

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

**РОЗРОБКА ВЕБСАЙТУ «ПРОБЛЕМИ ЕКОЛОГІЇ» З
ВИКОРИСТАННЯМ SPRING FRAMEWORK**
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-18-1

Мітьков М.Є.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Машталір В.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Мітькову Максиму Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебсайту «Проблеми екології» з використанням Spring Framework

затверджена наказом університету від 16 травня 2022 року № 541Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 травня 2022 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, СУБД MySQL, середовище розробки програмних систем IntelliJ IDEA, редактор вихідного коду Visual Studio Code, фреймворк Spring.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Моделювання структури реляційної бази даних обраної предметної області.

2. Моделювання структури дизайну вебсайту.

3. Моделювання та реалізація клієнт-серверного застосунку «Проблеми екології» з інформаційною частиною з оглядом сучасних проблем екології та інтернет-магазину «екомагазин».

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Модель даних, згенерована в MySQL, UML-діаграма застосунку, ілюстрації макетів сторінок вебзастосунку, скріншоти роботи застосунку.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Белова Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.04.2022	
2	Аналіз завдання, підбір літератури	18.04.22-21.04.22	
3	Аналіз літератури з досліджуваної проблеми	22.04.22-25.04.22	
4	Аналіз технічних засобів	26.04.22-30.04.22	
5	Розробка інформаційної системи вибраної предметної області	01.05.22-14.05.22	
6	Програмна реалізація	15.05.22-23.05.22	
7	Оформлення пояснювальної записки	24.05.22-26.05.22	
8	Перевірка на плагіат	30.05.22	
9	Рецензування	31.05.22	
10	Підготовка презентації та доповіді	30.05.22-31.05.22	
11	Занесення роботи в електронний архів	31.05.22	
12	Попередній захист кваліфікаційної роботи	01.06.22	

Дата видачі завдання 18 квітня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Машталір В.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 54 с., 6 табл., 19 рис., 1 дод., 30 джерел.

КЛІЄНТ-СЕРВЕРНИЙ ЗАСТОСУНОК, ВЕБСАЙТ, ПРОБЛЕМИ ЕКОЛОГІЇ, JAVA, SPRING FRAMEWORK, BOOTSTRAP, VISUAL STUDIO CODE, MYSQL, МОВА ЗАПИТІВ SQL, HTML, CSS, JAVASCRIPT.

Об'єктом роботи є вебсайт «Проблеми екології».

Метою роботи є розробка клієнт-серверного застосунку, з реалізацією інформаційного аспекту з оглядом сучасних проблем екології та інтернет-магазину «екомагазин».

Для розробки Front-end частини проекту використовувались HTML, CSS, JavaScript та фреймворк BootStrap. Для розробки Back-end частини проекту використано мову програмування Java та такі вебтехнології як Java EE та Spring Framework, а саме Spring Core, Spring MVC, Spring JDBC, Spring Security. У якості СУБД було використано MySQL. Застосунок має 3 рівня доступу: незареєстрований користувач, зареєстрований користувач та адміністратор.

У результаті роботи здійснена програмна реалізація системи інтернет-магазину.

CLIENT SERVER APPLICATION, WEB-SITE, ECOLOGICAL PROBLEMS, JAVA, SPRING FRAMEWORK, BOOTSTRAP, VISUAL STUDIO CODE, MYSQL, МОВА ЗАПИТІВ SQL, HTML, CSS, JAVASCRIPT.

The object of the work is the website «Ecological Problems».

The purpose of the work is to develop a client-server application, with the implementation of the information aspect with an overview of current environmental issues and the online store «eco-shop».

HTML, CSS, JavaScript and the BootStrap framework were used to develop the Front-end part of the project. The Java programming language and web technologies such as Java EE and Spring Framework, namely Spring Core, Spring MVC, Spring JDBC, Spring Security, were used to develop the Back-end part of the project. MySQL was used as a database. The application has 3 levels of access: unregistered user, registered user and administrator.

As a result, the software implementation of the online store system was implemented.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Теоретичний огляд розробки вебсайтів	9
1.1 Історія вебсайтів	9
1.2 Огляд типів сайтів	10
1.3 Огляд типів інтернет-ресурсів	13
1.4 Огляд основних методів розробки вебсайтів	13
1.4.1 Конструктор	14
1.4.2 CMS	15
1.4.3 Самостійна розробка	17
1.5 Постановка задачі	19
2 Моделювання бази даних, структури та дизайну вебсайту	20
2.1 Загальний огляд концепції роботи застосунку	20
2.2 Розробка структури дизайну вебсайту «Проблеми екології»	21
2.3 Розробка бізнес-правил застосунку	22
2.4 Моделювання структури прав кожної ролі у системі	23
2.5 Розробка бази даних застосунку з врахуванням розроблених бізнес- правил	24
2.5.1 Визначення необхідних таблиць та полів бази даних	24
2.5.2 Моделювання бази даних	26
2.5.3 Створення фізичної моделі бази даних	34
3 Розробка вебсайту «Проблеми екології»	38
3.1 Обґрунтування вибору середовища програмної реалізації	38
3.1.1 IntelliJ IDEA	38
3.1.2 Visual Studio Code	39

3.1.3	BootStrap.....	40
3.1.4	Spring Framework	41
3.2	Інструкція користувача та дорожня карта проєкту	43
3.3	Тестування розробленого вебсайту «Проблеми екології»	45
3.4	Перспективи подальшої роботи.....	49
	Висновки.....	50
	Перелік джерел посилання	51
	Додаток А Додаткові ілюстрації вебсайту «Проблеми екології».....	53

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

WEB – World Wide Web (всесвітня павутина)

SQL – Structured Query Language (мова структурованих запитів)

ВСТУП

Складно переоцінити значення сайтів в сучасному світі, адже саме завдяки їм люди отримали чудові умови для навчання, роботи і, звичайно ж, розваг. Саме на сайтах можна знайти неосяжну кількість корисної та цікавої інформації, весело провести час на розважальних сайтах, поспілкуватися з друзями в соціальних мережах, знайти хороший фільм, серіал або музичні композиції. Сайти з'явилися порівняно недавно, а саме в 1990 році, що за мірками історії взагалі дрібниця. У той час сайти вважалися дивиною, і представляли вони собою вони простенькі сторінки, часом розбавлені гіперпосиланнями для переходу на інші сторінки.

Зараз сайти стали невід'ємною частиною переважної більшості людей, адже саме в мережі ми проводимо чималу частину свого життя.

У даній кваліфікаційній роботі мова йде про розробку клієнт-серверного вебзастосунку, який являє собою інтернет-магазин «Проблеми екології», на якому проведений розгляд актуальних сучасних проблем екології.

Метою роботи є розробка клієнт-серверного застосунку, з реалізацією інформаційного аспекту з оглядом сучасних проблем екології та інтернет-магазину «екомагазин».

Актуальність теми полягає в тому, що зараз проблеми екології стоять на порядку денному дуже гостро, тому було вирішено провести огляд кожної теми запропонувати до них рішення, в тому числі й завдяки товарам з розробленого інтернет-магазину.

1 ТЕОРЕТИЧНИЙ ОГЛЯД РОЗРОБКИ ВЕБСАЙТІВ

1.1 Історія вебсайтів

Перший в світі сайт `info.cern.ch` народився 6 серпня 1991 року. Його творець, Тім Бернерс-Лі, опублікував на ньому опис нової технології World Wide Web, заснованої на протоколі передачі даних HTTP, системі адресації URI і мові гіпертекстової розмітки HTML. Також на сайті були описані принципи установки і роботи серверів і браузерів. Сайт став і першим в світі інтернет-каталогом, оскільки пізніше Тім Бернерс-Лі розмістив на ньому список посилань на інші сайти [1].

Всі інструменти, необхідні для роботи першого сайту, Бернерс-Лі підготував ще раніше – в кінці 1990 року з'явилися перший гіпертекстовий браузер WorldWideWeb з функціоналом вебредактора, перший сервер на базі NeXTcube і перші вебсторінки.

«Батько» веба вважав, що гіпертекст може служити основою для мереж обміну даними, і йому вдалося втілити свою ідею в життя. Ще в 1980 році Тім Бернерс-Лі створив гіпертекстове програмне забезпечення Enquire, що використовує для зберігання даних випадкові асоціації. Потім, працюючи в Європейському центрі ядерних досліджень в Женеві (CERN), він запропонував колегам публікувати гіпертекстові документи, пов'язані між собою гіперпосиланнями. Бернерс-Лі продемонстрував можливість гіпертекстового доступу до внутрішніх пошукачеві документів, а також новинним ресурсам Інтернету. В результаті, в травні 1991 року в CERN був затверджений стандарт www [2].

Тім Бернерс-Лі є «батьком» основоположних технологій вебу – HTTP, URI/URL і HTML, хоча їх теоретичні основи були закладені ще раніше. У 1940-х роках Венівар Буш висунув ідеї розширення пам'яті людини за допомогою технічних пристроїв, а також індексації накопиченої людством інформації для її швидкого пошуку. Теодор Нельсон і Даг Енгельбарт запропонували технологію гіпертексту – «розгалуженого» тексту, який надає читачеві різні варіанти читання. Xanadu, так і не закінчена гіпертекстова система Нельсона, була призначена для зберігання і пошуку тексту, до якого введені взаємозв'язку і «вікна». Нельсон мріяв зв'язати перехресними посиланнями всі тексти, створені людством [3].

В даний час Тім Бернерс-Лі очолює заснований ним Консорціум Всесвітньої павутини (World Wide Web Consortium), який займається розробкою і впровадженням стандартів Інтернету [4].

1.2 Огляд типів сайтів

Певний клас сайтів інакше називають інтернет-представництвом людини чи організації. Як коментар до запланування може бути сторінка-візитка на повнофункціональному сайті (порталі). Коли говорять «своя сторінка в Інтернеті», то мають на увазі цілий сайт або особисту сторінку в складі чужого сайту (портал). Крім сайтів (порталів), в мережі Інтернет також доступні WAPсайти для мобільних телефонів [5].

Спочатку сайти представляли собою сукупність статичних документів, наприклад – сайт-візитка. У міру розвитку комунікацій, кількість внутрішніх і зовнішніх посилань збільшувалася. Сайт став виконувати не тільки роль довідки, анотації, а й функціонального офісу, новинного або медійного центру. В даний

час більшості з них властива динамічність і інтерактивність. Для таких випадків фахівці використовують термін вебзастосунок – готовий програмний комплекс для вирішення завдань сайту.

Вебзастосунок входить до складу сайту, але вебзастосунок без даних сайтом є тільки технічно. Оболонку (форму, шаблон) потрібно наповнити і активізувати [6].

У більшості випадків в Інтернеті одному сайту відповідає одне доменне ім'я. Саме по доменних іменах сайти ідентифікуються в глобальній мережі. Можливі інші варіанти: один сайт на декількох доменах або кілька сайтів під одним доменом.

Зазвичай кілька доменів використовують великі сайти (вебпортали), щоб логічно відокремити різні види послуг, що надаються (mail.google.com, news.google.com, maps.google.com).

Непоодинокі й випадки виділення окремих доменів для різних країн або мов. Наприклад, google.ru та google.fr логічно є сайтом Google на різних мовах, але технічно це різні сайти. Об'єднання декількох сайтів під одним доменом характерно для безкоштовних хостингів. Іноді для ідентифікації сайтів в адресі після вказівки хоста стоїть тильда і ім'я сайту: example.com/~my-site-name/, а найчастіше використовується домен третього рівня: my-site-name.example.com [7].

