

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Web-додаток підтримки замовлення систем розумний будинок

Здобувача групи ІТ-92-1/2 _____ Троценка Дмитра Сергійовича _____
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Дмитро ТРОЦЕНКО _____
(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, канд.техн.наук, доцент Юлія ПАРФЕНЕНКО
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) _____
(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ІТ

_____ Світлана ВАЩЕНКО

«_____» _____ 2023 р.

З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Троценко Дмитро Сергійович

1 Тема роботи Web-додаток підтримки замовлення систем розумний будинок

керівник роботи Парфененко Юлія Вікторівна, к.т.н., доцент,

затверджені наказом по університету від «29» травня 2023 р. №0588-VI

2 Строк подання студентом роботи «07» червня 2023 р.

3 Вхідні дані до роботи фото компонентів систем розумний будинок та систем розумний будинок, дані про їх технічні характеристики

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування web-додатку підтримки замовлення систем розумний будинок, розробка web-додатку підтримки замовлення систем розумний будинок

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 8 лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Планування робіт	11.11.2022 - 21.12.2022	
2	Оформлення технічного завдання	21.12.2023 - 27.12.2023	
3	Проведення аналізу предметної області	16.01.2023 - 27.01.2023	
4	Проведення проектування web-додатку	28.01.2023 - 21.02.2023	
5	Розробка web-додатку	06.03.2023 - 30.05.2023	
6	Тестування web-додатку	17.05.2023 - 30.05.2023	
7	Завантаження web-додатку на хостинг	19.05.2023 - 22.05.2023	
8	Оформлення пояснювальної записки	16.03.2023 - 05.06.2023	

Студент

(підпис)

Дмитро ТРОЦЕНКО

Керівник роботи

(підпис)

к.т.н., доц. Юлія ПАРФЕНЕНКО

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток підтримки замовлення систем розумний будинок».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку літератури із 23 найменувань, 4 додатків. Загальний обсяг складає 117, із яких 68 с. – основний текст, 53 рисунки, 11 таблиці.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку підтримки замовлення систем розумний будинок.

Під час роботи над першим розділом було досліджено предметну область підтримки замовлень систем розумний будинок. В ній існують такі проблеми, як незначна розповсюдженість фізичних магазинів, вибір серед приладів аналогів, а також складність підбору даних систем. Також було обрано засоби реалізації додатку, визначено мету та задачі, була виконана постановка задачі.

У другому розділі виконано проектування web-додатку підтримки замовлення систем розумний будинок. Створено діаграми варіантів використання розробки, а також фізичну модель бази даних. Виконано моделювання основного процесу web-додатку в нотаціях IDEF0. Побудовано дерево прийняття рішень для вибору систем розумний будинок.

У третьому розділі було представлено архітектуру для web-додатку. Також в ньому міститься інформація про процеси програмної реалізації та демонстрацію роботи web-додатку. В останньому підрозділі наведено результати тестування правильності роботи посилань та кнопок web-додатку.

Результатом роботи є web-додаток підтримки замовлення систем розумний будинок, який дозволить спростити процеси вибору розумних приладів їх систем.

Ключові слова: web-додаток, системи розумний будинок, дані, html, css, моделювання, розробка.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Огляд останніх досліджень і публікацій	9
1.2 Аналіз програмних продуктів - аналогів	15
1.3 Постановка задачі	20
1.4 Вибір засобів реалізації з розроблення web-додатку	21
2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ ПІДТРИМКИ ЗАМОВЛЕННЯ СИСТЕМ РОЗУМНИЙ БУДИНОК.....	23
2.1 Структурно-функціональне моделювання web-додатку підтримки замовлення систем розумний будинок в IDEF0.....	23
3.2 Моделювання варіантів використання	28
3.3 Моделювання послідовності оформлення замовлення.....	31
3.4 Проектування дерева рішень	33
3.5 Проектування бази даних.....	34
3 РОЗРОБКА WEB-ДОДАТКУ ПІДТРИМКИ ЗАМОВЛЕННЯ СИСТЕМ РОЗУМНИЙ БУДИНОК.....	41
3.1 Архітектура web-додатку підтримки замовлення систем розумний будинок.....	41
3.2 Програмна реалізація web-додатку	42
3.3 Демонстрація роботи web-додатку	47
3.4 Тестування web-додатку.....	62
ВИСНОВОК.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68

Додаток А.....	71
Додаток Б.....	84
Додаток В.....	98
Додаток Г.....	99

ВСТУП

Сьогодні суспільство дуже активно розвивається у всіх напрямках, але надзвичайні зусилля витрачаються на науково-технічний прогрес. Робиться це з очевидною метою – зробити життя кращим та простішим. Дане завдання можна виконувати різними способами та різних сферах людської діяльності, наприклад, в місцях нашого проживання – будинках та квартирах.

Після пандемії всі знову почали розуміти, наскільки домівка є важливою. У цьому місці люди зараз проводять дуже багато часу у зв'язку з карантинном та переходом багатьох компаній на віддалений режим роботи через нестабільний фінансовий стан у всьому світі. Тому було б розумно зробити перебування у своїй оселі максимально зручним, безпечним й ефективним. Саме для цього уже давно було започатковано напрям під назвою розумний будинок, що означає використання новітніх технологій для того, щоб зробити проживання в домівці комфортнішим. Так, мова зараз йде про розумні світильники, чайники, пилососи, системи опалення, захисту та багато інших речей, які працюють в одній мережі й доступні для управління за допомогою смартфона. Беззаперечно, це є актуальним та потрібним напрямом, товари якого мають попит на ринку.

На цей час часу існує ще одна тенденція, а саме перехід від фізичних угод в магазинах, до тих, що відбуваються в мережі Інтернет. Людині все частіше продають товари за межами стін торговельних павільйонів. Компанії робляться це в межах всього інформаційного простору. Так само покупка вже не здійснюється в певному приміщенні, куди потрібно прийти за товаром, а на сторінках інтернет-магазинів.

На стику цих трендів маємо проблему. Для якісного та максимально ефективного розповсюдження систем розумний будинок потрібно розробляти web-додаток підтримки замовлень таких систем. Це зробить даний напрям доступнішим для більшої маси людей з інформаційної точки зору. Адже вони зможуть переглядати та купувати товари, не виходячи зі своєї оселі, перебування в якій може стати ще комфортнішим. Також це розширить розуміння того, на що здатні інструменти даного

підходу до господарства та підвищить їх популярність, що у свою чергу сприятиме розвитку таких технологій. Саме тому було прийнято рішення створити web-додаток підтримки замовлень систем розумний будинок.

Метою кваліфікаційної роботи бакалавра є розробка web-додатку підтримки замовлення систем розумний будинок, який допоможе користувачам оптимізувати процес вибору розумних приладів та систем розумний будинок.

Для досягнення мети потрібно вирішити задачі:

- визначити актуальність розробки;
- дослідити предметну область;
- провести аналіз аналогів;
- визначити основні вимоги до web-додатка;
- розробити архітектуру web-додатку;
- виконати моделювання web-додатку;
- розробити функціонал web-додатку та провести його тестування.

Практичним значенням роботи є те, що за допомогою web-додатку можна буде як замовити окремі компоненти, так і переглянути типову конфігурацію систем розумного будинку та здійснити підбір конфігурації залежно від потреби користувача.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Системи розумний будинок є складовою технології Інтернет речей – індустрії, основним принципом якої є взаємодія різних пристроїв між собою за допомогою мережі Інтернет. Дані прилади, зазвичай, обладнані датчиками для вимірювання параметрів зовнішнього середовища та процесорами для обробки даних, а система загалом здатна приймати рішення відповідно до інформації, отриманої з оточення [1].

Якщо поставити питання, чому саме цей напрям є достатньо актуальним, відповідь буде достатньо очевидною. Системи розумного будинку містять дуже корисні, ергономічні, перспективні та прибуткові товари та технології. Дослідження даного ринку від різних компаній в Америці та Європі показує, що кількість домогосподарств, які можна віднести до групи «розумних» постійно зростає, як і прибутки, що отримуються від їх розповсюдження [2].

Досить складно знайти подібні дослідження, які стосувалися б виключно України, проте можна знайти наукові статті стосовно цієї тематики, в яких досліджуються перспективи розвитку Інтернету речей в державі. Загалом, вони вказують на підвищення економічних показників у разі успішного запровадження даних технологій у повсякденне життя людей та економіку країни в цілому. Хоча сам процес змін буде досить складним і може нести певні ризики для економіки та інформаційної безпеки держави [3].

Зміни в країні можуть відбуватися досить довго, але вже сьогодні є кілька представників, які займаються комерційною діяльністю в даному напрямку.

Дана ніша вже існує певний час на ринку та з кожним днем набирає потенційної сили, вивільнення якої, призведе до збільшення популярності даного сектору ринку, до того ж конкуренція доволі стримана тому є можливість зайняти лідерські позиції використовуючи ефективні підходи для ведення бізнесу.

Системи розумного будинку сам по собі є корисними та в перспективі несуть цінність для всіх людей, тому що можуть зробити їх життя значно простішим, позбавивши від набридливих, повсякденних побутових обов'язків.

Слід зазначити, що в даній предметній області є декілька взаємопов'язаних проблем. Оскільки продукт досить дорогий, це значить, що він не дуже розповсюджений. Скоріше за все, вийшовши на вулицю, не вдасться знайти магазин, який займається реалізацією даних систем. На додаток, створення правильної системи може виявитися досить складним завданням, а оскільки немає фізичного магазину, то немає і консультанта, який пояснить, яку комплектацію необхідно придбати для задоволення тих чи інших потреб. Навіть, якщо є потреба обрати не складну систему, при цьому, маючи розуміння як це працює, доведеться вибирати з кількох варіантів аналогів. У такі моменти виявиться ще одна, дуже розповсюджена в наш час задача, пов'язана з витратами часу на вибір між аналогами.

Оптимальним підходом до вирішення даних проблем є створення web-додатку підтримки замовлення систем розумного будинку. Створити такий продукт відносно не складно, а скористатися ним зможе багато людей, адже він підтримується різними пристроями від смартфонів до розумних телевізорів, незалежно від операційної системи. Враховуючи специфіку продукту, інші способи реалізації можуть обмежити кількість користувачів або збільшити витрати на розробку та підтримку.

Сам процес створення web-додатку також може відрізнятись. Адже існує багато варіантів їх побудови, починаючи від особливостей архітектури, закінчуючи мовами написання програмного коду.

Для вибору найкращого способу реалізації продукції, слід ознайомитися із літературними джерелами, де детально описано процес розробки аналогічних web-додатків. Вони повинні бути орієнтовані на комерційну діяльність, полегшення процесу оформлення замовлень клієнтами та їх подальшої обробки адміністрацією. Цікавим підходом є використання web-додатку разом з ERP (Enterprise Resource Planning) системою, яка допомогла б в управлінні внутрішніми ресурсами магазину. Найбільш проста реалізація - розробка спільної бази та backend частини для обох складових. Схема даного підходу зображена на рисунку 1.1 [4].

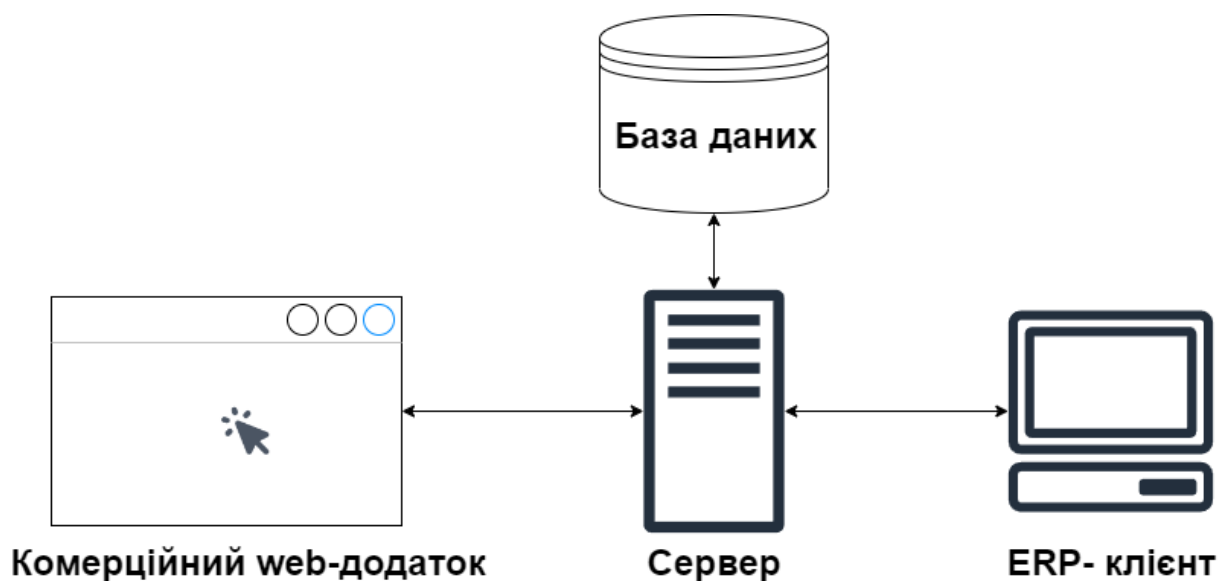


Рисунок 1.1 – ERP та web-додаток зі спільним сервером та базою даних

Проте, навіть у такому випадку збільшаться витрати на розробку. Тому було прийнято рішення зосередитися виключно на web-додатку і не створювати повноцінної ERP системи.

Методологія розробки також є дуже важливою частиною і визначає інструменти створення продукту та спосіб їх взаємодії. У побудові систем онлайн замовлень можна зустріти стек MERN.

MERN – це аббревіатура стека технологій, які використовуються при розробці web-додатків. До нього входять перелічені нижче інструменти розробки.

MongoDB – кросплатформенна програма з відкритим кодом, призначена для роботи з NoSQL базами даних, де використовують JSON-подібні документи з необов'язковими схемами. Документ є найпростішим типом даних у MongoDB, а структура записів складається з полів і значень.

Express – гнучкий фреймворк для Node.js, за допомогою якого можна створювати web-додатки та API. Має вбудовану систему шаблонів, а також надає можливість керування маршрутизацією.

ReactJS – бібліотека для створення інтерфейсу користувача. Базується на принципі компонентів – малих ізольованих частинах коду. Це робить процес розробки декларативним, гнучким і ефективним [5].

Node.js – середовище для написання серверної частини web-додатків мовою JavaScript. Це одне з найбільш поширених середовищ з відкритим кодом, яке можна запустити на різних платформах (Linux, Windows, macOS) [6].

За frontend частину в даному підході відповідає ReactJS, за backend ExpressJS та NodeJS, а за зберігання даних MongoDB. Перевагою такого підходу є використання єдиної мови JavaScript на всіх рівнях роботи web-додатку. Схематичне зображення стека MERN зображено на рисунку 1.2 [7].

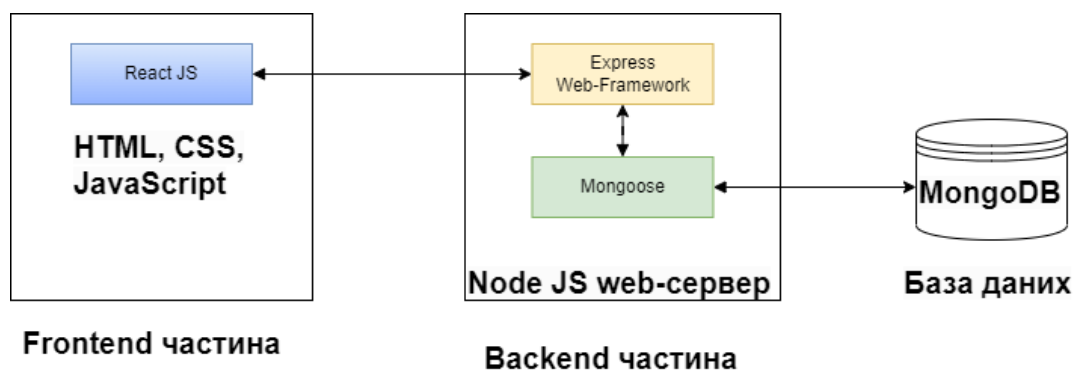


Рисунок 1.2 – Схематичне зображення стеку MERN

LNPM/LAPM – інший стек, який використовується для побудови web-додатків. Кожен елемент описано нижче, а схема зображена на рисунку 1.3.

Linux – операційна система, яка є середовищем виконання web-додатку.

Nginx/Apache – високопродуктивні HTTP-сервери.

PHP – це мова програмування, яка використовується на стороні сервера, взаємодіє з базою даних та може вбудовуватися до HTML документів.

MySQL – безкоштовне програмне забезпечення для керування реляційними базами даних з відкритим кодом, яке підтримує такі мови програмування як PHP, Java, Python та інші. Може бути встановленим на локальному сервері, що дозволяє використовувати його для розробки програмного забезпечення та web-додатків [8].

