроблювати власні сценарії та поведінки програми.

В самописному проекті можна використовувати сторонні бібліотеки, в них реалізовані деякі компоненти які полегшують, а іноді й прискорюють роботу усієї системи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Розробка моделі веб-додатку

Відобразимо проект використовуючи абстрактну модель[6] (рис 3.1). Під час розробки були прийняті рішення щодо удосконалення готового рішення:

- Відображення усіх даних для користувачів вибіркою з усіх можливих складів на одній сторінці з загальною кількістю одиниць;
- Шифрування даних користувачів;
- Можливість редагувати кількість елементів при відображенні на сторінці.

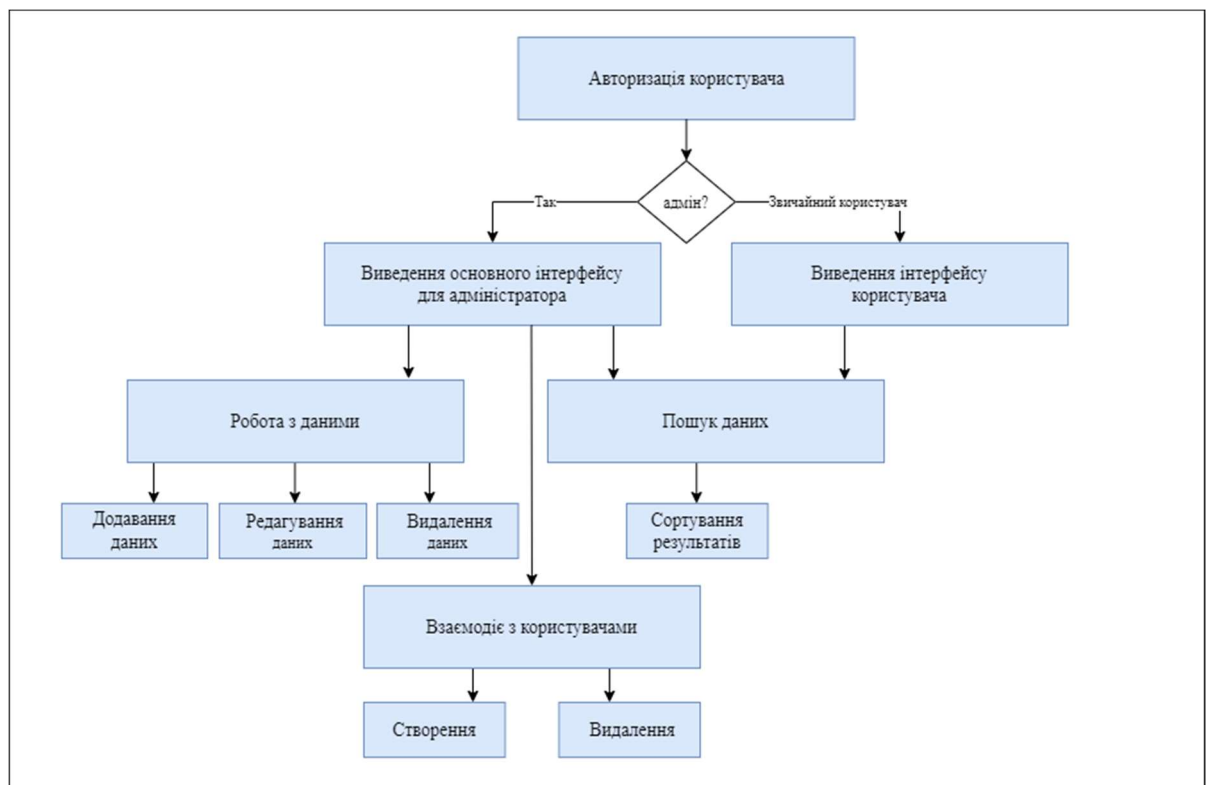


Рисунок 3.1 — Структура проекту

Звідси, використовуючи усі вимоги які були поставленні під час проектування, створимо поверхневу схему його роботи(рис 3.2), на ній показано основні дії при користуванні даним продуктом



Рисунок 3.2 — Загальна схема роботи програмного продукту

3.2 Структурно-функціональний аналіз проекту

Для більш грамотної розробки розіб'ємо процес на основні етапи розробки:

- Узгодження завдання, аналіз рішень;
- Планування роботи, постановка задач;
- Розроблення технічного завдання, обговорення нюансів;
- Формування інтерфейсу для проекту;
- Розроблення моделі програмного продукту, компоненти та їх взаємодія;
- Реалізація програмного продукту, логічна безперебійна робота системи;
- Тестування – робота та взаємодія з даними, позитивні сценарії;
- Внесення та опрацювання правок;
- Введення систему в дію для майбутньої експлуатації.

Розробимо приблизний графік роботи з термінами(рис 3.3 та рис 3.4),

для більш нормалізованої та якісної роботи.

	Назва завдання	Тривалість	Початок	Очікуваний кінець
1	Створення диплому	54 дні	05.04.21	30.05.21
2	Преддипломна практики	23 дні	05.04.21	27.04.21
3	Узгодження завдання	3 дні	05.04.21	07.04.21
4	Планування роботи	5 днів	08.04.21	12.04.21
5	Розроблення технічного завдання	3 дні	13.04.21	15.04.21
6	Формування інтерфейсу проекту	4 дні	16.04.21	19.04.21
7	Розроблення моделі програмного продукту	6 днів	20.04.21	25.04.21
8	Оформлення звіту з практики	2 дні	26.04.21	27.05.21
9	Реалізація програмного продукту	16 днів	5.06.21	20.06.21
10	Оформлення розділ 1	4 дні	15.04.21	18.05.21
11	Оформлення розділ 2	3 дні	19.04.21	21.05.21
12	Оформлення розділ 3	4 дні	22.05.21	27.05.21
13	Остаточне оформлення, внесення правок	3 дні	28.05.21	30.05.21

Рисунок 3.3 — Терміни та пункти роботи

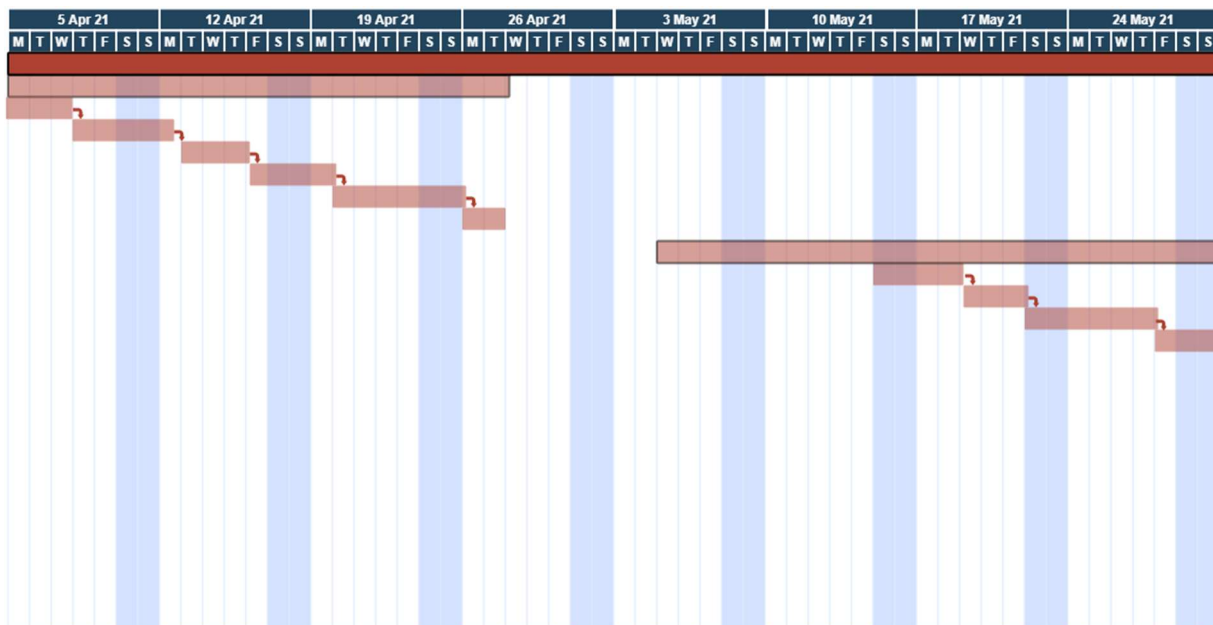


Рисунок 3.4 — Приблизний графік виконання термінів

Для більш детального розуміння розробимо діаграми діяльності та

декомпозицій[7](рис 3.5 та 3.6). За їх допомогою, маємо змогу побачити зв'язки між роботою певних функціональних одиниць, розділити програму на так звані компоненти для облегшення розуміння подальшої реалізації.