Один і той же сайт може бути доступний за різними адресами і зберігатися на різних серверах. Копія оригінального сайту в такому випадку називається дзеркалом. Існує також поняття офлайнова версія сайту – це копія сайту, яка може бути переглянута на будь-якому комп'ютері без підключення до комп'ютерної мережі і використання серверного програмного забезпечення (ПО).

При розробці сайту його тестують і налагоджують саме в офлайновій версії, для того, щоб не демонструвати нісенітницю і помилки, прорахунки великого проєкту. Саме для тестування в корпоративній мережі, або на початку в Інтернеті з обмеженим доступом під паролем запрошуються досвідчені «тестери». Це дозволяє прискорити виробництво великих проєктів і налагодити їх для масового відвідувача (користувача) [8].

Особливу роль виконують по розробці і обслуговуванню сайту (порталу) адміністратори (по-іншому – адміни, згідно інтернет-сленгу). Якщо виготовлення форми (оболонки) виконує група або дуже кваліфікований фахівець (програміст, вебдизайнер, системний адміністратор (згідно інтернет-сленгу – системний адміністратор), координатор, він же адміністратор проєкту), то обслуговування та інформаційне наповнення сайту підпорядковане, як правило, стратегічним завданням, вирішенням яких займається вся команда учасників проєкту під керуванням адміністратора проєкту (сайту, порталу).

Зараз напрацьовано багато програм і «пісалонок» в технології РНР, але це підвищило і вимоги до кваліфікації учасників проєкту, в зв'язку з многопрофільністю вирішуваних завдань [9].

Просто сторінка (сайт-візитка) може готуватися секретарем-референтом. Проєкти великих сайтів і порталів можуть зробити тільки обізнані і зацікавлені ІІ фахівці. Активна комунікація на сайті (порталі) часто виконує функцію директора напрямку і офісу зі службою супроводу (листування, комутатор прямого спілкування, оперативна довідка, і ін.).

Багато сайтів (порталів) оновлюють частіше ніж раз в день, а інтернет-магазини – за фактом руху товару (нові надходження, відсутності товару в наявності). Новинні сайти реально виставляють інформацію з точністю до хвилини, так як журналісти мають пріоритети на цитування першоджерел згідно з авторським правом, пріоритету посилань, рейтингу та ін.

1.3 Огляд типів інтернет-ресурсів

За доступності сервісів:

- відкриті – всі сервіси повністю доступні для будь-яких відвідувачів і користувачів;
- напіввідкриті – для доступу необхідно зареєструватися (зазвичай безкоштовно);
- закриті – повністю закриті службові сайти організацій (у тому числі корпоративні сайти), особисті сайти приватних осіб. Такі сайти доступні для вузького кола користувачів. Доступ новим користувачам зазвичай дається через так звані інвайт (запрошення) [10].

За фізичним розташуванням:

- загальнодоступні сайти мережі Інтернет;
- локальні сайти – доступні тільки в межах локальної мережі. Це можуть бути як корпоративні сайти організацій, так і сайти приватних осіб в локальній мережі провайдера.

1.4 Огляд основних методів розробки вебсайтів

В даний час існує безліч різних способів і засобів створити свій сайт, але будь-який із цих способів можна віднести до однієї з трьох категорій:

- розробка у конструкторі сайтів;
- розробка на CMS (особливо популярна WordPress);
- самостійна розробка, у тому числі з використанням популярних інструментів та фреймворків (Laravel, Django, Spring та ін.) [11].

1.4.1 Конструктор

Конструктор – програмне рішення (найчастіше онлайн), що дозволяє побудувати сайт за модульним принципом, коли розробник збирає всю конструкцію за допомогою готових «кубиків», які надає конструктор. Такий підхід дозволяє створити сайт взагалі без знань про веброзробку та супутні навички. Найбільше підійде тим, хто хоче зробити простенький сайт про себе чи свою компанію, без претензій на якість програмного коду та швидкість роботи сайту. Також, на конструкторах часто роблять прості лендинги для невеликих рекламних кампаній. Зробити серйозне корпоративне рішення чи інтернет-магазин на конструкторі неможливо [12].

Плюси:

- низька ціна. Майже всі конструктори спочатку безкоштовні, а вартість модулів, що підключаються, дуже низька;
- простота використання. Щоб створити свій сайт, достатньо вміти користуватися браузером комп'ютера і мати почуття смаку;
- уся рутинна робиться конструктором. Великі програмні скрипти, підключення модулів, розміщення на хостингу та інші приховані процеси здійснюються через зрозумілу панель керування кількома кліками на кнопках.

Мінуси:

- приховані витрати. За первинною дешевизною часто ховаються додаткові, часом значні витрати: розміщення на хостингу, домен другого рівня (на кшталт example.ru), електронна пошта з ім'ям домену та ін;
- домен третього чи вищого рівня. При використанні безкоштовних облікових записів у конструкторах, клієнт отримує розміщення свого сайту тільки на домені не нижче третього рівня, наприклад sitename.constructor.com. Такі домени не викликають великої довіри в інтернет-спільноті, а домени

другого рівня (напр. sitename.com) коштують дорожче, ніж якщо їх купувати у реєстраторів доменів безпосередньо;

- важкість сайту. Сайт, зроблений на конструкторі завжди завантажуватиметься довше за аналогічний сайт, зроблений на CMS або розроблений самостійно. Пояснюється це тим, що конструктор містить у собі величезну кількість програмного коду, який не відноситься до Вашого сайту, але необхідний для побудови його підсумкового вигляду;

- відсутність SEO. Незважаючи на заяви будь-якого конструктора сайтів, провести повноцінне SEO сайту, зробленого на конструкторі, не буде можливим через те, що SEO – це комплекс заходів, у тому числі, по роботі з кодом, індексації в пошукових системах, побудові структури сайту, та для цього потрібно доступ до програмного коду і розмітці, що неможливо в конструкторі [13].

1.4.2 CMS

У цьому випадку розглянемо найпопулярнішу CMS – WordPress. У неї є різні аналоги, які не такі популярні, але мають схожий набір інструментів та можливостей: Joomla, Drupal та інші.

CMS – це комплекс програмних інструментів для керування вебконтентом. Простими словами – це базовий каркас та набір додаткових інструментів та надбудов, який дозволяє не тільки створити вебсайт або вебпрограму, але й підтримувати його роботу, оновлювати контент та взаємодіяти з користувачами. Всі CMS мають панель керування з відносно дружнім інтерфейсом. Основна мова програмування – PHP. Будь-яка CMS може дозволити створити дуже складні

рішення, такі як інтернет-магазини або великі корпоративні сайти з глибокою вкладеністю сторінок, але і у них є свої особливості [14].

Плюси:

- безкоштовний доступ. Багато CMS (крім 1С-Бітрікс) спочатку безкоштовні, а крім того, в мережі існує безліч готових шаблонів сайтів під них (особливо під WordPress). Берете будь-який, що сподобався, міняєте дизайн під себе і сайт готовий;

- зручне керування контентом. За допомогою панелі керування керувати сайтом легко та просто. При використанні CMS не потрібні особливі навички для керування контентом;

- безліч готових рішень. У мережі існує маса модулів, плагінів, доповнень для різних завдань (від слайдерів для картинок до систем seo-оптимізації та супроводу користувача) [15].

Мінуси:

- вразливість сайту. Найважливіший мінус будь-якого поширеного CMS. Навіть сайти, зроблені на конструкторах, іноді краще захищені від злому, ніж ті, що знаходяться під управлінням WordPress та інших CMS. Навіть платна 1С-Бітрікс визнана далеко не найнадійнішою та захищеною від атак та проникнення [16];

- вимоги до знань. Розробка сайту на CMS вже вимагає від клієнта базових знань з верстки та програмування (в основному мовою PHP), що вже додає складнощів при створенні сайту. Хоча, варто зауважити, що знання можуть знадобитися тоді, коли клієнт хоче додати до наявного шаблону новий функціонал або створити повністю з нуля власний проєкт;

- складнощі з перенесенням. Хоча популярні CMS на даний момент мають автоматизовані засоби установки майже на будь-якому хостингу, у разі

необхідності перенесення сайту або управління його становищем можуть виникнути труднощі, так як доведеться робити всю процедуру установки заново;

- витрати на додатковий контент. Як і у випадку з конструкторами, додаткові модулі та розширення сайту для CMS коштують грошей, і в даному випадку вже більших, ніж для конструкторів. Майже будь-який модуль, який відноситься до постійно використовуваних, вимагатиме або пристойних витрат на початку або платної щомісячної підписки;

- великий сайт – великі витрати. Розробка складного та великого проєкту на CMS за витратами вийде не дешевше, а в деяких випадках, дорожче за розробку чистими мовами або з використанням фреймворків. Помилка, що з CMS Ви зможете зробити сайт дешевшим, ніж замовивши його у професіоналів, таке дуже часто зустрічається і потім обходиться клієнту в додаткові грошові витрати [17].

1.4.3 Самостійна розробка

Найбільш творчий і вільний, але й трудомісткий процес, який можна порівняти з пошиттям костюма по фігурі. Написання сайту або вебпрограми вимагає серйозних знань не тільки з самих мов програмування, але й розуміння архітектури, бізнес-процесів клієнта та багато іншого.

При цьому, створюючи сайт з нуля, клієнт отримає унікальний та персоналізований продукт, який вирішуватиме його завдання та не витратиме час на зайві процеси.

Самостійна розробка дозволяє створювати проєкти будь-якої складності та за будь-якими побажаннями клієнта.

Плюси:

- свобода вибору. Ви можете замовити все, що потрібно для ефективного вирішення бізнес-завдань. При цьому весь функціонал буде написаний саме під Ваші потреби, а не адаптований з будь-якого шаблону [18];
- широкі можливості просування. На відміну від CMS та конструкторів, просунути у природному пошуку самостійно розроблений сайт набагато легше;
- індивідуальний дизайн. Тільки чистий код дозволить створити той продукт, який Ви бачите і хочете отримати. Окремо можна відзначити, що грамотний підхід до UI/UX може бути тільки при використанні чистого коду, так як при використанні CMS клієнт змушений звертатися до готових рішень, які можуть не відповідати всім його потребам [19].

Мінуси:

- ціна. Спірний, але мінус чистого коду. Хоча у разі великих проєктів, розробка без конструкторів і CMS є найефективнішим і часом найдешевшим рішенням, тому що в результаті не доведеться отримувати не те, що хотілося, а згодом не переробляти при кожній необхідності;
- наявність знань. Самостійна розробка вимагає наявності великої кількості знань з мов програмування, побудови архітектури вебзастосунків, алгоритмів і структур даних, бізнес-процесів та багато іншого, що тягне за собою необхідність звертатися до найманих фахівців або в вебстудії;
- витрати часу. Хоч як крути, але використання CMS та конструкторів дозволяє зробити простий сайт швидше, ніж при розробці без них. У ситуаціях, коли час грає велику роль, звернення до чистого коду стає безглуздим [20].

1.5 Постановка задачі

Об'єктом розробки є вебсайт «Проблеми екології».

Метою роботи є реалізація клієнт-серверного застосунку для огляду сучасних проблем екології.

Таким чином, необхідно розробити методи і алгоритм створення динамічного вебсайту. Для цього необхідно вирішити наступні задачі:

- ознайомитися з сучасними Інтернет-технологіями і використовувати їх у своїй розробці;
- вивчити програмний інструментарій, застосовуваний для розробки і створення Web-сайтів;
- виявити і врахувати методи і способи подання на Web-сторінках різних видів інформації, що не перешкоджають їх доступності;
- ознайомитися з основними правилами і рекомендаціями по розробці і створенню Web-сайтів і неухильно дотримуватися їх у своїй практиці;
- вибрати стратегію розробки та створення Web-сайту.