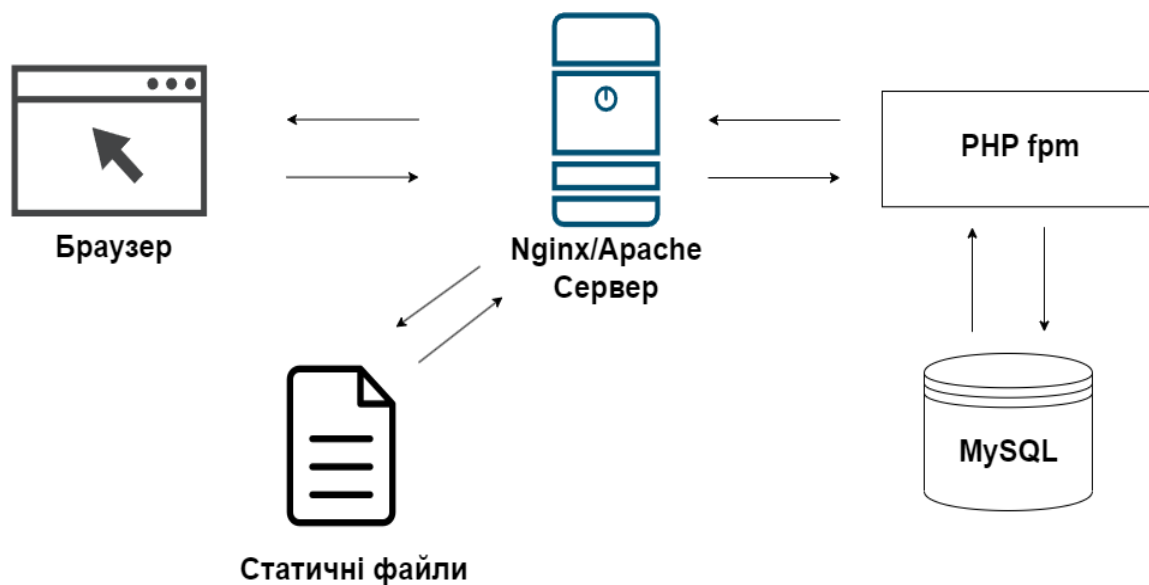


Рисунок 1.3 – Схематичне зображення стеку LNPM/LAPM

Даний стек описує методологію в загальному вигляді та акцентує увагу на backend частині. Для PHP коду можна, наприклад, використати Laravel фреймворк. Вибір використання чи не використання бібліотек та фреймворків залишається за розробником. Потрібно пам'ятати, що можна написати додаток і без знань додаткових технологій, вони, зазвичай, слугують лише для прискорення та спрощення процесу розробки. В літературі можна знайти приклади написання frontend частини лише за допомогою HTML, CSS та JavaScript [9].

Аналіз всіх вище описаних підходів дозволяє сформуванню загального бачення можливих варіантів створення архітектури проектів, надаючи конкретні приклади. Слід відмітити, що при написанні власного web-додатку краще відштовхуватися від конкретних потреб і приймати рішення, які будуть оптимальними в даному проекті. Враховуючи простоту підходу LNPM та відносно не складну, з технічної точки зору, ідею даної роботи, можна скористатися саме ним. При цьому не використовуючи ніяких бібліотек, що все ще дозволить реалізувати повноцінний продукт і при цьому вимагатиме лише базових знань HTML, CSS та JavaScript, PHP та MySQL.

Якщо стосовно інструментів та методології розробки все зрозуміло, тепер слід перейти до аналізу наповнення web-додатків, орієнтованих на онлайн замовлення

продуктів та послуг. Можна відмітити кілька тенденцій в структурі існуючих представників

Першою особливістю є наявність системи менеджменту користувачів, тобто розділення прав доступу та можливостей відвідувачів сайту відповідно до їх статусу. Наприклад, зареєстровані користувачі можуть переглядати деталі своїх замовлень, їх статус та власні персональні дані. Для реалізації даної особливості звичайно необхідно подбати про можливість реєстрації, авторизації та наявності профілю користувача [10].

Також слід відмітити, що відсутність повноцінної ERP, яка могла б впливати на наповнення сайту, змушує реалізувати новий інструмент, який дозволить адміністрації швидко змінювати інформацію про продукти без знань мов програмування. Мається на увазі графічна адміністративна панель. Її розробка значно простіша, ніж ERP, але при використанні даного інструменту, керування контентом стає значно ефективнішим і доступнішим.

У даному модулі слід також вмістити систему обробки заявок, яка відображатиме всі необхідні дані про статус замовлення. У деяких випадках адміністративна панель містить дані статистики для аналізу маркетингової складової web-додатку, але її якість також впливатиме на складність розробки всієї системи.

Зручним для користувачів є функція додавання в кошик та подальше створення замовлення саме з цього елемента інтерфейсу. Інколи при створенні web-додатку окремо виділяється інформація про доставку товару, де вказуються дані про кур'єра та місце відправки та видачі замовлення.

Загальна схема типового web-додатку підтримки замовлення зображена на рисунку 1.4. В ній враховано більшість необхідних функцій, реалізація яких, забезпечить базовий функціонал, і як наслідок, працездатність розробки [11].

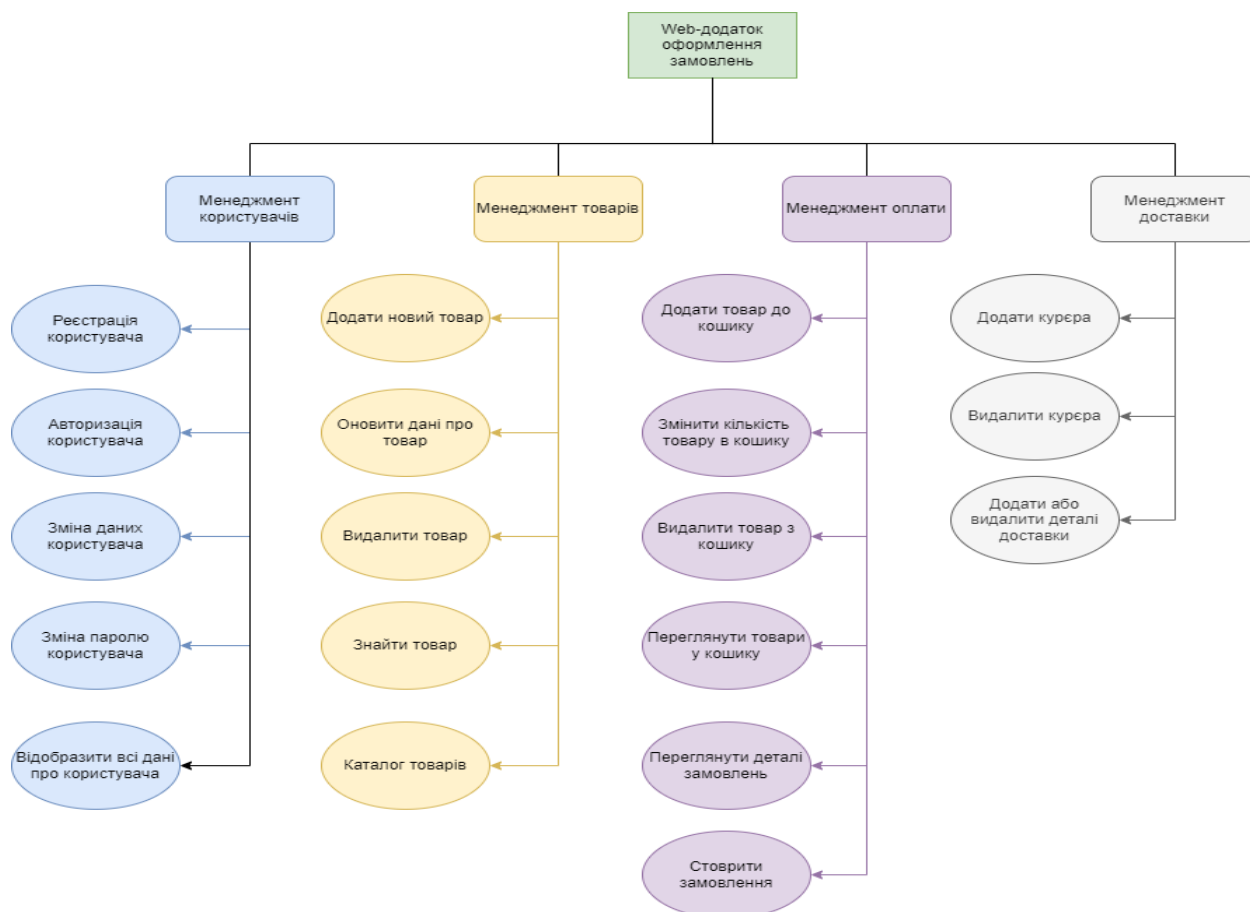


Рисунок 1.4 – Схематичне зображення типового web-додатку підтримки замовлення

У результаті роботи проведеної в даному підрозділі було дано загальну характеристику предметної області. Це дозволило виявити наявні в ній проблеми, а також визначитися з первинними потребами до продукту та підходом до його розробки.

1.2 Аналіз програмних продуктів - аналогів

При попередньому аналізі предметної області було розглянуто підходи до створення web-додатку підтримки замовлень, не прив'язаних до конкретної тематики.

У даному розділі наведено порівняльну характеристику аналогічних продуктів, які спеціалізуються на приладах та повноцінних системах розумний будинок.

Знайдено та досліджено декілька сайтів, які можна вважати схожими до розроблюваного web-додатку підтримки замовлення систем розумний будинок. Для створення конкурентоспроможного та корисного додатку, який зможе покрити всі або хоча б більшість потреб, необхідно виділити переваги та недоліки знайдених представників.

ilounge.ua – перший аналог, що має сучасний дизайн і досить великий перелік функцій. В наявності каталог товарів, їх класифікація на категорії та підкатегорії, а також можливість сортувати їх в певному порядку та фільтрувати за визначеним переліком параметрів. Магазин заохочує купувати продукти саме у них, вводячи систему бонусів, для використання якої необхідно зареєструватися. Також є кошик для товарів, автоматичне замовлення, розділ за акційними товарами та інформаційний блог орієнтований на технічну тематику. Сторінка каталогу сайту зображена на рисунку 1.5 [12].

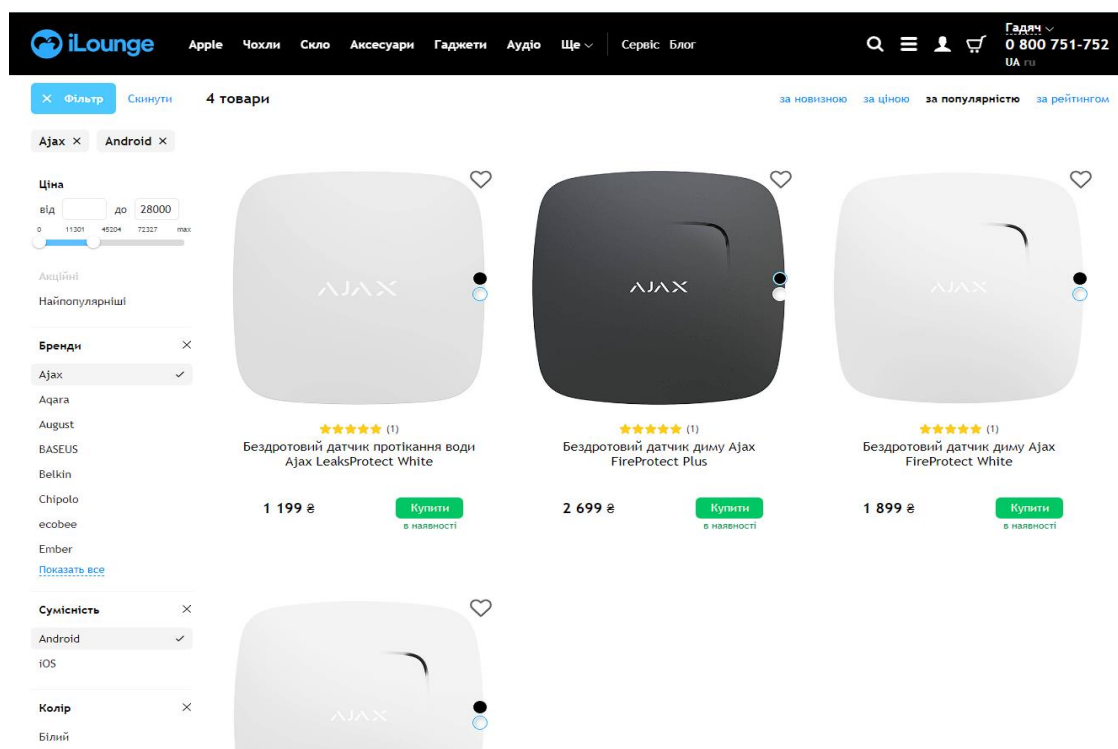


Рисунок 1.5 – Зображення сторінки каталогу аналога ilounge.ua

domos.ua – другий аналог, дизайн якого має направленість виключно на тематику розумних будинків. Відразу можна помітити відсутність можливості для створення облікового запису. Поділ на категорію реалізований. Сортування та фільтри відсутні, проте можна знайти товар через пошукове поле. Цікавою особливістю є приклади готових наборів, приклад сторінки із їх зображенням можна переглянути на рисунку 1.6 [13].

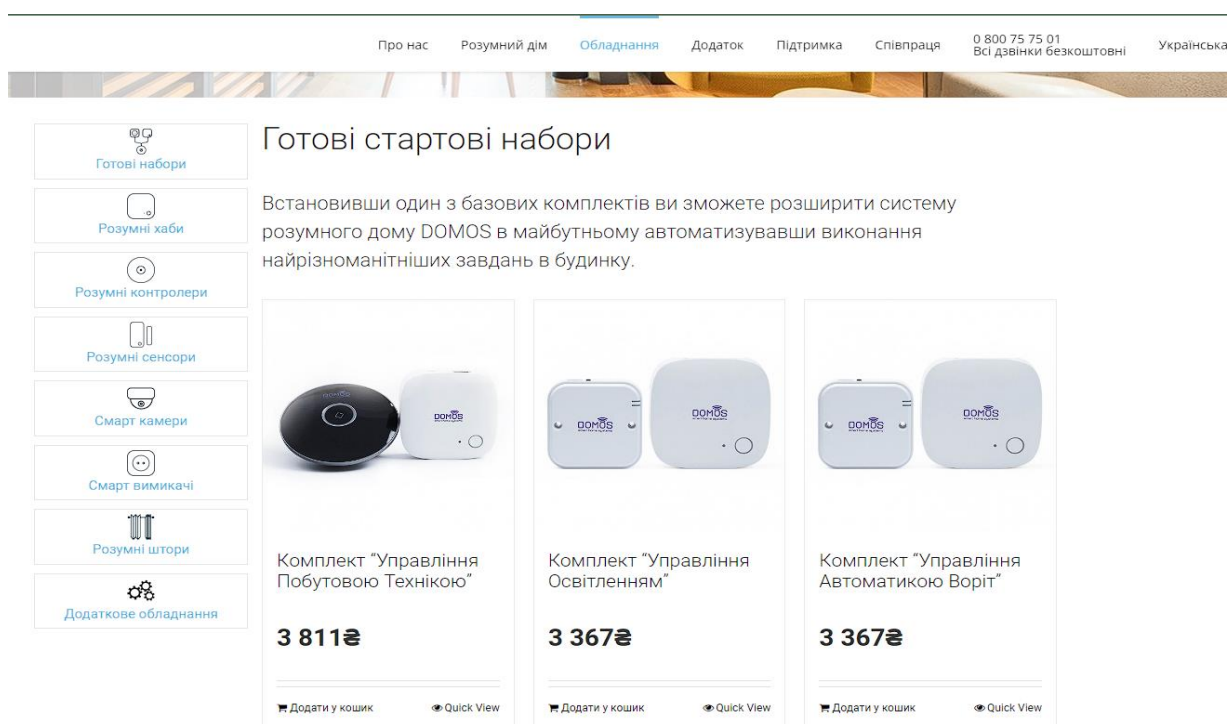


Рисунок 1.6 – Сторінка із готовими наборами аналога domos.ua

xterm.com.ua – ще один аналог, який не має специфічних функцій, які б виділяли його серед інших сайтів. Він має можливість автоматичного оформлення замовлення, інтерфейс зображено на рисунку 1.7. Також є кошик та можливість створити обліковий запис. Реалізовано каталог товарів з повноцінним функціоналом для вибору необхідних приладів. Наявна функція порівняння товарів, але вона працює не досить якісно [14].

Останнім досліджуваним аналогом є inteldim.lviv.ua. Сайт має застарілий дизайн. Інформації про системи та прилади дуже багато, їх опис досить детальний. Але даний сайт не призначений для автоматичного створення замовлень. Також

відсутня можливість купити прилади окремо від системи. Особливістю є форма для відправки параметрів свого будинку та бажаної системи, на основі яких буде проведено попередній розрахунок вартості системи, хоча підібраний набір в такому випадку можна вважати якісним, але він не буде автоматичним [15].

The screenshot shows the checkout page of the Xterm website. At the top, there is a navigation bar with links like 'Про магазин', 'Доставка та оплата', and 'Новини'. The main header features the Xterm logo and a search bar. Below the header, there are several category links: 'Розумний будинок', 'Тепла підлога', 'Клапани та приводи 2, 3, 4-х ходові', 'Автоматика', 'Запірна арматура', 'Контрольно-вимірювальні прилади', 'Насосні групи', 'Фільтри', and 'Автономне живлення'. The main content area is titled 'Оформлення замовлення' and contains a form for entering customer details (name, email, phone, address) and delivery options. On the right, a 'МОЄ ЗАМОВЛЕННЯ' section displays the items in the cart: a magnetic backflow filter (x4, 15744 UAH) and a thermostat (x2, 4526 UAH). The total amount to pay is 20270 UAH. A red button at the bottom says 'Оформити замовлення'.

Рисунок 1.7 – Сторінка оформлення замовлення аналога xterm.com.ua

The screenshot shows a form titled 'Замовлення попереднього розрахунку системи "Розумний дім"'. The form is in Ukrainian and includes the following fields and sections:

- Ваше ім'я***: Text input field.
- E-mail:***: Text input field.
- Текст**: Large text area for comments or additional information.
- Тип об'єкту***: Dropdown menu.
- Керування:** A grid of checkboxes for various smart home features:
 - освітлення
 - системою опалення
 - кондиціонерами
 - керовані розетки
 - аудіо-відео технікою по ІЧ
 - жалюзі, штори
 - система охорони (ОПС)
 - системи безпеки
 - ВСЕ РАЗОМ
 - інше (опишіть в полі "Текст")
- Ремонт (електромонтажні роботи)***: Dropdown menu.
- Поверхові плани (1)**, **(2)**, **(3)**: Each has a file selection button and a 'Файл не вибран' message.
- Buttons: 'Відправити' and 'Скасувати'.