Рисунок 3.5 — Контекстна діаграма

Для створення проекту потрібне обговорене технічне завдання. За ним встановили існуючих типів користувачів, які будуть користуватися продуктом. Для роботи програми потрібні певні бази даних та тестова інформація для перевірки працездатності та усунення можливих помилок, на виході отримаємо продукт, який є певним веб-сервісом.

Детальну роботу компонентів представлено на діаграмі декомпозиції(рис 3.6).

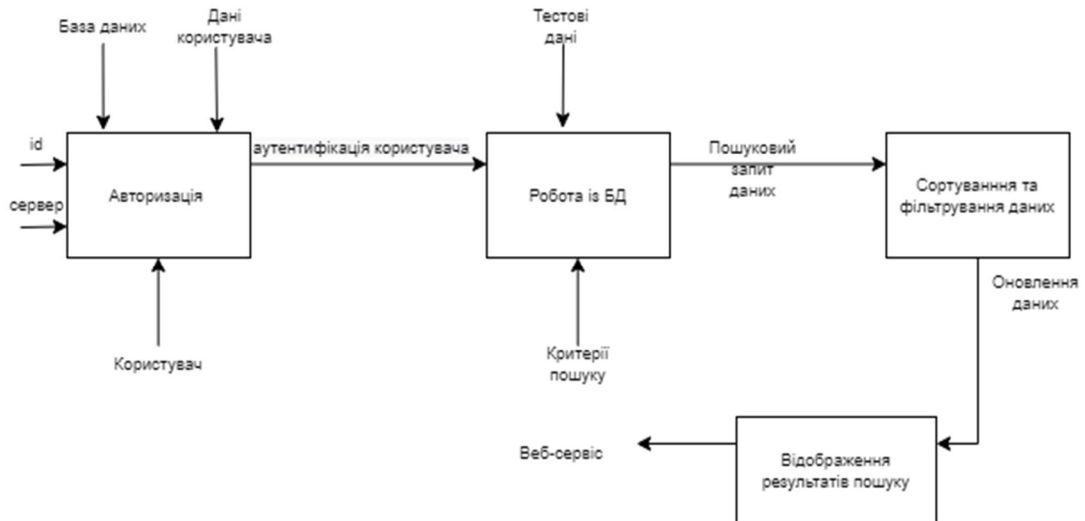


Рисунок 3.6 — Діаграма декомпозиції робіт

На етапі тестування проекту будуть створення певні тестові дані, в них будуть інформація, щодо майбутніх реальних даних, певні користувачі та заведені програмістом адміністратори. Перевірка всіх основних функцій та виявлення недоліків інтерфейсу.

Авторизація різних типів користувачів з різними сторінками відображення та можливостями у користуванні. Створення коректних записів адміністратором, відображення їх у користувачів, фільтрування інформації по певним фільтрам, перевірка пошуку, обробки на серверному рівні та оновлення інформації, удосконалення для більш зручного користування ресурсу на інтуїтивному рівні.

3.3 Функціональні вимоги

В результаті отриманий програмний продукт повинен бути інтуїтивно зрозумілим та зручним у використанні. Мати гарний дизайн, допоможуть у рішення цієї мети, велика кількість сучасних методів та інструментів, більш зрозумілі для швидкого освоєння існуючі бібліотеки з полегшенням деяких реалізацій та швидкісним розробленням для більш раціонального використання часу при створенні роботи.

Потрібно розробити декілька типів користувачів з різними інтерфейсами та доступними можливостями: зрозуміле редагування, щодо існуючої, інформації; створення та видалення її; виконання пошуку інформації використовуючи маючи фільтри пошуку; адміністратор має змогу до створення та видалення користувачів.

Для реалізації поставлених перед собою задач, було обрано мови програмування

- PHP, для виконання та обробки на стороні сервера;
- Javascript, для взаємодії на стороні клієнта (користувачі додатку).

3.4 Реалізація основних компонентів

На діаграмі варіантів використання[8], маємо змогу побачити усіх існуючих користувачів, а саме 2 типи, та їх можливості на сайті:

- адміністратор;
- користувач, якого зареєстрував адміністратор.

Набір функцій для кожного типу(див. рис 3.7).

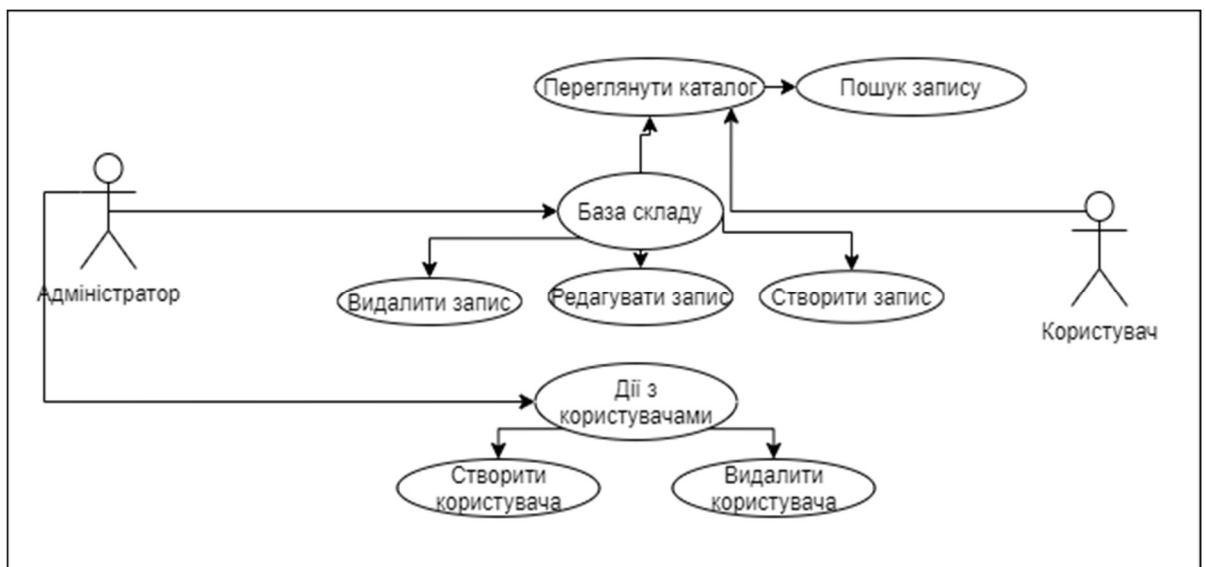
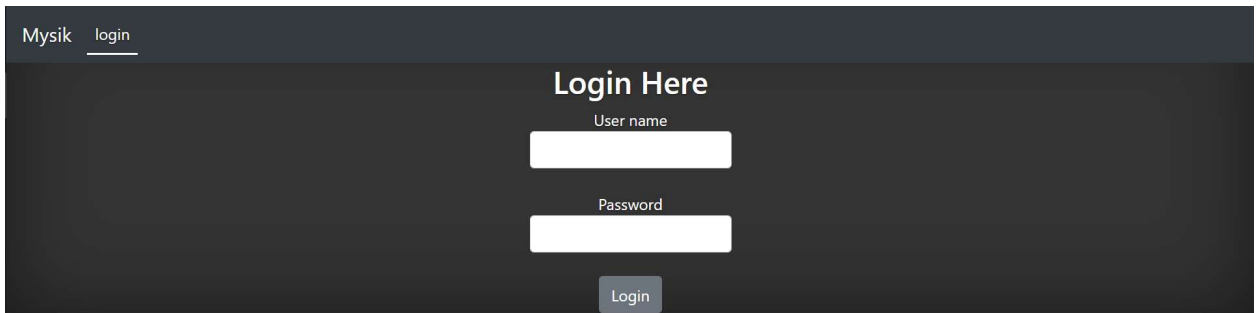


Рисунок 3.7 — Діаграма декомпозиції робіт

З початку користування ви потрапляєте на сторінку

авторизації(рис 3.8). Користувачі зі статусом адміністратор, отримали свої дані у програміста, так ресурс буде більш надійним, інформація, яка зберігаються на ньому, має редагуватися спеціалізованими людьми.

