

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

_____ Сергій Стіренко
(підпис)

“ ” _____ 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою “ Комп’ютерні системи та мережі ”
спеціальності 123 “Комп’ютерна інженерія”

на тему: WEB-сервіс онлайн оголошень

Виконав: студент 4 курсу, групи ІОЗ-91
(шифр групи)

Красулін Євгеній Сергійович

(прізвище, ім’я, по батькові)

(підпис)

Керівник _____ асистент Щур В. Ю.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) професор, д.т.н. Виноградов Ю. М

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент Доц. каф. ІСТ, к.т.н., доц. Алла Коган

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2023 р.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Інженерія програмного забезпечення комп’ютерних систем”

спеціальність 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій Стіренко

(підпис)

“ ” _____ 2023 р.

ЗАВДАННЯ

на бакалаврський дипломний проект студента

Красуліна Євгенія Сергійовича

1. Тема проекту WEB-сервіс онлайн оголошень

керівник проекту _____ асистент Щур В. Ю. _____,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 31 травня 2023 року №2102-с

2. Строк подання студентом проекту 8 червня 2023 р.

3. Вихідні дані до проекту технічна документація. теоретичні та статистичні дані, патенти на винахід

4. Зміст пояснювальної записки (перелік завдань, які потрібно розробити)

Розділ 1: Аналіз предметної області

Розділ 2: Теоретичні відомості інформаційної системи веб сервісу розміщення оголошень

Розділ 3: Проектування та розробка сервісу для покупки або продажу товарів

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) структурна схема системи, функціональна схема (діаграма класів), алгоритм дій програмного забезпечення.

6. Консультанти розділів проєкту

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	професор, д.т.н. Виноградов Ю. М		

7. Дата видачі завдання 27.02.2023 р.

Календарний план

№ п/п	Назва етапу дипломного проєкту	Строк виконання етапу проєкту	Примітка
1.	<i>Затвердження теми проєкту</i>	<i>31.03.2023</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>01.04.2023-15.04.2023</i>	
3.	<i>Програмна реалізація системи</i>	<i>15.04.2023-20.05.2023</i>	
4.	<i>Оформлення пояснювальної</i>	<i>15.04.2023-31.05.2023</i>	
5.	<i>Захист програмного продукту</i>	<i>05.06.2023</i>	
6.	<i>Передзахист</i>	<i>08.06.2023</i>	
7.	<i>Захист</i>	<i>23.06.2023</i>	

Студент Красулін Є.С.

(підпис)

Керівник проєкту Щур Ю. В.

(підпис)

Анотація

Кваліфікаційна робота присвячена детальному аналізу та розробці веб-сервісу для розміщення оголошень в онлайн-середовищі. У роботі проводиться огляд існуючих веб-застосунків, зокрема OLX, Work.ua, AUTO.RIA, Booking та Trello, що дозволяє отримати уявлення про особливості функціонування подібних сервісів. Також досліджується можливість запозичення досвіду та навчання від аналогічних проектів під час розробки власного сервісу.

У другому розділі розглядаються теоретичні відомості щодо інформаційної системи веб-сервісу розміщення оголошень. Детально описується процес розробки такої системи, структура програми та використані технології. Цей розділ надає фундаментальні знання для розуміння принципів функціонування веб-сервісу розміщення оголошень.

У третьому розділі розглядаються питання проектування та розробки самого сервісу для покупки або продажу товарів. Аналізується архітектура застосунку, проводиться огляд використаних технологій і здійснюється тестування розробленого застосунку. Цей розділ надає практичні відомості щодо розробки та функціонування веб-сервісу розміщення оголошень.

Annotation

The qualification work is devoted to the detailed analysis and development of a web service for posting ads in the online environment. The work reviews existing web applications, in particular OLX, Work.ua, AUTO.RIA, Booking and Trello, which allows you to get an idea of the features of the functioning of such services. The possibility of borrowing experience and learning from similar projects during the development of one's own service is also being explored.

The second section deals with theoretical information about the information system of the web service of placing ads. The process of developing such a system,

the structure of the program and the technologies used are described in detail. This section provides the fundamental knowledge to understand the principles of operation of the web service of placing ads.

The third section deals with design and development of the service itself for buying or selling goods. The architecture of the application is analyzed, the used technologies are reviewed, and the developed application is tested. This section provides practical information on the development and operation of a web-based ad serving service.

[illegible]

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ
на тему: «WEB-сервіс онлайн оголошень»

Київ – 2023

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Огляд існуючих вебзастосунків.....	6
1.1.1 OLX	6
1.1.2 Work.ua	9
1.1.3 AUTO.RIA	13
1.1.4 Booking	16
1.1.5 Trello.....	18
1.2 Запозичення досвіду аналогів при створенні свого проекту.....	19
РОЗДІЛ 2. ТЕОРЕТИЧНІ ВІДОМОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЕБ СЕРВІСУ РОЗМІЩЕННЯ ОГолошень.....	21
2.1 Опис процесу розробки інформаційної системи Веб-сервісу розміщення оголошень.....	21
2.2 Опис структури програми	23
2.3 Опис технологій для створення веб сервісу.....	26
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РОЗРОБКА СЕРВІСУ ДЛЯ ПОКУПКИ АБО ПРОДАЖУ ТОВАРІВ.....	32
3.1 Архітектура застосунку	32
3.2 Огляд використаних технологій	34
3.3 Тестування розробленого застосунку	37
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
ДОДАТОК 1.....	47

ВСТУП

Оголошення - це короткі тексти, які пропонують різноманітну інформацію. Як правило, оголошення спрямовані на комерційні операції (продаж або купівля різноманітних товарів, оренда тимчасового житла, афіші культурно-освітніх заходів, працевлаштування). Стало менше публічних повідомлень, які висвітлюють суспільно важливі проблеми або служать нагадуваннями чи застереженнями.

Раніше газети були найпоширенішим засобом для розміщення реклами. Кілька друкованих видавництв були зацікавлені у тому, щоб присвятити сторінку чи, можливо, цілий розділ друку сповіщень. Так розповсюджуються рекламні оголошення або повідомлення про продажі. Проте привабливість друкованих ЗМІ з часом впала, і вони втрачають свою аудиторію.

Таблички на автобусних зупинках і біля житлових будинків були ще одним застарілим способом фокусування оголошень. Однак такий підхід до розміщення реклами має ряд недоліків. В першу чергу, це близькість таких інструментів, а також захаращеність форумів неактуальними зауваженнями.

Сьогодні інтеграція реклами в життя сучасної людини можлива лише за допомогою технологій веб-розробки, які дозволяють охопити найбільшу аудиторію та реалізувати всі необхідні інструменти для виправданої та максимально раціональної роботи принципів дошки оголошень.

Метою проекту є створення веб-додатку дошки оголошень, який містить інструменти та канали зв'язку, необхідні для ефективного використання сервісу.

Робота вимагає від вас виконання наступних завдань:

- аналіз поточних рішень веб-додатків;
- визначення потреб веб-додатків;
- побудова архітектури власної програми;
- впровадження веб-додатку дошки оголошень.

					ІАЛЦ.467200.002 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Інтернет-дошки оголошень завжди використовували подібні функціональні можливості, щоб допомогти користувачам виконати необхідні завдання. Роль користувача загальна для таких систем, оскільки є лише дві сторони. Одна сторона створює пропозицію, а інша забезпечує попит. Немає різниці в структурі, яка відображає пропозиції користувачеві у стилі формату, адже користувач може детальніше вивчити кожен товар чи послугу та зв'язатися з організацією, яка створила конкретну пропозицію.

1.1 Огляд існуючих вебзастосунків

Практично кожен користувач Інтернету стикався з онлайн-сервісами, які допомагають створювати або продавати різні види реклами. Більшість із цих додатків створені задля однієї мети, але використовуються в різних секторах чи місцях. Можна виділити кілька найбільш поширених сфер застосування таких послуг.

1.1.1 OLX

OLX – це платформа для розміщення оголошень. Фабріс Гранд і Александр Оксенфор, двоє французьких підприємців, створили фірму в 2006 році. Корпорація почала діяльність в Індії, яка на той час була дуже перспективним ринком. Протягом 10 років послуга поширювалася на ринки інших країн по всьому світу. OLX набув популярності в Україні після злиття з уже створеною місцевою дошкою онлайн-оголошень Slando у 2015 році [1].

Пропонований веб-додаток надає приватним особам і компаніям платформу для купівлі та продажу продуктів і послуг в Інтернеті.

					ІАЛЦ.467200.002 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

Навіть якщо користувач не має акаунту, він може переглядати пропозиції на домашній сторінці сайту. 1.1 [1].

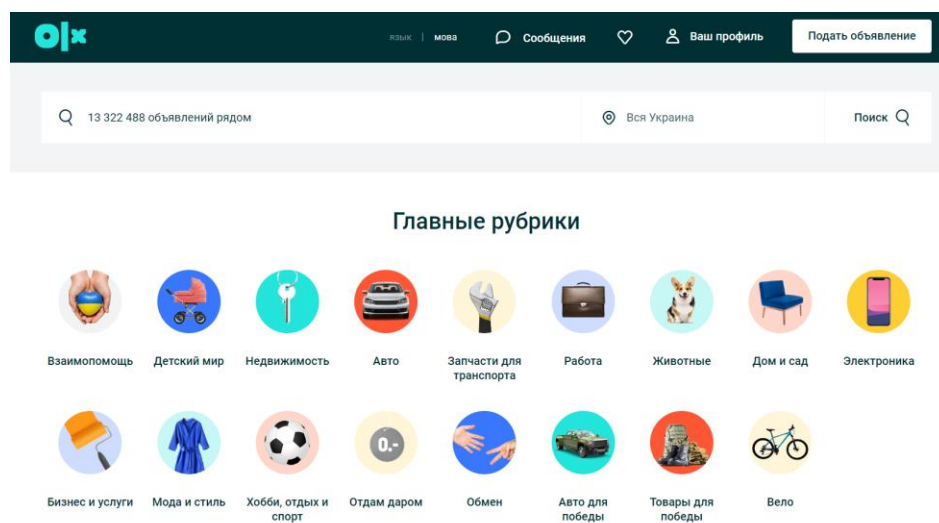


Рисунок 1.1 – Головна сторінка OLX

Неавторизовані користувачі також можуть використовувати можливості пошуку, фільтрації та сортування сторінки.

З точки зору фільтрів, користувач може вибрати товари чи послуги за місцем розташування, порядком ціни або певною категорією. Рис. 1.2 [1].

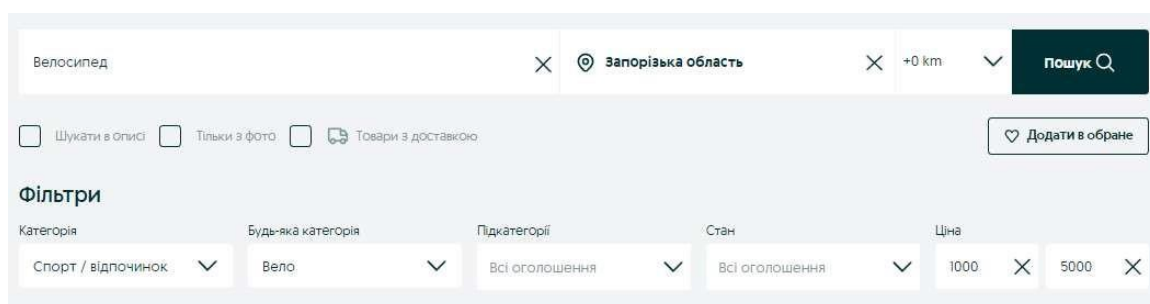


Рисунок 1.2 – Фільтри OLX

Сайти перераховані в списку за замовчуванням від новіших до старіших оголошень. Проте є певні платні можливості, щоб підняти оголошення на вершину дошки.

Сайт також дозволяє особам, які мають акаунт або компаніям створювати публікації своїх послуг чи товарів. Для цього треба заповнити необхідні поля назви товару або послуги, опис і контактну інформацію продавця, а також вибрати категорію товару або послуги у формі подання оголошення рис. 1.3 [1]. Можна завантажити одну або кілька фотографій і розмістити їх як додаткові дані.

Створити оголошення

Опишіть у подробицях

Вкажіть назву*

Наприклад, iPhone 8

Заголовок відіграє важливу роль, не забудьте додати його

Категорія*

Виберіть категорію

Фото

Перше фото буде на обкладинці оголошення. Перетягніть, щоб змінити порядок фото

Додати фото

Опис*

Подумайте, що ви хотіли би дізнатися з оголошення. І додайте це в опис

Напишіть ще 90 символів

0/9000

Ваші контактні дані

Місцезнаходження*

Запоріжжя, Шевченківський

Email-адреса

nsidomon.nsss@gmail.com

Номер телефону

0967253067

Рисунок 1.3 – Сторінка створення оголошень OLX

Особа, яка подала оголошення, може переглядати, редагувати або видаляти фотографії. 1.4 [1]. Також можна відповісти на зауваження або

коментарі щодо продукту чи послуги, залишені іншими користувачами рис.1.5 [1].