Для даної предметної області було обрано наступні технології:

- для розробки Front-end частини застосунку будуть використовуватися HTML, CSS, JavaScript та фреймворк BootStrap;
- для розробки Back-end частини застосунку будуть використовуватися технології Java EE та Spring Framework, а саме Spring Core, Spring MVC, Spring JDBC та Spring Security;
- для моделювання та реалізації бази даних вебзастосунку обрано СУБД MySQL.

2 МОДЕЛЮВАННЯ БАЗИ ДАНИХ, СТРУКТУРИ ТА ДИЗАЙНУ ВЕБСАЙТУ

2.1 Загальний огляд концепції роботи застосунку

Основною ідеєю вебсайту є клієнт-серверний вебзастосунок, який поєднує інформаційну частину з оглядом різних сучасних проблем екології (рис. А.1) та реалізацію інтернет-магазину різних екологічних речей, здебільшого, екосумок.

У межах цієї програми користувач може мати три ролі: незареєстрований користувач, зареєстрований користувач і адміністратор. Кожна роль має свій відповідний перелік можливостей і функціоналу.

Незареєстрований користувач має повне право на вивчення інформаційної частини вебсайту, пограти в міні-гру, а також має доступ до екомагазину і може додати товар, що сподобався, в кошик, але для здійснення замовлення вже необхідно авторизуватися.

Зареєстрований користувач (клієнт) має всі привілеї незареєстрованого користувача, а також має можливість здійснити замовлення вибраних товарів та продивлятися список своїх зроблених замовлень.

Адміністратор програми має доступ до таблиць користувачів, товарів та угод. Він має можливість додавати нових користувачів, видаляти існуючих, додавати нові товари, редагувати та видаляти вже існуючі, а також може видаляти оформлені користувачами замовлення.

2.2 Розробка структури дизайну вебсайту «Проблеми екології»

Дизайн сайту буде збудований методом секційної верстки (рис. 2.1, 2.2). Для цього необхідно виділити основні секції документа, такі як header, footer (рис. А.2), та декілька необхідних функціональних блоків.

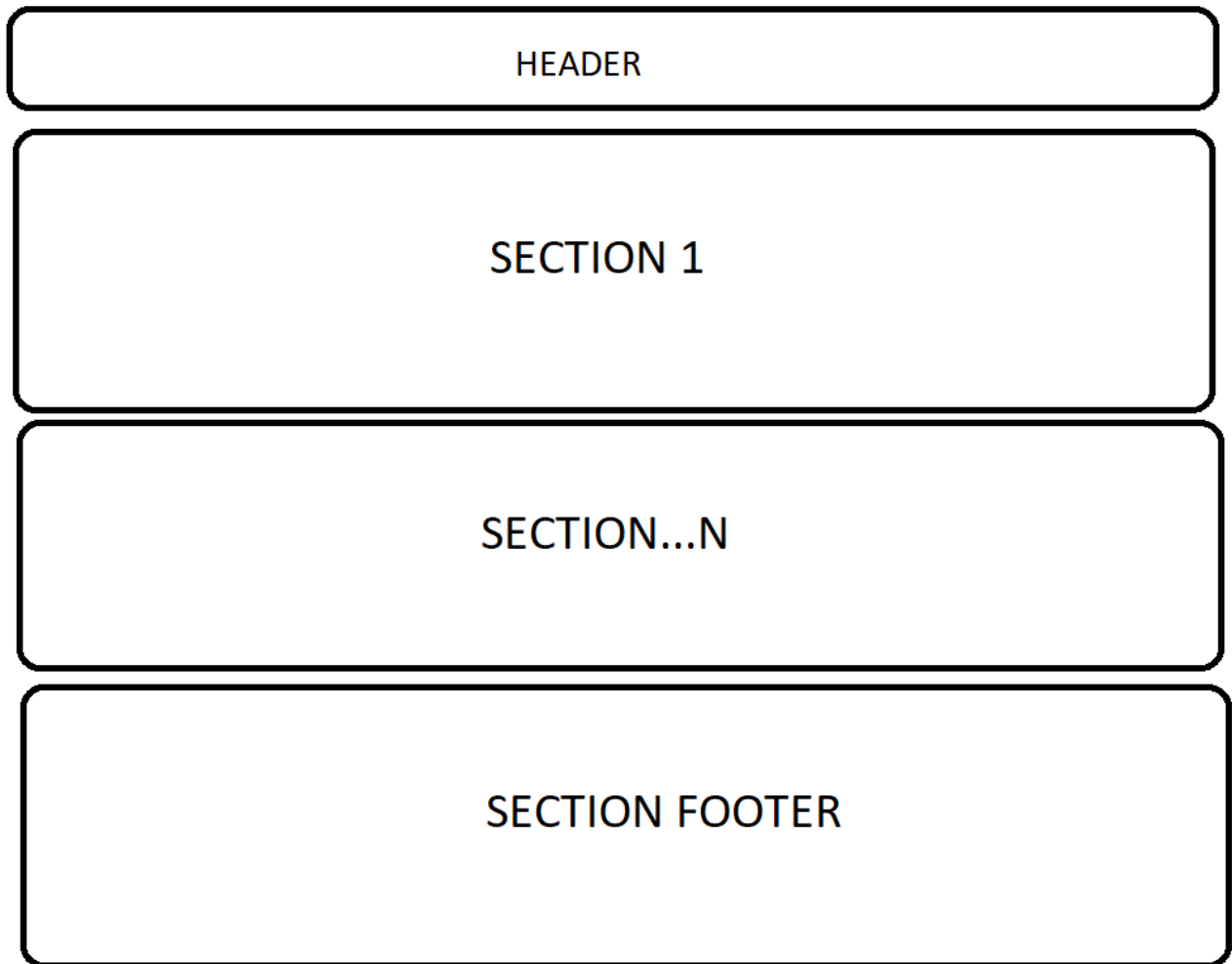


Рисунок 2.1 – Макет головної інформаційної сторінки вебсайту «Проблеми екології»

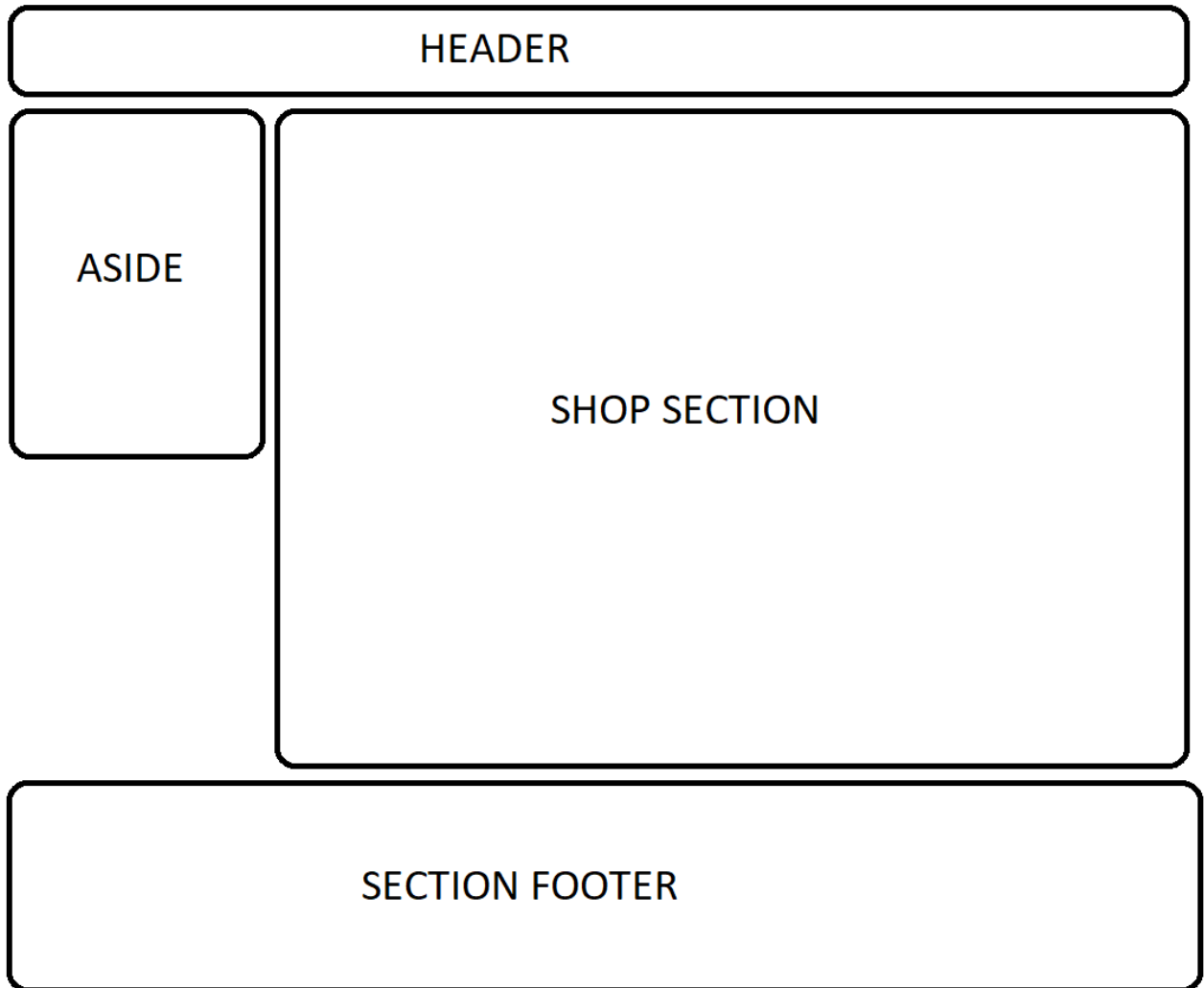


Рисунок 2.2 – Макет сторінки екомагазину вебсайту «Проблеми екології»

2.3 Розробка бізнес-правил застосунку

Враховуючи концепцію проєкту, необхідно побудувати бізнес-правила до нього:

- незареєстрований користувач може користуватися інформаційною частиною вебсайту;
- незареєстрований користувач може додати товари до кошика;
- незареєстрований користувач не може оформити замовлення;

- зареєстрований користувач може оформити замовлення товарів;
- адміністратор може додавати нових користувачів у систему та видаляти існуючих;
- адміністратор може додавати нові товари в систему, а також редагувати і видаляти вже існуючі;
- адміністратор може видаляти оформлені клієнтами замовлення.

2.4 Моделювання структури прав кожної ролі у системі

За визначеними бізнес-правилами побудуємо UML-діаграму прецедентів сайту «Проблеми екології» (рис. 2.3).