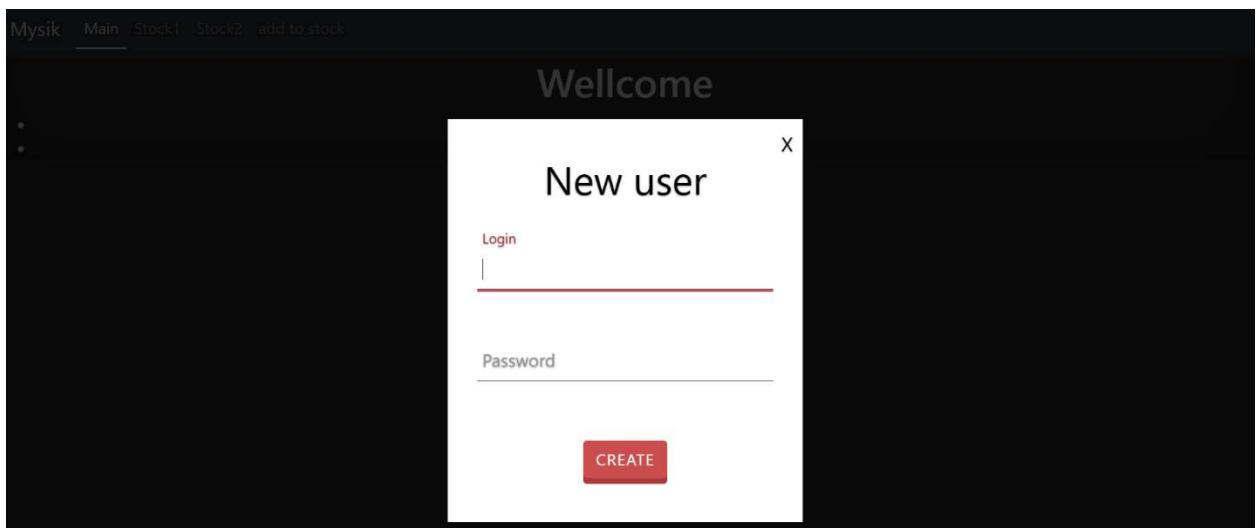


The screenshot shows a dark-themed login page. At the top left, there is a navigation menu with 'Mysik' and 'login' (underlined). The main heading is 'Login Here'. Below it are two white input fields: 'User name' and 'Password'. At the bottom center is a 'Login' button.

Рисунок 3.8 — Сторінка авторизації

На сторінці адміністратора ви маєте вся доступні привілеї розробленого ресурсу. Він може створювати користувачів, дивитись та видаляти існуючих.

Для створення користувача доступна форма (рис 3.9).



The screenshot shows a 'New user' form in a modal window. The window title is 'New user' with a close button 'X'. The form has two input fields: 'Login' and 'Password'. At the bottom is a red 'CREATE' button. The background shows a dark navigation menu with 'Mysik', 'Main', 'Stock1', 'Stock2', and 'add to stock'.

Рисунок 3.9 — Форма створення користувача

Для забезпечення безпеки всі паролі хешуються:

```
$name = mysqli_real_escape_string($con,$_POST['user']) ;
$pass =
mysqli_real_escape_string($con,password_hash($_POST['password
'],PASSWORD_BCRYPT));
e($name);
e($pass);
$s = " select * from users where name = '$name'";
$result = mysqli_query($con, $s);
$num = mysqli_num_rows($result);
if($num == 1){
print("<script language=javascript>
```



```

window.alert('пользователь уже существует');
window.location.href='http://pract.com/main.php#popup'
</script>");
}else{
$reg= " insert into users(name , password, ad) values
('$name', '$pass','0')";
mysqli_query($con, $reg);
print("<script language=javascript>
window.alert('Успешно');
window.location.href='http://pract.com/main.php'
</script>");
}

```

Э змога передивитись існуючих користувачів та можливість їх видалення (рис 3.10).

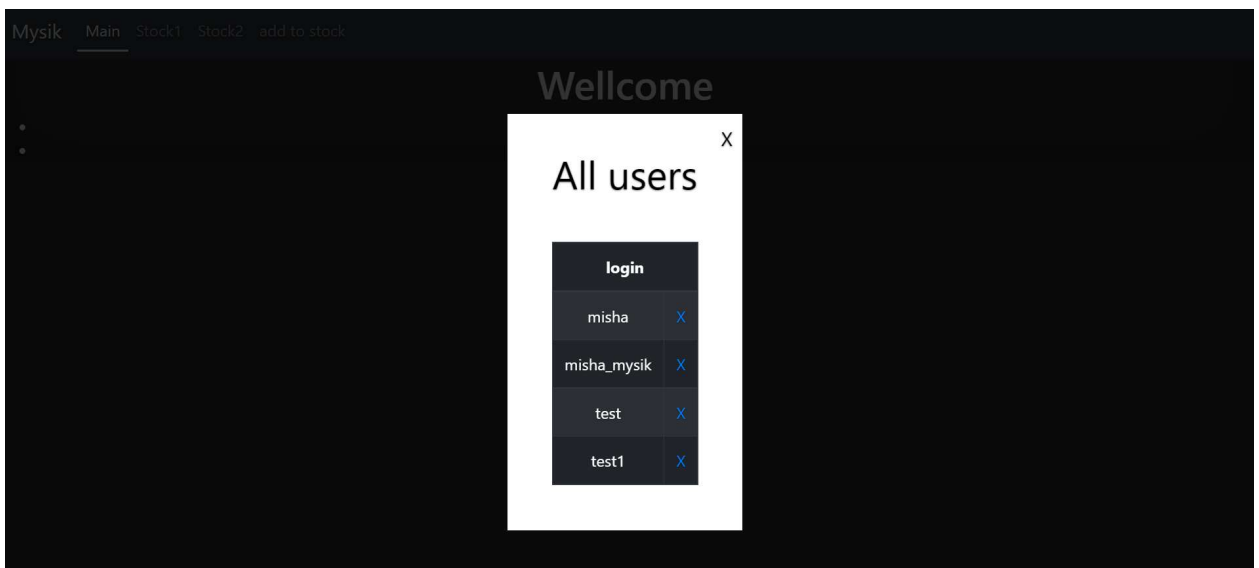


Рисунок 3.10 – Вікно існуючих користувачів

Нині існує 2 комори у яких є певні одиниці товару, інформація яка показана:

- категорія товару;
- назва товару, є унікальною;
- кількість у наявності;
- остання дата оновлення даних.

Вигляд їх на сторінці можемо побачити у вигляді таблиці (рис 3.11).

category	name ▼	amount	date	delete
2	yy-13	115	2021-05-20	X
5	w-5	56	2021-05-12	X
12	test	222	2021-05-20	X
2	tem1603	644	2019-01-02	X
4	rr-12	45	2021-05-11	X
1	qq	211	2021-05-20	X
2	pp-16	115	2021-05-20	X

Page 1 of 2

1 2

Рисунок 3.11 – Виведення даних

Сторінку можна сортувати за категоріями, алфавітним порядком, кількістю та датою останнього оновлення, для цього у заголовках розміщені посилання які змінюють запит на виведення даних:

```

$sort_list = array(
    'category_asc'    => '`category`',
    'category_desc'  => '`category` DESC',
    'name_asc'       => '`names`',
    'name_desc'      => '`names` DESC',
    'amount_asc'     => '`amount`',
    'amount_desc'    => '`amount` DESC',
    'date_asc'       => '`datearrival`',
    'date_desc'      => '`datearrival` DESC',
);
/* ФУНКЦІЯ ВИВОДА ССЫЛОК */
function sort_link_th($title, $a, $b) {
    $sort = @$_GET['sort'];

    if ($sort == $a) {
        return '<a class="active" href="?sort=' . $b
. '>' . $title . ' <i>▲</i></a>';
    } elseif ($sort == $b) {
        return '<a class="active" href="?sort=' . $a
. '>' . $title . ' <i>▼</i></a>';
    } else {
        return '<a href="?sort=' . $a . '>' . $title
. '</a>';
    }
}