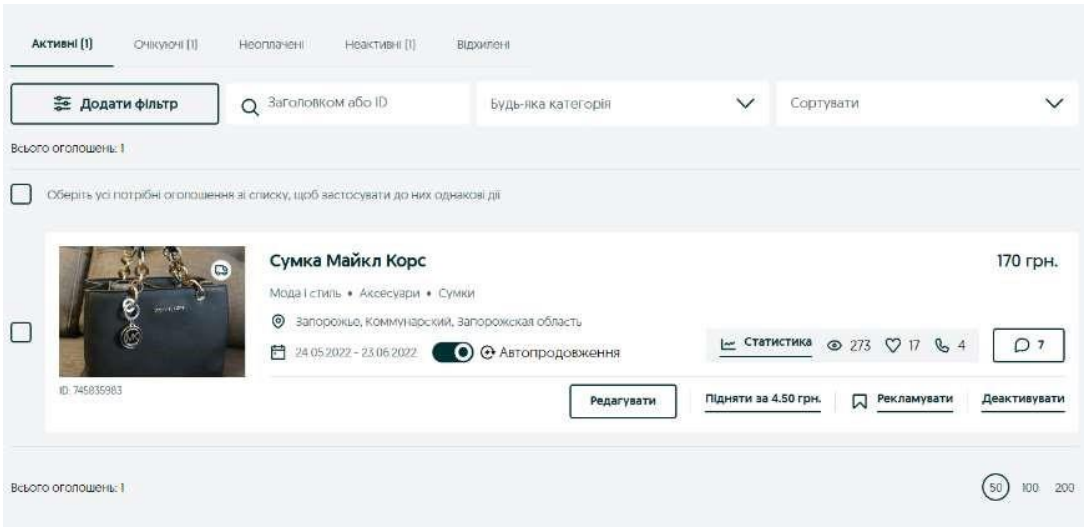


Рисунок 1.4 – Сторінка зі списком оголошень користувача OLX

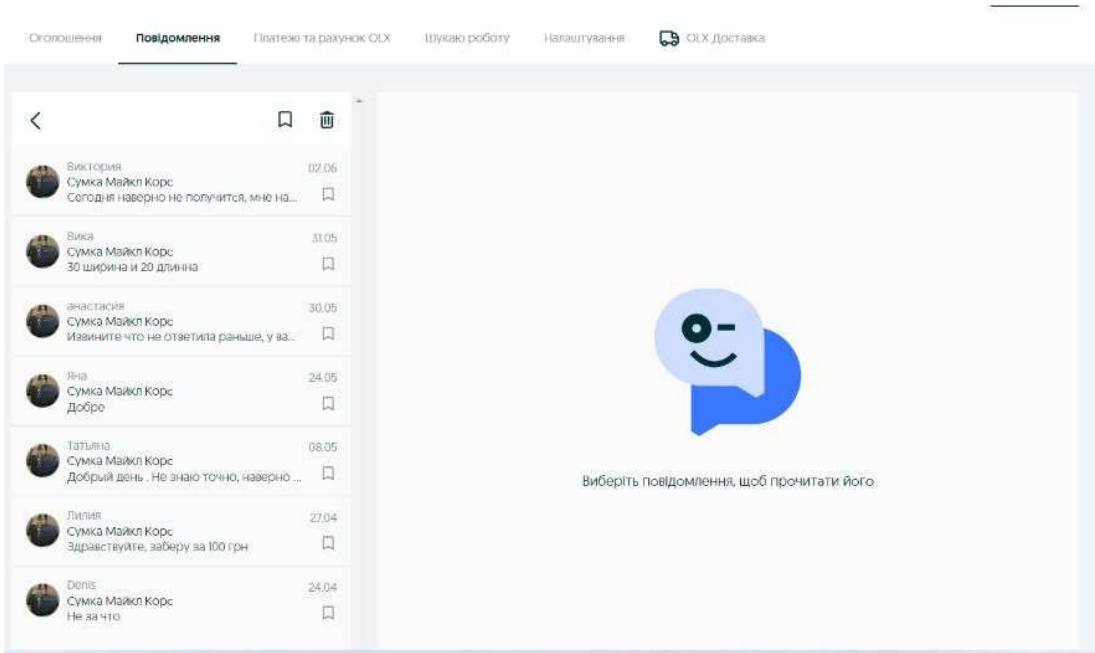


Рисунок 1.5 – Сторінка повідомлень та чатів OLX

1.1.2 Work.ua

Work.ua – це сервіс для пошуку роботи або співробітника. Дніпровська фірма «Реактор» побудувала її в 2006 році. Зараз платформа доступна тільки в Україні і є флагманом серед аналогів. [2].

Функціональні можливості сайту Work.ua розділяють користувачів на дві ролі, які користувач вибирає на головній сторінці, у верхньому правому куті рис. 1.6 [2].

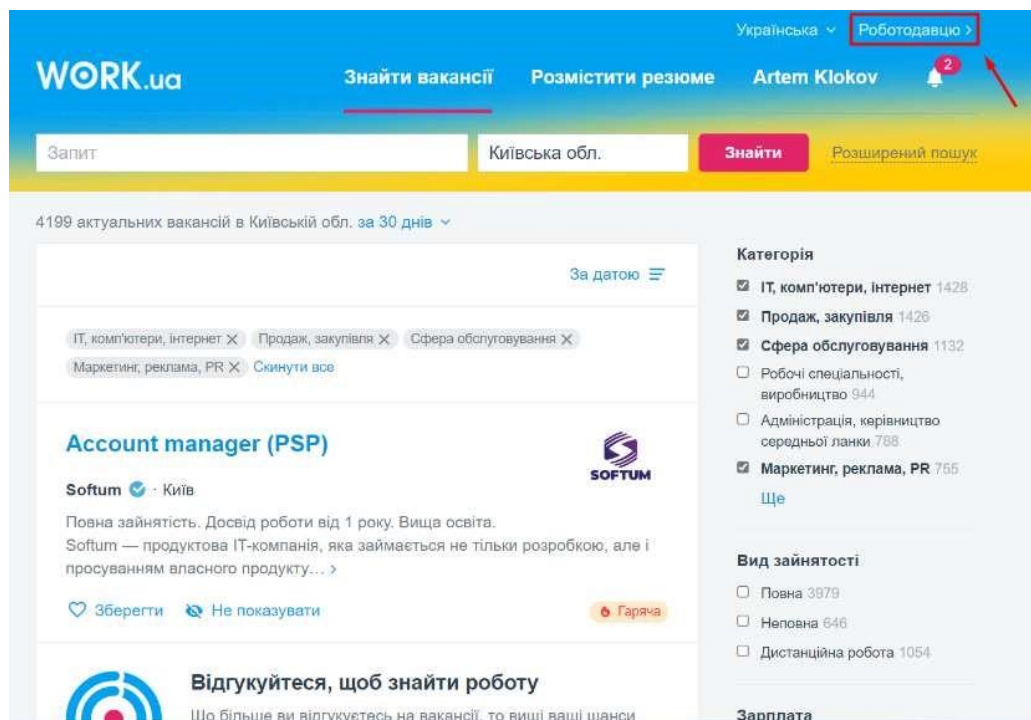


Рисунок 1.6 – Головна сторінка Work.ua

У позиції шукача користувач шукає вакансії, що задовольняють критеріям. Це можна зробити двома способами: моніторинг відкритих вакансій (активний пошук) і відправка резюме через веб-сайт (пасивний пошук).

Що стосується моніторингу вакансій, користувач має доступ до відкритих списків вакансій, фільтрів і пошуку. Пошук здійснюється як за назвою посади або галузі, так і за ключовими словами, пов'язаними з нею. Вакансії можна відфільтрувати за категоріями, професіями, населеними пунктами та конкретними критеріями роботодавця (рис. 1.7 [2]).

Категорія

☒ ІТ, комп'ютери, інтернет 1428
 ☒ Продаж, закупівля 1426
 ☒ Сфера обслуговування 1132
 ☐ Робочі спеціальності, виробництво 944
 ☐ Адміністрація, керівництво середньої ланки 788
 ☒ Маркетинг, реклама, PR 755

Ще

Вид зайнятості

☐ Повна 3979
 ☐ Неповна 646
 ☐ Дистанційна робота 1054

Зарплата

від

будь-яка

до

будь-яка

☐ Не вказана 1626

Підходить

☐ Кандидатам без досвіду 674
 ☐ Кандидатам без резюме 895
 ☐ Студентам 682
 ☐ Людям з інвалідністю 366
 ☐ Ветеранам АТО/ООС 198
 ☐ Пенсіонерам 75

Рисунок 1.7 – Фільтри пошуку Work.ua

За замовчуванням вакансії в списках упорядковано за датою їх публікації (від новіших то старих), однак користувач може сортувати доступні вакансії за рівнем оплати рис. 1.8 [2].

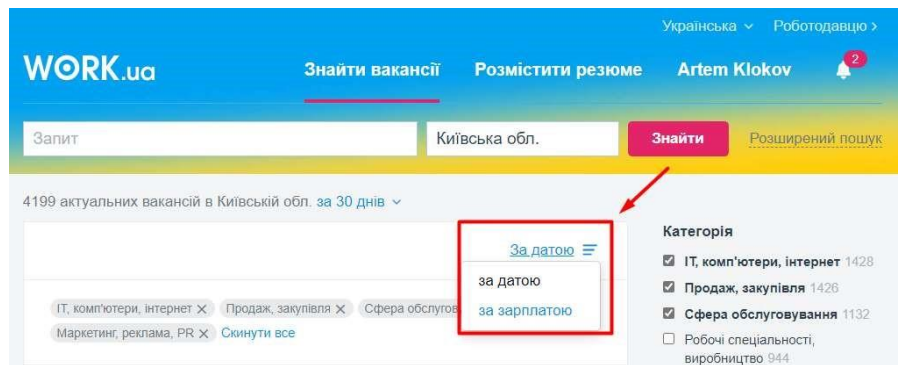


Рисунок 1.8 – Варіанти сортування Work.ua

Сценарій пасивного пошуку передбачає, що шукач надсилає резюме в кілька етапів. По-перше, необхідно заповнити основну інформацію на візитній картці, показаній на рис. 1.9 [2]. Після цього ви можете надати деяку інформацію про себе як про співробітника, якщо хочете. В подальшому, відправляючи резюме на сайт, користувач дає згоду на публікацію цієї інформації серед роботодавців, а також можливість зв'язку з ним для зацікавлених фірм.

Рисунок 1.9 – Сторінка створення резюме Work.ua

Участь роботодавця також охоплює численні обставини. Серед них: - розміщення запиту на пошук працівників на конкретну вакансію; - розгляд резюме, наданих відвідувачами сайту.

Обидва сценарії намагаються знайти можливого працівника, який відповідає кваліфікації.

Моніторинг резюме здійснюється за допомогою інструментів пошуку за галуззю або посадою рис. 1.10 [2].

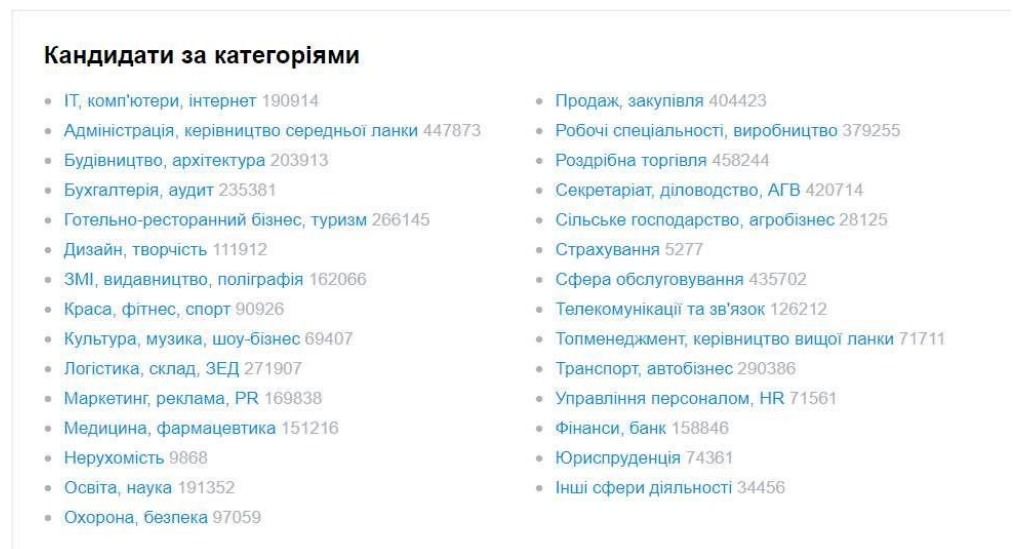


Рисунок 1.10 – Список категорій пошуку резюме Work.ua

Розміщення запиту вимагає окремої реєстрації вашої організації в мережі. Тоді компанія може подати оголошення з основними умовами для шукачів роботи та декларацією про умови їхньої можливої роботи.

1.1.3 AUTO.RIA

AUTO.RIA — український бренд автомобільної інтернет-торгівлі та одне з підприємств RIA.com. З 2014 року сайт є платформою для приватних осіб і компаній для розміщення оголошень про продаж нових і старих автомобілів. В Україні є офіси у Вінниці та Києві. Веб-сайт слугує віртуальною торговою

платформою для автомобілістів, де розміщені публічні оголошення про продаж автомобілів і товарів для них [3].

Без схвалення сервіс дозволяє користувачеві переглядати транзит на головній сторінці. Фільтри для пошуку товару доступні користувачам на головній сторінці. 1.11 [3].

Рисунок 1.11 – Фільтри пошуку AUTO.RIA

Для більш точного пошуку передбачена розширена база фільтрації, яка включає наступні критерії:

- стан автомобіля (б/у, новий, б/у, після ДТП та ін.);
- спосіб транспортування (автомобілі, мотоцикли, вантажівки, спецтехніка і так далі);
- тип кузова (універсал, седан, хетчбек і так далі);
- виробнича нація;
- марка автомобіля (BMW, Audi і так далі);
- модель автомобіля;
- ціна;
- місце, де знаходиться автомобіль;
- технічні параметри (паливо, коробка передач, тип приводу, колір і так далі);
- наявність додаткових опцій.

За винятком товарів, які виставляється за плату, список оголошень на сайті впорядковано від останньої до найновішої.

Після авторизації користувач може розміщувати онлайн-оголошення про купівлю транспортних та автомобільних товарів. У формі подання оголошення мал. 1.13 [3], є обов'язкові поля, де ви повинні ввести основну інформацію про автомобіль, і додаткові поля, де ви можете додати фотографію, номер VIN та іншу інформацію. Щоб розмістити оголошення, необхідно також вказати контактну інформацію продавця.

Послуга є платною для продавців, тому після заповнення всіх необхідних елементів необхідно оплатити публікацію оголошення. Продавці також можуть отримати доступ до додаткових платних функцій для просування реклами інжиру. 1.12 [3].

Особа, яка опублікувала оголошення, може переглядати, змінювати та видаляти раніше створені оголошення. Особисті повідомлення та коментарі поряд з рекламою дозволяють користувачам взаємодіяти один з одним.

The image shows three vertical panels representing different advertising packages on the AUTO.RIA website. Each panel has a title, an icon, a price per day, a list of features, a price per day for the selected option, and a total price at the bottom.