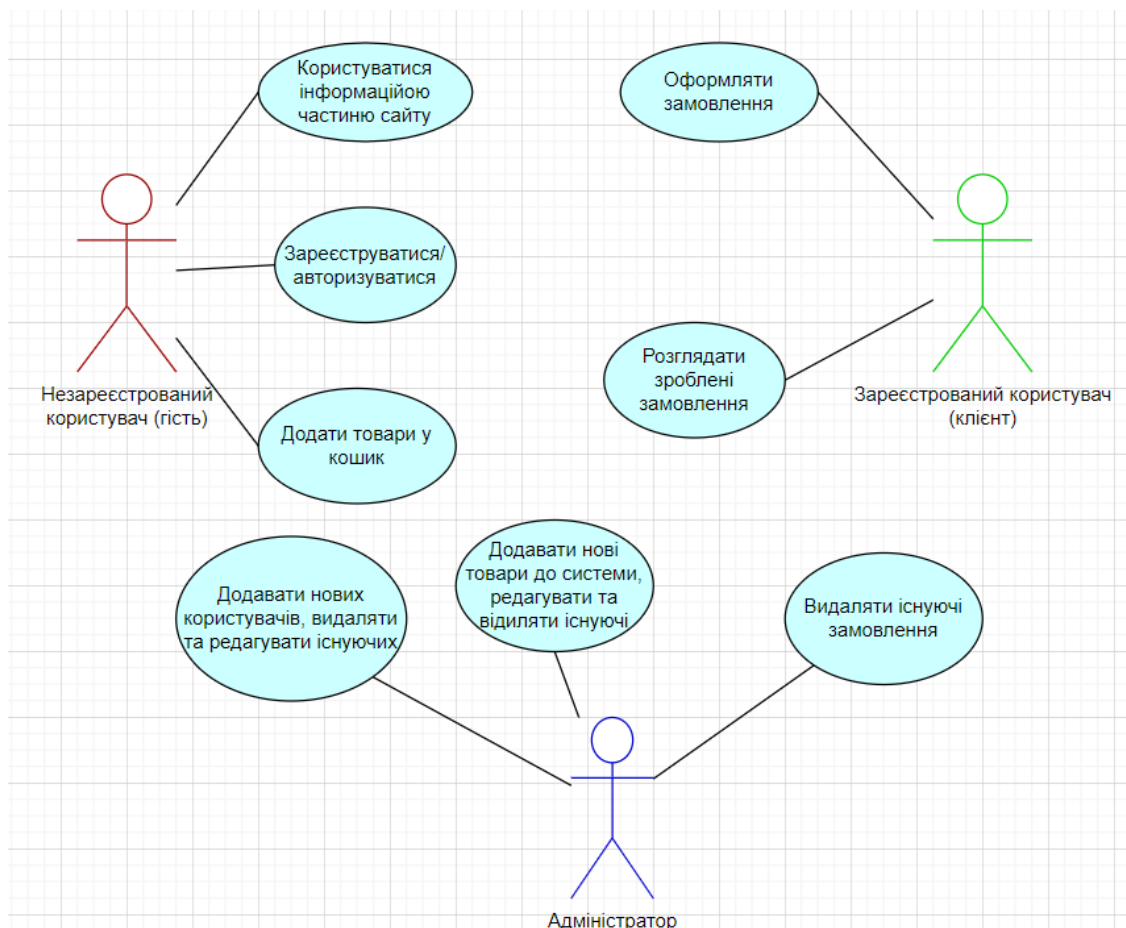


Рисунок 2.3 – UML-діаграма прецедентів сайту «Проблеми екології»

2.5 Розробка бази даних застосунку з врахуванням розроблених бізнес-правил

2.5.1 Визначення необхідних таблиць та полів бази даних

Для розробки бази даних програми необхідно визначити, які сутності будуть використовуватися в роботі.

Так як у нас буде присутній магазин екопродуктів, значить нам необхідно їх десь зберігати. Для цього створимо таблицю products з наступними полями:

- product_id (ідентифікатор товарів);
- product_brand (бренд товару);
- product_model (назва моделі товару);
- product_material (матеріал товару);
- product_length (значення довжини товару);
- product_price (ціна товару);
- product_add_date (дата додавання продукту до бази даних);
- product_update_date (дата останнього редагування товару).

Оскільки в системі будуть присутні користувачі різних ролей, необхідно створити для них відповідну таблицю – users, за наступними полями:

- user_id (ідентифікатор користувача);
- user_firstName (ім'я користувача);
- user_lastName (прізвище користувача);
- user_login (логін користувача);
- user_mobile (номер телефону користувача);
- user_email (email користувача);
- user_password (пароль користувача);
- user_registered_date (дата реєстрації користувача);
- user_role (роль користувача).

Користувач додає товари до корзини. Отже, потрібна таблиця, що зберігає дані про сесію користувача, адже додати товар до кошика може навіть незареєстрований користувач. Для цього буде таблиця `shopping_cart` з наступними полями:

- `cart_id` (ідентифікатор корзини);
- `cart_sessionId` (ідентифікатор сесії користувача);
- `cart_created_date` (дата створення корзини);
- `user_id` (зовнішній ключ користувача, власника корзини).

Товари, які додані до кошика потрапляють до таблиці `products_in_order` з наступними полями:

- `prod_order_id` (ідентифікатор товарів в корзині);
- `prod_order_name` (назва товарів в корзині);
- `prod_order_price` (ціна товарів в корзині);
- `prod_order_count` (кількість товарів в корзині);
- `cart_user_id` (зовнішній ключ користувача, власника корзини);
- `cart_id` (зовнішній ключ корзини);
- `product_id` (зовнішній ключ товару).

Інформація про замовлення зберігається у таблиці `orders` з наступними полями:

- `order_id` (ідентифікатор замовлення);
- `order_create_date` (дата створення замовлення);
- `customer_firstName` (ім'я клієнта);
- `customer_lastName` (прізвище клієнта);
- `product_name` (назва товарів);
- `product_cost` (вартість замовлення);
- `product_order_id` (зовнішній ключ товарів замовлення);
- `user_id` (зовнішній ключ користувача, власника замовлення).

На основі цього побудуємо таблицю зв'язків між сутностями (табл 2.1).

Таблиця 2.1 – Типи зв'язків між сутностями

Назва сутності	Назва сутності	Зв'язок
Users	Shopping_cart	1:M
Users	Orders	1:M
Shopping_cart	Products_id_order	1:M
Products	Products_id_order	1:M
Products_id_order	Orders	1:M

2.5.2 Моделювання бази даних

На основі визначених таблиць можна приступити до моделювання бази даних. Для цього чудово підійде ER-діаграма.

Діаграма прив'язки сутності ER – це блок-схеми, які ілюструють, як «сутності» (люди, об'єкти чи концепції) відносяться один до одного в системі. ER-діаграма - це та модель, яка найчастіше використовується для розробки або налагодження реляційних баз даних у галузях ПЗ, бізнес-інформаційних систем та досліджень. Вона використовує набір геометричних символів, таких як прямокутник, ромб, овал та лінії, для відображення взаємозв'язку об'єктів, відносин та їх атрибутів.

Розробка цієї діаграми ER приписується Пітер Чен, викладачеві університету Карнегі-Меллона в Піттсбурзі. Він почав її використовувати для проектування баз даних у 70-х роках минулого сторіччя. Будучи помічником професора в Школі управління Sloan MIT, в 1976 він опублікував статтю під назвою «Модель відношення сутностей до єдиного представлення даних». У

ширшому історичному сенсі, розуміння зображення взаємозв'язку речей перегукується з древньої Греції, принципи яких виявлено у працях Аристотеля, Сократа і Платона [21].

Про них згадується у роботах філософів-логіків пізнішого історичного періоду – Чарльза Сандерс Пірса та Готлофа Фреге. До 1960-х і 1970-х років Чарльз Бахман з однопумцями працювали над теоретичною моделлю Чена. Бахман розробив діаграму структури даних, яку він назвав. Браун опублікував роботи з моделювання реальних систем.

Джеймс Мартін додав уточнення до цієї діаграми ER. Робота Чена, Бахмана, Брауна, Мартіна та інших учасників призвела до розробки уніфікованої мови моделювання (UML), що застосовується у розробці програмного забезпечення.

Цей метод використовується для моделювання та розробки реляційних баз даних в ER-діаграмах. Особливо часто застосовується в логічній моделі та з точки зору конкретної технології, яка буде реалізована в моделі фізичних даних. У розробці ПЗ, діаграма ER часто є початковим етапом встановлення вимог до проекту інформ-систем. Реляційна база даних має еквівалентну реляційну таблицю і таким чином виражається за необхідності [22].

Області застосування:

- виправлення несправностей бази даних. Діаграми ER використовуються для аналізу існуючих баз даних для пошуку та усунення проблем у логіці чи розгортанні. Малюнок діаграми має показати де це відбувається;

- системи бізнес-інформації. Діаграми використовуються для розробки або аналізу баз даних, що використовуються в бізнес-процесах. Будь-який бізнес-процес, який використовує польові дані, що включають сутності, дії

та взаємодію, може потенційно отримати вигоду з реляційної бази даних. Він може оптимізувати процеси, спростити інформацію та покращити результати;

- реорганізація бізнес-процесів (BPR). Моделі діаграм ER допомагають у аналізі баз даних, що використовуються під час реорганізації бізнес-процесів;

- освіта. База даних сьогодні є методом зберігання реляційної інформації для освітніх цілей та подальшого пошуку, тому діаграми ER можуть бути корисними при плануванні цих структур;

- дослідження. Оскільки так багато досліджень зосереджено на структурованих фактах, ER може відігравати ключову роль у створенні корисних баз даних для їх аналізу.

ER діаграми баз даних є дуже корисною основою для створення та управління масивами даних. По-перше, діаграму ER легко зрозуміти. Це означає, що, наприклад, дизайнери можуть використовувати діаграми ER для простого спілкування з розробниками, клієнтами та кінцевими користувачами, незалежно від їхнього професіоналізму в області ІТ.

Вони легко переводяться в реляційні таблиці, які можна використовувати для швидкого збирання баз даних. Крім того, діаграми ER можуть безпосередньо використовуватися розробниками БД як план для впровадження даних у конкретні програмні програми. Вони можуть бути використані в інших контекстах, таких як опис різних відносин і операцій всередині організації.

Діаграма ER дуже популярна, оскільки має безліч переваг:

- ефективна комунікація дозволяє читачам легко розуміти відносини між різними предметними галузями діаграми ER;

- символи використовуються для ефективного представлення інформації, а також допомагають розуміти роботу бази даних;

- візуальне представлення діаграми потоків даних разом із діаграмами ER можуть ефективно використовуватися для візуального представлення макету;
- просте розуміння дизайну за допомогою діаграм ER;
- висока гнучкість. Діаграму ER можна ефективно використовувати для встановлення відносин із існуючими системами. Для виконання цієї операції можуть використовуватись математичні формули та реляційні таблиці [23].

Побудуємо ER-діаграму (рис. 2.4) для бази даних (табл 2.2-2.6).