```

Адміністратор може додавати дані до існуючої бази(рис 3.12)

Рисунок 3.12 – Додавання даних

Після створення позиції його буде переправлено на сторінку з оновленою інформацією, під час зміни автоматично оновлюється дата на поточну:

```

$category =
mysqli_real_escape_string($con,$_POST['category']);
$name= mysqli_real_escape_string($con,$_POST['thename']);
$amount = mysqli_real_escape_string($con,$_POST['amount']);
$hiredate = mysqli_real_escape_string($con,$_POST['dateof']);

$stock = mysqli_real_escape_string($con,$_POST['stock']);

if($stock == '1'){
    $sql = "INSERT INTO stock1 (category, names, amount,
datearrival)
VALUES ($category, '$name' , $amount ,
STR_TO_DATE('$hiredate', '%Y-%m-%d'))";
    if ($con->query($sql) === TRUE) {
        echo header('location:stock1.php');
    } else {
        echo "Error: " . $sql . "<br>" . $con->error;
    }
} else{
    $sql = "INSERT INTO stock2 (category, names, amount,
datearrival)
VALUES ($category, '$name' , $amount ,
STR_TO_DATE('$hiredate', '%Y-%m-%d'))";
    if ($con->query($sql) === TRUE) {
        echo header('location:stock2.php');
    } else {
        echo "Error: " . $sql . "<br>" . $con->error;
    }
}
$con->close();

```

Оновлення інформації виконується у режимі реального часу (рис3.13), також дата змінюється на поточну:

category	name	amount	date	delete
1	gl-69	19	2021-05-27	X
1	mk-13	100	2021-05-20	X

Рисунок 3.13 – Оновлення інформації

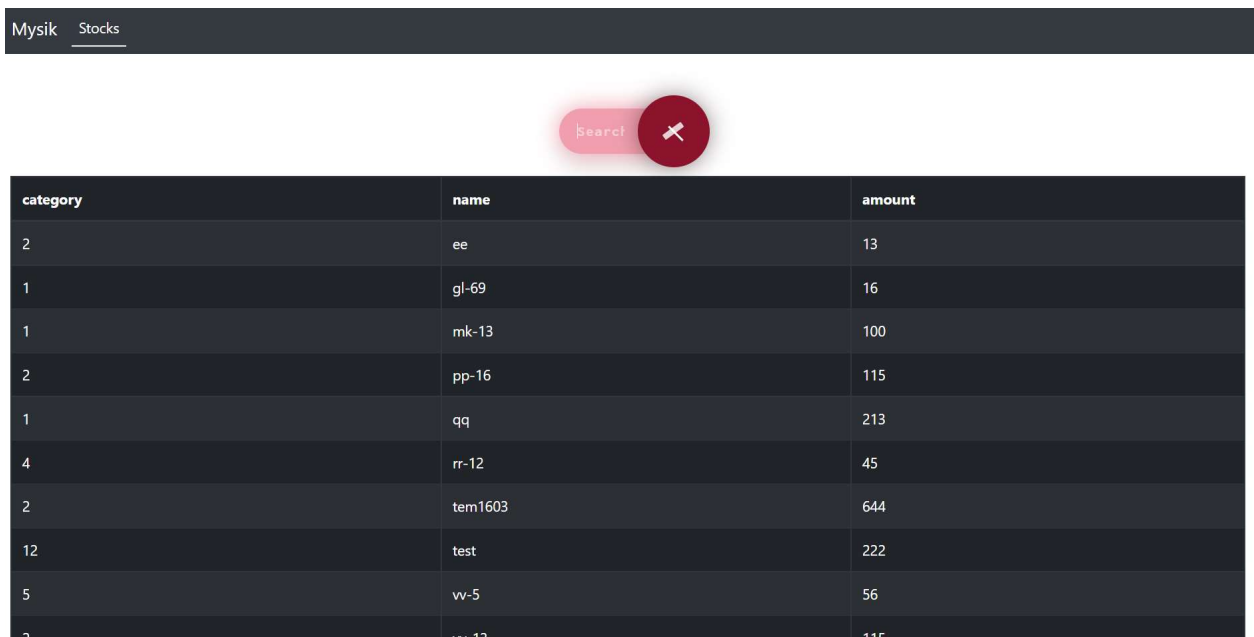
Функція, що оновлює дані:

```
$(function () {
    var oldVal, newVal, names, updateDate;
    var today = new Date();
    var dd = String(today.getDate()).padStart(2, '0');
    var mm = String(today.getMonth() + 1).padStart(2, '0');
    //January is 0!
    var yyyy = today.getFullYear();
    //2020-05-20
    $(' .edit').keypress(function(e) {
        if(e.which == 13){
            return false;
        }
    });
    $(' .edit').focus(function() {
        oldVal = $(this).text();
        names = $(this).data('id');
        today = yyyy + '-' + mm + '-' + dd;

    }).blur(function () {
        newVal = $(this).text();
        if(newVal != oldVal && newVal >= 0){
            $.ajax({
                url: 'stock1.php',
                type: 'POST',
                data: {new_val: newVal, name: names,
new_date: today},
                success: function(res) {
                    console.log(res);
                    location.reload();
                },
                error: function() {
                    alert('Error!');
                }
            });
        }
    });
    else{
        console.log('problem')
    }
});
```

```
});
```

Після авторизації звичайного користувача, його перенаправляє на сторінку з вибіркою усіх існуючих позицій(рис 3.14).



The screenshot shows a dark-themed user interface. At the top, there is a navigation bar with 'Mysik' and 'Stocks' links. Below it is a search bar with a red 'Search' button and a red circular icon with a white 'X'. The main content is a table with three columns: 'category', 'name', and 'amount'. The table contains the following data:

category	name	amount
2	ee	13
1	gl-69	16
1	mk-13	100
2	pp-16	115
1	qq	213
4	rr-12	45
2	tem1603	644
12	test	222
5	w-5	56
2	w-13	115

Рисунок 3.14 – Сторінка користувача

На ній відображається вибірка з усіх існуючих таблиць та при отриманні однакових позицій їх сума підсумовується:

```
$result = mysqli_query($con, "SELECT `stock1`.`category`,
`stock1`.`names`, (stock1.amount+NVL(stock2.amount, 0)) AS
amount from stock1 LEFT JOIN stock2 on
stock1.`names`=stock2.`names` ");
```

```
while ($row = mysqli_fetch_array($result)) {
    echo "<tr class='ul'>
    <td>" . $row['category'] . "</td>
    <td class='name'>" . $row['names'] . "</td>
    <td>" . $row['amount'] . "</td>
    </tr>";
}
?>
```

Для більш зручного пошуку інформації, була впроваджена функція пошуку (рис 3.15), вона відображає варіанти запиту у режимі на льоту:

```
let input = document.querySelector('#search__input');
input.oninput = function(){
    let value = this.value.trim();
    let list = document.querySelectorAll('.ul');

    if(value != ''){
```

```
list.forEach(elem => {
  if(elem.innerText.search(value) == -1){
    elem.classList.add('hide__search');
  }
});
} else{
  list.forEach(elem => {
    elem.classList.remove('hide__search');
  })
}

console.log(this.value);
}
```

Mysik Stocks



category	name	amount
2	ee	13
2	tem1603	644
12	test	222

Page 1 of 1

1

Рисунок 3.15 – Пошук даних

ВИСНОВКИ

У цій роботі було створено програмний продукт, хмарне сховище, з вимогами які були розроблені під час аналізу та розробки технічного завдання. Робота була розбита на етапи з приблизними оцінками на її виконання, було розроблені допоміжні діаграми для більш детальної реалізації. Створено декілька типів користувачів з різним виглядом інтерфейсів та доступними можливостями: редагування, щодо існуючої, інформації; створення та видалення її; пошук інформації використовуючи певні фільтри пошуку; створення та видалення користувачів, що було розроблено за допомогою:

- OS(open server panel) – відповідає за створення та впровадження серверного середовища;
- Maria DB та PhpMyAdmin – впровадження субд для проекту;
- PHP, Javascript – логіка, між серверною та клієнтською частинами, бізнес-логіка роботи;
- PhpMyStorm – редактор коду.