- Простий продаж (Simple sale):** Icon of a blue arrow. Price: 6 грн за день. Features: 1 місяць у пошуку AUTO.RIA, Перевірка по VIN коду, Реклама на ТОП позиціях в пошуку. Selected option: Без реклами. Total price: 200 грн.
- Швидкий продаж (Fast sale):** Icon of a red airplane. Price: 8 грн за день. Features: 1 місяць у пошуку AUTO.RIA, Перевірка по VIN коду, Реклама на ТОП позиціях 4 дні. Selected option: ТОП 1 - № 1 - 10 грн за день. Total price: 240 грн. This panel is highlighted with a red border and a 'Частіше обирають' badge.
- Турбо продаж (Turbo sale):** Icon of a yellow bell. Price: 9 грн за день. Features: 1 місяць у пошуку AUTO.RIA, Перевірка по VIN коду, Реклама на ТОП позиціях 4 дні. Selected option: ТОП 2 - № 1 - 20 грн за день. Total price: 280 грн.

Рисунок 1.12 – Можливості просування оголошення AUTO.RIA

Рисунок 1.14 – Фільтри пошуку Booking.com

Також є інструмент пошуку за типом житла (готелі, квартири, шале, котеджі тощо) рис. 1.15 [4].



Рисунок 1.15 – Рубрикатор Booking.com

Користувач, який шукає житло, може скористатися розширеним пошуком за такими критеріями:

- бюджет (за добу);
- рівень ековідповідальності житла;
- кількість зірок;
- відстань до центру міста;
- правила скасування;
- наявність харчування;
- наявність визначних місць поблизу;
- тип ліжка;
- наявність додаткових послуг.

Посилання у верхньому правому куті відкриває функцію, яка дозволяє зберегти ваше житло для подальшого використання користувачами, які зацікавлені в цій пропозиції рис. 1.16 [4].

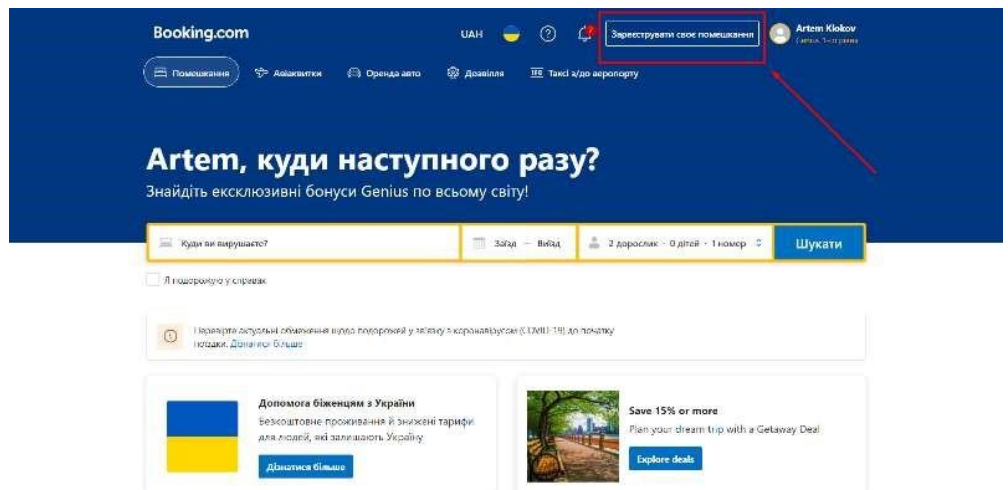


Рисунок 1.16 – Посилання для зміни ролевої моделі користувача
Booking.com

Щоб вийти з цієї функції, ви повинні увійти та вибрати інший вид. Після цього користувач повинен зареєструватися, вказавши адресу, кількість гостей, тип ліжка, кількість ванних кімнат, наявний комфорт, інформацію про сніданок (вид сніданку), наявність паркінгу для відвідувачів, мову персоналу та порядок проживання. Інший користувач може створити профіль хоста після реєстрації, хоча це не обов'язково. Також потрібно надати фото кожного та вартість проживання (за одну ніч). Якщо ви не задекларуєте, що ви зареєстровані як платник податку на товари та послуги, у реєстрації буде відмовлено.

1.1.5 Trello

Trello — програма для управління проектами. Fog Creek Software створила програму. Діяльність організації спрямована на полегшення командної взаємодії під час роботи над проектами різного напрямку, складності та галузі. Однак системи Trello, подібні до дошки оголошень на рис. 1.17 [5], пропонують одну з альтернатив для взаємодії користувачів з онлайн-сервісами.

Trello дозволяє користувачеві створювати та контролювати повідомлення, завдання та терміни. Користувачі, як і в інших програмах, поділяються на два типи. Однак, на користь розглянутих раніше послуг, ці обов’язки не пов’язані з ролями покупця та продавця [5].

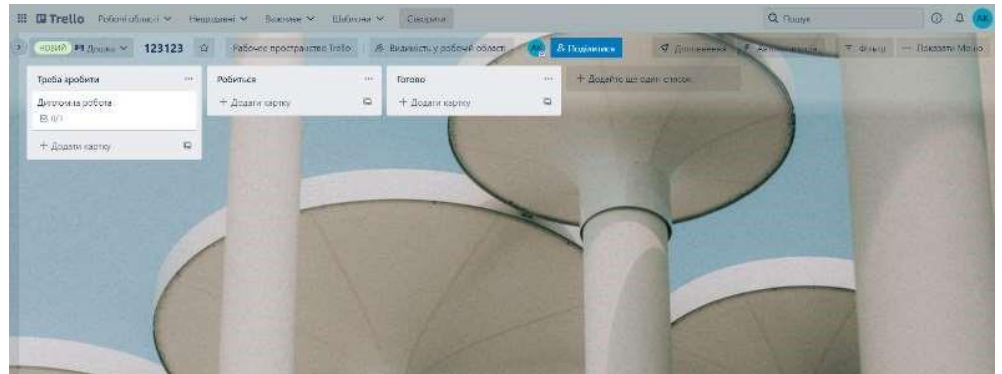


Рисунок 1.17 – Інтерфейс Trello

Перша з ролей — це роль працівника, який прагне виконувати покладені на нього обов’язки. Це завдання, які працівник спостерігає на дошці; він може реагувати на завдання, змінюючи статус його виконання, дозволяючи керівнику проекту постійно відстежувати всі доступні завдання та їх прогрес.

Ще одна функція – це керівник, який розподіляє роботу, визначає терміни виконання, задає інші параметри. Він має можливість призначати завдання окремим користувачам, а також сповіщати свою команду.

Інструменти Trello для створення та націлювання завдань або оголошень будуть актуальними в контексті цієї роботи, що триває. Естетична складова Trello нагадує дошку оголошень у її оригінальному вигляді.

1.2 Запозичення досвіду аналогів при створенні свого проекту

Кожен із розглянутих веб-додатків має певне призначення. Так, Work.ua займається виключно пропозиціями роботи [2], орендою Booking.com,

інформацією та організацією процесів Trello [5] та пропозиціями комерційних автомобілів AUTO.RIA [3]. Серед обраних прикладів OLX виділяється як платформа з найбільшою кількістю можливостей, оскільки її функціонал дозволяє користувачам публікувати або переглядати рекламу не лише про продаж речей, але й про продаж послуг, найму та оренди [1].

Інтерфейс між користувачем і системою мало відрізняється в цих службах з точки зору функціонування. В результаті користувач може зробити рекламу, переглянути свої списки або кожен детально, скористатися пошуком, критеріями пошуку та рубрикою. Через свої характеристики онлайн-додаток Trello виявився найбільш незвичайним у цьому відношенні. По-перше, оголошення в Trello має певну аудиторію та визначає одержувачів певного повідомлення. По-друге, кожному оголошенню (завданню) надається набір обставин, таких як терміни виконання, поточний стан виконання тощо.

Для простоти користування OLX, AUTO.RIA та Work.ua дотримуються філософії односторінкового (рубрика, пошук і список оголошень знаходяться на одній сторінці, і доступ до них доступний без завантаження інших). Перехід на іншу сторінку виконується лише для того, щоб створити власну рекламу або переглянути специфіку реклами в списку.

Слід підкреслити, що процес створення реклами в кожному з розглянутих онлайн-додатків має жорстку структуру, яка диктує вимоги до користувача для написання нової реклами (виняток становить Trello, через різницю в спрямованості та функціональному навантаженні). Крім того, кожна послуга містить етапи машинної та ручної модерації реклами протягом усього процесу створення реклами, щоб уникнути порушень закону, фейків і спаму. Такі запобіжні заходи необхідні для захисту користувачів від шахраїв та інших потенційних небезпек під час використання продукту, а також для уникнення шкоди репутації [10].

					ІАЛЦ.467200.002 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ТЕОРЕТИЧНІ ВІДОМОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЕБ СЕРВІСУ РОЗМІЩЕННЯ ОГОЛОШЕНЬ

2.1 Опис процесу розробки інформаційної системи Веб-сервісу розміщення оголошень

Для виконання роботи прийнято рішення використовувати триланкову архітектуру, яка складається з наступних компонентів: клієнт, сервер і база даних. Схема даної архітектури зображена на рисунку 2.1.

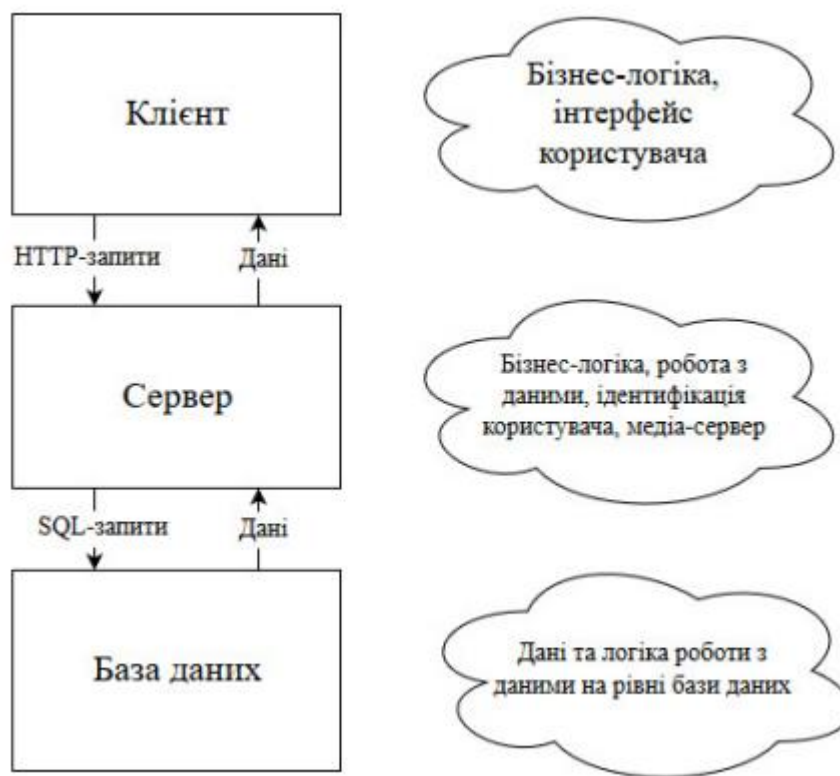


Рисунок 2.1 — Триланкова архітектура програмного комплексу

Розробка інформаційної системи веб-сервісу розміщення оголошень - це складний та багатокроковий процес, який передбачає роботу з великою

кількістю компонентів. У даному описі розглянемо основні етапи процесу розробки веб-сервісу розміщення оголошень.

1. Аналіз вимог

Першим етапом розробки інформаційної системи є аналіз вимог. На цьому етапі необхідно збір вимог від користувачів, розуміння їх потреб, визначення основних функціональних та нефункціональних вимог до системи. Функціональні вимоги включають у себе функції, які система повинна виконувати, наприклад, можливість розміщення та перегляду оголошень. Нефункціональні вимоги включають у себе вимоги до надійності, швидкості роботи, безпеки даних та ін.

2. Проектування

На другому етапі необхідно розробити детальний план системи, визначити архітектуру, вибрати технології, які будуть використовуватися під час розробки, визначити складові системи, створити прототипи інтерфейсу користувача. Для зберігання даних необхідно вибрати підходящу систему управління базами даних. На цьому етапі розроблюються діаграми класів, секвенцій та інші необхідні документи [14].

3. Розробка

Третій етап полягає в написанні програмного коду, розробці інтерфейсу та дизайну. Під час цього етапу використовуються вибрані на попередньому етапі технології та інструменти. Розробка може виконуватися в різних методологіях, наприклад, в залежності від вибраної командою розробників. Наприклад, можна використовувати Agile-методологію, яка полягає у розробці системи у вигляді ітерацій або спринтів. Кожен спринт може мати тривалість від 1 до 4 тижнів, під час якого розробники працюють над конкретними завданнями. Після завершення спринту проходить огляд результатів роботи команди та можливі корективи до планування.

4. Тестування та відладка

					ІАЛЦ.467200.002 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

На цьому етапі виконується перевірка розробленої системи на відповідність вимогам, які були зібрані на попередньому етапі. Тестування включає у себе ручне та автоматизоване тестування, під час якого перевіряються основні функції системи, її безпека та надійність. Після виявлення помилок та недоліків виконується відладка та їх виправлення.

5. Впровадження та підтримка

Після успішного завершення тестування та відладки система готова до впровадження в роботу. На цьому етапі виконується інтеграція системи з іншими системами, які використовуються в організації, а також проводиться навчання користувачів. Після впровадження системи у роботу виконується її підтримка, яка включає у себе виправлення помилок, вдосконалення системи та додавання нових функцій.

У процесі розробки інформаційної системи веб-сервісу розміщення оголошень важливо дотримуватися методологій та процесів, які забезпечують успішне виконання проекту. Команда розробників повинна працювати у злагоді та злагодженості, а також забезпечити взаємодію зі замовником та користувачами, щоб забезпечити досягнення бізнес-цілей проекту [19].

2.2 Опис структури програми

Веб-сервіс для розміщення оголошень є складною інформаційною системою, яка має свою структуру та функціональність. В цілому, структура веб-сервісу складається з таких основних компонентів: клієнтської частини (фронтенду), серверної частини (бекенду) та бази даних.

Клієнтська частина (фронтенд) відповідає за взаємодію з користувачем та відображення інформації на стороні клієнта. Фронтенд має свою структуру та включає такі елементи, як стилі, скрипти та інтерфейс користувача. Для розміщення оголошень веб-сервіс може містити такі елементи фронтенду, як

					ІАЛЦ.467200.002 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

форми введення інформації, список оголошень та фільтри для пошуку оголошень за різними критеріями.