 The footer contains contact information for 'розумний ДІМ' in Lviv, including a phone number (032-2470945), a mobile number (067-9900045), and an email address (rozdim@ukr.net).

Рисунок 1.7 – Сторінка із формою попереднього розрахунку вартості системи розумний будинок аналога inteldim.lviv.ua

Для порівняння обраних аналогів було створено таблицю 1.1. У ній відображено особливості досліджуваних продуктів.

Таблиця 1.1 - Порівняння показників сайтів-аналогів розроблюваного web-додатку підтримки замовлення систем розумний будинок

Показник/можливість	ilounge.ua	domos.ua	xterm.com.ua	inteldim.lviv.ua
Наявність типових наборів систем розумний будинок	Немає	Є	Є	Є
Підбір системи по заповненій формі	Немає	Немає	Немає	Є
Наявність системи фільтрів та пошуку товарів	Є	Немає	Є	Немає
Наявність каталогу товарів	Є	Є	Є	Немає
Автоматичне оформлення замовлення	Є	Є	Є	Немає

Як видно з таблиці 1.1, жоден зі знайдених аналогів немає всіх досліджуваних функцій. Тому розробку web-додатку підтримки замовлень систем розумний будинок можна вважати раціональним рішенням, якщо в результаті буде реалізовано всі особливі параметри схожих продуктів.

1.3 Постановка задачі

Мета проекту полягає у розробці web-додатку підтримки замовлення систем розумний будинок. Користувачі даного продукту зможуть оптимізувати процес замовлення систем розумний будинок. Відбуватиметься це за рахунок інструментів, які надаватиме web-додаток. Так, наприклад, клієнту не обов'язково досконально розумітися у складній архітектурі систем того чи іншого напрямку. Достатньо буде просто ввести необхідні параметри. Web-додаток запропонує оптимальний варіант.

Для скорочення витрат часу на пошук конкретних пристроїв можна буде застосувати каталог з пошуком та фільтрами. У web-додатку також можна переглянути кілька типових наборів, що відповідатимуть популярним параметрам середовища для встановлення. Це допоможе зорієнтуватися при створенні власної системи.

Для розробки web-додатку необхідно виконати взаємопов'язані етапи:

- розробити базу даних;
- розробити модуль авторизації;
- розробити інструмент для підбору систем розумний будинок;
- розробити каталог товарів;
- розробити каталог типових наборів;
- розробити дизайн web-додатку;
- розробити графічний інтерфейс web-додатку;
- розробити адміністративну панель.

Цільовою аудиторією web-додатку будуть потенційні покупці, котрі прагнуть оптимізувати процеси замовлення і покупки систем розумний будинок. Користувачами будуть адміністратори web-додатку.

Проаналізувавши потреби користувачів та особливості продуктів-аналогів, було визначено перелік основних функціональних вимог:

- наявність реєстрації відвідувачів та авторизації клієнтів;

- поділ товарів на категорії;
- наявність переліку типових наборів;
- наявність інструменту для підбору системи розумний будинок;
- можливість клієнта здійснювати замовлення самостійно;
- наявність повної інформації про продукцію магазину;
- наявність сторінки з персональними даними клієнта;
- наявність панелі адміністратора для роботи з товарами.

Технічне завдання на розробку продукту у повному обсязі наведено у додатку

А.

1.4 Вибір засобів реалізації з розроблення web-додатку

Слід розуміти, що завдання зі створення web-додатку підтримки замовлення систем розумний будинок носить комплексний характер. Для його виконання потрібно застосовувати різні технології та програми залежно від етапів розробки, починаючи від верстки і закінчуючи тестуванням. Для того, щоб продукт надавав людям цінність, потрібно застосовувати актуальні та ефективні для вирішення поставлених проблем підходи.

Оскільки розроблюваний web-додаток повинен зберігати, як мінімум перелік товарів та інформацію про них, потрібно подбати про базу даних. Для її розробки та створення було обрано технологію MySQL, адже вона працює на всіх популярних операційних системах, є безкоштовною та не вимагає великих потужностей від апаратної частини.

Для написання частини, яка відповідатиме за логіку, було обрано мову php. Вона здатна інтегруватися в HTML та динамічно змінювати контент. Крім цього мова є досить простою, масштабованою, гнучкою та кросплатформенною, запускаючись на різному програмному забезпеченні сервера.

Візуальну складову в свою чергу можна поділити мінімум на дві частини. Тому створення каркасу web-додатку та його оздоблення буде виконано за допомогою HTML та CSS, а для того, щоб придати роботі web-додатку динаміки буде застосовано JavaScript, який може робити все, що пов'язано з маніпуляцією web-сторінками, взаємодією з користувачем та веб-сервером.

Можливості JavaScript:

- змінювати наповнення та стилі сторінки;
- інтерактивна взаємодія користувачів з web-інтерфейсом;
- надсилати запити до сервера через мережу;
- робота з куками;
- зберігати дані користувача [16].

При розробці дизайну web-додатку буде використано програму Figma – редактор для створення графічних зображень та верстки сайтів. Дизайнери користуються цією програмою для відтворення повноцінних інтерфейсів та їх складових, реалізації рухомих макетів та прототипів [17]. Однією з головних переваг є його популярність, що відразу говорить про те, що буде досить легко за потреби отримати підтримку від спільноти його користувачів. Більше того, в самому продукті є вкладки з корисними публікаціями та готовими макетами, якими діляться інші дизайнери. Також слід відмітити зручний та впорядкований інтерфейс.

Для локального тестування продукту допоможе програма MAMP. Вона надає усі інструменти, необхідні для запуску сервера на настільному ПК та тестування чи розробки веб-додатків. Є можливість протестувати проекти на мобільних пристроях. Можна використовувати веб-сервер Apache або ж Nginx. На додаток до MySQL як сервера бази даних, можна працювати з PHP, Python, Perl або Ruby [18].

Також буде використано PHPMailer – бібліотеку надсилання пошти, написану мовою php. Вона є досить простою у встановленні та налаштуванні. Рівень безпеки кращий, ніж у звичайної функції mail() – єдиної, яку надає PHP для надсилання електронних листів. До того ж, бібліотека надасть можливість відправляти їх з локального комп'ютера, що буде корисним на етапах активної розробки [19].

2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ ПІДТРИМКИ ЗАМОВЛЕННЯ СИСТЕМ РОЗУМНИЙ БУДИНОК

2.1 Структурно-функціональне моделювання web-додатку підтримки замовлення систем розумний будинок в IDEF0

Моделювання специфікацій, таких як класичні специфікації інтегрованих методів визначення (IDEF), сприяють узгодженості, доступності та агностицизму у процесі моделювання. Одна з таких специфікацій - IDEF0, є відомою та широко використовуваною специфікацією функціонального моделювання, особливо корисною для початкуючих інженерів у цій сфері. IDEF0 входить до складу більш широкого набору специфікацій моделювання IDEF, які охоплюють інформаційне моделювання, моделювання даних та моделювання процесів [20].

IDEF0 є частиною більш широкого набору специфікацій моделювання, відомого як IDEF (Integrated DEFinition). IDEF включає різні підходи до моделювання, такі як інформаційне моделювання, моделювання даних та моделювання процесів. Отже, IDEF0 не є самостійною специфікацією, але вписується в цей ширший набір специфікацій, які використовуються для розробки і аналізу функцій, даних та процесів в системах [20].

Базовою структурною одиницею функціонального моделювання IDEF0 є однокомпонентна функція. Вона перетворює вхідні дані у виходи за допомогою кількох елементів керування та механізмів:

- входи опрацьовуються функцією;
- виходи є результатами функції;
- елементи керування направляють функцію та задають їй обмеження;
- механізми показують, як функція виконується [20].

Візуальне схематичне представлення того, як слід виконувати моделювання IDEF0 представлено на рисунку 3.1.

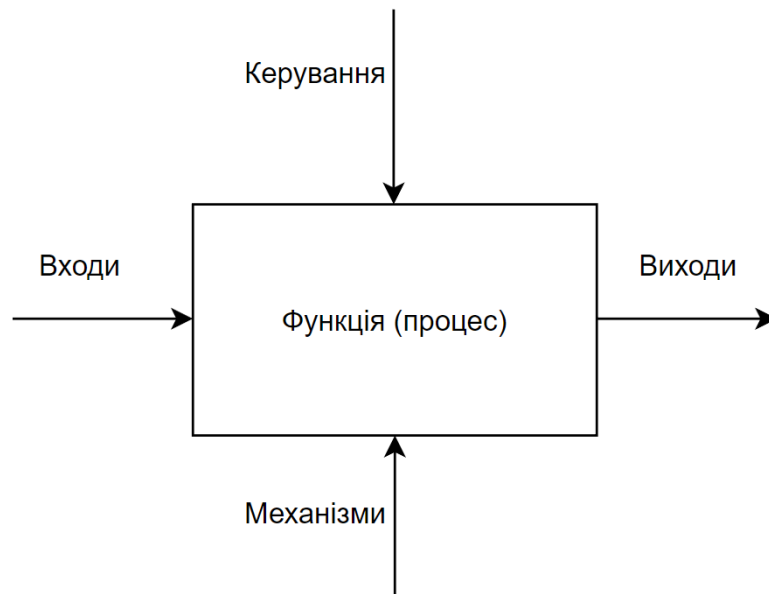


Рисунок 3.1 – Схема моделювання в нотатції IDEF0

Кожна функція, що використовується в IDEF0, має включати елементи перелічені нижче.

Назва функції – це опис діяльності, яку виконує функція, в формі дієслівно-іменникової фрази. Ця назва повинна чітко вказувати на призначення функції.

Дослідження часу: якщо можливо, до функції можуть бути додані відомості про часові параметри, пов'язані з виконанням цієї функції. Це може включати тривалість або періодичність виконання функції.

Ключові показники ефективності: ці показники пов'язані з іншими важливими метриками, які виходять за межі часу виконання функції.

Включення цих елементів допомагає забезпечити більш повну та зрозумілу специфікацію функцій в нотатції IDEF0 [20].

Функціональне моделювання процесу підтримки замовлення систем розумний будинок в нотатції IDEF0 з точки зору клієнта представлено на рисунку 3.2.

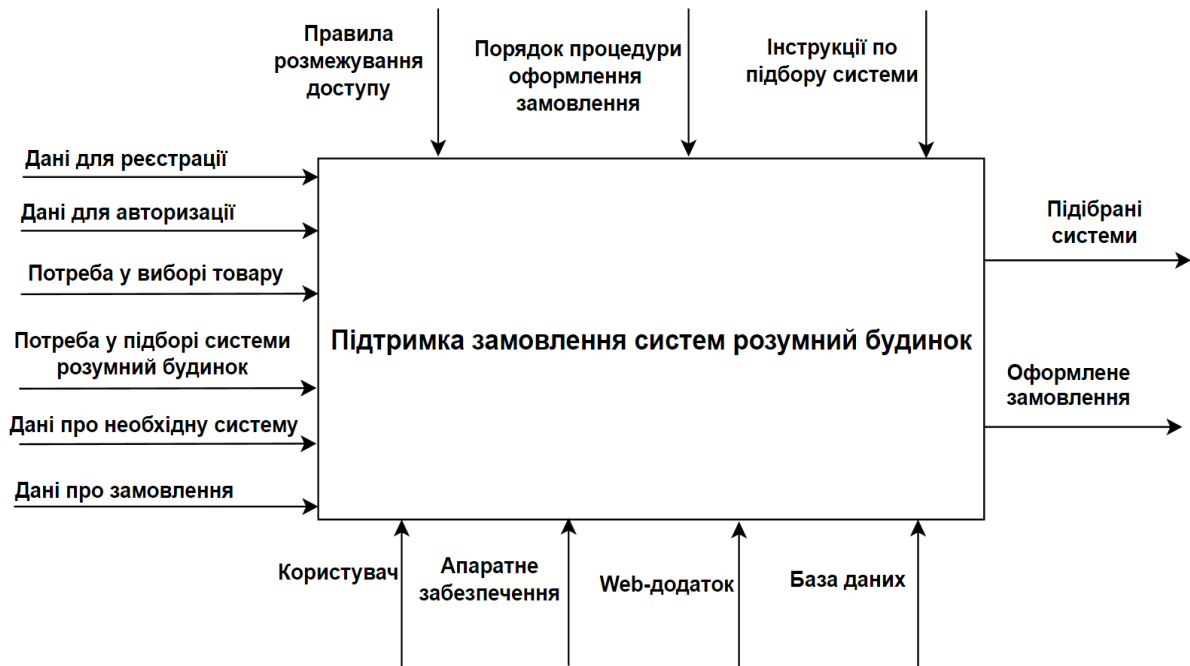


Рисунок 3.2 – Функціональне моделювання процесу підтримки замовлення систем розумний будинок в нотації IDEF0 з точки зору клієнта

В центральному блоці діаграми IDEF0 розміщено основну функцію web-додатку, а саме підтримка замовлення систем розумний будинок.

Для його виконання маємо вхідні дані – дані для реєстрації та авторизації, потреби у виборі товару та підборі системи розумний будинок. Також сюди відносяться дані про необхідну систему та дані про замовлення.

Механізмами будуть: користувач, який взаємодіятиме з розробкою, база даних, яка зберігатиме всю інформацію, web-додаток і апаратне забезпечення, на якому відбуватиметься реалізація головного процесу.

На виході отримуємо підібрані системи та оформлене замовлення.

До управління відносяться правила розмежування доступу, які визначатимуть можливості кожного, хто користуватиметься додатком, залежно від їх ролей. Порядок процедури оформлення замовлення визначатиме загальні правила і послідовність дій, які необхідно виконати для створення замовлення. Інструкції по підборі систему будуть використані при підборі системи

Наступним етапом структурно-функціонального моделювання буде декомпозиція. Вона дозволить більш конкретно зрозуміти будову попередньої функції. Також буде детальніше відображено її взаємодію з елементами. Це сприятиме загальному розумінню того, як відбуватимуться процеси підтримки замовлення систем розумний будинок. Нижче приведено основні правила функціональної декомпозиції.

Кожна функція розділяється на 3-6 внутрішніх функцій, які також можуть бути поділені на менші функціональні моделі. Оптимальним рівнем декомпозиції функціонального моделювання є момент, коли подальший поділ перестане приносити користь і де не буде багато некерованих, складних компонентних моделей. Мета, точка зору та контекст, визначені до початку моделювання, визначатимуть необхідний рівень декомпозиції.[20]

Декомпозиція функціональної моделі процесу підтримки замовлення систем розумний будинок в IDEF0 представлена на рисунку 3.3.

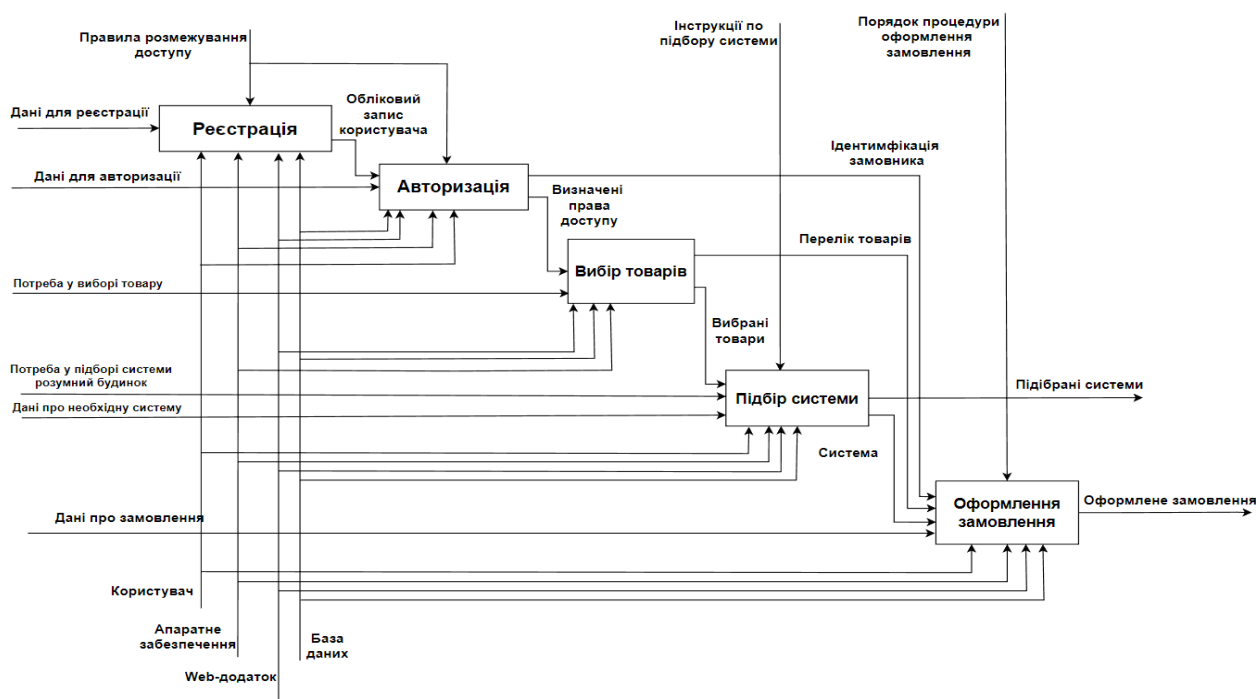


Рисунок 3.3 – Декомпозиція функціонального моделювання процесу підтримки замовлення систем розумний будинок в нотатції IDEF0 з точки зору клієнта

Як видно з діаграми, основна функція була декомпована на 5 процесів: «Реєстрація», «Авторизація», «Вибір товару», «Підбір системи» та «Оформлення замовлення». Користувачу потрібно пройти етап реєстрації, при цьому, на вхід подаються відповідні дані. В результаті цього процесу буде створено новий обліковий запис і з'явиться можливість авторизації. Після цього буде можливість перейти до вибору товарів та підбору системи. Виходи, отримані на цих етапах, будуть використані як входи для п'ятого процесу – Оформлення замовлення. Дана декомпозиція задіяла всі входи, виходи, механізми та елементи керування, які були представлені в попередній діаграмі і, по суті, лише уточнила головний процес.