Надалі проект можна розширювати, а також використовувати у взаємодії з інтернет-магазином, додавання іншого можливого функціоналу.

СПИСОК ЛІТЕРАТУРИ

1. “Что такое протокол сайта? Как работает HTTP и HTTPS — Словарь — Линкбилдер.” <https://linkbuilder.su/glossary/protokol-sayta> (accessed May 31, 2021).
2. “Найпоширеніші типи сайтів 6 типів сайтів | Інфографіка — Internetdevels офіційний блог.” <https://internetdevels.ua/blog/most-common-types-of-websites> (accessed May 31, 2021).
3. “Open Server — профессиональный инструмент веб-разработчика под Windows / Хабр.” <https://habr.com/ru/post/137388/> (accessed May 31, 2021).
4. “Сравнение 5 популярных CMS: особенности, плюсы, минусы / Блог компании VDSina.ru / Хабр.” <https://habr.com/ru/company/vdsina/blog/514764/> (accessed May 31, 2021).
5. “CMS – Системы управления сайтом / Хабр.” <https://habr.com/ru/hub/cms/> (accessed May 31, 2021).
6. “Как создать сайт? Структура и контент / Хабр.” <https://habr.com/ru/post/501430/> (accessed May 31, 2021).
7. ramil_trinion, “Перевод стандарта IDEF0 на русском языке / Хабр.” <https://habr.com/ru/post/515466/> (accessed May 31, 2021).
8. “Двадцать лет с юзкейсами: выжимаем практический опыт / Блог компании QIWI / Хабр.” <https://habr.com/ru/company/qiwi/blog/346438/> (accessed May 31, 2021).
9. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Санкт-Петербург — 2019. — 832 с.
10. Сидельников Г. Визуальный CSS / Вильямс — 2021. — 240с.
11. Дакетт Дж. HTML и CSS. Разработка и дизайн веб-сайтов/ Эксмо — 2020. — 408с.
12. Морган Н. Javascript для дітей. / Львів— 2016. — 288с.
13. Тейлор А. SQL для чайников. / Москва — 2020. — 544с.
14. Бьюли А. Изучаем SQL./ Санкт-Петербург — 2015. — 312с.

15. Мартін Р. Чистий код./ Львів— 2019. — 368с.
16. Стефанов С. JavaScript. Шаблоны / Санкт-Петербург — 2018. — 272с.

ДОДАТОК

ЛІСТИНГ МОДУЛІВ ЕКСПЕРТНОЇ СИСТЕМИ

Лістинг коду **index.php**:

```
<?php
require 'xss.php';
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>dizel</title>

  <style type="text/css">
    body{
      background-color: #333 !important;
      text-shadow: 0 0.05rem 0.1rem rgba(0, 0, 0, .5)
!important;
      box-shadow: inset 0 0 5rem rgba(0, 0, 0, .5)
!important;
      color: #fff !important;
    }
  }
  .container-fluid {
    padding-right: 20px;
    padding: 1em;
    border-radius: 0;
    width: 95%;
    margin: auto !important;
    margin-top: 20px !important;
    margin-bottom: 1em !important;
  }
  .active{
    border-bottom: 2px solid white;
  }
  .btn-lg{
    padding: .75rem 1.25rem;
    font-weight: 700;
    color: #333;
    text-shadow: none;
    background-color: #fff;
    border: .05rem solid #fff;
  }
</style>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo
tstrap.min.css" integrity="sha384-
```

```

Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6
JXm" crossorigin="anonymous">
</head>

<body class="text-center">
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
  <a class="navbar-brand" href="#">Mysik</a>
  <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#collapsibleNavbar">
  <span class="navbar-toggler-icon"></span>
</button>
  <div class="collapse navbar-collapse"
id="collapsibleNavbar">
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link active" href="index.php">login</a>
    </li>
  </ul>
</div>
</nav>
<div class="container ">
  <div class='row'>
    <div class="col-md-12 justify-content-center">
      <h2> Login Here </h2>
      <form action="validation.php" method="post">
        <div class="form-group">
          <label><?php echo _e($user->user);?>Username
          <input type="text" name="user" class="form-control"
required> </label>
        </div>
        <input type = "hidden" name = "csrf" value = "<?php
echo $csrf ?>">
        <div class="form-group">
          <label>Password <?php echo _e($user->password);?>
          <input type="password" name="password" class="form-
control" required> </label>
        </div>
        <button type="submit" class="btn btn-secondary">
Login </button>
      </form>

    </div>

  </div>

</div>

</div>

```

```
</body>
</html>
```

Лістинг коду **main.php**:

```
<?php
session_start();
require 'xss.php';
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "practt";

$con = new mysqli($servername, $username, $password,
$dbname);

if ($con->connect_error) {
    die("Connection failed: " . $con->connect_error);
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="magnific/magnific-
popup.css">
    <link rel="stylesheet" href="css/style.css">
    <title>dizel</title>

    <style type="text/css">
        body{
            background-color: #333 !important;
            text-shadow: 0 0.05rem 0.1rem rgba(0, 0, 0, .5)
!important;
            box-shadow: inset 0 0 5rem rgba(0, 0, 0, .5)
!important;
            color: #fff !important;
        }
    }
    .container-fluid {
        padding-right: 20px;
        padding: 1em;
        border-radius: 0;
        width: 95%;
        margin: auto !important;
        margin-top: 20px !important;
        margin-bottom: 1em !important;
    }
    .active{
```

```

        border-bottom: 2px solid white;
    }
    .btn-lg{
        padding: .75rem 1.25rem;
        font-weight: 700;
        color: #333;
        text-shadow: none;
        background-color: #fff;
        border: .05rem solid #fff;
    }
    .hidden{
        display: none;
    }
</style>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo
tstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6
JXm" crossorigin="anonymous">
</head>

<body class="text-center">
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
  <a class="navbar-brand" href="#">Mysik</a>
  <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse"
id="collapsibleNavbar">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link active" href="#">Main</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="stock1.php">Stock1</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="stock2.php">Stock2</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="new_stock.php">add to
stock</a>
      </li>
    </ul>
  </div>
</nav>

<div>
  <h1>Wellcome <?php echo $_SESSION['username']; ?></h1>

```

```

        <nav>
            <ul>
                <li><a href="#popup"
class="create__popup">create user</a></li>
                <li><a href="#popup-user">list
user</a></li>
            </ul>
            <div id="popup" class="popup">
                <div class="popup__body">
                    <div class="popup__content">
                        <a href="#" class="popup__close
close-popup">X</a>
                        <div class="popup__title">
                            New user
                        </div>
                        <div class="popup__text">
                            <form action="registration.php"
method="post" id="create">
                                <div class="group">
                                    <input name="user"
type="text" required>
                                    <span
class="highlight"></span>
                                    <span class="bar"></span>
                                    <label>Login</label>
                                </div>
                                <div class="group">
                                    <input name="password"
type="password" required>
                                    <span
class="highlight"></span>
                                    <span class="bar"></span>
                                    <label>Password</label>
                                </div>
                                <button type="submit"
class="btn btn-2 btn-2h">create</button>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
            <div id="popup-user" class="popup">
                <?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "practt";

$con = new mysqli($servername, $username, $password,
$dbname);

```

```

if ($con->connect_error) {
    die("Connection failed: " . $con->connect_error);
}
php?>
        <div class="popup_body">
            <div class="popup_content">
                <a href="#" class="popup_close
close-popup">X</a>
                <div class="popup_title">
                    All users
                </div>
                <div class="popup_text">
                    <div class="container-fluid">
                        <table class="table table-
dark table-striped table-bordered">
                            <thead>
                                <tr>
                                    <th
colspan="2">login</th>
                                </tr>
                            </thead>
                            <tbody>
                                <?php
                                if (isset($_GET['del' ]))
                                {
                                    $id = $_GET['del'];
                                    $query = "DELETE FROM
users WHERE name='$id'";
                                    mysqli_query($con,
$query) or die(mysqli_error($con));
                                    $result =
                                    mysqli_query($con, "SELECT name FROM users ");
                                    while ($row =
mysqli_fetch_array($result)) {
                                        echo "<tr>
                                        <td>" .
                                        $row['name'] . "</td>
                                        <td><a
href='?del=". $row['name'] ."#popup-user
'>X</a></td>
                                        </tr>";
                                    }
                                ?>
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>