На рисунку 2.2 зображено клієнтську частину популярного веб-сервісу для розміщення оголошень OLX

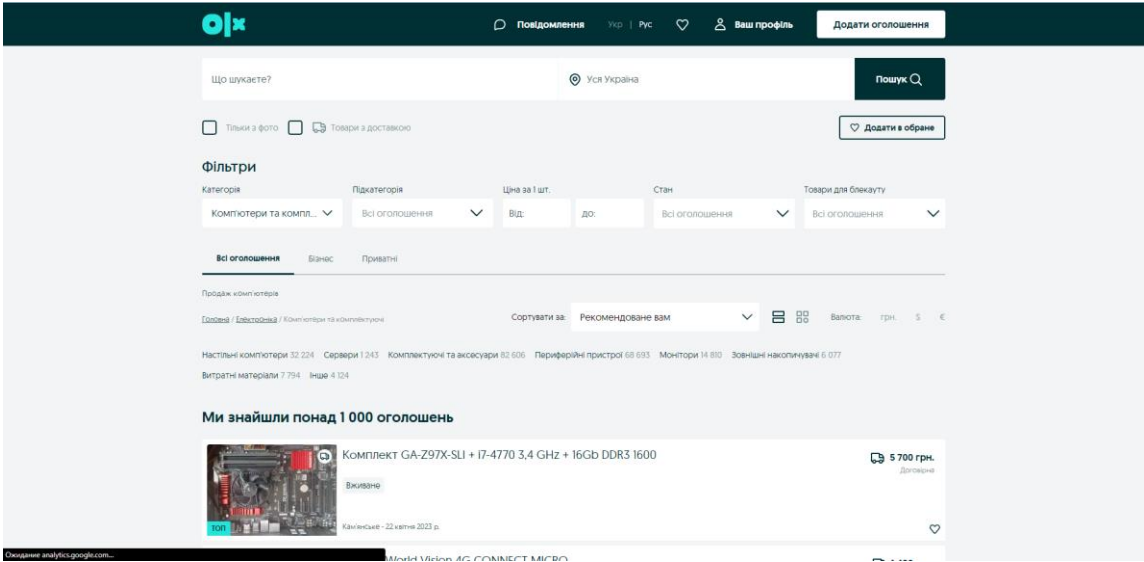


Рисунок 2.2 – Клієнтська частина веб-сервісу OLX

Серверна частина (бекенд) відповідає за обробку запитів користувача та забезпечення роботи бази даних. Бекенд має свою структуру та включає такі елементи, як маршрутизатори запитів, контролери, сервіси та моделі даних. Маршрутизатор відповідає за розподіл запитів до відповідних контролерів, які в свою чергу виконують необхідні дії та повертають відповідь до клієнта. Сервіси містять логіку бізнес-процесів, які виконуються на бекенді, а моделі даних містять структуру та спосіб збереження даних в базі даних [23].

На рисунку 2.3 зображено взаємодію між Front-end та Back-end

бази даних. Кожен компонент має свою структуру та функціональність, яка дозволяє забезпечувати роботу системи. Веб-сервіс може містити різні елементи, які доповнюють його можливості та забезпечують безпеку його роботи. Описана структура є базовою та може бути розширена залежно від потреб веб-сервісу для розміщення оголошень [30].

2.3 Опис технологій для створення веб сервісу

Для розробки веб-сервісу онлайн розміщення оголошень використовуються різні технології, мови програмування, фреймворки та бібліотеки.

1. Java є об'єктно-орієнтованою мовою програмування, що дозволяє програмістам створювати складні програми, використовуючи різні класи та об'єкти. Крім того, Java має вбудовану систему збору сміття, що дозволяє програмістам не турбуватися про вивільнення пам'яті після використання об'єктів.

Java також володіє багатим набором стандартних бібліотек, які дозволяють програмістам взаємодіяти з базами даних, створювати мережеві додатки та обробляти різноманітні формати даних, такі як XML та JSON.

Ще однією з переваг Java є те, що вона є переносимою мовою програмування, що дозволяє запускати програми на різних платформах без необхідності переписувати код. Це робить Java популярним вибором для створення веб-додатків, які повинні працювати на різних операційних системах.

Java також має велику спільноту розробників та підтримується багатьма компаніями, включаючи Oracle та IBM. Це забезпечує стабільність та розвиток мови, а також підтримку програмістів, які працюють з Java.

					ІАЛЦ.467200.002 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

Загалом, Java є потужною та гнучкою мовою програмування, яка є однією з найпопулярніших виборів для створення веб-додатків. Вона дозволяє розробникам створювати складні програми, які можуть масштабуватися та працювати на різних платформах, а також використовувати багатий набір стандартних бібліотек та інструментів [22].

2. SQL (Structured Query Language) є мовою програмування, яка використовується для управління базами даних. SQL дозволяє програмістам створювати, змінювати та видаляти дані з баз даних, виконувати запити на вибірку даних, створювати таблиці та встановлювати зв'язки між ними.

SQL є стандартом, який визначає способи взаємодії з базами даних. Це означає, що бази даних, які використовують SQL, можуть бути реалізовані на різних платформах та системах управління базами даних.

SQL має багато різних команд та функцій, які дозволяють виконувати різноманітні операції з даними, такі як вибірка, вставка, оновлення та видалення. Крім того, SQL дозволяє програмістам створювати складні запити, які містять умови, об'єднання таблиць, підзапити та інші операції [15].

Для веб-розробки, SQL використовується для створення та управління базами даних, які зберігають інформацію про користувачів, продукти, замовлення та інші дані, які використовуються в веб-сервісі. Наприклад, в онлайн сервісі розміщення оголошень, SQL використовується для створення таблиць, що містять інформацію про користувачів та оголошення, а також для вибірки даних під час відображення списку оголошень та пошуку за певними критеріями.

Отже, SQL є незамінною частиною веб-розробки та дозволяє програмістам ефективно управляти даними, що зберігаються в базах даних.

3. Spring Boot є одним з найпопулярніших фреймворків Java для розробки веб-додатків та мікросервісів. Він забезпечує швидкий старт для розробки, що дозволяє програмістам швидко розгорнути та запустити додаток з мінімальними налаштуваннями. Spring Boot базується на фреймворку Spring

					ІАЛЦ.467200.002 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

Framework, інтегруючи усі його основні функції та додаткові бібліотеки. Він дозволяє зосередитись на розробці бізнес-логіки, забезпечуючи готове середовище для підтримки різних інструментів, таких як бази даних, веб-сервери, шаблонізатори та багато іншого.

Spring MVC є фреймворком, який забезпечує реалізацію шаблону проектування Модель-Представлення-Контролер (MVC) для створення веб-додатків. Він дозволяє розділити логіку додатку на три складові: модель, яка представляє бізнес-логіку; представлення, яке відповідає за відображення даних користувачам; та контролер, який обробляє запити користувачів та відповідає на них. Spring MVC дозволяє програмістам розробляти веб-додатки, що можуть бути масштабовані та підтримувані з легкістю.

Spring Security є фреймворком для забезпечення безпеки веб-додатків. Він дозволяє програмістам захистити веб-додатки від атак, таких як перехоплення даних, внесення змін у дані та інші. Spring Security забезпечує відправку запитів на автентифікацію та авторизацію, забезпечуючи, що тільки авторизовані користувачі мають доступ до ресурсів. Крім того, Spring Security надає можливість налаштувати різні види аутентифікації та авторизації, включаючи базову автентифікацію, форму логіну, JWT (JSON Web Token) автентифікацію та інші. Фреймворк також дозволяє налаштовувати рівні доступу до ресурсів та забезпечує захист від CSRF (Cross-Site Request Forgery) атак [21].

Разом з Spring Boot та Spring MVC, Spring Security створює повноцінну стек технологій для створення веб-додатків Java. Використання цих фреймворків дозволяє програмістам створювати високоякісні веб-додатки, які можуть бути масштабовані та підтримувані з легкістю. Spring Boot, Spring MVC та Spring Security є популярними серед програмістів Java завдяки своїй простоті використання та широкому спектру функціональності.

4. HTML є основною мовою, яка використовується для створення структури веб-сторінок та додавання вмісту до них. Вона дозволяє відображати різні типи даних, такі як текст, зображення, відео, аудіо та інші. HTML

					ІАЛЦ.467200.002 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

використовує теги для відображення вмісту та його організації на веб-сторінках.

CSS використовується для стилізації веб-сторінок та надання їм зовнішнього вигляду. Вона дозволяє програмістам відділяти вміст веб-сторінки від її візуального представлення. CSS дозволяє задавати кольори, шрифти, розміри, відступи, рамки та інші властивості елементів веб-сторінки [23].

JavaScript є мовою програмування, яка використовується для створення динамічних та інтерактивних елементів на веб-сторінках. Вона дозволяє програмістам додавати взаємодію користувача з елементами веб-сторінки, наприклад, обробляти події, які виникають при кліку на кнопку або руху курсора миші, виконувати асинхронні запити до сервера та змінювати вміст веб-сторінки без перезавантаження її.

Разом, HTML, CSS та JavaScript допомагають програмістам створювати привабливі, інтерактивні та користувачем дружні веб-додатки.

5. Bootstrap був розроблений компанією Twitter з метою полегшення розробки веб-інтерфейсів та забезпечення їхньої адаптивності на різних пристроях, включаючи мобільні пристрої. Його використання дозволяє програмістам швидко та ефективно створювати інтерфейси з використанням заздалегідь визначених класів та компонентів.

Bootstrap містить багато вбудованих CSS та JavaScript компонентів, таких як кнопки, форми, меню навігації, каруселі, модальні вікна та багато інших. Компоненти можна легко налаштувати та змінювати за допомогою відповідних класів та параметрів [11].

Один з головних переваг Bootstrap полягає в його адаптивності. За допомогою відповідних класів, можна забезпечити оптимальний вигляд веб-інтерфейсу на різних розмірах екрану, що дозволяє користувачам насолоджуватися зручним та привабливим виглядом інтерфейсу на будь-яких пристроях.

Крім того, Bootstrap має велику і активну спільноту розробників, що забезпечує постійну підтримку та розвиток фреймворку. Завдяки цьому, Bootstrap став популярним фреймворком для розробки веб-додатків та сайтів, що забезпечує ефективність та якість розробки.

6. Hibernate - це фреймворк для роботи з базами даних, що дозволяє спростити процес взаємодії з базою даних та підвищити продуктивність.

Spring Data JPA - це фреймворк для спрощення роботи з базами даних, що дозволяє працювати з базою даних на рівні об'єктів та зменшує кількість коду, необхідного для виконання операцій.

7. MySQL – це система управління реляційними базами даних, яка використовується для зберігання інформації.

8. Apache Freemarker є відкритим програмним забезпеченням, написаним на мові Java, і використовується для генерації вихідного коду веб-сторінок. Він дозволяє розробникам створювати шаблони веб-сторінок, які потім можна заповнювати динамічним контентом з даних, що надходять з бази даних або інших джерел даних.

Одна з переваг Freemarker полягає в тому, що він дозволяє відокремлювати логіку від представлення, що сприяє підтримці чистого коду і спрощує розробку та підтримку веб-додатків. Крім того, Freemarker має багато корисних функцій, таких як умовні оператори, цикли, фільтри та інші, які дозволяють легко створювати складні шаблони [17].

Використання Apache Freemarker є особливо корисним для веб-розробників, які працюють з фреймворками, такими як Spring або Struts, оскільки вони можуть використовувати Freemarker для генерації HTML-коду відповідей на запити користувачів.

9. Lombok - це бібліотека для спрощення написання коду на Java, яка дозволяє програмістам зосередитися на бізнес-логіці додатка, а не на написанні повторюваного та стандартного коду. Завдяки Lombok можна автоматично генерувати гетери, сеттери, конструктори, методи toString, equals та hashCode,

що значно зменшує кількість коду, який потрібно писати. Крім того, Lombok дозволяє створювати анотації, які автоматично генерують код для методів, що зменшує час, необхідний для написання коду та покращує читабельність програмного коду.

Lombok також дозволяє використовувати лямбда-вирази, забезпечуючи зручний та простий спосіб використання функцій, а також підтримується рядом інтегрованих середовищ розробки, таких як Eclipse та IntelliJ IDEA. Бібліотека допомагає зменшити кількість написаного коду та робить розробку на Java більш зручною та продуктивною.

Всі ці технології використовуються для розробки веб-сервісу онлайн розміщення оголошень. Spring Boot, Spring MVC та Spring Security забезпечують основу для розробки серверної частини програми, а Hibernate та Spring Data JPA дозволяють взаємодіяти з базою даних MySQL на рівні об'єктів, що спрощує процес роботи з базою даних [9].

HTML, CSS, JavaScript і Bootstrap використовуються для створення інтерфейсу користувача. Apache Freemarker використовується для створення шаблонів веб-сторінок, які відображатимуться користувачеві. Lombok використовується для спрощення написання повторюваного коду Java.

Результатом розробки програми буде веб-сервіс, який дозволить користувачам розміщувати оголошення та переглядати вже розміщені оголошення. Інтерфейс користувача буде створено з використанням HTML, CSS, JavaScript і Bootstrap, а серверна частина програми буде розроблена з використанням Spring Boot, Spring MVC, Spring Security, Hibernate і Spring Data JPA. В результаті, користувачі зможуть взаємодіяти з додатком, розміщувати та переглядати оголошення, а власник програми зможе керувати базою даних та забезпечувати безпеку програми [7].

					ІАЛЦ.467200.002 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РОЗРОБКА СЕРВІСУ ДЛЯ ПОКУПКИ АБО ПРОДАЖУ ТОВАРІВ

3.1 Архітектура застосунку

Розроблений сервіс призначений для купівлі або продажу різних предметів. Для запуску сервісу використовувався фреймворк spring-boot, який відповідає за серверну частину. Він був налаштований за допомогою сторінок Apache freemarker ftlh, які відповідають за інтерфейс. Конфігурація сервісу була створена в Apache Maven, фреймворку для автоматизації збирання проектів на основі опису їх структури у файлах POM, яка є підмножиною XML, до якої додано фреймворк і бібліотеки. Структура папок проекту є похідною від стандартного веб-сервісу на springboot. Код розробленого застосунку зазначено в додатку А.

В таблиці 3.1 представлені порти, які використовуються для сервісу.