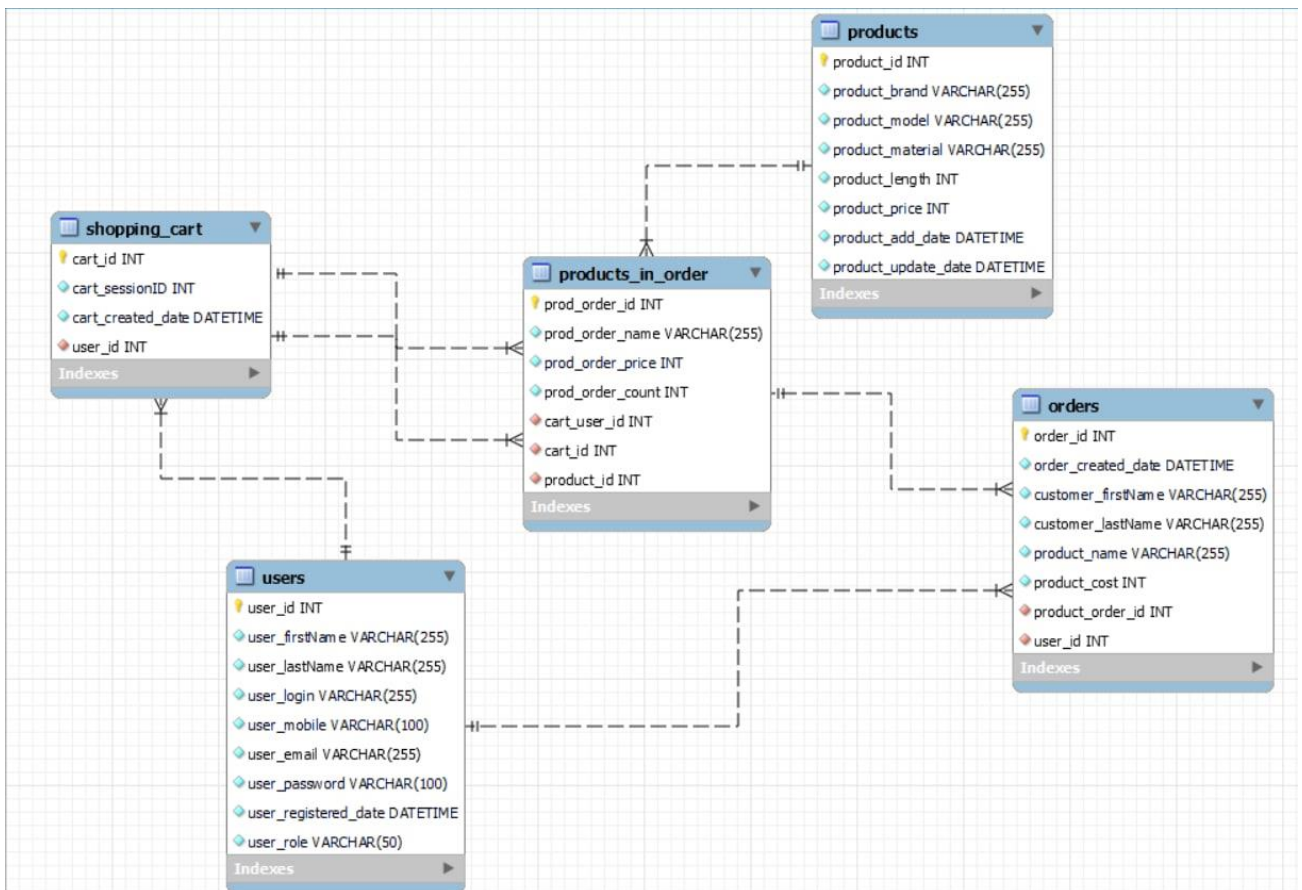


Рисунок 2.4 – Згенерована в MySQL модель бази даних

Таблиця 2.2 – Опис структури таблиці «Products»

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	Product_id	INT		Ідентифікатор товарів
	Product_brand	VARCHAR	255	Бренд товару
	Product_model	VARCHAR	255	Назва моделі товару
	Product_material	VARCHAR	255	Матеріал товару
	Product_length	INT		Значення довжини товару
	Product_price	INT		Ціна товару
	Product_add_date	DATETIME		Дата додавання продукту до бази даних
	Product_update_date	DATETIME		Дата останнього редагування товару

Таблиця 2.3 – Опис структури таблиці «Users»

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	user_id	INT		Ідентифікатор користувача
	user_firstName	VARCHAR	255	Ім'я користувача
	user_lastName	VARCHAR	255	Прізвище користувача
	user_login	VARCHAR	255	Логін користувача
	user_mobile	VARCHAR	100	Номер телефону користувача
	user_email	VARCHAR	255	Email користувача
	user_password	VARCHAR	100	Пароль користувача
	user_registered_date	DATETIME		Дата реєстрації користувача
	user_role	VARCHAR	50	Роль користувача

Таблиця 2.4 – Опис структури таблиці «Shopping_cart»

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	cart_id	INT		Ідентифікатор корзини
	cart_sessionId	INT		Ідентифікатор сесії користувача
	cart_created_date	DATETIME		Дата створення корзини
FK	user_id	INT		Зовнішній ключ Користувача, власника корзини

Таблиця 2.5 – Опис структури таблиці «Products_in_order»

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	prod_order_id	INT		Ідентифікатор товарів в корзині
	prod_order_name	VARCHAR	255	Назва товарів в корзині
	prod_order_price	INT		Ціна товарів в корзині

Продовження таблиці 2.5

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
	prod_order_count	INT		Кількість товарів в корзині
FK	cart_user_id	INT		Зовнішній ключ користувача, власника корзини
FK	cart_id	INT		Зовнішній ключ корзини
FK	product_id	INT		Зовнішній ключ товару

Таблиця 2.6 – Опис структури таблиці «Products_in_order»

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	order_id	INT		Ідентифікатор замовлення
	order_create_date	DATETIME		Дата створення замовлення
	customer_firstName	VARCHAR	255	Ім'я клієнта

Продовження таблиці 2.6

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
	customer_lastName	VARCHAR	255	Прізвище клієнта
	product_name	VARCHAR	255	Назва товарів
	product_cost	INT		Вартість замовлення
FK	product_order_id	INT		Зовнішній ключ товарів замовлення
FK	user_id	INT		Зовнішній ключ користувача, власника замовлення

2.5.3 Створення фізичної моделі бази даних

На основі визначених раніше моделей створимо структуру бази даних застосунку.

Лістинг 2.1 Створення таблиці «Products»:

```
CREATE TABLE products (  
    product_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    product_brand VARCHAR(255) NOT NULL,  
    product_model VARCHAR(255) NOT NULL,  
    product_material VARCHAR(255) NOT NULL,  
    product_length INT NOT NULL,  
    product_price INT NOT NULL,  
    product_add_date datetime NOT NULL,  
    product_update_date datetime NOT NULL) ENGINE = InnoDB;
```

Лістинг 2.2 Створення таблиці «Users»:

```
CREATE TABLE users (  
    user_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    user_firstName VARCHAR(255) NOT NULL,  
    user_lastName VARCHAR(255) NOT NULL,  
    user_login VARCHAR(255) NOT NULL,  
    user_mobile VARCHAR(100) NOT NULL,  
    user_email VARCHAR(255) NOT NULL,  
    user_password VARCHAR(100) NOT NULL,  
    user_registered_date datetime NOT NULL,  
    user_role VARCHAR(50) NOT NULL  
)  
ENGINE = InnoDB;
```

Лістинг 2.3 Створення таблиці «shopping_cart»:

```
CREATE TABLE shopping_cart (  
    cart_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    cart_sessionID INT NOT NULL,  
    cart_created_date datetime NOT NULL,  
    user_id INT NOT NULL,  
  
    FOREIGN KEY (user_id) REFERENCES users (user_id),  
  
    UNIQUE (user_id)  
) ENGINE = InnoDB;
```

Лістинг 2.4 Створення таблиці «products_in_order»:

```
CREATE TABLE products_in_order (  
    prod_order_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    prod_order_name VARCHAR(255) NOT NULL,  
    prod_order_price INT NOT NULL,  
    prod_order_count INT NOT NULL,  
    cart_user_id INT NOT NULL,  
    cart_id INT NOT NULL,  
    product_id INT NOT NULL,  
  
    FOREIGN KEY (cart_user_id) REFERENCES shopping_cart(user_id),  
    FOREIGN KEY (cart_id) REFERENCES shopping_cart(cart_id),  
    FOREIGN KEY (product_id) REFERENCES products(product_id),
```

```

    UNIQUE (cart_user_id, cart_id, product_id)
) ENGINE = InnoDB;

```

Лістинг 2.5 Створення таблиці «orders»:

```

CREATE TABLE orders (
    order_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    order_created_date datetime NOT NULL,
    customer_firstName VARCHAR(255) NOT NULL,
    customer_lastName VARCHAR(255) NOT NULL,
    product_name VARCHAR(255) NOT NULL,
    product_cost INT NOT NULL,
    product_order_id INT NOT NULL,
    user_id INT NOT NULL,

    FOREIGN KEY (product_order_id) REFERENCES
products_in_order(prod_order_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id),

    UNIQUE (product_order_id, user_id)
)
ENGINE = InnoDB;

```

3 РОЗРОБКА ВЕБСАЙТУ «ПРОБЛЕМИ ЕКОЛОГІЇ»

3.1 Обґрунтування вибору середовища програмної реалізації

Для реалізації вебзастосунку в якості IDE були обрані IntelliJ IDEA та Visual Studio Code. Для розробки Front-end частини були використані наступні технології: HTML, CSS, JavaScript та фреймворк Bootstrap. Для розробки Back-end частини використовувалися технології Java EE та Spring Framework.

3.1.1 IntelliJ IDEA

Програмне забезпечення JetBrains IntelliJ IDEA є провідним середовищем швидкої розробки на мові Java. IntelliJ IDEA є високотехнологічним комплексом тісно інтегрованих інструментів програмування, що включає інтелектуальний редактор вихідних текстів з розвиненими засобами автоматизації, потужні інструменти рефакторингу коду, вбудовану підтримку технологій J2EE, механізми інтеграції з середовищем тестування Ant/JUnit та системами керування версіями, унікальний інструмент оптим Code Inspection та інноваційний візуальний конструктор графічних інтерфейсів.

Унікальні можливості JetBrains IntelliJ IDEA позбавляють програміста вантажу рутинної роботи, допомагають своєчасно усунути помилки і підвищити якість коду, піднімаючи продуктивність розробника на нову висоту [24].

Хоча середовище спочатку створювалося для максимальної оптимізації Java-розробки, зараз у ній є опції для роботи з більшістю затребуваних мов програмування, причому деякі інструменти використовують технологію

машинного навчання. IDE надає інтелектуальну допомогу під час написання коду:

- виконує глибокий аналіз та створює віртуальну карту проєкту;
- виявляє помилки та пропонує варіанти виправлення;
- автоматично доповнює код з огляду на контекст;
- проводить валідацію (перевірку на відповідність стандартам) коду;
- виконує рефакторинг коду – робить його простішим і зрозумілішим;
- підтримує роботу із вставками, написаними іншими мовами

програмування;

- дозволяє використовувати шаблони для вставки фрагментів коду, що повторюються;

- пропонує оптимізацію за допомогою профілювача – інструмента, який аналізує продуктивність коду та оцінює навантаження на процесор та оперативну пам'ять.

3.1.2 Visual Studio Code

Visual Studio Code (VS Code) – редактор вихідного коду, розроблений Microsoft для Windows, Linux та macOS. Позиціонується як «легкий» редактор коду для кросплатформної розробки веб- та хмарних програм. Включає в себе відладчик, інструменти для роботи з Git, підсвічування синтаксису, IntelliSense та засоби для рефакторингу. Має широкі можливості для кастомізації: теми користувача, поєднання клавіш і файли конфігурації. Поширюється безкоштовно, розробляється як програмне забезпечення з відкритим вихідним кодом [25].

3.1.3 BootStrap

Bootstrap – це відкритий та безкоштовний HTML, CSS та JS фреймворк, який використовується веброзробниками для швидкої верстки адаптивних дизайнів сайтів та вебзастосунків [26].

Bootstrap складається з:

- інструментів для створення макета (обгорткових контейнерів, потужної системи сіток, гнучких медіа-об’єктів, адаптивних утилітних класів);
- класів для стилізації базового контенту: тексту, зображень, коду, таблиць та figure;
- готових компонентів: кнопок, форм, горизонтальних і вертикальних навігаційних панелей, слайдерів, списків, акордеонів, модальних вікон, спливаючих підказок та ін;
- утилітних класів для вирішення традиційних завдань, що найбільш часто виникають перед веброзробниками: вирівнювання тексту, відображення та приховування елементів, завдання кольору, фону, margin і padding відступів, і т.д.