Діаграма декомпозиції процесу підбору систем розумний будинок показана на рисунку 3.4. Дана діаграма була побудована для кращого розуміння одного з головних процесів попереднього рівня декомпозиції.

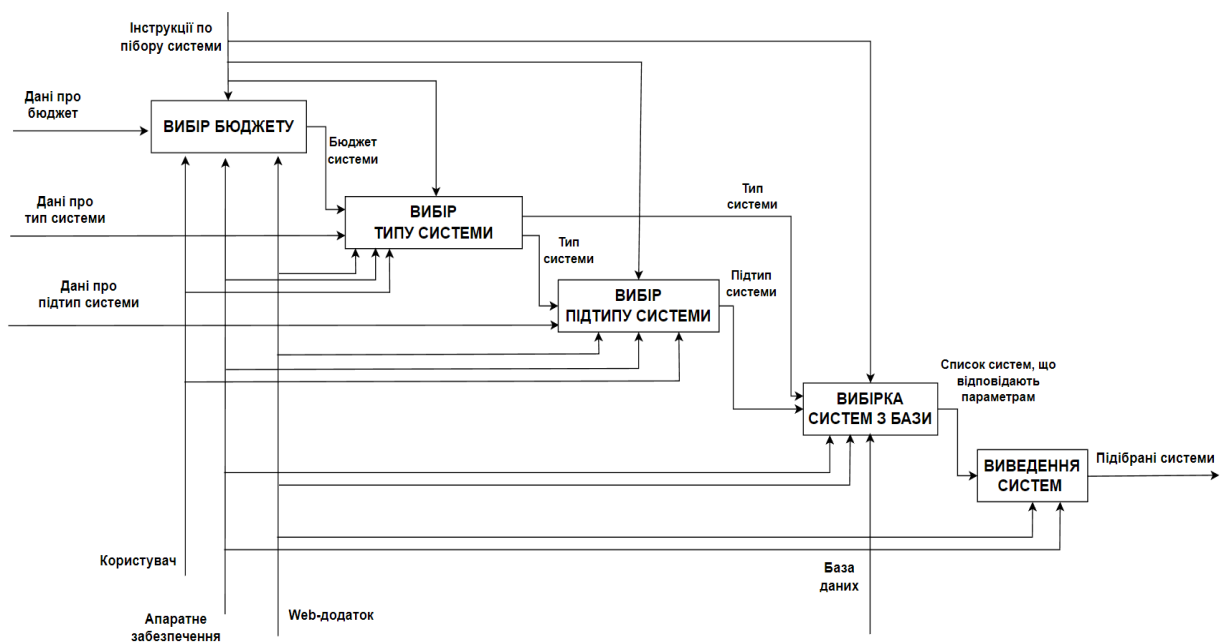


Рисунок 3.4 – Діаграма декомпозиції процесу підбору систем розумний будинок

Перелік основних функцій:

- вибір бюджету;
- вибір системи;

- вибір підтипу системи;
- вибірка систем з бази;
- виведення систем.

Керуючі елементи, механізми та вихід залишаються такими ж. Більш детально розкрито поняття вхідних даних, які використовуються для відповідних процесів. Також відображено, що отримується в результаті кожного з процесів.

3.2 Моделювання варіантів використання

Наступним етапом є моделювання варіантів використання. Даний підхід дозволяє краще усвідомити способи взаємодії основних елементів, пов'язаних з розроблюваним web-додатком. Також він допоможе ще раз переконатися, що всі необхідні вимоги до продукту були сформовані, не конфліктують одна з одною та можуть бути досягнуті. Основними елементами є актори та прецеденти.

Актор (учасник) – певна кількість взаємопов'язаних ролей, які виконуються при контакті з прецедентами. Актором може бути система, людина, клас, або ролі, які вони відіграють за межами сутності.

Прецедент – пояснення певної частини діяльності системи з погляду користувача. Він не відображає, як саме отримується певний результат, а лише вказує на те, що конкретно виконується [21].

Було побудовано дві діаграми варіантів використання. Одна відображає поведінку системи з точки зору Клієнта – людини, яка використовує web-додаток для оптимізації процесу вибору певного товару, та з точки зору Адміністратора – того, хто керує наповненням додатку та опрацьовує оформлені замовлення. Дані діаграми web-додатку підтримки замовлення систем розумний будинок в нотації UML представлено на рисунках 2.5. та 2.6.

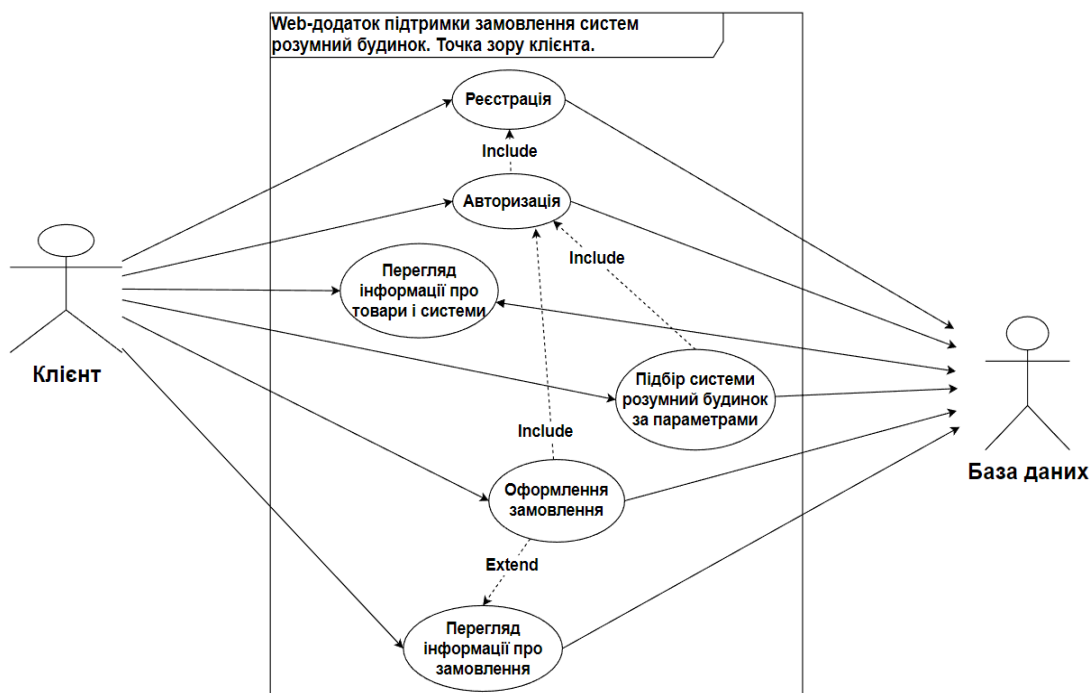


Рисунок 2.5 – Діаграма варіантів використання web-додатку підтримки замовлення систем розумний будинок з точки зору Клієнта

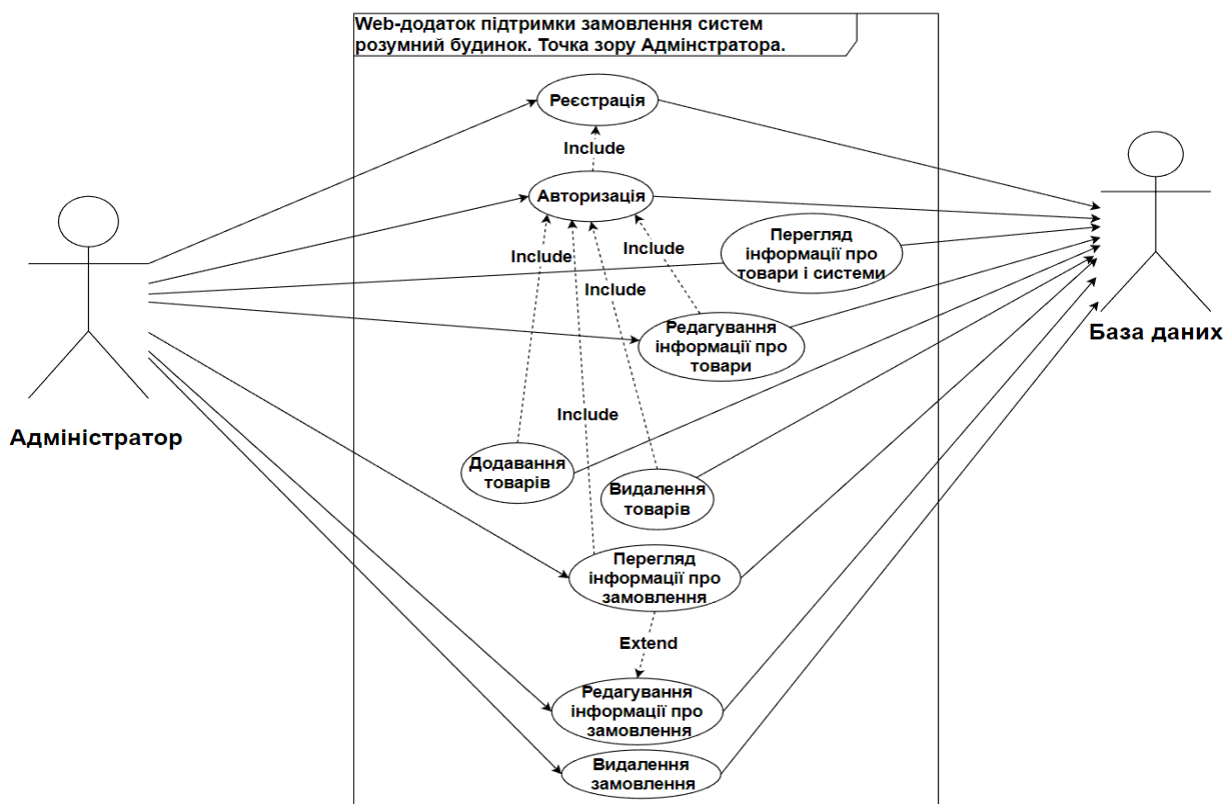


Рисунок 2.6 – Діаграма варіантів використання web-додатку підтримки замовлення систем розумний будинок з точки зору Адміністратора

На рисунку 2.4 на діаграмі вказані два актори. Зліва вгорі знаходиться користувач – особа, яка зацікавлена у спрощенні процесу замовлення і вибору товару. Для задоволення своїх потреб даний актор буде використовувати розроблений web-додаток, який в свою чергу повинен відповідати всім вимогам клієнта. Тому продукт повинен мати інструменти, які дозволять користувачу виконувати такі дії, як перегляд інформації про товари та системи розумний будинок, підбирати системи розумного будинку за введеними параметрами та оформлювати замовлення продукції, яка йому сподобалася.

На рисунку 2.5, актор, який зображено з правої сторони є Адміністратором. Це також користувач web-додатку, але в нього зовсім інші потреби та цілі, тому він має можливості виконання таких дій як, додавання, редагування та видалення інформації про товари. Даний актор також відповідає за обробку замовлень, які формує Клієнта. Тому він повинен мати можливість переглядати зміст замовлень, також редагувати певні дані про них, а за потреби повністю видаляти.

В обох діаграмах відображено реєстрацію та авторизацію – варіанти використання, які дозволять розподілити права доступу між акторами. Також з правої сторони на обох рисунках знаходиться актор База даних. З ним обидва користувачі взаємодіють через прецеденти, і будь-яка їх дія супроводжується запитом саме до цієї зовнішньої системи.

Дана схема використання web-додатку підтримки замовлення систем розумний будинок висвітлює основні шляхи його використання. Дослідивши дану діаграму, що було зроблено вище, можна зробити висновок, що всі необхідні функції оптимізації процесу вибору та замовлення систем розумний будинок розробка покриває.

3.3 Моделювання послідовності оформлення замовлення

Діаграма послідовності в нотації UML відображає порядок взаємодії певних елементів у часі. Взаємодія, при цьому, відображається у вигляді повідомлень. Об'єктами відображення цього процесу найчастіше виступають користувачі та системи, або частини систем [22].

На рисунку 2.7 представлена діаграма послідовності підтримки замовлення систем розумний будинок засобами web-додатку.

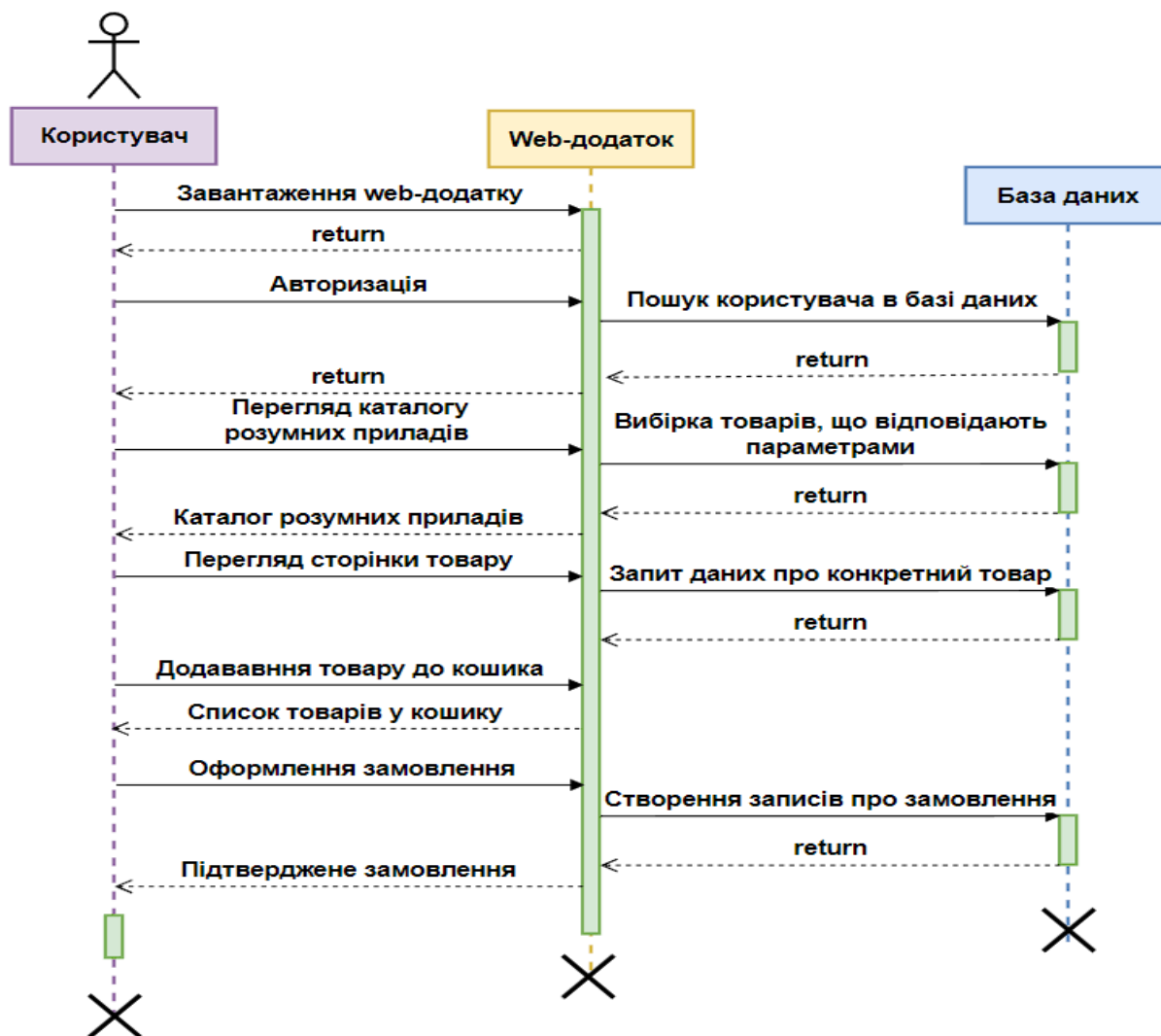


Рисунок 2.7 – Діаграма послідовності підтримки замовлення систем розумний будинок за допомогою web-додатку

На даній діаграмі зображено три об'єкти: користувач, web-додаток та база даних. Більшу частину часу перший із них надсилає запит до другого, а той обробляє його і звертається до сховища даних за потрібною для користувача інформацією. Розглянемо покроково всі етапи взаємодії перелічених об'єктів, результатом яких, буде створене замовлення.

Спочатку користувач надсилає запит на завантаження web-додатку і в разі успіху отримує певну відповідь від нього у вигляді відображення певного інтерфейсу. Після цього клієнт надішле запит на авторизацію. Дані, отримані від нього, будуть перевірені на наявність в базі даних. Якщо там існує відповідний запис, то відправнику буде надано відповідні права доступу.

Перегляд каталогу також потребує звернення як до web-додатку так і до сховища, яке і зберігає всю необхідну інформацію про додатки. При цьому можна задати особливі критерії для пошуку товарів на відображення. Результатом буде перелік компонентів розумного будинку, з якими можна ознайомитися детальніше.

Для цього, знову ж таки, через web-додаток, слід надіслати запит до бази даних. У разі вподобання певного продукту, можна помістити його до кошика. Оскільки даний механізм реалізовано без задіяння постійного сховища даних, то запит до нього не відправляється. Користувач, отримує у відповідь список необхідних товарів.