Таблиця 3.1

Порти для моніторингу SMTP-пакетів

HTTP-порт	Service
8080	Spring Backend-service

На рисунку 3.1 представлено програмний код для конфігурування бази даних MySQL, який знаходиться у файлі application properties

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/eshop?useSSL=false&serverTimezone=UTC
2 spring.datasource.username=root
3 spring.datasource.password=root
4
5 spring.jpa.hibernate.ddl-auto=update
6 spring.jpa.show-sql=true
7 spring.freemarker.expose-request-attributes=true
```

Рис. 3.1. – Програмний код для код для конфігурування бази даних.

На рисунку 3.2 продемонстрована база даних до підключення сервісу.

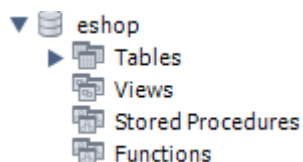


Рис. 3.2. – База даних до підключення сервісу.

На рисунку 3.3 база даних вже після підключення сервісу

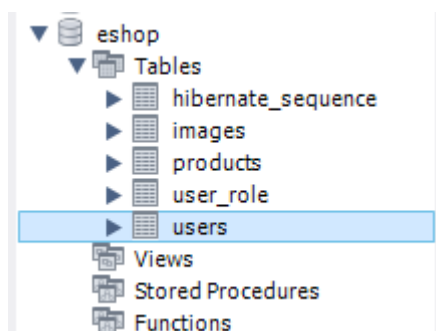


Рис. 3.3. – База даних після підключення сервісу.

Архітектура папок проекту продемонстрована на рисунку 3.4. Вона відповідає стандартам springboot та maven.

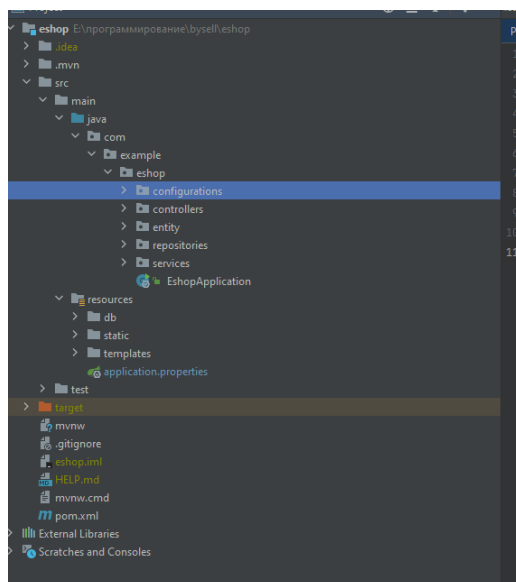


Рис. 3.4. – Архітектура папок сервісу.

3.2 Огляд використаних технологій

Java є основною мовою програмування, яка використовується в розробці. Для бекенд-фреймворку використовувалися Springboot і Apache Maven. Для показу сторінок або для інтерфейсу було вибрано шаблон Apache Freemarker..

На рисунках 3.5, 3.6 показано Pom.xml файл, який використовується для правильної конфігурації сервісу

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3       xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.6.7</version>
9     <relativePath/> <!-- lookup parent from repository -->
10  </parent>
11  <groupId>com.example</groupId>
12  <artifactId>eshop</artifactId>
13  <version>0.0.1-SNAPSHOT</version>
14  <name>eshop</name>
15  <description>eshop</description>
16  <properties>
17    <java.version>17</java.version>
18  </properties>
19  <dependencies>
20    <dependency>
21      <groupId>org.springframework.boot</groupId>
22      <artifactId>spring-boot-starter-data-jpa</artifactId>
23    </dependency>
24    <dependency>
25      <groupId>org.springframework.boot</groupId>
26      <artifactId>spring-boot-starter-freemarker</artifactId>
27    </dependency>
28    <dependency>
29      <groupId>org.springframework.boot</groupId>
30      <artifactId>spring-boot-starter-mail</artifactId>
31    </dependency>
32    <dependency>
33      <groupId>org.springframework.boot</groupId>
34      <artifactId>spring-boot-starter-security</artifactId>
35    </dependency>
36    <dependency>
37      <groupId>org.springframework.boot</groupId>
38      <artifactId>spring-boot-starter-thymeleaf</artifactId>
39    </dependency>
40    <dependency>
41      <groupId>org.springframework.boot</groupId>
42      <artifactId>spring-boot-starter-web</artifactId>
43    </dependency>
44    <dependency>
45      <groupId>org.thymeleaf.extras</groupId>
46      <artifactId>thymeleaf-extras-springsecurity5</artifactId>
47    </dependency>
48    <dependency>
49      <groupId>mysql</groupId>
50      <artifactId>mysql-connector-java</artifactId>
51      <scope>runtime</scope>
52    </dependency>
53  </dependencies>
```

Рис. 3.5. – Pom.xml файл для конфігурації

```

54      </dependency>
55      <dependency>
56          <groupId>org.springframework.boot</groupId>
57          <artifactId>spring-boot-starter-test</artifactId>
58          <scope>test</scope>
59      </dependency>
60      <dependency>
61          <groupId>org.springframework.security</groupId>
62          <artifactId>spring-security-test</artifactId>
63          <scope>test</scope>
64      </dependency>
65
66      <dependency>
67          <groupId>org.projectlombok</groupId>
68          <artifactId>lombok</artifactId>
69      </dependency>
70
71  </dependencies>
72
73  <build>
74      <plugins>
75          <plugin>
76              <groupId>org.springframework.boot</groupId>
77              <artifactId>spring-boot-maven-plugin</artifactId>
78          </plugin>
79      </plugins>
80  </build>
81
82 </project>
83

```

Рис. 3.6. – Pom.xml файл для конфігурації

В цьому файлі ключові ролі відіграють dependency, наприклад dependency org.springframework.boot на рисунку 3.7. Ця залежність підключає framework springboot на якому тримається сервіс.

```

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

```

Рис. 3.7. Dependency для конфігурації сервісу

Бібліотека Lombok була використана для швидкого створення сутностей та їх відповідальних потреб; це допомагає вам швидко згенерувати належне поняття сутності, заощаджуючи час на створення коду. На рисунку 3.8 , 3.9 продемонстровано використання бібліотеки Lombok.

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

Рис. 3.8. – Підключення бібліотеки Lombok

```
@Data
@AllArgsConstructor
@NoArgsConstructor
```

Рис. 3.9.Застосування анотації Lombok

Для безпеки сервісу було додано Spring Security framework, який дозволяє зробити реєстрацію та авторизацію користувача. Його конфігурація продемонстрована на рисунку 3.10

```
@EnableWebSecurity
@RequiredArgsConstructor
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    1 usage
    private final CustomUserDetailsService customUserDetailsService;

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests() .ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
            .antMatchers( ...antPatterns: "/", "/goods/**", "/images/**", "/registration", "/user/**", "/static/**", "/about-us")
            .permitAll() .ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
            .anyRequest().authenticated()
            .and() HttpSecurity
            .formLogin() .FormLoginConfigurer<HttpSecurity>
            .loginPage("/login")
            .permitAll()
            .and() HttpSecurity
            .logout() LogoutConfigurer<HttpSecurity>
            .permitAll();
    }

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.userDetailsService(customUserDetailsService)
            .passwordEncoder(passwordEncoder());
    }

    1 usage
    @Bean
    public PasswordEncoder passwordEncoder() { return new BCryptPasswordEncoder( strength: 8); }
```

Рис. 3.10. Security config для сервісу

3.3 Тестування розробленого застосунку

Щоб увімкнути web-сервіс треба підключити його до бази даних. Для цього треба ввести дані про MySQL сервер у файл application.properties. На рисунку 3.11 показано поля, які відповідають за підключення

```
spring.datasource.url=jdbc:mysql://localhost:3306/eshop?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=root
```

Рис. 3.11. – Поля для підключення до бази даних.

Якщо ввести ці дані помилково, або не ввімкнути MySQL сервер то буде помилка, яка вказана на рисунку 3.12.

```
java.sql.SQLNonTransientConnectionException Create breakpoint : Public Key Retrieval is not allowed
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLException.java:110) ~[mysql-connector-java-8.0.28.jar:8.0.28]
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122) ~[mysql-connector-java-8.0.28.jar:8.0.28]
at com.mysql.cj.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:829) ~[mysql-connector-java-8.0.28.jar:8.0.28]
at com.mysql.cj.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:449) ~[mysql-connector-java-8.0.28.jar:8.0.28]
at com.mysql.cj.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:242) ~[mysql-connector-java-8.0.28.jar:8.0.28]
at com.mysql.cj.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:198) ~[mysql-connector-java-8.0.28.jar:8.0.28]
at com.zaxxer.hikari.util.DriverDataSource.getConnection(DriverDataSource.java:138) ~[HikariCP-4.0.3.jar:na]
at com.zaxxer.hikari.pool.PoolBase.newConnection(PoolBase.java:364) ~[HikariCP-4.0.3.jar:na]
at com.zaxxer.hikari.pool.PoolBase.newPoolEntry(PoolBase.java:206) ~[HikariCP-4.0.3.jar:na]
at com.zaxxer.hikari.pool.HikariPool.createPoolEntry(HikariPool.java:476) ~[HikariCP-4.0.3.jar:na]
at com.zaxxer.hikari.pool.HikariPool.checkFailFast(HikariPool.java:561) ~[HikariCP-4.0.3.jar:na]
at com.zaxxer.hikari.pool.HikariPool.<init>(HikariPool.java:115) ~[HikariCP-4.0.3.jar:na]
```

Рис. 3.12. – Помилка при підключенні до бази даних

Після запуску сервісу на екрані буде головна сторінка сайту, на якій розташовані всі товари, які є у наявності. Щоб відсортувати ці товари було додано спеціальний фільтр товарів.

					ІАЛЦ.467200.002 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

На рисунку 3.13 представлений результат пошуку товарів завдяки фільтру за містом

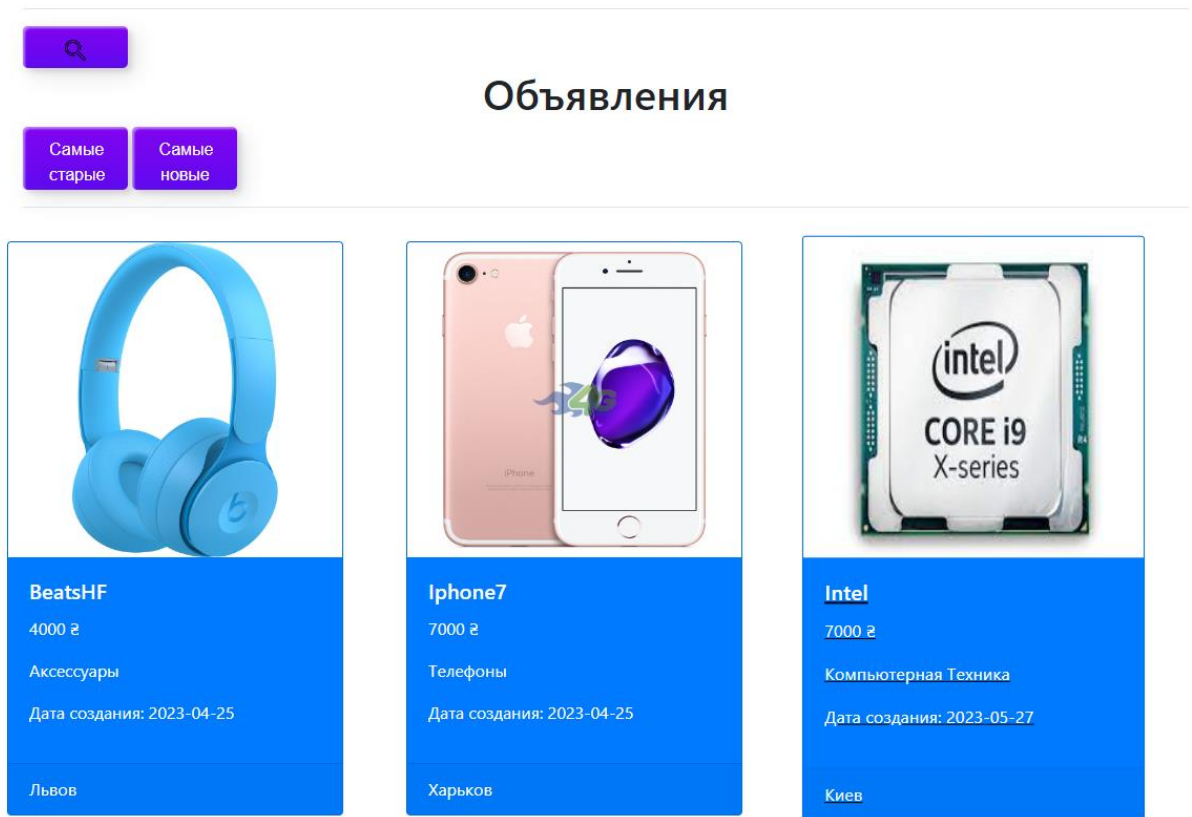


Рис. 3.13. – Сторінка товарів після застосування фільтра пошуку за містом

Також є пошук за назвою товару. Не треба завжди вводити назву для пошуку повністю, реалізовано частковий фільтр за назвою на рисунку 3.14, 3.15.