```

```

        </div>
    </div>
</div>
</nav>
</div>
    <script src="magnific/jquery.magnific-
popup.min.js"></script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquer
y.min.js"></script>
    <script src="script/main.js"></script>
</body>
</html>

```

Лістинг коду add_stock.php:

```

<?php
session_start();
require 'xss.php';
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "practt";

$con = new mysqli($servername, $username, $password,
$dbname);

if ($con->connect_error) {
    die("Connection failed: " . $con->connect_error);
}

//создаем ключ для функции hash_hmac
if (empty($_SESSION['key']))
    $_SESSION['key'] = bin2hex(random_bytes(32));
//создаем токен CSRF
$csrf = hash_hmac('sha256', 'this is some string:',
$_SESSION['key']);
//проверить токен
if (isset($_POST['submit'])) {
    if(hash_equals($csrf, $_POST['csrf'])) {
        echo 'Well done';
    } else
        echo 'CSRF Token Failed!' ;
}

$category =
mysqli_real_escape_string($con,$_POST['category']);
$name= mysqli_real_escape_string($con,$_POST['thename']);
$amount = mysqli_real_escape_string($con,$_POST['amount']);
$hireddate = mysqli_real_escape_string($con,$_POST['dateof']);

$stock = mysqli_real_escape_string($con,$_POST['stock']);

```



```

e($category);
e($theName);
e($amount);
e($hiredate);
if($stock == '1'){
    $sql = "INSERT INTO stock1 (category, names, amount,
datearrival)
VALUES ($category, '$theName' , $amount ,
STR_TO_DATE('$hiredate', '%Y-%m-%d'))";
    if ($con->query($sql) === TRUE) {
        echo header('location:stock1.php');
    } else {
        echo "Error: " . $sql . "<br>" . $con->error;
    }
} else{
    $sql = "INSERT INTO stock2 (category, names, amount,
datearrival)
VALUES ($category, '$theName' , $amount ,
STR_TO_DATE('$hiredate', '%Y-%m-%d'))";
    if ($con->query($sql) === TRUE) {
        echo header('location:stock2.php');
    } else {
        echo "Error: " . $sql . "<br>" . $con->error;
    }
}
}

```

```
$con->close();
```

```
?>
```

Лістинг коду **new_stock.php**:

```

<?php
session_start();
require 'xss.php';
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "practt";

$con = new mysqli($servername, $username, $password,
$dbname);

if ($con->connect_error) {
    die("Connection failed: " . $con->connect_error);
}

```

```

?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>dizel</title>
  <link rel="stylesheet" href="css/form_add.css">
  <style type="text/css">
    body{
      background-color: #333 !important;
      text-shadow: 0 0.05rem 0.1rem rgba(0, 0, 0, .5)
!important;
      box-shadow: inset 0 0 5rem rgba(0, 0, 0, .5)
!important;
      color: #fff !important;
    }
  }
  .container-fluid {
    padding-right: 20px;
    padding: 1em;
    border-radius: 0;
    width: 95%;
    margin: auto !important;
    margin-top: 20px !important;
    margin-bottom: 1em !important;
  }
  .active{
    border-bottom: 2px solid white;
  }
  .btn-lg{
    padding: .75rem 1.25rem;
    font-weight: 700;
    color: #333;
    text-shadow: none;
    background-color: #fff;
    border: .05rem solid #fff;
  }
  </style>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo
tstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6
JXm" crossorigin="anonymous">
</head>

<body class="text-center">
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
  <a class="navbar-brand" href="#">Mysik</a>
  <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#collapsibleNavbar">
  <span class="navbar-toggler-icon"></span>

```

```

        </button>
        <div class="collapse navbar-collapse"
id="collapsibleNavbar">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="main.php">Main</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link"
href="stock1.php">Stock1</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link"
href="stock2.php">Stock2</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link active" href="#">add to
stock</a>
                </li>
            </ul>
        </div>
    </nav>

<div>
    <div class="wrapper">
        <div class="container">

            <form class="form">

                <label for="exampleSelect1">Stock</label>
                <select name="stock"
id="exampleSelect1">
                    <option>1</option>
                    <option>2</option>
                </select>

                <input type="text" placeholder="category"
name="category">
                <input type="text" placeholder="name"
name="names">
                <input type="text" placeholder="amount"
name="amount">
                <input type="date" placeholder="date"
name="dateof">
                <button type="submit" id="login-
button">Insert</button>

            </form>
        </div>
    </div>

```

```

    </div></div>

</body>
</html>

```

Лістинг коду **registration.php**:

```

<?php
//header("location:index.php");
session_start();
require 'xss.php';
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "practt";

$con = new mysqli($servername, $username, $password,
$dbname);

if ($con->connect_error) {
    die("Connection failed: " . $con->connect_error);
}

$name = mysqli_real_escape_string($con,$_POST['user']) ;
$pass =
mysqli_real_escape_string($con,password_hash($_POST['password
'], PASSWORD_BCRYPT));
e($name);
e($pass);
$s = " select * from users where name = '$name'";
$result = mysqli_query($con, $s);
$num = mysqli_num_rows($result);
if($num == 1){
    print("<script language=javascript>
        window.alert('пользователь уже существует');
        window.location.href='http://pract.com/main.php#popup'
        </script>");
}else{
$reg= " insert into users(name , password, ad) values
('$name', '$pass','0')";
mysqli_query($con, $reg);
print("<script language=javascript>
        window.alert('Успешно');
        window.location.href='http://pract.com/main.php'
        </script>");
}

?>

```

Лістинг коду stock1.php:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "practt";

$con = new mysqli($servername, $username, $password,
$dbname);

if ($con->connect_error) {
    die("Connection failed: " . $con->connect_error);
}

// Изменение общего количества записей на страницу
if ($_GET['pagesize'] != "") {
    $total_records_per_page = $_GET['pagesize'];
    $ok = filter_var($total_records_per_page,
FILTER_VALIDATE_INT);
    if (is_null($ok) || ($ok === false)) {
        require_once('error_message.php');
        die();
    } else {
        if ($total_records_per_page <= 0) {
            require_once('error_message.php');
            die();
        }
    }
} else {
    $total_records_per_page = 7;
}

// Номер страницы
if ($_GET['page'] != "") {
    $page = $_GET['page'];
    $ok = filter_var($page, FILTER_VALIDATE_INT);
    if (is_null($ok) || ($ok === false)) {
        require_once('error_message.php');
        die();
    } else {
        if ($page <= 0) {
            require_once('error_message.php');
            die();
        }
    }
} else {
    $page = 1;
}

//Расчет значения OFFSET
$offset = ($page - 1) * $total_records_per_page;
$previous_page = $page - 1;
```

```

$next_page = $page + 1;

//Именения количества страниц
$result_count = mysqli_query($con, "SELECT COUNT(*) As
total_records FROM stock1");
$total_records = mysqli_fetch_array($result_count);
$total_records = $total_records['total_records'];
$total_no_of_pages = ceil($total_records /
$total_records_per_page);

//update table
function update_option(){
    $con = new mysqli("localhost", "root","", "practt");
    $newAmount = (int)$_POST['new_val'];
    $hireDate = mysqli_real_escape_string($con,
$_POST['new_date']);
    $names = mysqli_real_escape_string($con, $_POST['name']);
    $query = "update stock1 set amount = $newAmount,
datearrival = STR_TO_DATE('$hireDate', '%Y-%m-%d') where
names = '$names' ";
    $res = mysqli_query($con, $query);
    if(mysqli_affected_rows($con)) return true;
    else return false;
}

// redact table
if( isset($_POST['new_val'])) {
    if(update_option()){
        exit("saves");
    }{
        exit('some problem of save');
    }
    exit;
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>dizel</title>

    <style type="text/css">
        .container-fluid {
            padding-right: 20px;
            padding: 1em;
            border-radius: 0;
            width: 95%;
            margin: auto !important;
            margin-top: 20px !important;
            margin-bottom: 1em !important;
        }

```