Лише на етапі оформлення замовлення відбувається переміщення даних про замовлення до бази. Якщо це відбулося успішно, то web-додаток повідомить користувача про це.

Дана послідовність відображає лише один із можливих сценаріїв взаємодії об'єктів для створення замовлення, самий простий і зрозумілий. Адже до замовлення також можуть входити системи розумного будинку, крім цього, можна використати інструмент для підбору потрібної системи і вже після цього перейти до етапу формування замовлення. При цьому розглянуто варіанти з успішними відповідями від web-додатку та бази даних. Проте, ключову послідовність подій було відображено, що дозволить краще закріпити уявлення про функціонування розроблюваного web-додатку.

3.4 Проектування дерева рішень

В підрозділі структурно-функціонального моделювання в нотаціях IDEF0 та IDEF1 було розглянуто функцію вибору системи розумний будинок. Це дало розуміння того, як в загальному випадку повинен проходити даний процес. Для ще більш детального та конкретизованого представлення принципу роботи даного інструменту web-додатку було побудоване схематичне представлення дерева рішень підбору системи розумний будинок, зображене на рисунку 2.8.

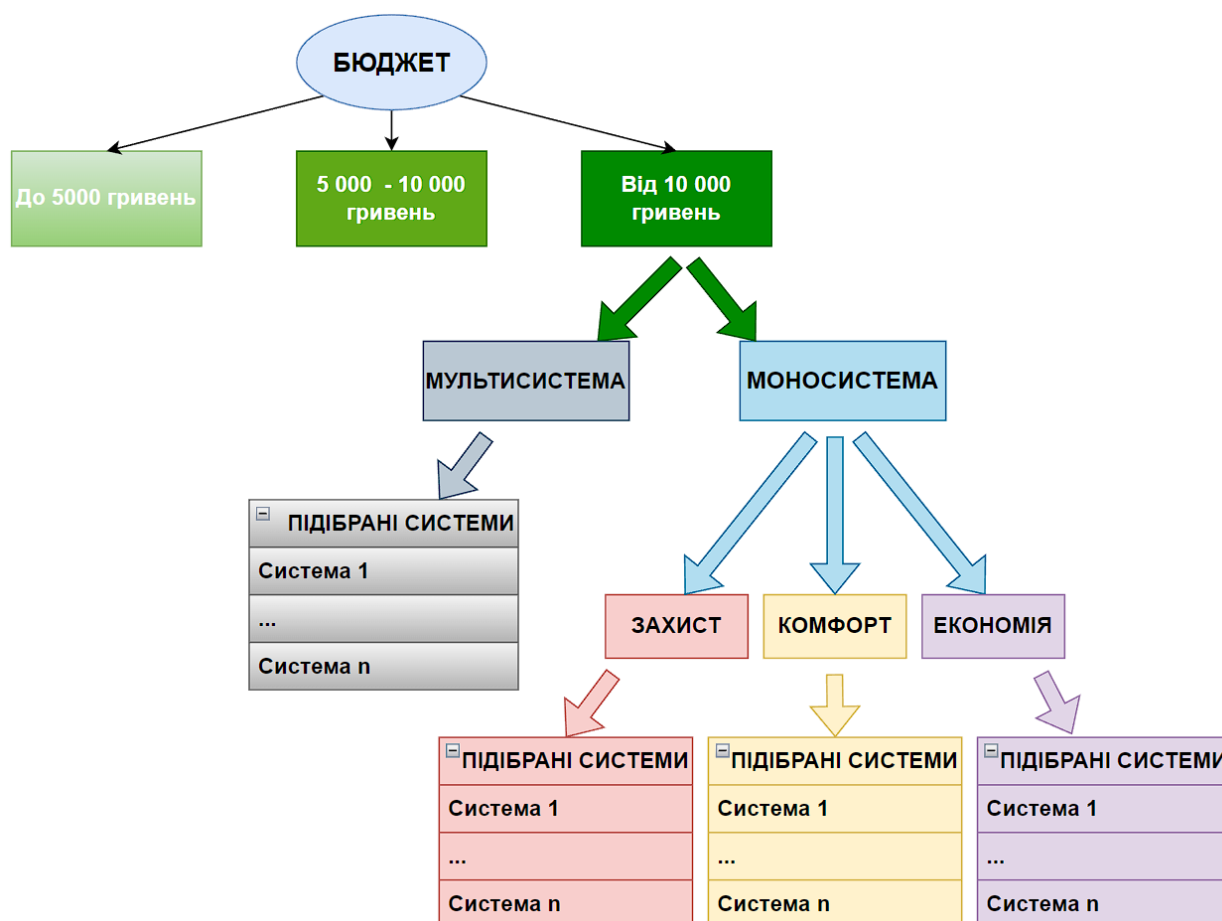


Рисунок 2.8 – Дерево прийняття рішень вибору системи розумний будинок

Як видно зі схеми, користувачу на кожному етапі потрібно буде зробити рішення, обравши один зі трьох варіантів, як у випадку із бюджетом та підтипом моносистеми, або із двох варіантів у випадку вибору типу системи. У результаті

невеликої кількості прийнятих, чітких і явно визначених рішень буде отримано список систем, які відповідатимуть потребам користувача.

3.5 Проектування бази даних

Фізична модель являє собою макет фактичної бази даних із усіма її компонентами та службами. Вона заснована на уже зробленому виборі до технологій, в яких база даних буде виконана, як наприклад SQL, Ingres чи ORACLE.

Фізична модель – це розширення та доповнення логічної моделі, яке відображає більше деталей. Наприклад, назви таблиць для сутностей та більший перелік специфікацій для кожного з атрибутів, таких, як довжина, тип даних та їх точність. Дана модель призначена для розробників, для тих, хто хоче знати, де фактично зберігаються дані, та як отримати доступ до них.

Також на фізичній моделі можуть бути представлені ролі, первинні ключі, індекси, обмеження, представлення та процедури. Крім того, фізична модель може зберігати проміжні дані, згенеровані програмними процесами, які будуть використані пізніше [23].

На рисунку 2.9 представлена фізична модель даних web-додатку підтримки замовлення систем розумний будинок.

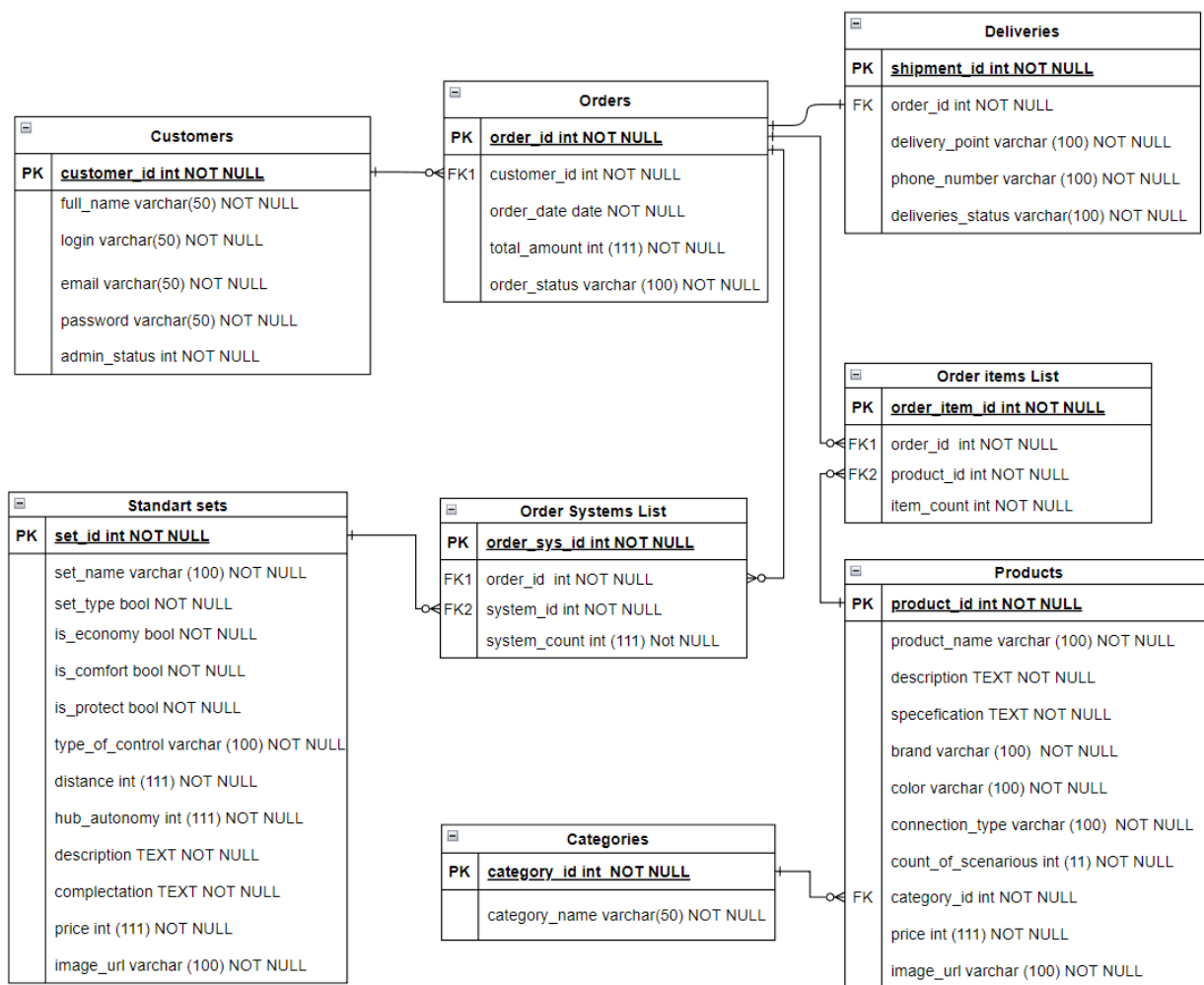


Рисунок 2.9 – Фізична модель бази даних web-додатку підтримки замовлення систем розумний будинок

Представлена вище модель містить таблицю замовлень Orders, яка зберігатиме основні дані про всі сформовані замовлення. Вона міститиме дані про загальну суму покупок, дату створення та поточний статус замовлення та посилання на клієнта, який його зробив.

Таблиця Customers міститиме дані про користувача, які знадобляться для авторизації та контакту з ним. Поле admin_status визначатиме, чи є матиме обліковий запис доступ до функцій адміністрування web-додатку.

Таблиця Deliveries міститиме дані про доставку, які стосуються конкретного замовлення. Зв'язок буде встановлено через зовнішній ключ order_id.

Order Systems List та Order Items List міститимуть списки продуктів та систем розумний будинок, які відповідають конкретному замовленню. Перелік

формуватиметься із полів, які є посиланнями на те, що було замовлено - system_id та product_id, та їх кількості - system_count_id та item_count_id.

Standart sets – таблиця, що міститиме дані про системи розумний будинок. Поле complectation міститиме перелік приладів, що входять до набору, а set_type, is_comfort, is_economy, is_protect – відповідатимуть логічному типу даних і визначатимуть чи це мультисистема, чи вона покриває потреби в комфорті, економії або ж захисті, відповідно.

Таблиця Products міститиме дані про окремі розумні товари. Вона містить посилання на дані з таблиці Categories через зовнішній ключ category_id. За допомогою цього зв'язку буде визначатися належність товару до певної категорії приладів.

В таблицях 2.1 – 2.8 наведено повний перелік атрибутів для кожної з таблиць фізичної моделі бази даних. Дані таблиці містять стовпчик, який вказує назву поля та стовпчик, який пояснює призначення поля.

Таблиця 2.1 – Пояснення призначення полів таблиці Customers

Назва поля	Пояснення призначення поля
customer_id	Зберігає номер користувача в базі даних.
full_name	Зберігає повне ім'я користувача.
login	Зберігає логін користувача.
email	Зберігає електронну пошту користувача.
password	Зберігає пароль користувача.
admin_status	Зберігає інформацію про те, чи є користувач адміністратором.

Таблиця 2.2 – Пояснення призначення полів таблиці Orders

Назва поля	Пояснення призначення поля
order_id	Зберігає номер замовлення.
customer_id	Зберігає номер користувача, який зробив дане замовлення.
order_date	Зберігає дату створення замовлення користувачем.
total_amount	Зберігає значення повної вартості замовлення.
order_status	Зберігає значення статусу замовлення.

Таблиця 2.3 – Пояснення призначення полів таблиці Deliveries

Назва поля	Пояснення призначення поля
shipment_id	Зберігає номер доставки в базі даних.
order_id	Зберігає номер замовлення, для якого виконується доставка.
delivery_point	Зберігає місце доставки даного замовлення (order_id).
phone_number	Зберігає номер телефону
deliveries_status	Зберігає значення статусу доставки.

Таблиця 2.4 – Пояснення призначення полів таблиці Order items List

Назва поля	Пояснення призначення поля
category_id	Зберігає номер категорії в базі даних.
category_name	Зберігає ім'я категорії.

Таблиця 2.5 – Пояснення призначення полів таблиці Order items List

Назва поля	Пояснення призначення поля
item_id	Зберігає номер запису про продукт, що входить до замовлення (order_id).
order_id	Зберігає номер замовлення, якому належить даний запис.
product_id	Зберігає номер продукту.
item_count	Зберігає кількість даного товару (product_id) в замовленні (order_id).

Таблиця 2.6 – Пояснення призначення полів таблиці Order Systems List

Назва поля	Пояснення призначення поля
order_sys_id	Зберігає номер запису про систему, що входить до замовлення (order_id).
order_id	Зберігає номер замовлення, якому належить даний запис.
system_id	Зберігає номер системи розумний будинок.
system_count	Зберігає кількість даних систем розумного будинку (system_id) в замовленні (order_id).

Таблиця 2.7 – Пояснення призначення полів таблиці Products

Назва поля	Пояснення призначення поля
product_id	Зберігає номер товару в базі даних.
product_name	Зберігає назву товару.
description	Зберігає опис товару.
brand	Зберігає назву бренду товару.
color	Зберігає значення кольору товару.
connection_type	Зберігає інформацію про тип підключення товару.
count_of_scenarios	Зберігає значення кількості сценаріїв наявних в товарі.
specification	Зберігає перелік специфікацій товару.
category_id	Зберігає номер категорії, до якої належить товар
price	Зберігає значення ціни товару.
image_url	Зберігає посилання на розміщення файлу зображення товару на сервері.
is_active	Зберігає інформацію про те, чи видимий товар для клієнтів.

Таблиця 2.8 – Пояснення призначення полів таблиці Stendart sets

Назва поля	Пояснення призначення поля
set_id	Зберігає номер системи розумний будинок в базі даних.
set_name	Зберігає назву системи розумний будинок.

Продовження таблиці 2.8 – Пояснення призначення полів таблиці Stendart sets

Назва поля	Пояснення призначення поля
is_multisystem	Зберігає інформацію про відношення системи розумний будинок до типу мультисистем.
is_economy	Зберігає інформацію про відношення системи розумний будинок до підтипу економія.
is_protect	Зберігає інформацію про відношення системи розумний будинок до підтипу захист.
is_comfort	Зберігає інформацію про відношення системи розумний будинок до підтипу комфорт.
type_of_control	Зберігає інформацію про тип контролю даної системи розумний будинок.
distance	Зберігає інформацію про дальність роботи даної системи розумний будинок.
hub_autonomy	Зберігає інформацію про автономність хабу даної системи розумний будинок
description	Зберігає опис системи розумний будинок.
complectation	Зберігає перелік приладів, що входять до системи розумний будинок.
price	Зберігає значення ціни системи розумний будинок.
img_url	Зберігає посилання на розміщення файлу зображення системи розумний будинок на сервері.

3 РОЗРОБКА WEB-ДОДАТКУ ПІДТРИМКИ ЗАМОВЛЕННЯ СИСТЕМ РОЗУМНИЙ БУДИНОК

3.1 Архітектура web-додатку підтримки замовлення систем розумний будинок

Архітектура web-додатку описує його структуру та організацію у широкому масштабі. Вона визначає основні компоненти та взаємодію між ними, а також встановлює принципи, за якими будується додаток. Основною метою є забезпечення ефективності, масштабованості, безпеки та розширюваності продукту.

Архітектура web-додатку для системи підтримки замовлення розумного будинку складається з кількох основних елементів. Відповідна діаграма зображена на рисунку 3.1.

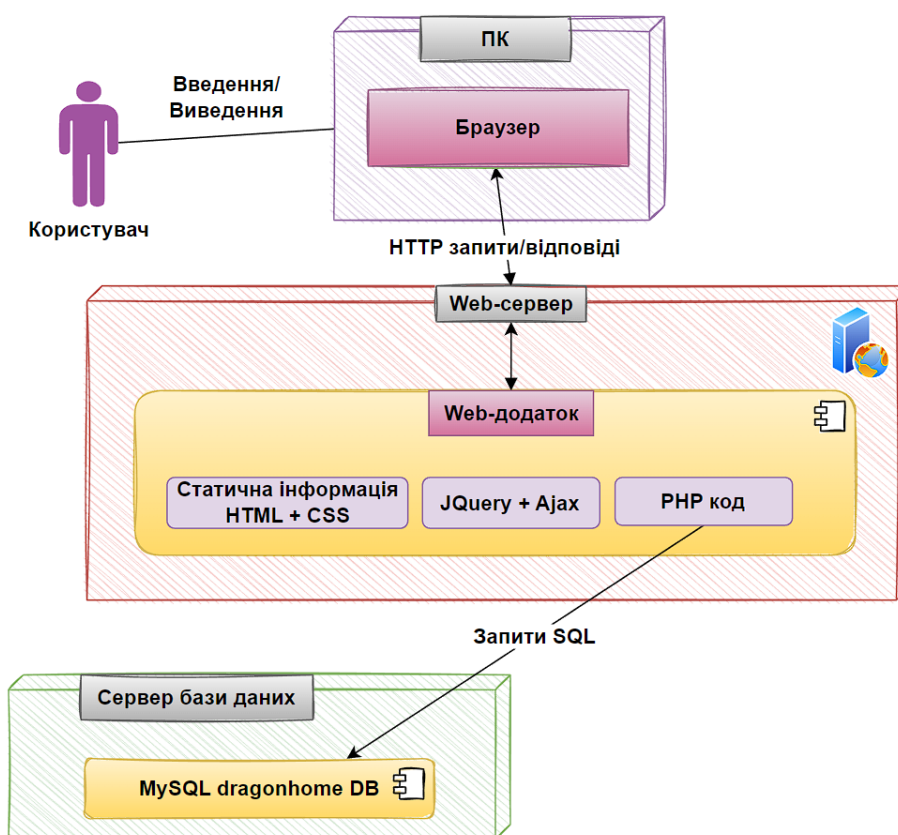


Рисунок 3.1 – Архітектура web-додатку підтримки замовлення систем розумний будинок

Користувач взаємодіє з додатком через браузер, установлений на персональний комп'ютер. На web-сервері містяться дані HTML та CSS для відображення статичного контенту, а JavaScript бібліотека jQuery використовується для обробки дій користувача. Технологія Ajax дозволяє взаємодіяти з сервером та отримувати від нього відповіді без перезавантаження сторінок.