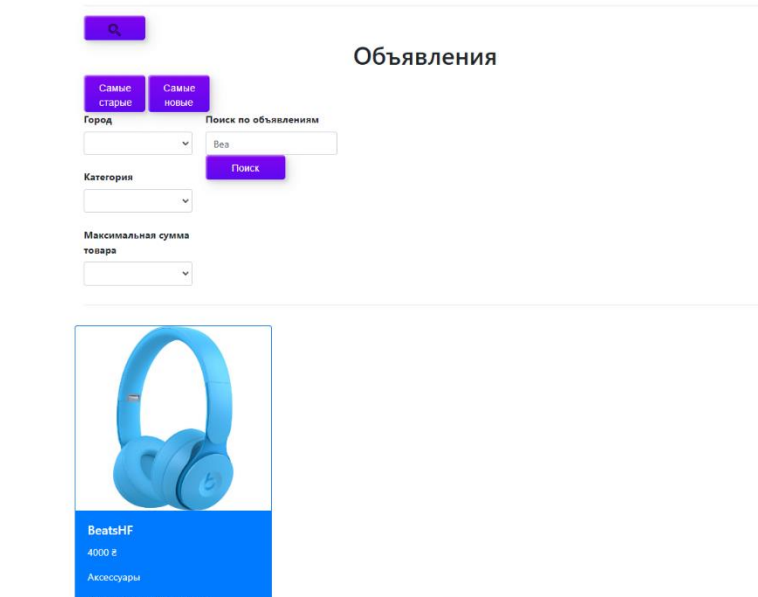


Рис. 3.14. – Результат пошуку за назвою

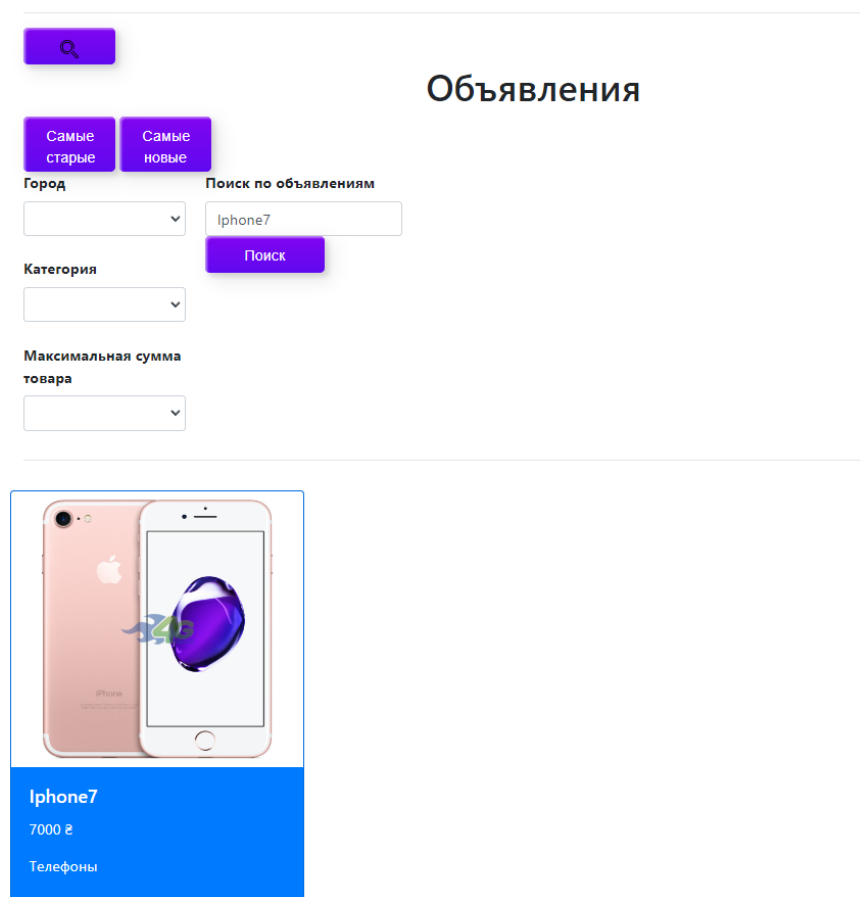


Рис. 3.15. – Результат пошуку за назвою

Процес реєстрації повинен пройти кожен, хто хоче продавати товар. Для Того, щоб зареєструватися треба пройти по адресу /registration та заповнити форму реєстрації. Після проходження валідації, та заповнювання всіх полів дані заносяться в базу даних завдяки User service та методу createUser на рисунку 3.16

```
public boolean createUser(User user) {
    String email = user.getEmail();
    if (userRepository.findByEmail(email) != null) return false;
    user.setActive(true);
    user.setPassword(passwordEncoder.encode(user.getPassword()));
    user.setPassword(user.getPassword());
    user.getRoles().add(Role.ROLE_ADMIN);
    log.info("Saving new User with email: {}", email);
    userRepository.save(user);
    return true;
}
```

Рис. 3.16-Метод createUser

На рисунку 3.17 представлена валідація для полів реєстрації. Якщо валідація не буде пройдена, то новий користувач не додається в базу даних та не зможе авторизуватися на сайті.

Регистрация

Имя пользователя

Электронная почта

Номер телефона

 Адрес электронной почты должен содержать символ "@". В адресе "user" отсутствует символ "@".

Рис. 3.17. – Валідація полів при реєстрації.

Розглянемо процес реєстрації користувача, на рисунках 3.18,3.19, 3.20 буде продемонстрована функція реєстрація користувача.

Регистрация

Имя пользователя

use

Электронная почта

Номер телефона

Пароль

Зарегистрироваться

Рис. 3.18. – Форма реєстрації на сайті.

Регистрация

Имя пользователя

user

Электронная почта

user@gmail.com

Номер телефона

666246755

Пароль

....

Зарегистрироваться

Рис. 3.19. – Заповнення форми реєстрації на сайті.

Result Grid								
Filter Rows:								
	id	active	date_of_created	email	name	password	phone_number	image_id
	1	1	2023-05-08 14:02:43.795329	admin@gmail.com	admin	\$2a\$08\$WIA.jkZ4bnvYwyfKRaiMwOWrY2J5Zzr...	-	NULL
▶	6	1	2023-05-08 14:20:11.313727	user@gmail.com	user	\$2a\$08\$tLbre4BFByOdBO3FJgR.Khlg5EGR5y...	0660871999	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 3.20. – Результат заповнення форми реєстрації на сайті.

Процес додавання товару на сайт є найголовнішим процесом, розглянемо його на рисунках 3.21, 3.22, 3.23

Мои товары

Добавить новый товар

Название товара

Цена товара

₴

Город

Город не выбран

▼

Категория товара

Категория не выбрана

▼

Описание товара

Выберите файл

Файл не выбран

Первая фотография к товару

Выберите файл

Файл не выбран

Вторая фотография к товару

Выберите файл

Файл не выбран

Третья фотография к товару

Добавить

Рис. 3.21. Форма додавання товарів на сайт

Мои товары

Добавить новый товар

Название товара

BeatsHF

Цена товара

4000

₴

Город

Львов

▼

Категория товара

Аксессуары

▼

Описание товара

Неплохие наушники

Выберите файл

IMG_0027.PNG

Первая фотография к товару

Выберите файл

Файл не выбран

Вторая фотография к товару

Выберите файл

Файл не выбран

Третья фотография к товару

Добавить

Рис. 3.22. Заповнена форма додавання товарів на сайт

					ІАЛЦ.467200.002 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

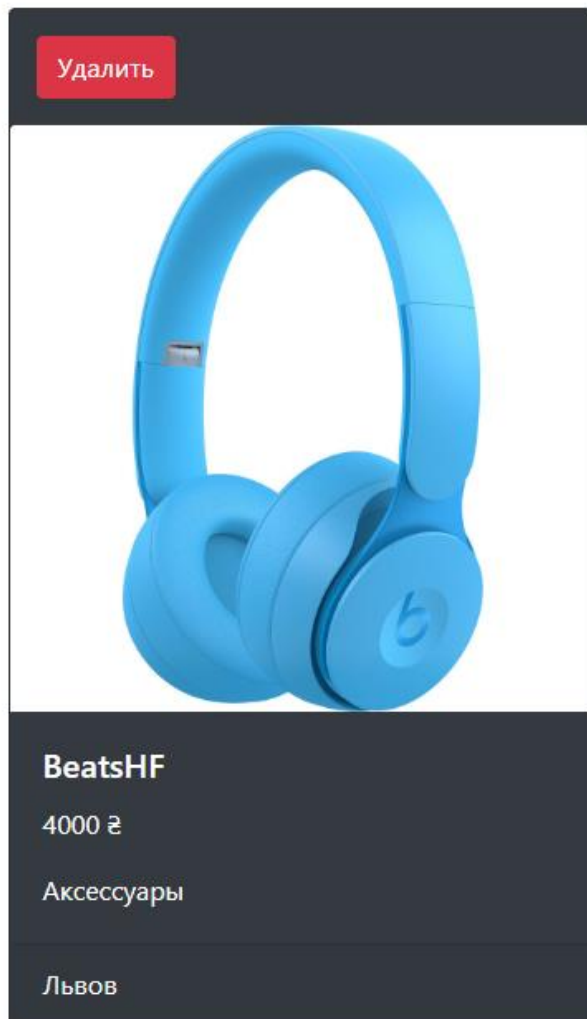


Рис. 3.23. Результат додавання товару

Архітектура та розробка веб-сервісу була представлена в цьому розділі, як і певні тести додатків. Сервіс написаний на Java і побудований на фреймворку springboot і Apache Maven. Для показу веб-сторінок використовувався шаблон freemarker. Коли з'єднання з базою даних не вдається, проект використовує спеціальний файл конфігурації, який створює помилку. Також був включений фільтр, щоб дозволити користувачам переміщатися між різними товарами.

ВИСНОВКИ

У цій кваліфікаційній роботі був проведений аналіз предметної області, пов'язаної з веб-сервісами онлайн оголошень. Були розглянуті існуючі веб-застосунки, такі як OLX, Work.ua, AUTO.RIA, Booking та Trello, що надають користувачам можливість розміщувати оголошення у різних сферах.

Запозичення досвіду аналогів при створенні свого проекту є важливим етапом розробки веб-сервісу. Це дозволяє використати кращі практики та уникнути можливих помилок. В даній роботі проведено аналіз інших аналогічних веб-сервісів, що дало змогу здобути цінний досвід та врахувати його при розробці проекту.

Теоретичні відомості про інформаційну систему веб-сервісу розміщення оголошень були розглянуті в розділі 2. Описано процес розробки інформаційної системи, структуру програми та використані технології. Це надало зрозуміння процесу розробки веб-сервісу та його основних складових.

У розділі 3 було проведено проектування та розробку сервісу для покупки або продажу товарів. Була розглянута архітектура застосунку та оглянуті використані технології. Також проведено тестування розробленого застосунку для перевірки його функціональності та коректності роботи.

Загальним результатом дипломної роботи є розроблений веб-сервіс онлайн оголошень, який надає можливість користувачам розміщувати та переглядати оголошення у зручний спосіб. Робота виконана згідно поставлених цілей та завдань.

Дипломна робота може бути корисною для подальшого розвитку веб-сервісів онлайн оголошень та дослідження в сфері інформаційних технологій.

					ІАЛЦ.467200.002 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ігнатенко О. В. "Огляд існуючих вебзастосунків для розміщення оголошень".
2. Петренко І. М. "Роль OLX в українському ринку онлайн оголошень".
3. Коваленко В. С. "Розвиток та особливості функціонування Work.ua".
4. Литвиненко І. Г. "Аналіз інформаційних систем AUTO.RIA для автомобільного ринку".
5. Мартиненко Н. О. "Роль Booking.com у сфері онлайн-бронювання готелів".
6. Лещенко О. В. "Використання Trello для керування проектами в онлайн середовищі".
7. Куліковська І. П. "Запозичення досвіду аналогів при створенні власного проекту".
8. Сидоренко В. М. "Опис процесу розробки інформаційної системи Веб-сервісу розміщення оголошень".
9. Кравченко О. В. "Опис структури програми веб-сервісу розміщення оголошень".
10. Грищенко І. І. "Опис технологій для створення веб-сервісу розміщення оголошень".
11. Коваленко А. С. "Архітектура застосунку веб-сервісу для покупки або продажу товарів".
12. Левченко В. Г. "Огляд використаних технологій веб-сервісу розміщення оголошень".
13. Павленко М. В. "Тестування розробленого веб-сервісу для покупки або продажу товарів".

					ІАЛЦ.467200.002 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

14. Козлова О. С. "Тенденції розвитку веб-сервісів розміщення оголошень".
15. Семенова І. П. "Аналіз ефективності веб-сервісів розміщення оголошень".
16. Гончаренко Т. В. "Інтеграція зовнішніх сервісів веб-сервісу розміщення оголошень".
17. Клименко Д. І. "Безпека та захист даних веб-сервісу розміщення оголошень".
18. Борисенко В. В. "Використання машинного навчання в веб-сервісах розміщення оголошень".
19. Пономаренко Н. В. "Інтерфейс та дизайн веб-сервісу розміщення оголошень".
20. Мельник О. М. "Аналіз ринку онлайн оголошень в Україні".
21. Smith, J. "Overview of Existing Web Applications for Classified Ads".
22. Johnson, A. "The Role of OLX in the Online Classifieds Market".
23. Thompson, R. "Development and Features of Work.ua".
24. Brown, M. "Analysis of Information Systems in the Automotive Classifieds Market".
25. Wilson, K. "The Impact of Booking.com in the Online Hotel Booking Industry".
26. Davis, S. "Utilizing Trello for Project Management in an Online Environment".
27. Roberts, L. "Learning from Analogues in Creating Your Own Project".
28. Johnson, M. "Description of the Development Process of the Information System for Online Classifieds Web Service".
29. Adams, E. "Description of the Program Structure for a Web Classifieds Service".
30. Thompson, S. "Description of Technologies for Creating a Web Classifieds Service

					ІАЛЦ.467200.002 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК 1

Лістінг

Лістінг коду розробленого застосунку

ІАЛЦ.467200.003 Д1

Аркушів 37

Київ 2023 р

ProductRepository.java

```
package com.example.eshop.repositories;

import com.example.eshop.entity.Product;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

import javax.xml.crypto.Data;
import java.time.LocalDate;
import java.util.List;

public interface ProductRepository extends JpaRepository<Product, Long> {

    List<Product> findAllByTitleAndCity(String title, String city);
    List<Product> findAllByOrderByDateOfCreatedDesc();
    List<Product> findAllByOrderByDateOfCreatedAsc();
}
```

ProductService.java

```
package com.example.eshop.services;

import com.example.eshop.entity.Image;
import com.example.eshop.entity.Product;
import com.example.eshop.entity.User;
import com.example.eshop.repositories.ProductRepository;
```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірив	Щур В. Ю..						48	86	
						НТУУ “КПІ” ФІОТ ІОЗ-91			
Н. Контр.	Виноградов Ю.М.								
Затвердив									

```

import com.example.eshop.repositories.UserRepository;

import lombok.extern.slf4j.Slf4j;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;


import java.io.IOException;
import java.security.Principal;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;
import java.util.Optional;

```

```
@Service
```

```
@Slf4j
```

```
public class ProductService {
```

```
    @Autowired
```

```
    private ProductRepository productRepository;
```

```
    @Autowired
```

```
    private UserRepository userRepository;
```

```
    //Поиск по названию продукта
```

```
    public List<Product> findAllByTitleStartsWith(String title) {
```

```
        List<Product> allProducts = productRepository.findAll();
```

```
        List<Product> allProductsStartsWith = new ArrayList<>();
```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						49	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

if (title == null) {
    return allProducts;
}
for (Product product : allProducts) {
    if (product.getTitle().toLowerCase().startsWith(title.toLowerCase()))
{
        allProductsStartsWith.add(product);
    }
}
return allProductsStartsWith;
}