Переваги Bootstrap при його використанні для frontend розробки сайтів:

- висока швидкість створення якісної адаптивної верстки навіть початківцями веброзробниками (досягається це завдяки використанню готових класів та компонентів, створених професіоналами);
- кросбраузерність та кросплатформенність (коректне відображення та робота сайту у всіх підтримуваних цим фреймворком браузерах та операційних системах);
- наявність великої кількості готових добре продуманих компонентів, протестованих величезним співтовариством веброзробників на різних пристроях;
- можливість налаштування під свій проєкт, досягається це за допомогою зміни SCSS змінних та використання міксинів (можна змінити кількість колонок, кольору, радіус заокруглень, відступи між колонками тощо);

- низький поріг входження; для роботи з фреймворком не обов'язково мати «глибокі» знання з HTML, CSS, JavaScript та jQuery (досить знати лише основи цих технологій);
- однорідність дизайну та його узгодженість між різними компонентами (у Bootstrap всі компоненти виконані в єдиному стилі);
- наявність величезної кількості спільнот та навчальних матеріалів; при бажанні це допоможе не тільки добре розібратися у фреймворку, але і знайти відповіді практично на будь-які питання, що у вас виникають [27].

3.1.4 Spring Framework

Spring – найпопулярніше середовище розробки програм для корпоративної Java. Мільйони розробників по всьому світу використовують Spring Framework для створення високопродуктивного, легко тестованого та багаторазово використовуваного коду.

Spring Framework – це платформа Java із відкритим вихідним кодом. Спочатку його було написано Родом Джонсоном і вперше випущено під ліцензією Apache 2.0 у червні 2003 року.

Основні функції Spring Framework (рис. 3.1) можуть бути використані при розробці будь-якого Java-програми, але є розширення для створення вебдодатків поверх платформи Java EE. Платформа Spring призначена для спрощення використання J2EE-розробки та просування добрих методів програмування за рахунок використання моделі програмування на основі POJO [28].

Нижче наведено список деяких переваг використання Spring Framework:

- Spring дозволяє розробникам розробляти програми корпоративного класу з використанням POJO. Перевага використання лише POJO полягає в тому,

що вам не потрібен продукт EJB-контейнера, такий як сервер додатків, але у вас є можливість використовувати тільки надійний контейнер сервлета, такий як Tomcat, або комерційний продукт;

- Spring організований за модульним принципом. Незважаючи на те, що кількість пакетів та класів суттєво, вам потрібно турбуватися лише про ті, які вам потрібні, та ігнорувати все інше [29];

- тестування програми, написаної на Spring, дуже просто, тому що в цю середу переміщений код, що залежить від середовища. Крім того, завдяки використанню JavaBeanstyle POJO простіше використовувати впровадження залежностей для введення тестових даних;

- вебфреймворк Spring – це добре спроектований вебфреймворк, який надає відмінну альтернативу вебфреймворкам, таким як Struts або іншим надмірно спроектованим або менш популярним вебфреймворкам;

- Spring надає зручний API-інтерфейс для перетворення специфічних для технології виключень (наприклад, JDBC, Hibernate або JDO) у погоджені, неперевірені винятки;

- легкі контейнери IoC мають тенденцію бути легкими, особливо, порівняно, наприклад, з контейнерами EJB. Це корисно для розробки та розгортання програм на комп'ютерах з обмеженими пам'яттю та ресурсами ЦП;

- Spring надає узгоджений інтерфейс керування транзакціями, який може масштабуватись до локальної транзакції (наприклад, з використанням однієї бази даних) та масштабуватись до глобальних транзакцій (наприклад, з використанням JTA) [30].

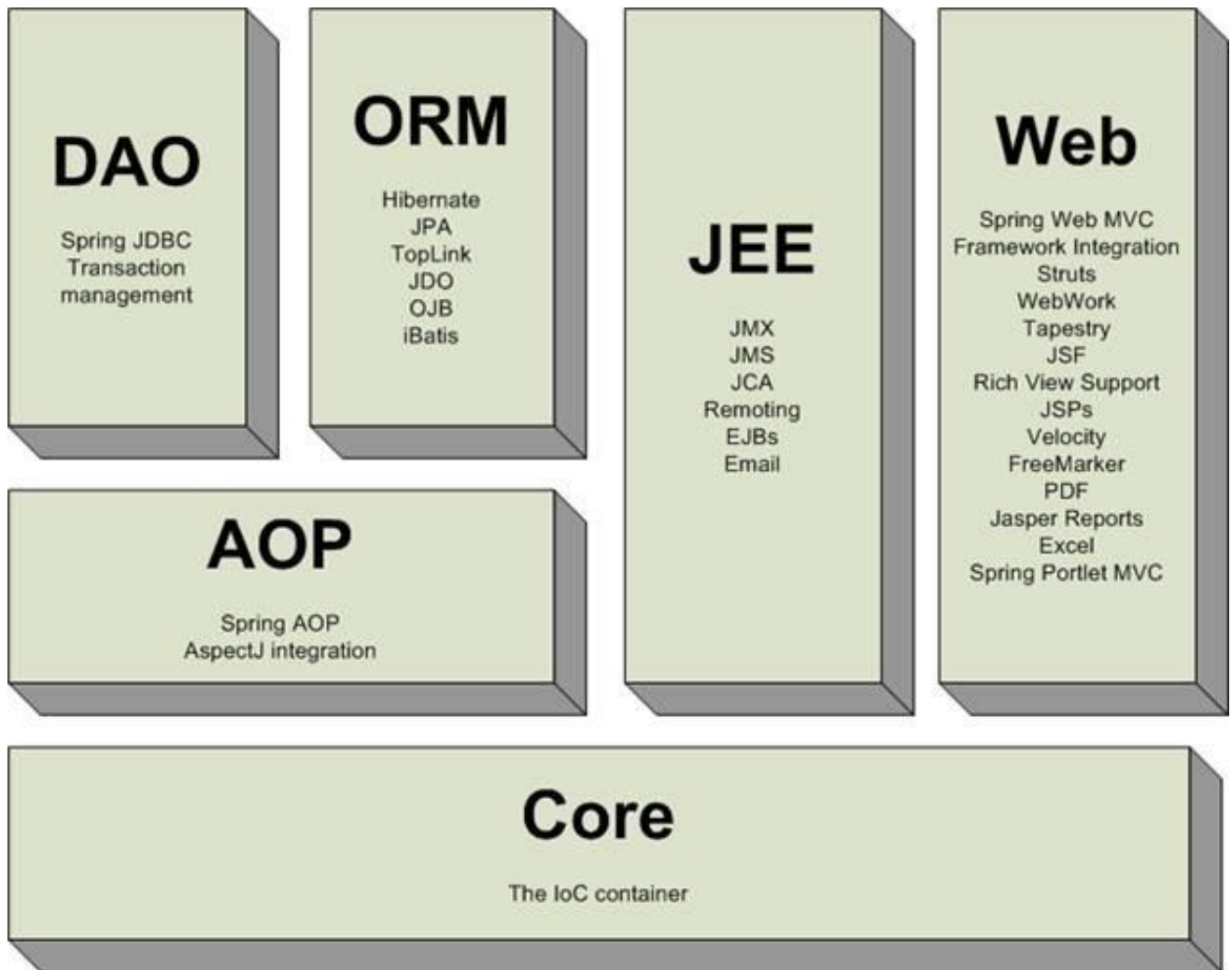


Рисунок 3.1 – Компоненти Spring Framework

3.2 Інструкція користувача та дорожня карта проєкту

При заході на початкову сторінку для незареєстрованого користувача є досить багато функціоналу. Він може вивчати всю інформацію на головній інформаційній сторінці та пограти на ній у міні-гру (рис. А.3). Після цього є можливість зайти на сторінку екомагазину, вибрати товар, що сподобався, і додати його в кошик. Щоб підтвердити замовлення, користувачеві необхідно авторизуватися, або, якщо у нього ще немає облікового запису – зареєструватися.

Існуючі замовлення можна відстежити на сторінці замовлень. Також адміністратор може авторизуватися та потрапити на сторінку адміністратора, з відповідним функціоналом (рис. 3.2).

В рамках вебсайту, що розробляється, існують наступні сторінки:

- головна сторінка з інформацією;
- сторінка авторизації користувача;
- сторінка реєстрації користувача;
- сторінка адміністратора;
- сторінка екомагазину;
- сторінка кошика;
- сторінка замовлень користувача.

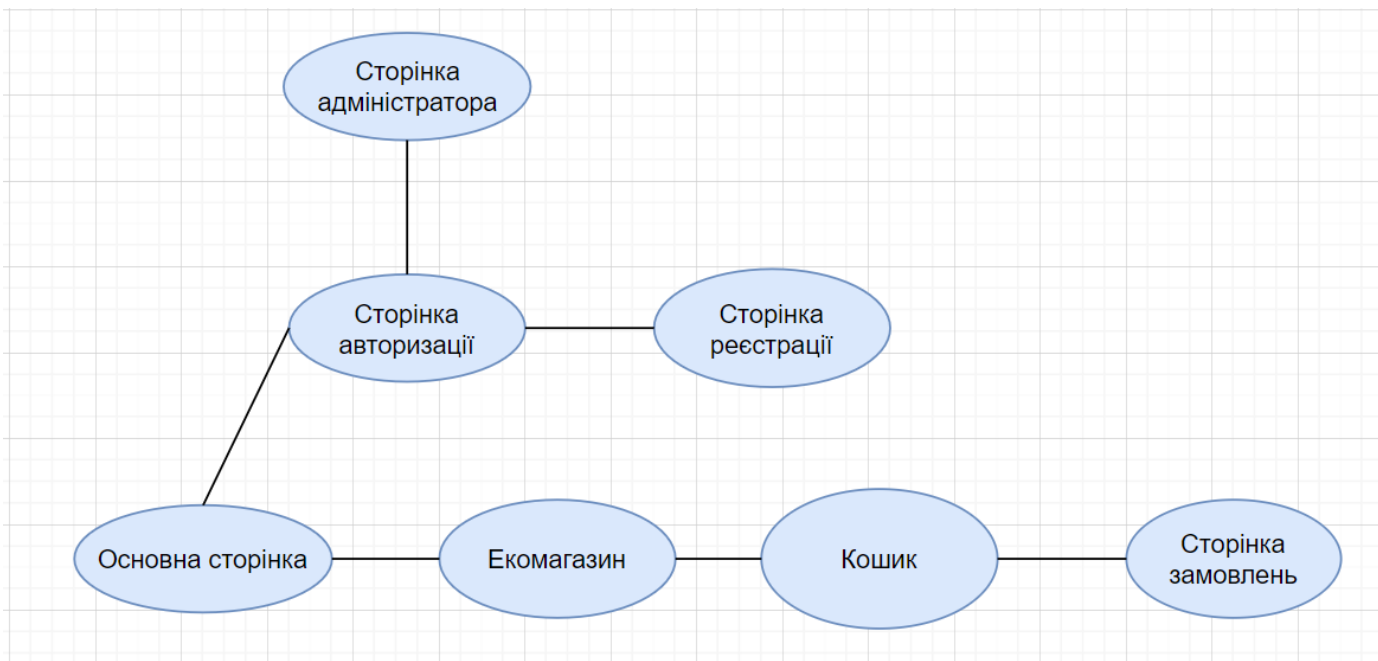


Рисунок 3.2 – Дорожня карта вебсайту «Проблеми екології»

3.3 Тестування розробленого вебсайту «Проблеми екології»

У цьому розділі наведено скріншоти (рис. 3.3 – 3.12) процесу роботи вебзастосунку:

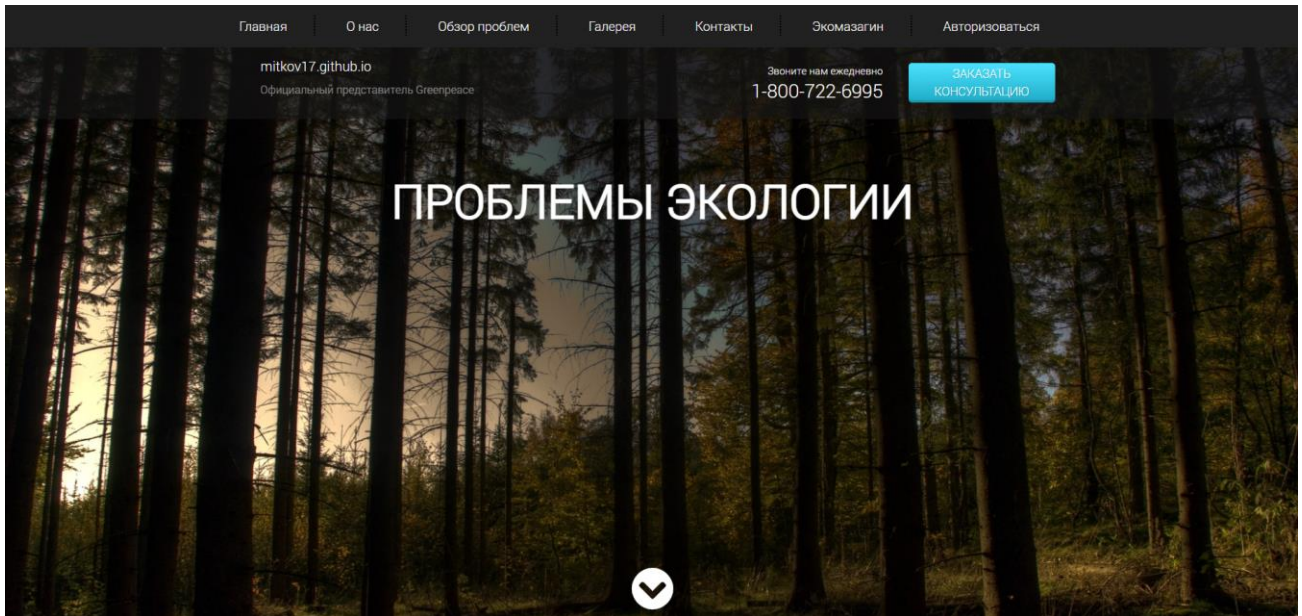


Рисунок 3.3 – Головной экран вебсайта «Проблеми екології»

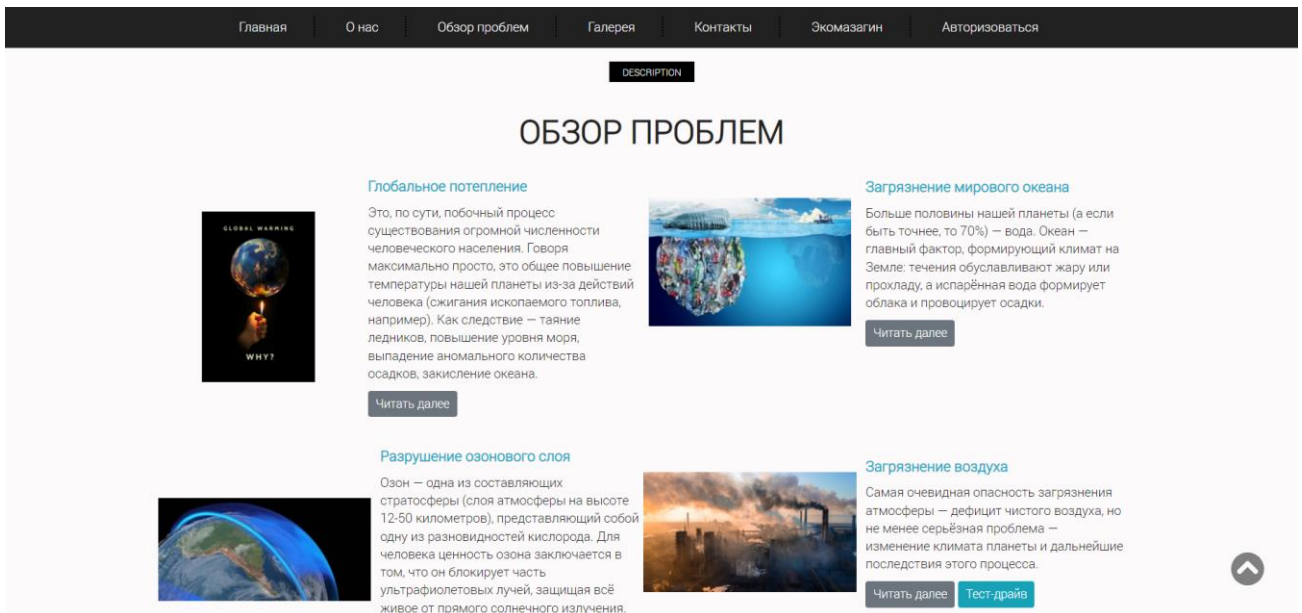


Рисунок 3.4 – Секция с обзором проблем экології

Главная	О нас	Обзор проблем	Галерея	Контакты	Экомагазин	Авторизоваться
ТАБЛИЦА С ХАРАКТЕРИСТИКАМИ ПРОДУКЦИИ						
Модель	Длина ручек	Цена(\$)	Плотность(г/м2)	Сумка выдерживает до(кг)		
White Eco Bag	60	70	135	15		
Black Eco Bag	60	107	260	20		
Green Eco Bag	70	99	205	17		
Eco Bag with a Pattern	60	210	320	20		
Black Eco Bag	60	210	210	10		
White Eco Bag	75	187	411	22		
Eco Bag with a Pattern	65	217	411	19		
White Eco Bag with a Pattern	65	120	350	18	Удалить	Изменить

Редактирование

Рисунок 3.5 – Секция з таблицею характеристик товарів екомагазину

Редактирование
×

Модель

Длина ручек

Цена(\$)

Плотность(г/м2)

Сумка выдерживает до(кг)