Запити HTTP відправляються з персонального комп'ютера до web-сервера, який у відповідь надсилає інформацію, частина якої, згенерована PHP-кодом. При цьому web-сервер взаємодіє з сервером бази даних, використовуючи запити SQL, для отримання потрібних даних.

3.2 Програмна реалізація web-додатку

Програмна реалізація web-додатку починається із підготовки середовищ розробки. У підрозділі вибору засобів реалізації було визначено, що на етапі розробки буде задіяно локальний сервер, налаштований за допомогою програми MAMP. Він дозволить працювати з PHP кодом і базою MySQL. Встановивши налаштування та вказавши шлях до папки з проектом, слід натиснути «Start Servers». Якщо навпроти написів Apache Server та MySQL Server кільця стали зеленими, як на рисунку 3.2, то сервери web-додатку та бази даних працюють. Після цього слід перейти на стартову сторінку, яка зображена на рисунку 3.3, де вказано додаткові дані для підключення бази даних та посилання на phpMyAdmin.

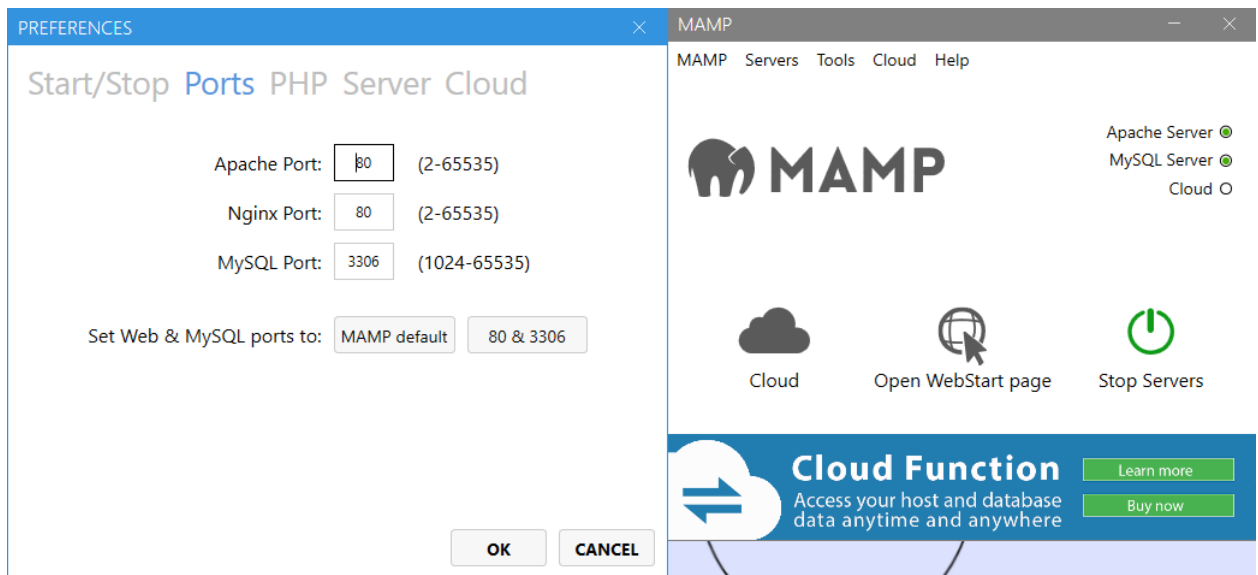


Рисунок 3.2 – Вікна запуску та налаштування МАМР

MAMP PRO

With MAMP PRO you can create a separate host for each of your web projects. Install WordPress with only one click and after completion of your page, publish it directly from MAMP PRO on your live server. [Learn more](#)

PHP

phpinfo shows the current configuration of PHP.

PHP-Caches

APC (not loaded)
eAccelerator (not loaded)
XCache (not loaded)
OPcache (not loaded)

phpMyAdmin

Configure your MySQL databases with [phpMyAdmin](#).

phpLiteAdmin

phpLiteAdmin needs PHP 5.2.4 to 7.0.x

MAMP Cloud

You can easily save your host and database data in the cloud and later retrieve it on another Mac, when backing up data before making changes on your host. [Learn more](#)

MySQL

MySQL can be administered with [phpMyAdmin](#).

To connect to the MySQL server from your own scripts use the following connection parameters:

Host	localhost
Port	3306
User	root
Password	root

Examples:

[PHP >= 5.6.x](#) [PHP <= 5.5.x](#) [Python](#) [Perl](#)


Connect via network:

```
$user = 'root';
$password = 'root';
$db = 'inventory';
$host = 'localhost';
$port = 3306;

$link = mysqli_init();
$success = mysqli_real_connect(
    $link,
    $host,
    $user,
    $password,
    $db,
    $port
);
```

MAMP News

Tweets from @mamp_en



Nothing to see here - yet

When they Tweet, their Tweets will show up here.

[View on Twitter](#)

Рисунок 3.3 – Стартова сторінка МАМР

У phpMyAdmin було створено базу даних, яка відповідає фізичній моделі, розміщеній в розділі проектування web-додатку. Структура створеної бази даних представлена на рисунку 3.4.

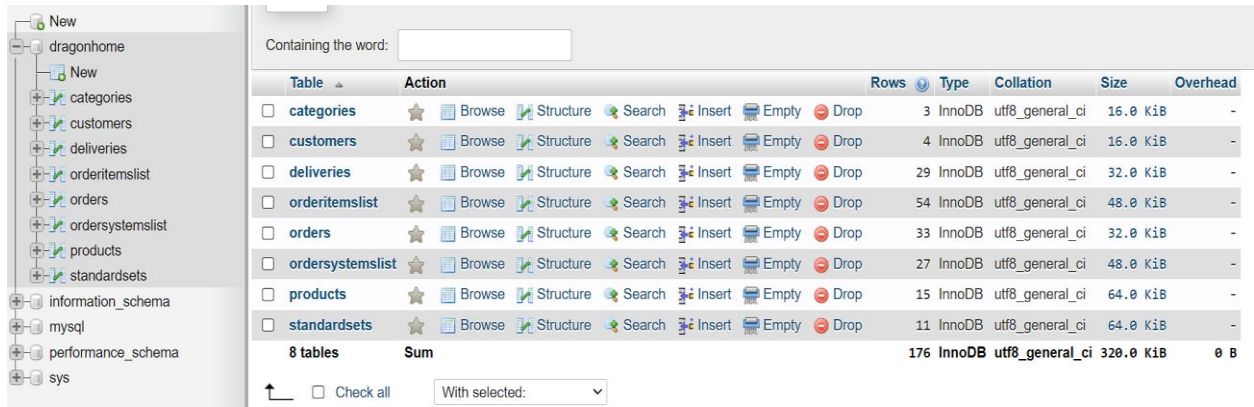


Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> categories	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> customers	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> deliveries	★ Browse Structure Search Insert Empty Drop	29	InnoDB	utf8_general_ci	32.0 KiB	-
<input type="checkbox"/> orderitemslst	★ Browse Structure Search Insert Empty Drop	54	InnoDB	utf8_general_ci	48.0 KiB	-
<input type="checkbox"/> orders	★ Browse Structure Search Insert Empty Drop	33	InnoDB	utf8_general_ci	32.0 KiB	-
<input type="checkbox"/> ordersystemslst	★ Browse Structure Search Insert Empty Drop	27	InnoDB	utf8_general_ci	48.0 KiB	-
<input type="checkbox"/> products	★ Browse Structure Search Insert Empty Drop	15	InnoDB	utf8_general_ci	64.0 KiB	-
<input type="checkbox"/> standardsets	★ Browse Structure Search Insert Empty Drop	11	InnoDB	utf8_general_ci	64.0 KiB	-
8 tables	Sum	176	InnoDB	utf8_general_ci	320.0 KiB	0 B

Рисунок 3.4 – Структура бази даних dragonhome в phpMyAdmin

Для роботи з кодом було обрано Visual Studio Code, адже програма є досить універсальною, тобто в ній можна працювати з будь якою мовою програмування. Вона досить гнучка і може бути модифікованою за допомогою розширень.

Дане програмне забезпечення досить популярне і має велику спільноту розробників, які користуються нею, що сприяє розвитку продукту в цілому. Приклад інтерфейсу зображено на рисунку 3.5.

Також на даному зображенні відкрито файл, який є шаблоном для головного меню web-додатку. За допомогою php він підключається майже до кожної сторінки, а для внесення змін потрібно відредагувати вміст лише одного файлу.

Зліва можна спостерігати структуру проекту. Папка content містить файли, які можна використати при наповненні web-додатку контентом, а в uploads зберігаються всі картинки, завантажені, як клієнтами так і адміністрацією сайту. Файли в кореневій папці проекту, як той, що відкрито на рисунку 3.5, формують структуру відображення контенту на сторінках.

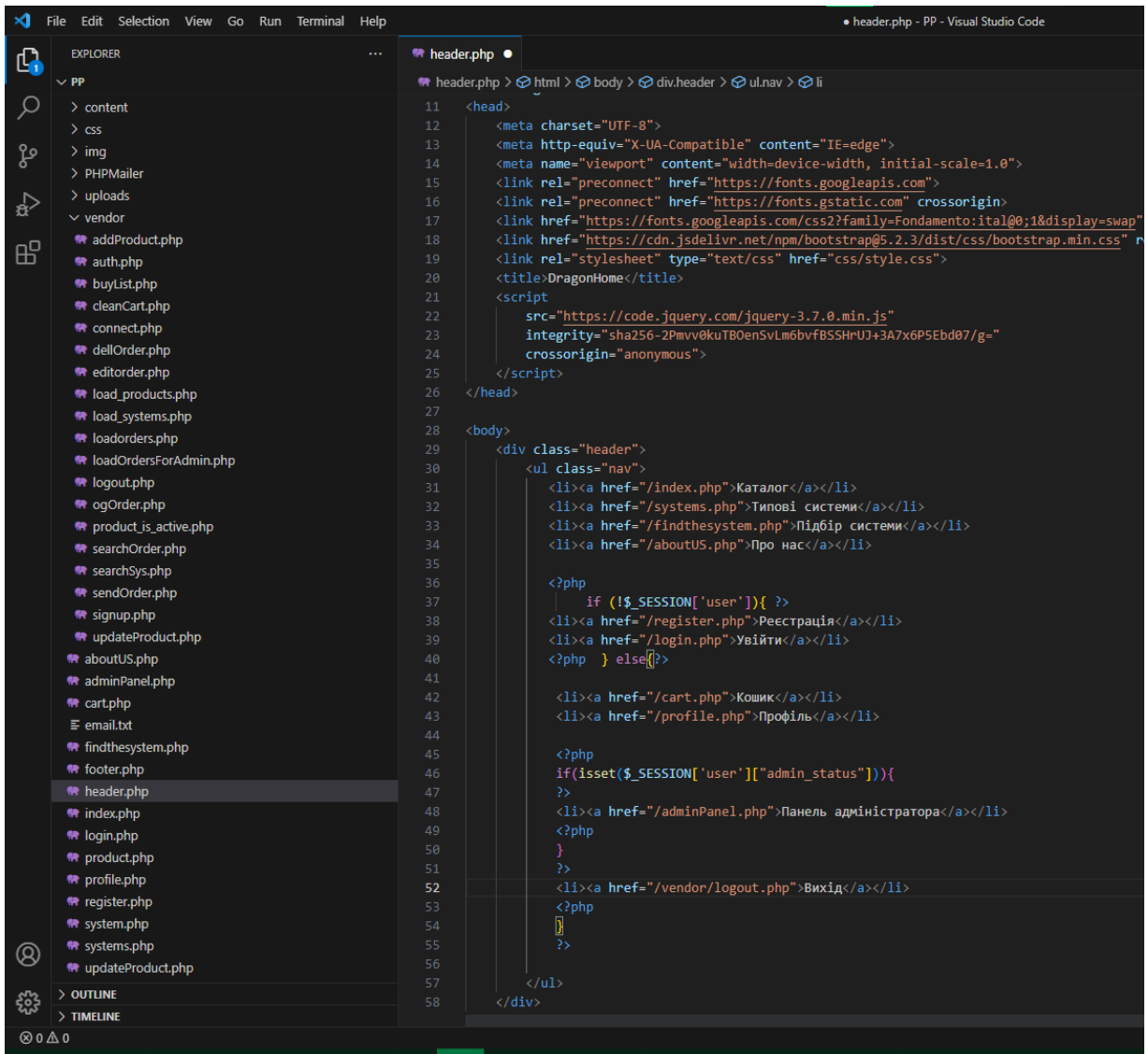


Рисунок 3.5 – Файлова структура web-додатку в інтерфейсі Visual Studio Code

В папці vendor зберігаються файли php, які відповідають за запити до бази даних. Також вони створюють динамічне наповнення для сторінки, на основі отриманих з бази рядків. Приклад такого файлу зображено на рисунку 3.6. loadorders.php призначений для завантаження переліку замовлень, зроблених користувачем.

```

17     </div>
18
19     <div class="order-info">
20         <div class="info-label">Статус доставки</div>
21     </div>
22
23     <div class="order-info">
24         <div class="info-label">Статус замовлення</div>
25     </div>
26
27 </div>
28
29 <?php
30
31 session_start();
32 include_once('connect.php');
33
34 $userID = $_POST['userIDfield'];
35 $offset = $_POST['offsetfield'];
36 $limit = 5;
37
38
39 $query = "SELECT o.order_id, d.phone_number, o.order_date, d.delivery_point, d.deliveries_status, o.order_status
40 FROM customers c
41 JOIN orders o ON c.customer_id = o.customer_id
42 JOIN deliveries d ON o.order_id = d.order_id
43 WHERE c.customer_id = $userID
44 ORDER BY o.order_id DESC
45 LIMIT $offset, $limit";
46
47
48 $orders = mysqli_query($connect, $query);
49 $orders = mysqli_fetch_all($orders);
50
51 foreach ($orders as $order) {
52     >>
53     <div class="order">
54         <div class="info-value">?<= $order[0] ?></div>
55
56         <div class="info-value">?<= $order[1] ?></div>
57
58         <div class="info-value">?<= $order[2] ?></div>
59
60         <div class="info-value">?<= $order[3] ?></div>
61
62         <div class="info-value">?<= $order[4] ?></div>
63
64         <div class="info-value">?<= $order[5] ?></div>
65

```

Рисунок 3.6 – Вміст файлу loadorders.php

Для надсилення електронних листів використано бібліотеку PHPMailer. Для цього її було завантажено з github.com у вигляді архіву та розпаковано до кореневої папки проекту. Саме налаштування обмежилось кодом, написаним у файлі `sendOrder.php`, який зображено на рисунку 3.7.