//Поиск по названию города
public List<Product> findAllByCity(String city) {
    List<Product> allProducts = productRepository.findAll();
    List<Product> allProductsByCity = new ArrayList<>();
    if (city == null) {
        return allProducts;
    }
    for (Product product : allProducts) {
        if (product.getCity().toLowerCase().startsWith(city.toLowerCase()))
    {
        allProductsByCity.add(product);
    }
}
return allProductsByCity;
}

```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірив	Щур В. Ю..							50	86
						НТУУ “КПІ” ФІОТ ІОЗ-91			
Н. Контр.	Виноградов Ю.М.								
Затвердив									

//Поиск по цене

```
public List<Product> findAllByPrice(String price) {  
    List<Product> allProducts = productRepository.findAll();  
    List<Product> allProductsWithPriceLow = new ArrayList<>();  
    if (price == null) {  
        return allProducts;  
    }  
    for (Product product : allProducts) {  
        if (Integer.parseInt(product.getPrice()) <= Integer.parseInt(price)) {  
            allProductsWithPriceLow.add(product);  
        }  
    }  
    return allProductsWithPriceLow;  
}
```

//Поиск по категории

```
public List<Product> findAllByCategory(String category) {  
    List<Product> allProducts = productRepository.findAll();  
    List<Product> allProductsWithCategory = new ArrayList<>();  
    if (category == null) {  
        return allProducts;  
    }  
    for (Product product : allProducts) {  
        if (product.getCategory().equals(category)) {  
            allProductsWithCategory.add(product);  
        }  
    }  
}
```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						51	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

return allProductsWithCategory;
}

//Поиск по названию и городу
public List<Product> findAllByTitleStartingWithAndCity(String title,
String city) {
    List<Product> allProducts = productRepository.findAll();
    List<Product> allProductsStartsWithAndCity = new ArrayList<>();
    for (Product product : allProducts) {
        if (product.getTitle().toLowerCase().startsWith(title.toLowerCase()) &&
product.getCity().equals(city)) {
            allProductsStartsWithAndCity.add(product);
        }
    }
    return allProductsStartsWithAndCity;
}

//Поиск по названию и цене
public List<Product> findAllByTitleStartingWithAndPrice(String title,
String price) {
    List<Product> allProducts = productRepository.findAll();
    List<Product> allProductsStartsWithAndPrice = new ArrayList<>();
    for (Product product : allProducts) {
        if (product.getTitle().toLowerCase().startsWith(title.toLowerCase()) &&
Integer.parseInt(product.getPrice()) <= Integer.parseInt(price)) {
            allProductsStartsWithAndPrice.add(product);
        }
    }
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						52	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

    }

    return allProductsStartsWithAndPrice;
}

//Поиск по названию и категории

public List<Product> findAllByTitleStartingWithAndCategory(String title,
String category) {
    List<Product> allProducts = productRepository.findAll();
    List<Product> allProductsStartsWithAndCategory = new
ArrayList<>();
    for (Product product : allProducts) {
        if (product.getTitle().toLowerCase().startsWith(title.toLowerCase()) &&
product.getCategory().equals(category)) {
            allProductsStartsWithAndCategory.add(product);
        }
    }
    return allProductsStartsWithAndCategory;
}

//Поиск по городу и цене

public List<Product> findAllByCityAndPrice(String city, String price) {
    List<Product> allProducts = productRepository.findAll();
    List<Product> allProductsByCityAndPrice = new ArrayList<>();
    for (Product product : allProducts) {

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						53	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

        if(product.getCity().equals(city) && Integer.parseInt(product.getPrice()) <=
Integer.parseInt(price)) {
            allProductsByCityAndPrice.add(product);
        }
    }
    return allProductsByCityAndPrice;
}

```

//Поиск по городу и категории

```

public List<Product> findAllByCityAndCategory(String city, String
category) {
    List<Product> allProducts = productRepository.findAll();
    List<Product> allProductsByCityAndCategory = new ArrayList<>();
    for (Product product : allProducts) {
        if (product.getCity().equals(city) &&
product.getCategory().equals(category)) {
            allProductsByCityAndCategory.add(product);
        }
    }
    return allProductsByCityAndCategory;
}

```

//Поиск по цене и категории

```

public List<Product> findAllByPriceAndCategory(String price, String
category) {

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						54	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

List<Product> allProducts = productRepository.findAll();

List<Product> allProductsByPriceAndCategory = new ArrayList<>();
for (Product product : allProducts) {
    if (Integer.parseInt(product.getPrice()) <= Integer.parseInt(price) &&
product.getCategory().equals(category)) {
        allProductsByPriceAndCategory.add(product);
    }
}

return allProductsByPriceAndCategory;
}

```

```

//Поиск по названию городу и цене

public List<Product> findAllByTitleAndCityAndPrice(String title, String city,
String price) {

    List<Product> allProducts = productRepository.findAll();

    List<Product> allProductsByTitleAndCityAndPrice = new
ArrayList<>();

    for (Product product : allProducts) {
        if (Integer.parseInt(product.getPrice()) <= Integer.parseInt(price) &&
product.getTitle()
                .toLowerCase().startsWith(title.toLowerCase()) &&
product.getCity().equals(city)) {
            allProductsByTitleAndCityAndPrice.add(product);
        }
    }

    return allProductsByTitleAndCityAndPrice;
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						55	86
						НТУУ “КПІ” ФІОТ		
Н. Контр.	Виноградов Ю.М.					ІОЗ-91		
Затвердив								

```

//Поиск по названию городу и категории

public List<Product> findAllByTitleAndCityAndCategory(String title, String
city, String category) {

    List<Product> allProducts = productRepository.findAll();

    List<Product>    allProductsByTitleAndCityAndCategory    =    new
ArrayList<>();

    for (Product product : allProducts) {

        if                (product.getCategory().equals(category)                &&
product.getTitle().toLowerCase().startsWith(title.toLowerCase())                &&
product.getCity().equals(city)) {

            allProductsByTitleAndCityAndCategory.add(product);

        }

    }

    return allProductsByTitleAndCityAndCategory;

}

```

```

//Поиск по городу цене и категории

public List<Product> findAllByPriceAndCityAndCategory(String price, String
city, String category) {

    List<Product> allProducts = productRepository.findAll();

    List<Product>    allProductsByPriceAndCityAndCategory    =    new
ArrayList<>();

    for (Product product : allProducts) {

        if (Integer.parseInt(product.getPrice()) <= Integer.parseInt(price) &&
product.getCity().equals(city) && product.getCategory().equals(category)) {

            allProductsByPriceAndCityAndCategory.add(product);

        }

    }

}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						56	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

    }

    return allProductsByPriceAndCityAndCategory;
}

//Поиск по цене категории и названию
public List<Product> findAllByPriceAndCategoryAndTitle(String price, String
title, String category) {
    List<Product> allProducts = productRepository.findAll();
    List<Product> allProductsByPriceAndCategoryAndTitle = new
ArrayList<>();
    for (Product product : allProducts) {
        if (Integer.parseInt(product.getPrice()) <= Integer.parseInt(price) &&
product.getTitle()
            .toLowerCase().startsWith(title.toLowerCase()) &&
product.getCategory().equals(category)) {
            allProductsByPriceAndCategoryAndTitle.add(product);
        }
    }
    return allProductsByPriceAndCategoryAndTitle;
}

// public List<Product> findAllByTitleStartsWithAndCity(String title,
String city) {
//     if (Objects.equals(city, "")) {
//         return findAllByTitleStartsWith(title);
//     }

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						57	86
						НТУУ “КПІ” ФІОТ		
Н. Контр.	Виноградов Ю.М.					ІОЗ-91		
Затвердив								

```

//                                List<Product>  productsByTitleStartWithAndCity  =
findAllByTitleStartingWithAndCity(title, city);

//    return productsByTitleStartWithAndCity;
// }

/*////////////////////////////////////*/

public                                List<Product>
findAllByTitleStartsWithAndCityAndCategoryAndPrice(String title, String city,
String category, String price) {
    //Вывод всех товаров Y
    if  (Objects.equals(title,  "")  &&  Objects.equals(city,  "")  &&
Objects.equals(category,  "") && Objects.equals(price,  "")) {
        System.out.println("Вывод всех товаров");
        return productRepository.findAll();
    }

    //////////////////////////////////////

    //Поиск товаров по названию Y
    if  (!Objects.equals(title,  "")  &&  Objects.equals(category,  "")  &&
Objects.equals(price,  "") && Objects.equals(city,  "")) {
        System.out.println("Поиск товаров по названию");
        return findAllByTitleStartsWith(title);
    }

    //Поиск товаров по городу Y
    if  (Objects.equals(title,  "")  &&  Objects.equals(category,  "")  &&
Objects.equals(price,  "") && !Objects.equals(city,  "")) {

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						58	86
						НТУУ “КПІ” ФІОТ		
Н. Контр.	Виноградов Ю.М.					ІОЗ-91		
Затвердив								

```

        System.out.println("Поиск товаров по городу");
        return findAllByCity(city);
    }

    //Поиск товаров по цене Y
    if (Objects.equals(title, "") && Objects.equals(category, "") && !
Objects.equals(price, "") && Objects.equals(city, "")) {
        System.out.println("Поиск товаров по цене");
        return findAllByPrice(price);
    }

    //Поиск товаров по категориям Y
    if (Objects.equals(title, "") && !Objects.equals(category, "") &&
Objects.equals(price, "") && Objects.equals(city, "")) {
        System.out.println("Поиск товаров по категориям");
        return findAllByCategory(category);
    }

    //////////////////////////////////////

    //Поиск по названию и городу Y
    if (!Objects.equals(title, "") && Objects.equals(category, "") &&
Objects.equals(price, "") && !Objects.equals(city, "")) {
        System.out.println("Поиск по названию и городу");
        return findAllByTitleStartingWithAndCity(title, city);
    }

    //Поиск по названию и цене Y
    if (!Objects.equals(title, "") && Objects.equals(category, "") && !
Objects.equals(price, "") && Objects.equals(city, "")) {
        System.out.println("Поиск по названию и цене");
    }

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						59	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

return findAllByTitleStartingWithAndPrice(title, price);
}

//Поиск по названию и категории Y
if (!Objects.equals(title, "") && !Objects.equals(category, "") &&
Objects.equals(price, "") && Objects.equals(city, "")) {
    System.out.println("Поиск по названию и категории");
    return findAllByTitleStartingWithAndCategory(title, category);
}

//Поиск по городу и цене Y
if (Objects.equals(title, "") && Objects.equals(category, "") && !
Objects.equals(price, "") && !Objects.equals(city, "")) {
    System.out.println("Поиск по городу и цене");
    return findAllByCityAndPrice(city, price);
}

//Поиск по городу и категории Y
if (Objects.equals(title, "") && !Objects.equals(city, "") && !
Objects.equals(category, "") && Objects.equals(price, "")) {
    System.out.println("Поиск по городу и категории");
    return findAllByCityAndCategory(city, category);
}

//Поиск по цене и категории Y
if (Objects.equals(title, "") && Objects.equals(city, "") &&
!Objects.equals(category, "") && !Objects.equals(price, "")) {
    System.out.println("Поиск по цене и категории");
    return findAllByPriceAndCategory(price, category);
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						60	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

////////////////////////////////////

//Поиск по названию городу и цене Y
if (!Objects.equals(title, "") && !Objects.equals(city, "") &&
Objects.equals(category, "") && !Objects.equals(price, "")) {
    System.out.println("Поиск по названию городу и цене");
    return findAllByTitleAndCityAndPrice(title, city, price);
}

//Поиск по названию городу и категории Y
if (!Objects.equals(title, "") && !Objects.equals(city, "") && !
Objects.equals(category, "") && Objects.equals(price, "")) {
    System.out.println("Поиск по названию городу и категории");
    return findAllByTitleAndCityAndCategory(title, city, category);
}

//Поиск по городу цене и категории N
if (Objects.equals(title, "") && !Objects.equals(city, "") && !
Objects.equals(category, "") && !Objects.equals(price, "")) {
    System.out.println("Поиск по городу цене и категории");
    return findAllByPriceAndCityAndCategory(price, city, category);
}

//Поиск по цене категории и названию Y
if (!Objects.equals(title, "") && Objects.equals(city, "") && !
Objects.equals(category, "") && !Objects.equals(price, "")) {
    System.out.println("Поиск по цене категории и названию");
    return findAllByPriceAndCategoryAndTitle(price, title, category);
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						61	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```
//      if (Objects.equals(city, "") && (Objects.equals(category, ""))) {
//          return findAllByTitleStartsWith(title);
//      }
//      if (!Objects.equals(category, "")) {
//          return findAllByCategory(category);
//      }
```

```
return findAllByPriceAndCategory("10", "123");
```

```
}
```

```
/*////////////////////////////////////////*/
```

```
public void saveProduct(Principal principal, Product product, MultipartFile file1,
MultipartFile file2, MultipartFile file3) throws IOException {
```

```
    product.setUser(getUserByPrincipal(principal));
```

```
    Image image1;
```

```
    Image image2;
```

```
    Image image3;
```

```
    if (file1.getSize() != 0) {
```

```
        image1 = toImageEntity(file1);
```

```
        image1.setPreviewImage(true);
```

```
        product.addToProduct(image1);
```

```
    }
```

```
    if (file2.getSize() != 0) {
```

```
        image2 = toImageEntity(file2);
```

```
        product.addToProduct(image2);
```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірив	Щур В. Ю..						62	86	
						НТУУ “КПІ” ФІОТ ІОЗ-91			
Н. Контр.	Виноградов Ю.М.								
Затвердив									

```

    }

    if (file3.getSize() != 0) {
        image3 = toImageEntity(file3);
        product.addToProduct(image3);
    }

    log.info("Saving new Product. Title: {}; Author email: {}", product.getTitle(),
product.getUser().getEmail());