```

    .active{
        border-bottom: 2px solid white;
    }
    .page-item a{
        color: #333;
    }
    .page-item.active a{
        color: #fff !important;
        background: #333 !important;
        border: #333 !important;
    }
</style>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo
tstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6
JXm" crossorigin="anonymous">
    <link rel="stylesheet" href="./css/redact.css">
</head>

<body>
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
    <a class="navbar-brand" href="#">Mysik</a>
    <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#collapsibleNavbar">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse"
id="collapsibleNavbar">
        <ul class="navbar-nav">
            <li class="nav-item">
                <a class="nav-link " href="main.php">Main</a>
            </li>
            <li class="nav-item active">
                <a class="nav-link"
href="stock1.php">Stock1</a>
            </li>
            <li class="nav-item">
                <a class="nav-link"
href="stock2.php">Stock2</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="new_stock.php">add
to stock</a>
            </li>
        </ul>
    </div>
</nav>
<div class="container-fluid">
    <table class="table table-dark table-striped table-
bordered">

```

```

        <thead>
        <tr>
            <th><?php echo sort_link_th('category',
'category_asc', 'category_desc'); ?></th>
            <th><?php echo sort_link_th('name', 'name_asc',
'name_desc'); ?></th>
            <th><?php echo sort_link_th('amount',
'amount_asc', 'amount_desc'); ?></th>
            <th><?php echo sort_link_th('date', 'date_asc',
'date_desc'); ?></th>
            <th>delete</th>
<!--         <th>category</th>-->
<!--         <th>name</th>-->
<!--         <th>amount</th>-->
<!--         <th>date of arrival</th>-->
        </tr>
    </thead>
    <tbody>
    <?php
    /* Все варианты сортировки */
    $sort_list = array(
        'category_asc'    => '`category`',
        'category_desc'  => '`category` DESC',
        'name_asc'       => '`names`',
        'name_desc'      => '`names` DESC',
        'amount_asc'     => '`amount`',
        'amount_desc'    => '`amount` DESC',
        'date_asc'       => '`datearrival`',
        'date_desc'      => '`datearrival` DESC',
    );
    /* Функция вывода ссылок */
    function sort_link_th($title, $a, $b) {
        $sort = @$_GET['sort'];

        if ($sort == $a) {
            return '<a class="active" href="?sort=' . $b
. '>' . $title . ' <i>▲</i></a>';
        } elseif ($sort == $b) {
            return '<a class="active" href="?sort=' . $a
. '>' . $title . ' <i>▼</i></a>';
        } else {
            return '<a href="?sort=' . $a . '>' . $title
. '</a>';
        }
    }

    /* Проверка GET-переменной */
    $sort = @$_GET['sort'];
    if (array_key_exists($sort, $sort_list)) {
        $sort_sql = $sort_list[$sort];
    } else {
        $sort_sql = reset($sort_list);
    }

```



```

    }
    if (isset($_GET['del'])) {
        $id = $_GET['del'];
        $query = "DELETE FROM stock1 WHERE names='$id'";
        mysqli_query($con, $query) or
die(mysqli_error($con));
    }
    $result = mysqli_query($con, "SELECT * FROM stock1
ORDER BY {$sort_sql} LIMIT $offset, $total_records_per_page
");

    while ($row = mysqli_fetch_array($result)) {
        echo "<tr>
<td>" . $row['category'] . "</td>
<td>" . $row['names'] . "</td>
<td><div class='edit' contenteditable data-
id=\".$row['names'].\">" . $row['amount'] . "</div></td>
<td>" . $row['datearrival'] . "</td>
<td><a href='?del=\". $row['names'] . \" ' >X</a></td>
</tr>";
    }
    ?>
</tbody>
</table>
</div>

<div class="pagination pg-blue justify-content-center">
    <strong>Page <?php echo $page . " of " .
$total_no_of_pages; ?></strong>
</div>

<nav aria-label="Page navigation example">
    <ul class="pagination pg-blue justify-content-center">
        <?php
        for ($i = 1; $i <= $total_no_of_pages; $i++):?>
            <li class=<?php if ($page == $i) {
                echo "\"page-item active \"";
            } else {
                echo "\"page-item\"";
            } ?>>
                <a class="page-link"
                    <?php echo
"href='?page=$i&pagesize=$total_records_per_page'"; ?>>
                    <?php echo $i; ?>
                </a>
            </li>

            <?php endfor; ?>
        </ul>
    </nav>

```

```

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquer
y.min.js"></script>
<script src="script/table_redact.js"></script>
</body>
</html>

```

Лістинг коду **stock_list.php**:

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "practt";

$con = new mysqli($servername, $username, $password,
$dbname);

if ($con->connect_error) {
    die("Connection failed: " . $con->connect_error);
}

// Изменение общего количества записей на страницу
if ($_GET['pagesize'] != "") {
    $total_records_per_page = $_GET['pagesize'];
    $ok = filter_var($total_records_per_page,
FILTER_VALIDATE_INT);
    if (is_null($ok) || ($ok === false)) {
        require_once('error_message.php');
        die();
    } else {
        if ($total_records_per_page <= 0) {
            require_once('error_message.php');
            die();
        }
    }
} else {
    $total_records_per_page = 20;
}

// Номер страницы
if ($_GET['page'] != "") {
    $page = $_GET['page'];
    $ok = filter_var($page, FILTER_VALIDATE_INT);
    if (is_null($ok) || ($ok === false)) {
        require_once('error_message.php');
        die();
    } else {
        if ($page <= 0) {
            require_once('error_message.php');

```

```

        die();
    }
}
} else {
    $page = 1;
}

//Расчет значения OFFSET
$offset = ($page - 1) * $total_records_per_page;
$previous_page = $page - 1;
$next_page = $page + 1;

//Именения количества страниц
$result_count = mysqli_query($con, "SELECT COUNT(*) As
total_records FROM stock1");
$total_records = mysqli_fetch_array($result_count);
$total_records = $total_records['total_records'];
$total_no_of_pages = ceil($total_records /
$total_records_per_page);

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>dizel</title>
    <link rel="stylesheet" href="css/search.css">

    <style type="text/css">
        .container-fluid {
            padding-right: 20px;
            padding: 1em;
            border-radius: 0;
            width: 95%;
            margin: auto !important;
            margin-top: 20px !important;
            margin-bottom: 1em !important;
        }
        .active{
            border-bottom: 2px solid white;
        }
        .page-item a{
            color: #333;
        }
        .page-item.active a{
            color: #fff !important;
            background: #333 !important;
            border: #333 !important;
        }

        .hide__search{

```

```

        display: none;
    }
</style>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo
tstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6
JXm" crossorigin="anonymous">
</head>

<body>
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
  <a class="navbar-brand" href="#">Mysik</a>
  <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse"
id="collapsibleNavbar">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="#">Stocks</a>
      </li>
    </ul>
  </div>
</nav>
<div class="container-fluid">
  <div class="container">
    <input id="search_input" class="search_input"
type="text" placeholder="Search...">
    <div class="search"></div>
  </div>
  <table class="table table-dark table-striped table-
bordered">
    <thead>
      <tr>
<!--          <th>--><?php //echo sort_link_th('category',
'category_asc', 'category_desc'); ?><!--</th>-->
<!--          <th>--><?php //echo sort_link_th('name',
'name_asc', 'name_desc'); ?><!--</th>-->
<!--          <th>--><?php //echo sort_link_th('amount',
'amount_asc', 'amount_desc'); ?><!--</th>-->
<!--          <th>--><?php //echo sort_link_th('date',
'date_asc', 'date_desc'); ?><!--</th>-->
<!--          <th>Stock #</th>-->
          <th>category</th>
          <th>name</th>
          <th>amount</th>
<!--          <th>date of arrival</th>-->
        </tr>
    </thead>
    <tbody>