Спочатку відбувається підключення бібліотеки з вказанням шляху до неї. Потім створюється об'єкт `PHPMailer()`. Залежно від провайдера поштових послуг, потрібно вказати порт та хост. У випадку з `gmail` це відповідні значення `587` та «`smtp.gmail.com`». Для відправлення потрібно було створити пошту, для якої встановити подвійну аутентифікацію та створити програмний ключ для входу. Його значення присвоєно змінній `Password`. Адреса, на яку слід відправити листа, отримується із сесії створеної поточним користувачем.

```

178
179 use PHPMailer\PHPMailer\PHPMailer;
180 use PHPMailer\PHPMailer\Exception;
181
182 // Включаємо необхідні файли для використання PHPMailer
183 require '../PHPMailer/src/Exception.php';
184 require '../PHPMailer/src/PHPMailer.php';
185 require '../PHPMailer/src/SMTP.php';
186
187 $phpmailer = new PHPMailer(); // Створюємо новий об'єкт PHPMailer
188 $phpmailer->CharSet = 'UTF-8'; // Налаштовуємо кодування символів на UTF-8
189 $phpmailer->isHTML(true); // Встановлюємо формат повідомлення HTML
190 $phpmailer->isSMTP(); // Використовуємо SMTP для відправки листів
191 $phpmailer->Host = 'smtp.gmail.com'; // Вказуємо хост SMTP-сервера (у цьому випадку використовується Gmail)
192 $phpmailer->SMTPAuth = true; // Вимагаємо аутентифікацію на SMTP-сервері
193
194 $phpmailer->Port = 587; // Вказуємо порт SMTP-сервера (у випадку Gmail - 587)
195 $phpmailer->Username = 'dragonhomeua@gmail.com'; // Вказуємо логін (адресу електронної пошти) для авторизації на SMTP-сервері
196 $phpmailer->Password = ' '; // Вказуємо пароль для авторизації на SMTP-сервері
197 $phpmailer->SMTPSecure = 'tls'; // Встановлюємо тип безпеки з'єднання з SMTP-сервером (у випадку Gmail - tls)
198 $phpmailer->SMTPKeepAlive = true; // Залишаємо з'єднання з SMTP-сервером активним після відправки кожного листа
199 $phpmailer->setFrom('dragonhomeua@gmail.com', 'List manager'); // Встановлюємо адресу та ім'я відправника
200 $phpmailer->Subject = "Замовлення на сайті DragonHome"; // Встановлюємо тему листа
201 // Масив користувачів, яким буде надіслано листи
202 $users = [
203     ['email' => $_SESSION['user']['email'], 'name' => $_SESSION['user']['full_name']]
204 ];
205 // Ітеруємося по користувачам та надсилаємо листи
206 foreach ($users as $user) {
207     // Додаємо адресу отримувача
208     $phpmailer->addAddress($user['email'], $user['name']);
209
210     // Встановлюємо тіло повідомлення (HTML-версію)
211     $phpmailer->Body = $body;
212
213     // Встановлюємо альтернативний текст повідомлення (текстову версію)
214     $phpmailer->AltBody = $body;
215
216     try {
217         // Надсилаємо лист
218         $phpmailer->send();
219         echo "Повідомлення надіслано на: ({$user['email']}) {$phpmailer->ErrorInfo}\n";
220     } catch (Exception $e) {
221         echo "Помилка відправки ({$user['email']}) {$phpmailer->ErrorInfo}\n";
222     }
223     $phpmailer->clearAddresses(); // Очищуємо адреси отримувачів для наступного листа
224 }
225
226 $phpmailer->smtpClose(); // Закриваємо з'єднання з SMTP-сервером

```

Рисунок 3.7 – Вміст файлу sendOrder.php

3.3 Демонстрація роботи web-додатку

Головною сторінкою web-додатку є каталог розумних приладів, зображений на рисунку 3.8. Реалізовано можливість переглянути товари, які відносяться до певної категорії, як, наприклад, хаби або ж переглянути пристрої, незалежно від приналежності до однієї з представлених груп, при натисканні на кнопку «Всі».

Нижче знаходиться панель, яка дозволяє сортувати товари за ціною та назвою. В цьому блоці також розміщено поле для пошуку товарів за назвою.

З лівої частини знаходяться фільтри, де можна вказати діапазон цін, бренди та кількість сценаріїв. При натисканні на кнопку пошуку, буде відображено виключно ті товари, що відповідають даним параметрам.

В центральній частині можна побачити перелік товарів. На даній сторінці відображається зображення, назва та ціна пристрою. При натисканні на кнопку «Переглянути» відкривається сторінка з більш детальною інформацією про товар. Під цим блоком знаходяться кнопки переходу на наступну та попередню сторінки каталогу, де буде відображено інші товари.

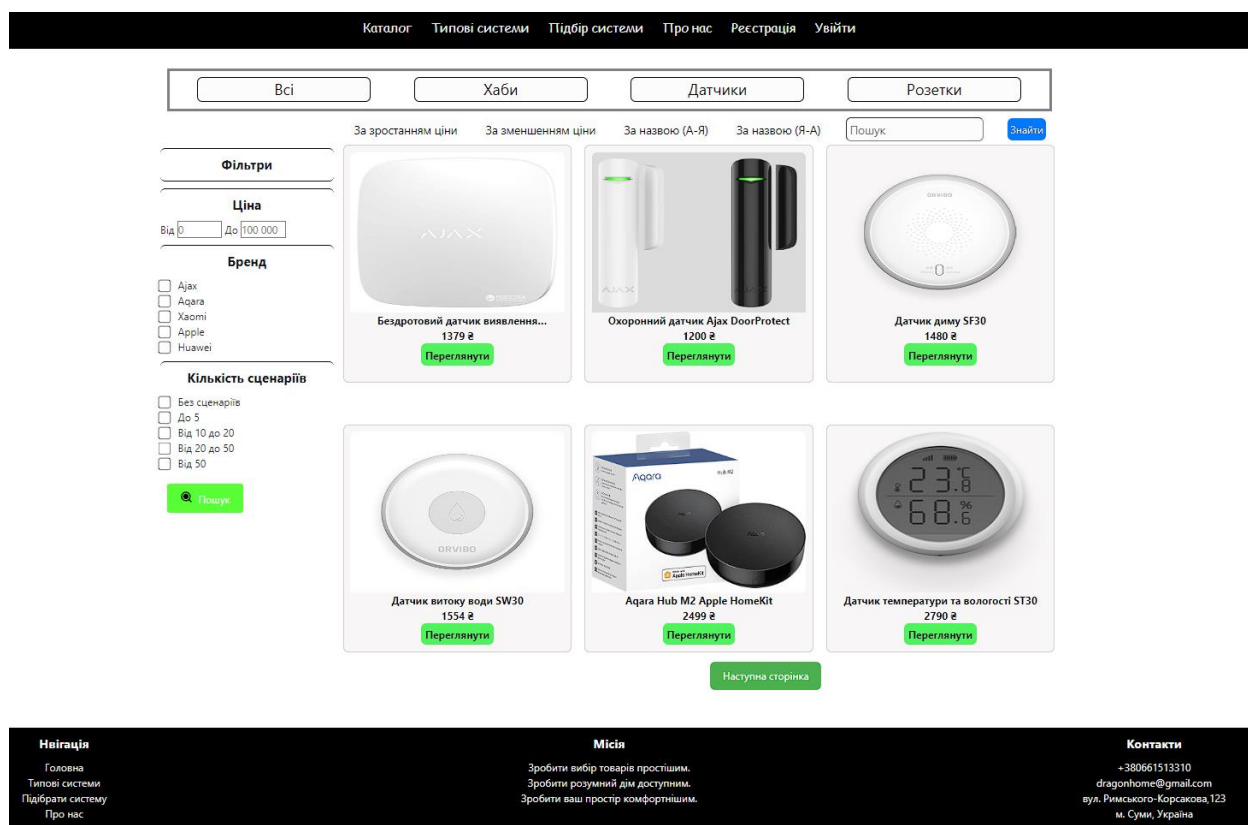


Рисунок 3.8 – Головна сторінка web-додатку

Для отримання більшої кількості можливостей слід авторизуватися. Для цього створено сторінку для входу в обліковий запис, яка зображена на рисунку 3.9. На ній потрібно ввести логін та пароль та натиснути кнопку «Увійти». Якщо дані вірні, то вхід буде виконано успішно, користувачу виведеться сторінка каталогу уже з додатковими можливостями, а якщо ні, то буде виведено повідомлення про помилку.

Каталог Типові системи Підбір системи Про нас Реєстрація Увійти

Увійти

Логін

Пароль

Увійти

Немає акаунта? - [Зареєструйтеся](#)

Нвігація <ul style="list-style-type: none">ГоловнаТипові системиПідбрати системуПро нас	Місія <ul style="list-style-type: none">Зробити вибір товарів простішим.Зробити розумний дім доступним.Зробити ваш простір комфортнішим.	Контакти <ul style="list-style-type: none">+380661513310dragonhome@gmail.comвул. Римського-Корсакова,123м. Суми, Україна
---	---	--

Рисунок 3.9 – Сторінка авторизації

Якщо учасник системи ще не зареєстрований, під кнопкою увійти є посилання на форму створення акаунту, зображену на рисунку 3.10. Вона містить більше полів, в які потрібно ввести дані, що будуть застосовані пізніше, при користуванні web-додатком, такі як повне ім'я та електронна адреса. Також слід двічі ввести пароль, для того, щоб бути впевненим у його правильності.

Зареєструйтеся

Повне ім'я

Логін

Електронна адреса

Пароль

Підтвердження пароля

[Зареєструватися](#)

[Є акаунт? - Увійти](#)

Рисунок 3.10 – Сторінка створення облікового запису

Якщо авторизація пройшла успішно, то в меню у верхній частині web-додатку зміняться надписи. З'явиться кнопка виходу з облікового запису, кнопка перегляду інформації про користувача, кошик із товарами для замовлення та панелью адміністратора, звичайно, якщо в наявності відповідний рівень доступу. Саме такий випадок наведено на рисунку 3.11.

Каталог Типові системи Підбір системи Про нас Кошик Профіль Панель адміністратора Вихід

Фільтри

Ціна
Від До

Бренд

Ajax
 Aqara
 Xiaomi
 Apple
 Huawei

Кількість сценаріїв


Без сценаріїв
 До 5
 Від 10 до 20
 Від 20 до 50
 Від 50

[Пошук](#)

Всі
Хаби
Датчики
Розетки

За зростанням ціни
За зменшенням ціни
За назвою (А-Я)
За назвою (Я-А)


[Знайти](#)



Бездротовий датчик виявлення...

1379 ₴


[Переглянути](#) [Редагувати](#)



Охоронний датчик Ajax DoorProtect

1200 ₴


[Переглянути](#) [Редагувати](#)



Датчик диму SF30

1480 ₴


[Переглянути](#) [Редагувати](#)



Датчик витoku води SW30

1554 ₴


[Переглянути](#) [Редагувати](#)



Aqara Hub M2 Apple HomeKit

2499 ₴

[Переглянути](#) [Редагувати](#)



Датчик температури та вологості ST30

2790 ₴

[Переглянути](#) [Редагувати](#)

[Наступна сторінка](#)

Рисунок 3.11 – Головна сторінка та меню, після входу в обліковий запис

Можна помітити, що на головній сторінці, для кожної картки приладу, біля кнопки перегляду, в адміністратора є кнопка для редагування інформації про товар. Якщо користувач web-додатку натисне на неї, то буде відкрито сторінку редагування даних про товар. Приклад можна переглянути на рисунку 3.12.

Початкові дані про товар зберігаються в полях, куди можна вводити нову інформацію, виключеннями є поля зображення та категорії. При натисканні кнопки оновити дані зміняться та відкриється головна сторінка web-додатку.

Каталог Типові системи Підбір системи Про нас Кошик Профіль Панель адміністратора Вихід

Оновити дані про товар

Назва товару:

Опис:

Бренд:

Колір:

Тип з'єднання:

Кількість сценаріїв:

Специфікація:

Категорія:

Ціна:


Зображення:
 Файл не выбран

Рисунок 3.12 – Сторінка оновлення даних про товар

Якщо натиснути на кнопку «Переглянути» в блоці картки товару, на головній сторінці відбудеться перехід до більш детальної інформації про обраний пристрій. В даній частині web-додатку можна ознайомитися зі специфікацією та детальним описом товару, дізнатися його бренд, колір, тип з'єднання та кількість сценаріїв.

На цій сторінці також можна додати прилад до кошику. Якщо користувач має роль адміністратора, то він може активувати товар, тим самим, зробивши його видимим для всіх клієнтів, або ж деактивувати його, щоб він відображався тільки для адміністрації. На рисунку 3.13 зображено приклад такої сторінки.

Каталог
Типові системи
Підбір системи
Про нас
Кошик
Профіль
Панель адміністратора
Вихід



Охоронний датчик Ajax DoorProtect

Бренд: Ajax
 Колір: Чорний та білий
 З'єднання: Jeweller
 Кількість сценаріїв: 0

1200 €

Деактивувати
Додати до кошика

Опис товару

Бездротовий датчик відкриття дверей/вікна Ajax DoorProtect strong призначений для детектування відкриття дверей, вікон і т.д. Датчик обладнаний клемною колодкою для підключення додаткових дротяних датчиків, в тому числі дротяних датчиків відкриття, призначених для установки на металеві ворота і люки. Конструктивно датчик складається з двох частин - магніту і блоку з герконом. Передбачено два варіанти магніту - для паралельної (великий магніт) і перпендикулярної (малий магніт) установки. Принцип роботи датчика відкриття заснований на властивостях геркона - елемента, здатного ставати провідником під впливом магнітного поля. У нормальному стані магніт і блок з герконом зімкнуті. Як тільки двері, на яких встановлений датчик, відкриваються - магніт віддаляється від геркона, контакти розмикаються і геркон перестає проводити струм. При закритті дверей відбуваються зворотні процеси: магніт наближається до геркона, контакти замикаються і геркон починає проводити струм. В обох випадках датчик спрацьовує і миттєво відсилає повідомлення про тривогу на центральний блок. У датчику встановлений англійський геркон, який не залипає і відпрацьовує понад мільйон замикань/розмикань.

Специфікації товару

Поріг спрацьовування датчика - 2 см.
 Корпус датчика захищений від розтину тампером.
 Датчик вимірює температуру навколишнього середовища і передає інформацію про поточну температуру.
 Датчик відкриття.

Навігація

- Головна
- Типові системи
- Підбрати систему
- Про нас

Місія

- Зробити вибір товарів простішим.
- Зробити розумний дім доступним.
- Зробити ваш простір комфортнішим.









Контакти

+380661513310
 dragonhome@gmail.com
 вул. Римського-Корсакова,123
 м. Сімферополь

Рисунок 3.13 – Сторінка товару

Сторінка типових систем містить набори розумних приладів. Саме вона зображена на рисунку 3.14. Будь-який набір, як і товар в каталозі можна переглянути, для ознайомлення з детальною інформацією про нього.

Приклад сторінки з детальною інформацією про систему розумний будинок приведено на рисунку нижче. Вона містить кнопку додавання товару до кошику, але на ній відсутня можливість приховати систему. Розділ комплектація містить дані про пристрої, які входять в систему. Як і для приладів в каталозі, наявний детальний словесний опис, а також кілька стислих та конкретних характеристик, таких як з'єднання та тип системи.

 <p>Система захисту від витоків газу та... 7249 ₴ Переглянути</p>	 <p>Комплект незалежної системи охорон... 9247 ₴ Переглянути</p>	 <p>Philips Hue White (Bridge, Dimmer,... 5090 ₴ Переглянути</p>	 <p>Управління теплою підлогою Tervix... 16849 ₴ Переглянути</p>
 <p>Комплект "Управління Опаленням... 3885 ₴ Переглянути</p>	 <p>Розумна система освітлення Nanoleaf.. 18269 ₴ Переглянути</p>	 <p>Bron Comfort 24907 ₴ Переглянути</p>	 <p>Комплект для Розумного будинку... 5485 ₴ Переглянути</p>

[Наступна сторінка](#)**Нвігація**

Головна
Типові системи
Підбрати систему

Місія

Зробити вибір товарів простішим.
Зробити розумний дім доступним.
Зробити ваш простір комфортнішим.

Контакти

+380661513310
dragonhome@gmail.com
вул. Римського-Корсакова, 123

Рисунок 3.14 – Сторінка типових систем розумний будинок

**Комплект незалежної системи охорони квартири**

Тип системи: Захист
З'єднання: ZigBee, WiFi
Зона роботи (в метрах): 60
Автономність хабу (в годинах): 0

9247 ₴

[Додати до кошика](#)**Опис товару**

Система охорони квартири/будинку від компанії Tervix - це частина розумного будинку направлена на захист помешкання, та забезпечення контролю над важливими зонами помешкання. Система працює по енергоефективному протоколу передачі даних ZigBee. Разом з цією системою Ви отримуєте «фундамент» Розумного будинку, адже контролер, що йде в комплекті, може одночасно підтримувати до 85 різних пристроїв ZigBee. В комплекті йде контролер, датчик руху, датчик відкриття дверей вікон, сирена та WiFi камера. Всі ці пристрої працюють по протоколу ZigBee. І після налаштування функціонують автономно незалежно від наявності мережі WiFi та інтернету. В комплекті системи також йде WiFi камера з датчиком руху та записом подій на SD карту до 64Гб. Після налаштування систему можна буде активувати або деактивувати. Коли система активована, то при спрацюванні датчика руху або датчика відкриття дверей прийде повідомлення на телефон та спрацює сирена. Тривалість роботи сирени налаштується. WiFi камера буде записувати всі події, а якщо активований датчик руху, то буде надсилати повідомлення на телефон. Коли система деактивована, то датчики неспрацюють та повідомлення не приходять на телефон, камера не записує.

Комплектація

Контролер безпроводний Tervix ProLine ZigBee Gateway
Розумна сирена Tervix Pro Line ZigBee Siren
Внутрішня панорамна WiFi камера Tervix Pro Line Minion Cam WiFi
Безпроводний датчик руху Tervix Pro Line ZigBee EYE PIR Sensor
Безпроводний датчик відкриття вікон / дверей Tervix Pro Line ZigBee

Нвігація

Головна
Типові системи
Підбрати систему

Місія

Зробити вибір товарів простішим.
Зробити розумний дім доступним.
Зробити ваш простір комфортнішим.

Контакти

+380661513310
dragonhome@gmail.com
вул. Римського-Корсакова, 123

Рисунок 3.15 – Сторінка системи розумний будинок

Інструмент для підбору системи можна знайти, натиснувши на пункт «Підбір системи» в головному меню. Першим питанням до користувача буде грошовий діапазон, в якому потрібно знайти систему. Відповідне зображення сторінки можна переглянути на рисунку 3.16. Потрібно обрати на один із трьох варіантів, після чого натиснути на кнопку «Далі».

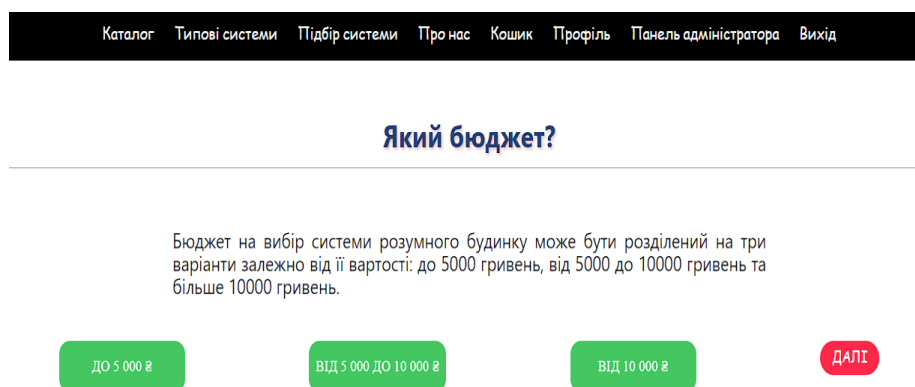


Рисунок 3.16 – Перший вибір в інструменті для підбору системи розумний будинок

Наступним питанням при підборі системи розумний будинок буде тип системи. На вибір два варіанти: моносистеми або ж мультисистеми. Зображення відповідної сторінки на рисунку 3.17.