Добавить

Рисунок 3.6 – Модальне вікно додавання нового запису до таблиці характеристик

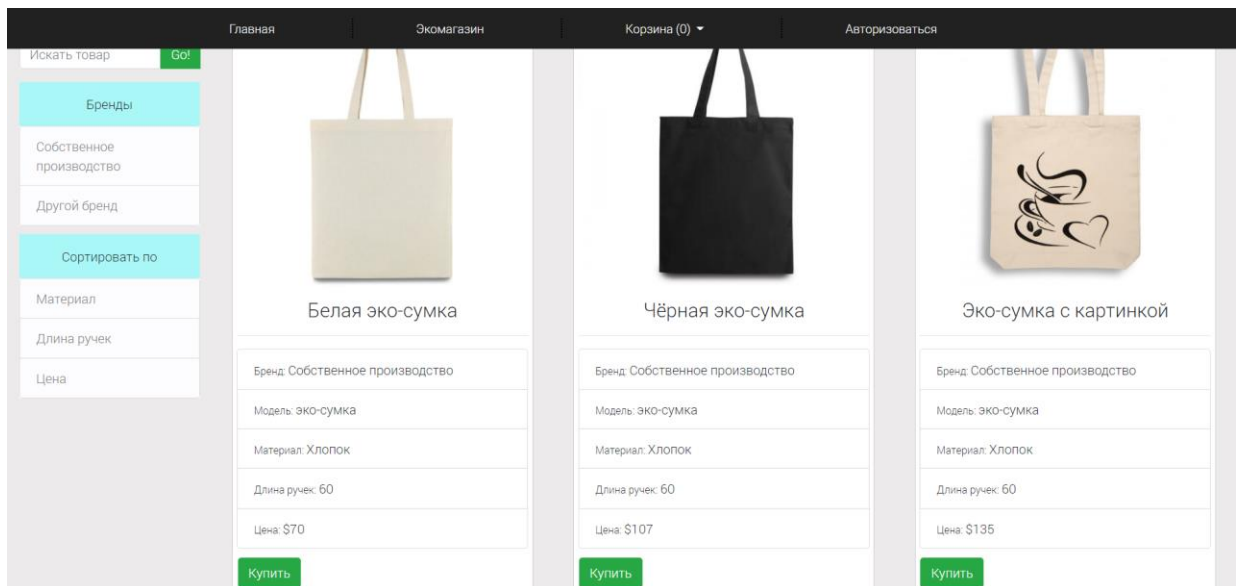


Рисунок 3.7 – Сторінка екомагазину з товарами

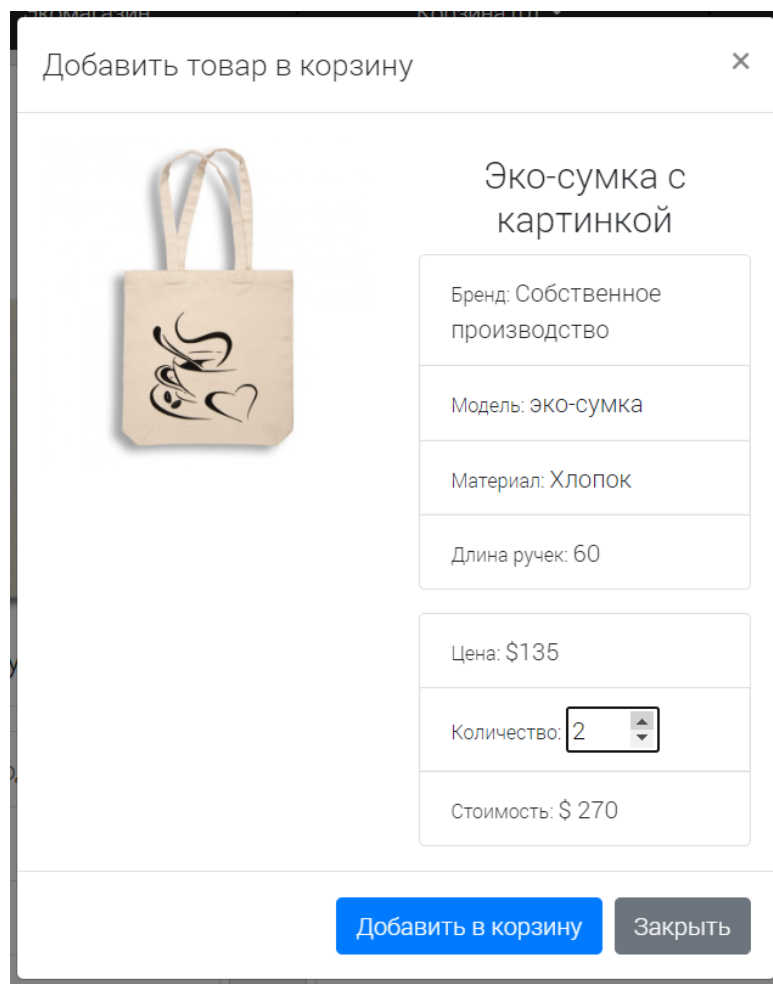


Рисунок 3.8 – Модальне вікно з додаванням товару до кошика

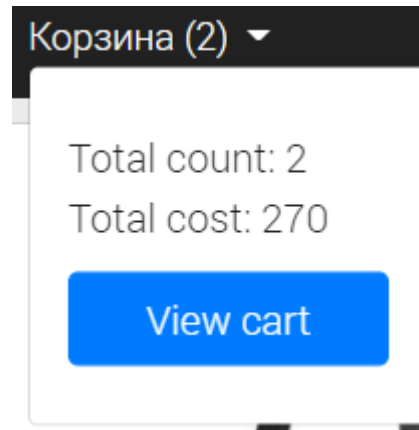


Рисунок 3.9 – Перехід до кошика

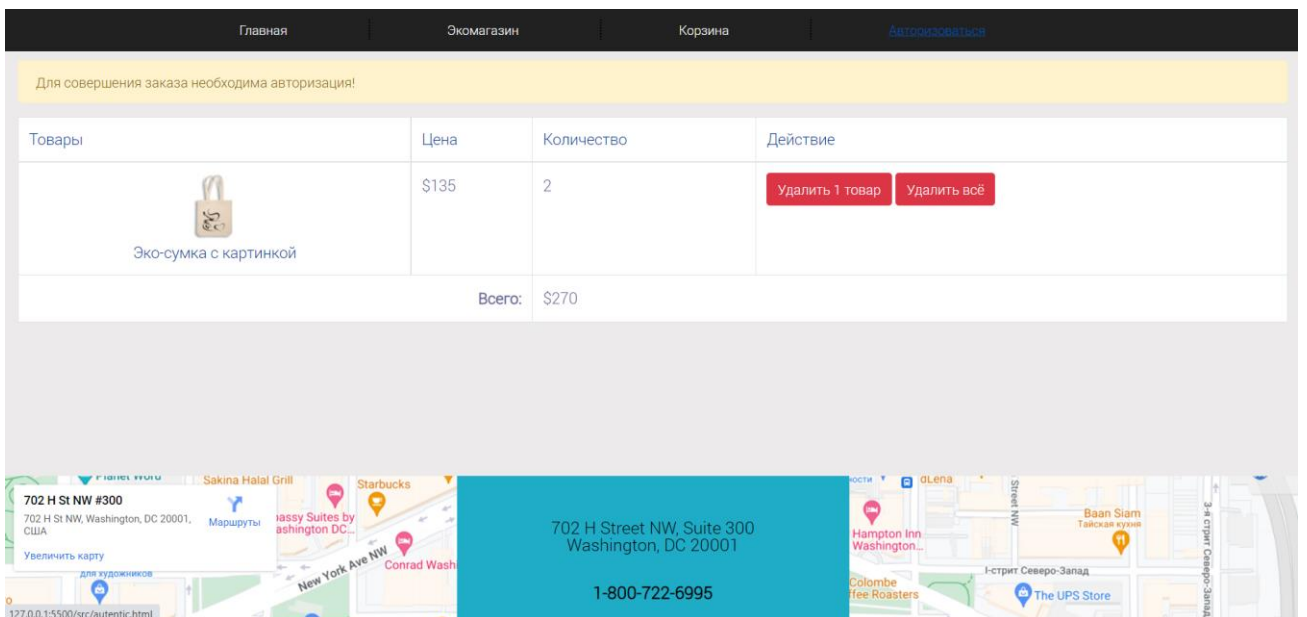


Рисунок 3.10 – Сторінка кошика від лиця незареєстрованого користувача

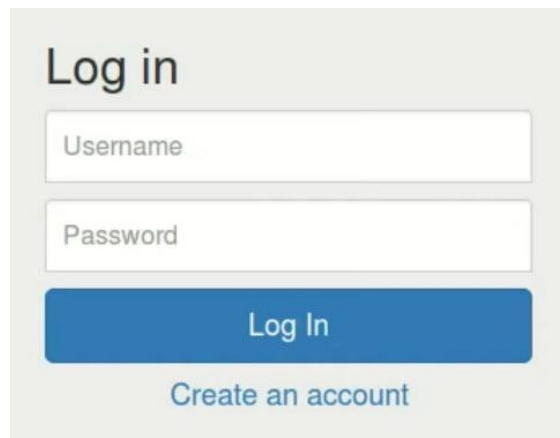


Рисунок 3.11 – Авторизація користувача

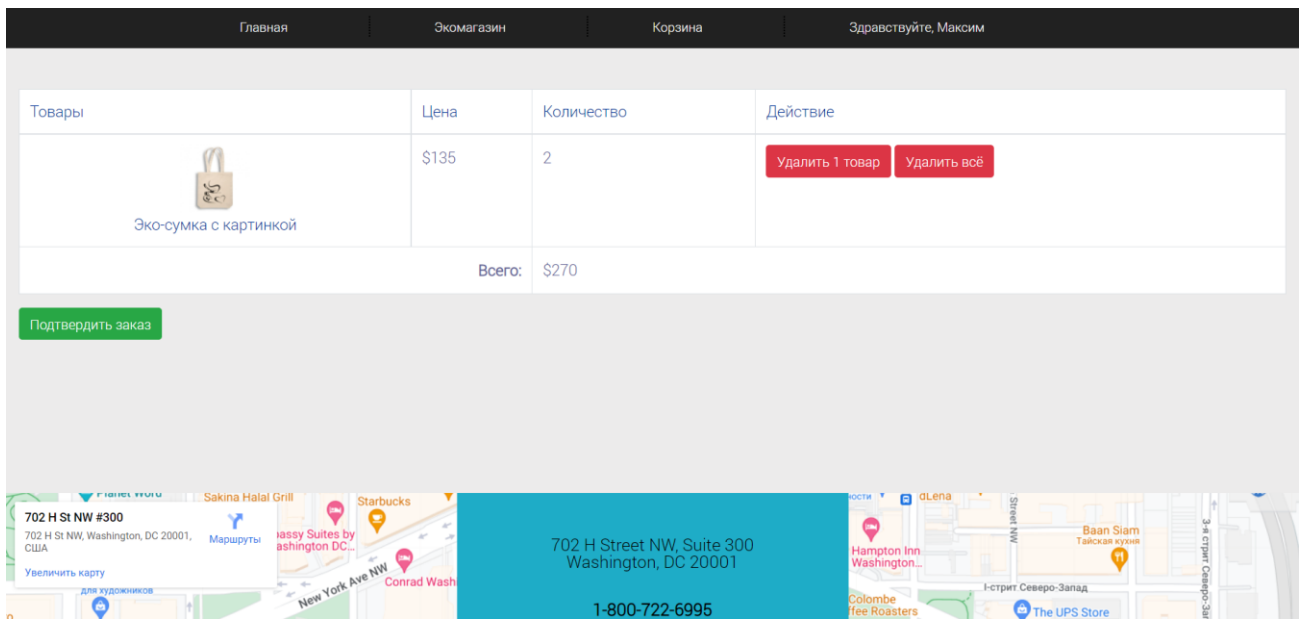


Рисунок 3.12 – Кошик від лица клієнта

3.4 Перспективи подальшої роботи

У процесі розробки вебсайту було виявлено варіанти для подальшого розвитку програми:

- додавання нових видів товарів у магазин;
- додати завантаження аватарок для користувачів;
- інтегрувати авторизацію у системі з соц. мережами;
- оптимізувати верстку для різних роздільних здатностей екрана;
- викласти сайт на реальний хостинг.

ВИСНОВКИ

В рамках кваліфікаційної роботи було змодельовано та реалізовано клієнт-серверний застосунок «Проблеми екології».

У процесі розробки було вивчено і втілено в реальність багато технологій розробки клієнт-серверних додатків, таких як HTML, CSS, JavaScript та фреймворк BootStrap для реалізації Front-end частини проєкту. Для реалізації Back-end частини проєкту було вивчено та використано різноманітні технології Java EE та, в особливості, технології Spring Framework: Spring Core, Spring MVC, Spring JDBC та Spring Security. Для проєктування бази даних проєкту було вивчено мову SQL та різноманітні реляційні системи управління базами даних. Для розробки застосунку був обраний MySQL.

Розроблений вебзастосунок дозволяє користувачам вивчати важливі сучасні проблеми екології та шляхи їх вирішення. Також, реалізований інтернет-магазин «екомагазин», як ще один шлях вирішення проблем екології – покупка екотоварів.

Систему було протестовано, та за результатами тестування були виявлені варіанти та перспективи подальшої роботи над сайтом та його розвитку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ю. Козміна, Р. Харроп, К. Шефер, К. Хо. (2019) Spring 5 для професіоналів = Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools, 1120 с.
2. Богданенко Д. А. (2018) Підходи до архітектурного проєктування веб додатків, 195 с.
3. Бекірова Е. А., Халілова З. Е. (2019) Основні етапи створення вебзастосунків (укр.) - Державна бюджетна освітня установа вищої освіти Республіки Крим "Кримський інженерно-педагогічний університет імені Февзі Якубова" (Сімферополь), С. 84-91. - ISSN 2658-5944.
4. Раджпут Д. (2019) Spring. Усі патерни проєктування, 320 с.
5. Лонг Д., Бастані До. (2019) Cloud Native Java: Designing Resilient Systems with Spring Boot, Spring Cloud, and Cloud Foundry., 624 с.
6. Волков А. С., Волкова К. А. (2019) Огляд архітектурних компонентів сучасного вебзастосунку, 961 с.
7. Jake Spurlock (2013) Bootstrap. Responsive Web-Development, 128 с.
8. David Cochran, Ian Whitley (2014) Bootstrap Site Blueprints., 304 с.
9. David Cochran. (2012) Twitter Bootstrap Web Development., 68 с.
10. Фрімен Ерік, Фрімен Елізабет (2010) Head First HTML with CSS & XHTML., 656 с.
11. Пітер Лабберс, Брайан Олберс, Френк Салім (2011) HTML5 для професіоналів: потужні інструменти для розробки сучасних вебдодатків., 272 с.
12. Макфарланд, Девід. (2015) JavaScript та jQuery: вичерпне керівництво., 880с.
13. Герберт Шілдт. (2018) Java. The Complete Reference., 1488 с.

14. Кей С. Хорстман. (2017) Core Java. Volume II - Advanced Feature (Tenth Edition)., 976 с.
15. Стівен Шафер. (2011) HTML, XHTML, та CSS Bible., 656 с.
16. Ніксон Р. (2016) Створюємо динамічні вебсайти за допомогою PHP, MySQL, JavaScript, CSS та HTML5. 4-те вид., 768 с.
17. Peter Shaw. (2014) witter Bootstrap 3 Succinctly, 110 с.
18. Джеймс Р. Грофф, Пол Н. Вайнберг, Ендрю Дж. Оппель. (2014) SQL: The Complete Reference, Third Edition., 960 с.
19. Кріс Фіайлі. (2003) SQL: Посібник з вивчення мови., 456 с.
20. К. Дж. Дейт. (2005) Введення у системи баз даних, 1328 с.
21. Мік Кнутсон, Пітер Муларьєн, і Роберт Вінч (2017) Spring Security - Third Edition, 742 с.
22. Нанда Натімутху (2016) Spring Security Essentials, 560 с.
23. Лаурентіу Спілка (2020) Spring Security in Action, 824 с.
24. Сет Ледд (2016) Expert Spring MVC and Web Flow, 710 с.
25. Жоффруа Уорин (2015) Mastering Spring MVC 4, 520 с.
26. Мадхусудхан Конда (2012) Just Spring Data Access, 650 с.
27. Петрі Кайнулайнен (2012) Spring Data, 587 с.
28. Зайцев Петро, Ткаченко Вадим, та Шварц Берон (2018) MySQL по максимуму. 3-тє видання, 864 с.
29. Чинтан Мехта, Кандарп Патель, Шаббір Чаллавала, Джадіп Лакхатарія (2022) MySQL 8 для великих даних, 228 с.
30. Ларі Ульман (2022) MySQL: Посібник з вивчення мови, 354 с.