    Product productFromDb = productRepository.save(product);

productFromDb.setPreviewImageId(productFromDb.getImages().get(0).getId());

    productRepository.save(product);
}

public List<Product> sortProductsByDateDesc(){
    return productRepository.findAllByOrderByDateOfCreatedDesc();
}

public List<Product> sortProductsByDateAsc(){
    return productRepository.findAllByOrderByDateOfCreatedAsc();
}

private Image toImageEntity(MultipartFile file) throws IOException {
    Image image = new Image();
    image.setName(file.getName());
    image.setOriginalFileName(file.getOriginalFilename());
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						63	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

image.setContentType(file.getContentType());

    image.setSize(file.getSize());
    image.setBytes(file.getBytes());
    return image;
}

public void deleteProduct(Long id) {
    Product product = productRepository.findById(id)
        .orElse(null);
    if (product != null) {
        productRepository.delete(product);
    }
}

public User getUserByPrincipal(Principal principal) {
    if (principal == null) return new User();
    return userRepository.findByEmail(principal.getName());
}

public Product getProductById(Long id) {
    return productRepository.findById(id).orElse(null);
}
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду росробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						64	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

MvcConfig.java

```
package com.example.eshop.configurations;

import org.springframework.context.annotation.Configuration;
import
org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import
org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration

public class MvcConfig implements WebMvcConfigurer {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/static/**")
            .addResourceLocations("classpath:/static/");
    }
}
```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						65	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

Product.java

```
package com.example.eshop.entity;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
import javax.persistence.*;
```

```
import java.time.LocalDate;
```

```
import java.time.LocalDateTime;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
@Entity
```

```
@Table(name = "products")
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
public class Product {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    @Column(name = "id")
```

```
    private Long id;
```

```
    @Column(name = "title")
```

```
    private String title;
```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірів	Щур В. Ю..						66	86	
						НТУУ “КПІ” ФІОТ			
Н. Контр.	Виноградов Ю.М.					ІОЗ-91			
Затвердив									

```

@Column(name = "description", columnDefinition = "text")
private String description;
@Column(name = "price")
private String price;
@Column(name = "city")
private String city;
@Column(name = "category")
private String category;
@OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY,
mappedBy = "product")
private List<Image> images = new ArrayList<>();
private Long previewImageId;
@ManyToOne(cascade = CascadeType.REFRESH, fetch =
FetchType.LAZY)
@JoinColumn
private User user;
private LocalDate dateOfCreated;

@PrePersist
private void init() {
    dateOfCreated = LocalDate.now();
}

public void addImageToProduct(Image image) {
    image.setProduct(this);
    images.add(image);
}
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						67	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

UserService.java

```
package com.example.eshop.services;
```

```
import com.example.eshop.entity.User;
```

```
import com.example.eshop.entity.enums.Role;
```

```
import com.example.eshop.repositories.UserRepository;
```

```
import lombok.extern.slf4j.Slf4j;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.security.crypto.password.PasswordEncoder;
```

```
import org.springframework.stereotype.Service;
```

```
import java.security.Principal;
```

```
import java.util.Arrays;
```

```
import java.util.List;
```

```
import java.util.Map;
```

```
import java.util.Set;
```

```
import java.util.stream.Collectors;
```

```
@Service
```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив		Красулін Є.С.			Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірив		Щур В. Ю..					68	86	
						НТУУ “КПІ” ФІОТ			
Н. Контр.		Виноградов Ю.М.				ІОЗ-91			
Затвердив									

```

@Slf4j

public class UserService {

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private PasswordEncoder passwordEncoder;

    public boolean createUser(User user) {

        String email = user.getEmail();

        if (userRepository.findByEmail(email) != null) return false;

        user.setActive(true);

        user.setPassword(passwordEncoder.encode(user.getPassword()));

        user.setPassword(user.getPassword());

        user.getRoles().add(Role.ROLE_ADMIN);

        log.info("Saving new User with email: {}", email);

        userRepository.save(user);

        return true;

    }

    public List<User> getAllUsers(){

        return userRepository.findAll();

    }

    public void banUser(Long id) {

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						69	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

    if (id!=null){
        User user = userRepository.findById(id).orElse(null);
        assert user != null;
        user.setActive(false);
        log.info("Ban User with id ={}; email: {}", user.getId(),
user.getEmail());
        userRepository.save(user);
    }
}

public void unBanUser(Long id) {
    if (id!=null){
        User user = userRepository.findById(id).orElse(null);
        assert user != null;
        user.setActive(true);
        log.info("Unban User with id ={}; email: {}", user.getId(),
user.getEmail());
        userRepository.save(user);
    }
}

public void changeUserRole(User user, Map<String, String> form) {
    Set<String> roles = Arrays.stream(Role.values())
        .map(Role::name)
        .collect(Collectors.toSet());

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						70	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

user.getRoles().clear();

for (String key: form.keySet()){
    if (roles.contains(key)){
        user.getRoles().add(Role.valueOf(key));
    }
}

userRepository.save(user);
}

public User getUserByPrincipal(Principal principal) {
    if (principal == null) return new User();
    return userRepository.findByEmail(principal.getName());
}
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						71	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

User.java

```
package com.example.eshop.entity;

import com.example.eshop.entity.enums.Role;
import lombok.Data;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;

import javax.persistence.*;
import java.time.LocalDateTime;
import java.util.*;
```

```
@Entity
@Table(name = "users")
```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив	Красулін Є.С.				Лістинг Лістинг коду росзробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірів	Щур В. Ю..						72	86	
						НТУУ “КПІ” ФІОТ ІОЗ-91			
Н. Контр.	Виноградов Ю.М.								
Затвердив									

```

@Data
public class User implements UserDetails {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;
    @Column(name = "email", unique = true)
    private String email;
    @Column(name = "phone_number")
    private String phoneNumber;
    @Column(name = "name")
    private String name;
    @Column(name = "active")
    private boolean active;
    @OneToOne(cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    @JoinColumn(name = "image_id")
    private Image avatar;
    @Column(name = "password", length = 1000)
    private String password;
    @ElementCollection(targetClass = Role.class, fetch = FetchType.EAGER)
    @CollectionTable(name = "user_role",
        joinColumns = @JoinColumn(name = "user_id"))
    @Enumerated(EnumType.STRING)
    private Set<Role> roles = new HashSet<>();

```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірив	Щур В. Ю..						73	86	
						НТУУ “КПІ” ФІОТ			
Н. Контр.	Виноградов Ю.М.					ІОЗ-91			
Затвердив									

```
@OneToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER,
mappedBy = "user")
```

```
private List<Product> products = new ArrayList<>();
```

```
private LocalDateTime dateOfCreated;
```

```
@PrePersist
```

```
private void init() {
```

```
    dateOfCreated = LocalDateTime.now();
```

```
}
```

```
public boolean isAdmin(){
```

```
    return roles.contains(Role.ROLE_ADMIN);
```

```
}
```

```
// security
```

```
@Override
```

```
public Collection<? extends GrantedAuthority> getAuthorities() {
```

```
    return roles;
```

```
}
```

```
@Override
```

```
public String getUsername() {
```

```
    return email;
```

```
}
```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						74	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

@Override

```
public boolean isAccountNonExpired() {  
    return true;  
}
```

@Override

```
public boolean isAccountNonLocked() {  
    return true;  
}
```

@Override

```
public boolean isCredentialsNonExpired() {  
    return true;  
}
```

@Override

```
public boolean isEnabled() {  
    return active;  
}  
}
```

					ІАЛЦ.467200.003 Д1							
		№ докум.	Підпис	Дата								
Розробив		Красулін Є.С.			Лістинг Лістинг коду росзробленого застосунку		Літ.		Аркуш		Аркушів	
Перевірів		Щур В. Ю..							75		86	
							НТУУ “КПІ” ФІОТ ІОЗ-91					
Н. Контр.		Виноградов Ю.М.										
Затвердив												

UserController.java

```
package com.example.eshop.controllers;
```

```
import com.example.eshop.entity.User;
```

```
import com.example.eshop.services.UserService;
```

```
import com.sun.xml.bind.v2.util.StackRecorder;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import java.security.Principal;
```

```
@Controller
```

```
public class UserController {
```

```
    @Autowired
```

```
    private UserService userService;
```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірів	Щур В. Ю..						76	86	
						НТУУ “КПІ” ФІОТ			
Н. Контр.	Виноградов Ю.М.					ІОЗ-91			
Затвердив									

```
@GetMapping("/login")
```

```
public String login(Principal principal, Model model) {  
    model.addAttribute("user", userService.getUserByPrincipal(principal));  
    return "login";  
}
```

```
@GetMapping("/profile")
```

```
public String profile(Principal principal, Model model) {  
    User user = userService.getUserByPrincipal(principal);  
    model.addAttribute("user", user);  
    return "profile";  
}
```

```
@GetMapping("/registration")
```

```
public String registration(Principal principal, Model model) {  
    model.addAttribute("user", userService.getUserByPrincipal(principal));  
    return "registration";  
}
```

```
@PostMapping("/registration")
```

```
public String createUser(User user, Model model) {  
    if (!userService.createUser(user)) {  
        model.addAttribute("errorMessage", "Пользователь с email: " +  
user.getEmail() + " уже существует");  
        return "registration";  
    }  
    return "redirect:/login";  
}
```

					ІАЛЦ.467200.003 Д1				
		№ докум.	Підпис	Дата					
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів	
Перевірив	Щур В. Ю..						77	86	
						НТУУ “КПІ” ФІОТ			
Н. Контр.	Виноградов Ю.М.					ІОЗ-91			
Затвердив									

```

    @GetMapping("/user/{user}")

    public String userInfo(@PathVariable("user") User user, Model model, Principal
principal) {

        model.addAttribute("user", user);

        model.addAttribute("userByPrincipal",
userService.getUserByPrincipal(principal));

        model.addAttribute("products", user.getProducts());

        return "user-info";

    }

}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						78	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

CustomUserDetailsService.java

```
package com.example.eshop.services;

import com.example.eshop.repositories.UserRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import
org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;

@Service
public class CustomUserDetailsService implements UserDetailsService {

    @Autowired
    private UserRepository userRepository;

    @Override
    public UserDetails loadUserByUsername(String email) throws
UsernameNotFoundException {
        return userRepository.findByEmail(email);
    }
}
```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						79	86
						НТУУ “КПІ” ФІОТ		
Н. Контр.	Виноградов Ю.М.					ІОЗ-91		
Затвердив								

UserRepository.java

```
package com.example.eshop.repositories;

import com.example.eshop.entity.User;
import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {

    User findByEmail(String email);

}
```

ProductController.java

```
package com.example.eshop.controllers;

import com.example.eshop.entity.Product;
import com.example.eshop.entity.User;
import com.example.eshop.repositories.ProductRepository;
import com.example.eshop.services.ProductService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;
import java.security.Principal;
```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						80	86
						HTUU “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

@Controller
public class ProductController {

    @Autowired
    private ProductService productService;

    @Autowired
    private ProductRepository productRepository;

    @GetMapping("/")
    //      public String products(@ModelAttribute("searchWord") String title,
    @ModelAttribute("searchCity") String city, Principal principal, Model model) {
        public String products(@ModelAttribute("searchWord") String title,
    @ModelAttribute("searchCity") String city,
    @ModelAttribute("searchCategory") String category,
    @ModelAttribute("searchPrice") String price, Principal principal, Model model) {
    //      model.addAttribute("products",
    productService.findAllByTitleStartsWithAndCity(title, city));
        model.addAttribute("products",
    productService.findAllByTitleStartsWithAndCityAndCategoryAndPrice(title, city,
    category, price));
        model.addAttribute("user",
    productService.getUserByPrincipal(principal));
        model.addAttribute("searchWord", title);
        model.addAttribute("searchCity", city);
        model.addAttribute("searchCategory", category);
        model.addAttribute("searchPrice", price);
        return "products";
    }

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						81	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

    @GetMapping("/orderByDateAsc")
    public String productsWithDateAsc(@ModelAttribute("searchWord")
String title, @ModelAttribute("searchCity") String city,
                                     @ModelAttribute("searchCategory") String category,
                                     @ModelAttribute("searchPrice") String price,
                                     Principal principal, Model model){
        model.addAttribute("searchWord", title);
        model.addAttribute("searchCity", city);
        model.addAttribute("searchCategory", category);
        model.addAttribute("searchPrice", price);
        model.addAttribute("products",
productService.sortProductsByDateAsc());
        model.addAttribute("user",
productService.getUserByPrincipal(principal));
        return "products";
    }

```

```

    @GetMapping("/orderByDateDesc")
    public String productsWithDateDesc(@ModelAttribute("searchWord")
String title, @ModelAttribute("searchCity") String city,
                                     @ModelAttribute("searchCategory") String category,

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірів	Щур В. Ю..						82	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

    @ModelAttribute("searchPrice") String price,
                    Principal principal, Model model){
        model.addAttribute("searchWord", title);
        model.addAttribute("searchCity", city);
        model.addAttribute("searchCategory", category);
        model.addAttribute("searchPrice", price);
        model.addAttribute("products",
productService.sortProductsByDateDesc());
        model.addAttribute("user",
productService.getUserByPrincipal(principal));
        return "products";
    }

    @GetMapping("/product/{id}")
    public String productInfo(@PathVariable Long id, Model model, Principal
principal) {

        Product product = productService.getProductById(id);
        model.addAttribute("user",
productService.getUserByPrincipal(principal));
        model.addAttribute("product", product);
        model.addAttribute("images", product.getImages());
        model.addAttribute("authorProduct", product.getUser());
        return "product-info";
    }

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						83	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								

```

@PostMapping("/product/create")
public String createProduct(@RequestParam("file1") MultipartFile file1,
    @RequestParam("file2") MultipartFile file2,
        @RequestParam("file3") MultipartFile file3, Product
product, Principal principal) throws IOException {
    productService.saveProduct(principal, product, file1, file2, file3);
    return "redirect:/";
}

```

```

@PostMapping("/product/delete/{id}")
public String deleteProduct(@PathVariable Long id) {
    productService.deleteProduct(id);
    return "redirect:/my/products";
}

```

```

@GetMapping("/my/products")
public String userProducts(Principal principal, Model model) {
    User user = productService.getUserByPrincipal(principal);
    model.addAttribute("user", user);
    model.addAttribute("products", user.getProducts());
    return "my-products";
}
}

```

					ІАЛЦ.467200.003 Д1			
		№ докум.	Підпис	Дата				
Розробив	Красулін Є.С.				Лістинг Лістинг коду розробленого застосунку	Літ.	Аркуш	Аркушів
Перевірив	Щур В. Ю..						84	86
						НТУУ “КПІ” ФІОТ ІОЗ-91		
Н. Контр.	Виноградов Ю.М.							
Затвердив								