```

```

<?php
/* Все варианты сортировки */
$sort_list = array(
    'category_asc'    => '`category`',
    'category_desc'  => '`category` DESC',
    'name_asc'       => '`names`',
    'name_desc'      => '`names` DESC',
    'amount_asc'     => '`amount`',
    'amount_desc'    => '`amount` DESC',
    'date_asc'       => '`datearrival`',
    'date_desc'      => '`datearrival` DESC',
);
/* Функция вывода ссылок */
function sort_link_th($title, $a, $b) {
    $sort = @$_GET['sort'];

    if ($sort == $a) {
        return '<a class="active" href="?sort=' . $b
. ' ">' . $title . ' <i>▲</i></a>';
    } elseif ($sort == $b) {
        return '<a class="active" href="?sort=' . $a
. ' ">' . $title . ' <i>▼</i></a>';
    } else {
        return '<a href="?sort=' . $a . ' ">' . $title
. '</a>';
    }
}

/* Проверка GET-переменной */
$sort = @$_GET['sort'];
if (array_key_exists($sort, $sort_list)) {
    $sort_sql = $sort_list[$sort];
} else {
    $sort_sql = reset($sort_list);
}
$result = mysqli_query($con, "SELECT
`stock1`.`category`, `stock1`.`names`,
(stock1.amount+NVL(stock2.amount, 0)) AS amount from stock1
LEFT JOIN stock2 on stock1.`names`=stock2.`names` ");

    while ($row = mysqli_fetch_array($result)) {
        echo "<tr class='ul'>
<td>" . $row['category'] . "</td>
<td class='name'>" . $row['names'] . "</td>
<td>" . $row['amount'] . "</td>
</tr>";
    }
    ?>
</tbody>
</table>
</div>

```

```

<div class="pagination pg-blue justify-content-center">
    <strong>Page <?php echo $page . " of " .
$total_no_of_pages; ?></strong>
</div>

<nav aria-label="Page navigation example">
    <ul class="pagination pg-blue justify-content-center">
        <?php
        for ($i = 1; $i <= $total_no_of_pages; $i++):?>
            <li class=<?php if ($page == $i) {
                echo "\"page-item active \"";
            } else {
                echo "\"page-item \"";
            } ?>>
                <a class="page-link"
                    <?php echo
"href='?page=$i&pagesize=$total_records_per_page'"; ?>>
                    <?php echo $i; ?>
                </a>
            </li>

            <?php endfor; ?>
        </ul>
    </nav>
<script src="script/search.js"></script>
</body>
</html>

```

Лістинг коду **validation.php**:

```

<?php
session_start();
require 'xss.php';
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "practt";

//создаем ключ для функции hash_hmac
if (empty($_SESSION['key']))
$_SESSION['key'] = bin2hex(random_bytes(32));
//создаем токен CSRF
$csrf = hash_hmac('sha256', 'this is some string:',
$_SESSION['key']);
//проверить токен
if (isset($_POST['submit'])) {
    if(hash_equals($csrf, $_POST['csrf'])) {
        echo 'Well done';
    } else

```

```

        echo 'CSRF Token Failed!' ;
    }

$con = new mysqli($servername, $username, $password,
$dbname);

if ($con->connect_error) {
    die("Connection failed: " . $con->connect_error);
}

$name = e(mysqli_real_escape_string($con,$_POST['user'])) ;
$pass = e(mysqli_real_escape_string($con,
$_POST['password']));
//$pass = password_hash($_POST['password'], PASSWORD_BCRYPT);
$s = " select * from users where name = '$name'";
$hash = " select password from users where name = '$name'";
$admin = "select ad from users where name = '$name'";
$result = mysqli_query($con, $s);
$resAdmin = mysqli_query($con, $admin);
$pass1 = mysqli_query($con, $hash) or die ("Some error");
$bdPass = $pass1->fetch_assoc();
$resPass = password_verify($pass,$bdPass['password']);
$num = mysqli_num_rows($result);
if($num == 1 && $resPass == true && $resAdmin == 1){
    $_SESSION['username'] = $name;
    header('location:main.php');
}
elseif($num == 1 && $resPass == true && $resAdmin == 0){
    $_SESSION['username'] = $name;
    header('location:stock_list.php');
}
elseif($num == 1 && $resPass == false){
    print("<script language=javascript>
    window.alert('Пароль не верен');
    window.location.href='http://pract.com/index.php'
    </script>");
} else{
    print("<script language=javascript>
    window.alert('Такого пользователя не существует');
    window.location.href='http://pract.com/index.php'
    </script>");
}

?>

```

Лістинг коду **main.js**:

```

const popupLinks = document.querySelector('.popup-link');
const body = document.querySelector('body');
const lockPadding = document.querySelectorAll('.lock-
padding');

```

```

let unlock = true;

if (popupLinks.length > 0) {
  for (let index = 0; index < popupLinks.length; index++) {
    const popupLink = popupLinks[index];
    popupLink.addEventListener("click", function (e) {
      const popupName =
popupLink.getAttribute('href').replace('#', '');
      const curentPopup =
document.getElementById(popupName);
      popupOpen(curentPopup);
      e.preventDefault();
    });
  }
}

const popupCloseIcon = document.querySelectorAll('.close-
popup');
if (popupCloseIcon.length > 0) {
  for (let index = 0; index < popupCloseIcon.length;
index++){
    const el = popupCloseIcon[index];
    el.addEventListener('click', function (e) {
      popupClose(el.closest('.popup'));
      e.preventDefault();
    });
  }
}

function popupOpen(curentPopup) {
  if (curentPopup && unlock) {
    const popupActive =
document.querySelector('.popup.open');
    if (popupActive) {
      popupClose(popupActive, false);
    } else {
      bodyLock();
    }
    curentPopup.classList.add('open');
    curentPopup.addEventListener("click", function
(e) {
      if (!e.target.closest('.popup__content')) {
        popupClose(e.target.closest('.popup'));
      }
    });
  }
}

function popupClose(popupActive, doUnlock = true){
  if(unlock){
    popupActive.classList.remove('open');
    if(doUnlock){
      bodyUnlock();
    }
  }
}

```



```

    }
}

function bodyLock() {
    const lockPaddingValue = window.innerWidth -
document.querySelector('.wrapper').offsetWidth + 'px';
    for (let index = 0; index < lockPadding.length; index++)
    {
        const el = lockPadding[index];
        el.style.paddingRight = lockPaddingValue;
    }
    body.style.paddingRight = lockPaddingValue;
    body.classList.add('lock');

    unlock = false;
    setTimeout(function () {
        unlock = true;
    }, timeout);
}

function bodyUnLock() {
    setTimeout(function () {
        for (let index = 0; index < lockPadding.length;
index++) {
            const el = lockPadding[index];
            el.style.paddingRight = '0px';
        }
        body.style.paddingRight = '0px';
        body.classList.remove('lock');
    }, timeout);
    unlock = false;
    setTimeout(function () {
        unlock = true;
    }, timeout);
}

```

Лістинг коду **search.js**:

```

window.onload = () =>{
    // console.log(1);
    let input = document.querySelector('#search__input');
    input.oninput = function(){
        let value = this.value.trim();
        let list = document.querySelectorAll('.ul');

        if(value != ''){

            list.forEach(elem => {
                if(elem.innerText.search(value) == -1){
                    elem.classList.add('hide__search');
                }
            });
        }
    };
}

```

```

    } else{
        list.forEach(elem => {
            elem.classList.remove('hide__search');
        })
    }

    console.log(this.value);

}

};

```

Лістинг коду **table_redact.js**:

```

$(function (){
    var oldVal, newVal, names, updateDate;
    var today = new Date();
    var dd = String(today.getDate()).padStart(2, '0');
    var mm = String(today.getMonth() + 1).padStart(2, '0');
    //January is 0!
    var yyyy = today.getFullYear();
    //2020-05-20
    $('input.edit').keypress(function(e){
        if(e.which == 13){
            return false;
        }
    });
    $('input.edit').focus(function(){
        oldVal = $(this).text();
        names = $(this).data('id');
        today = yyyy + '-' + mm + '-' + dd;

    }).blur(function (){
        newVal = $(this).text();
        if(newVal != oldVal && newVal >= 0){
            $.ajax({
                url:'stock1.php',
                type: 'POST',
                data: {new_val: newVal, name:names,
new_date:today},
                success: function(res){
                    console.log(res);
                    location.reload();
                },
                error: function(){
                    alert('Error!');
                }
            })
        }
    }
    else{
        console.log('problem')
    }
});
});

```