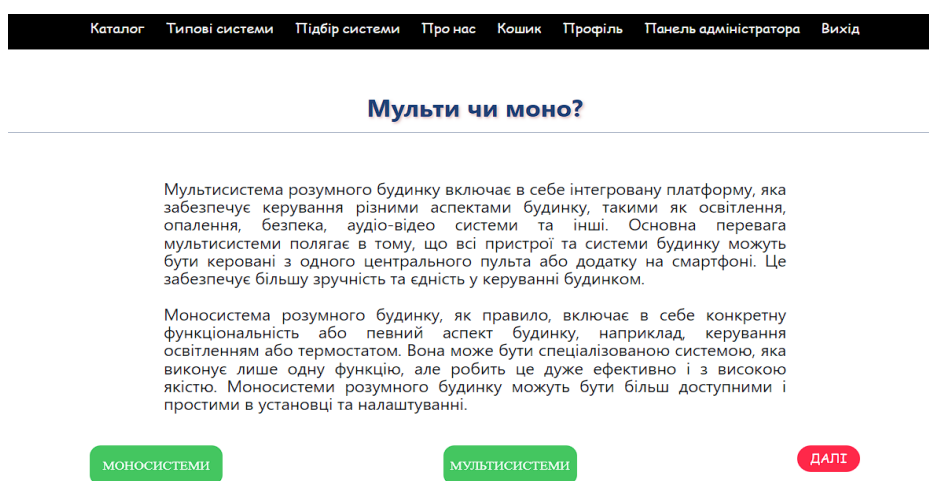


Рисунок 3.17 – Другий вибір в інструменті для підбору системи розумний будинок

Якщо було обрано моносистему, то слід визначитися з її підтипом. Вибір між трьома варіантами: комфорт, економія та захист. Приклад сторінки зображено на рисунку 3.18. Як і на попередніх етапах, перед вибором можна коротко ознайомитися, що собою являє кожен із запропонованих варіантів.

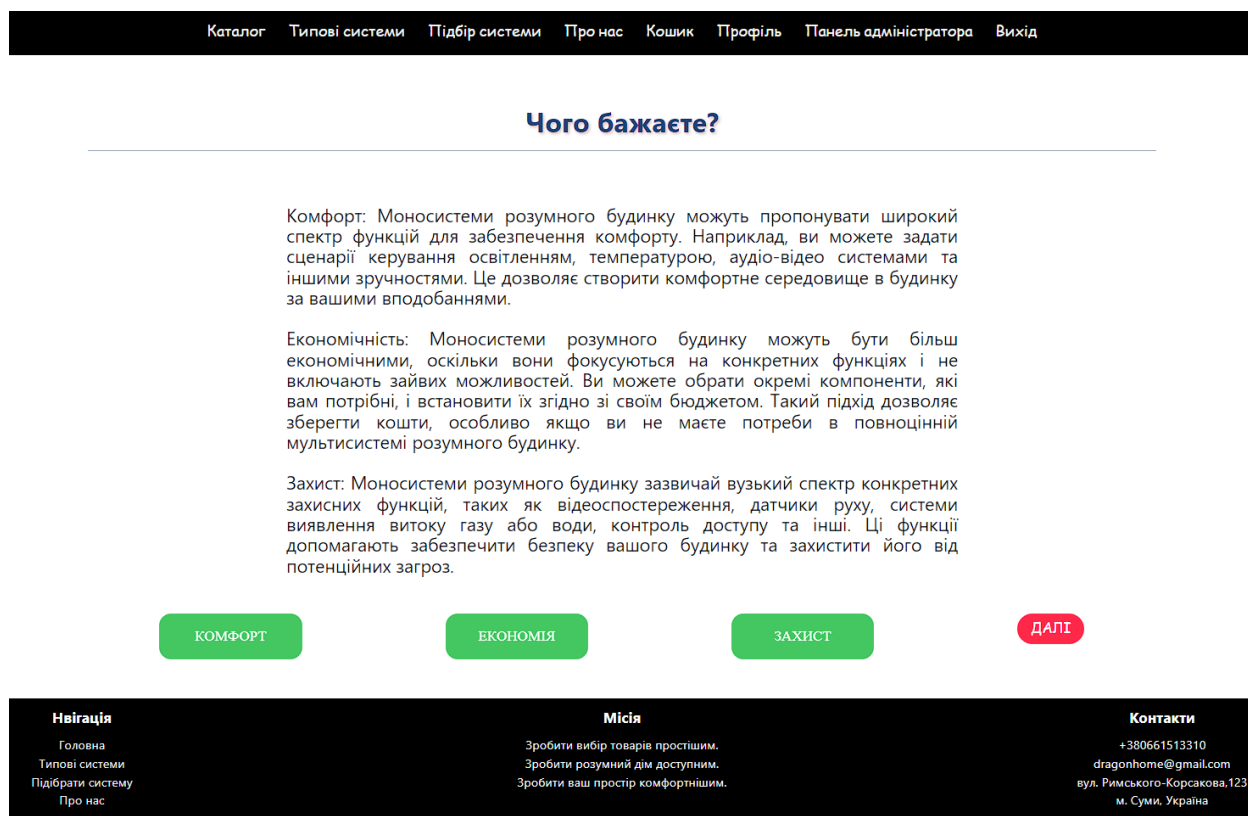


Рисунок 3.18 – Останній вибір в інструменті для підбору системи розумний будинок

Після вибору підтипу або ж типу (у випадку, якщо було обрано мультисистеми) буде відображено кілька варіантів, які відповідатимуть вимогам клієнта. Він може ознайомитися з ними та зробити вибір того, що йому найбільше сподобається. Даний підхід дозволяє скоротити діапазон запропонованих систем, але при цьому залишає подальшу свободу вибору для користувача. Приклад використання інструменту зображено на рисунку 3.19.

Теоретично можливо, що системи, які відповідатимуть запиту клієнта, не буде знайдено. У такому випадку буде виведено повідомлення, представлене на

рисунку 3.20. Тоді клієнт розумітиме, що наборів, які відповідають його бажанням, немає в наявності.

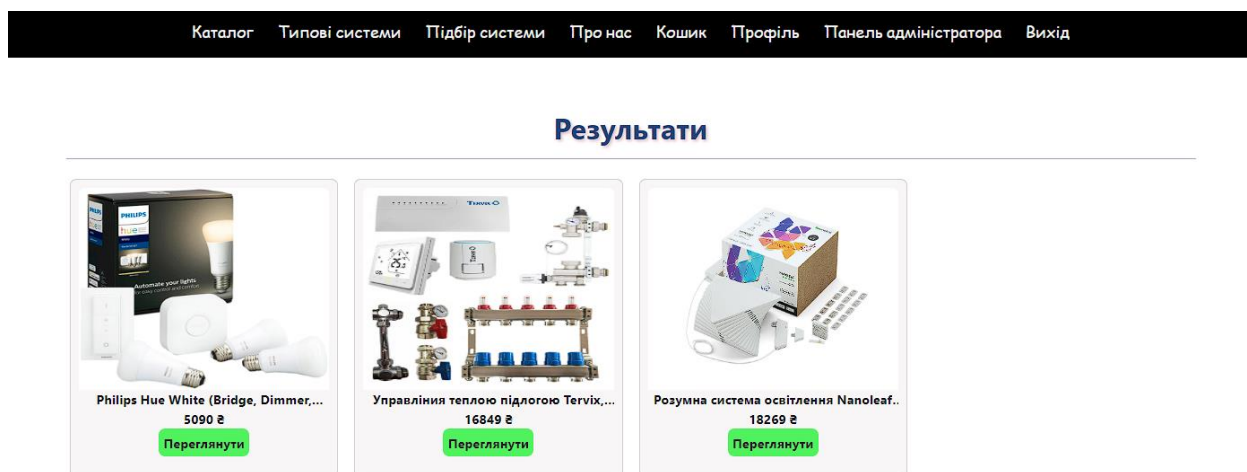


Рисунок 3.19 – Перший приклад результату використання інструменту для підбору системи розумний будинок

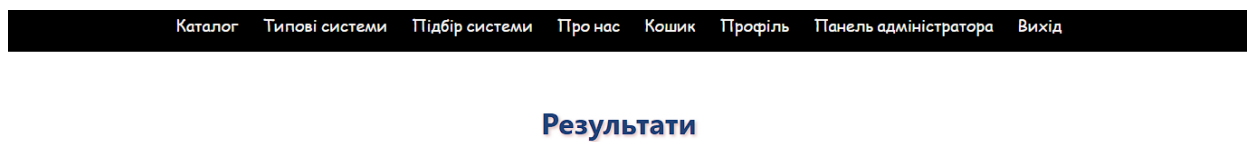


Рисунок 3.20 – Другий приклад результату використання інструменту для підбору системи розумний будинок

Після того, як всі необхідні товари було додано до кошику, можна перейти до сторінки оформлення замовлення і ознайомитися з ними та загальною ціною потенційної покупки. Приклад зображено на рисунку 3.21. Для створення замовлення необхідно ввести контактні дані, такі як номер телефону та адресу доставки, і

натиснути на кнопку «Створити замовлення». Або ж очистити корзину натиснувши відповідну кнопку.

Після оформлення замовлення користувач отримає повідомлення із номером замовлення та додатковою інформацією. Приклад зображено на рисунку 3.22.

Назва товару	Кількість	Ціна за одиницю	Ціна за все
Бездротовий датчик виявлення затоплення Ajax LeaksProtect White	2	1379 ₴	2758 ₴
Датчик диму SF30	1	1480 ₴	1480 ₴
Комплект розумного будинку Fibaro Starter Kit	1	24690 ₴	24690 ₴

Загальна вартість замовлення: 28928 ₴

Номер телефону

+380661516620

Адреса доставки

Нова пошта. Відділення номер 1, с. Синівка, Роменський район, Сумська область

Нвігація Головна Типові системи Підбрати систему Про нас	Місія Зробити вибір товарів простішим. Зробити розумний дім доступним. Зробити ваш простір комфортнішим.	Контакти +380661513310 dragonhome@gmail.com вул. Римського-Корсакова,123 м. Суми, Україна
---	--	--

Рисунок 3.21 – Сторінка корзини товарів

Назва товару	Кількість	Ціна за одиницю	Ціна за все
Бездротовий датчик виявлення затоплення Ajax LeaksProtect White	2	1379 ₴	2758 ₴

✕

Замовлення номер 49 створено!

Переглянути інформацію про нього ви можете на сторінці профілю.

Також інформацію про замовлення надіслано на вашу електронну адресу.

Загальна вартість замовлення: 28928 ₴

Рисунок 3.22 – Повідомлення при успішному оформленні замовлення

Після закриття вікна зі сповіщенням сторінка оформлення замовлення матиме вигляд, як на рисунку 3.23. Те ж саме користувач побачить, якщо просто не додаватиме товарів до кошику.

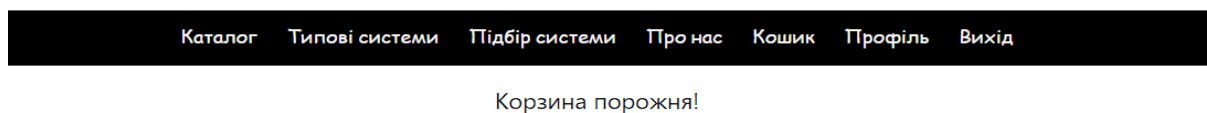


Рисунок 3.23 – Сторінка пустої кошика товарів

Як і було вказано в повідомленні у модальному вікні, ознайомитися з певними деталями про замовлення можна на сторінці користувача, як показано на рисунку 3.24 або ж в листі, який приходить на електронну пошту, як на рисунку 3.25.

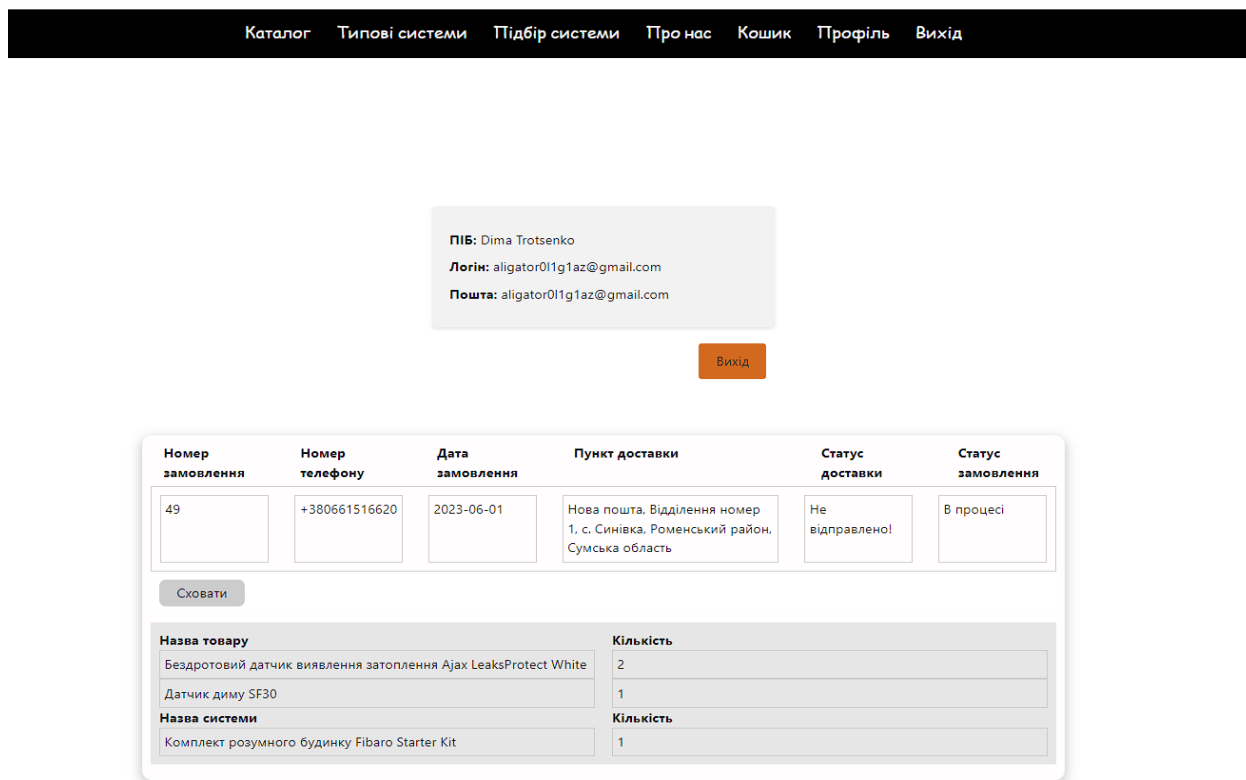


Рисунок 3.24 – Сторінка профілю користувача

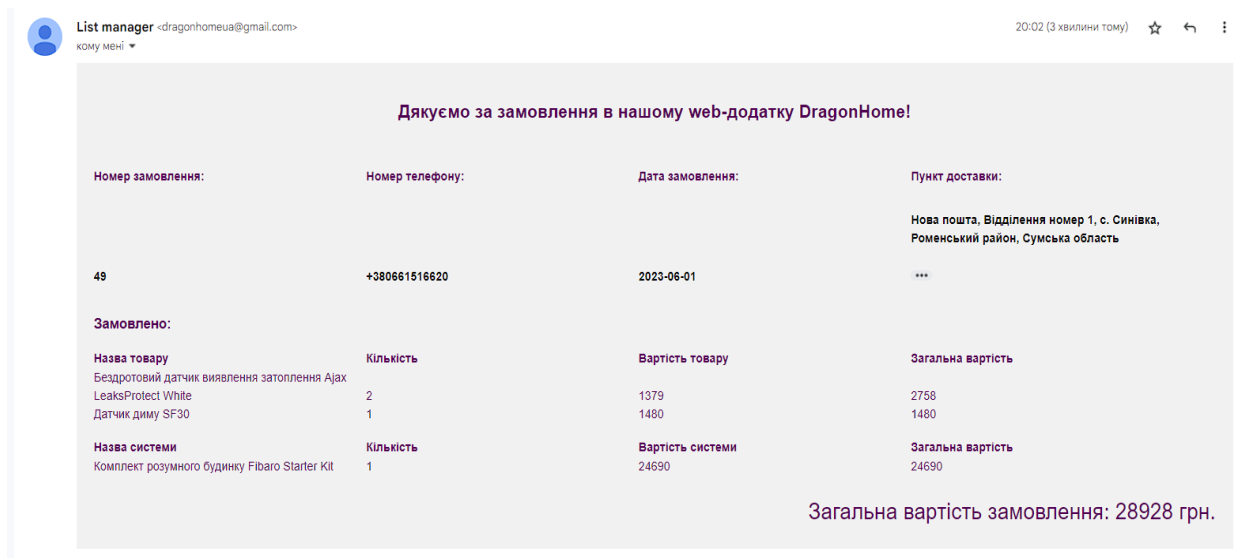


Рисунок 3.25 – Приклад листа, отриманого після оформлення замовлення

Для адміністрації створено сторінку, на якій відображаються деталі про створювані користувачами замовлення. Приклад можна переглянути на рисунку 3.26. Вона містить поле для пошуку по номеру замовлення.

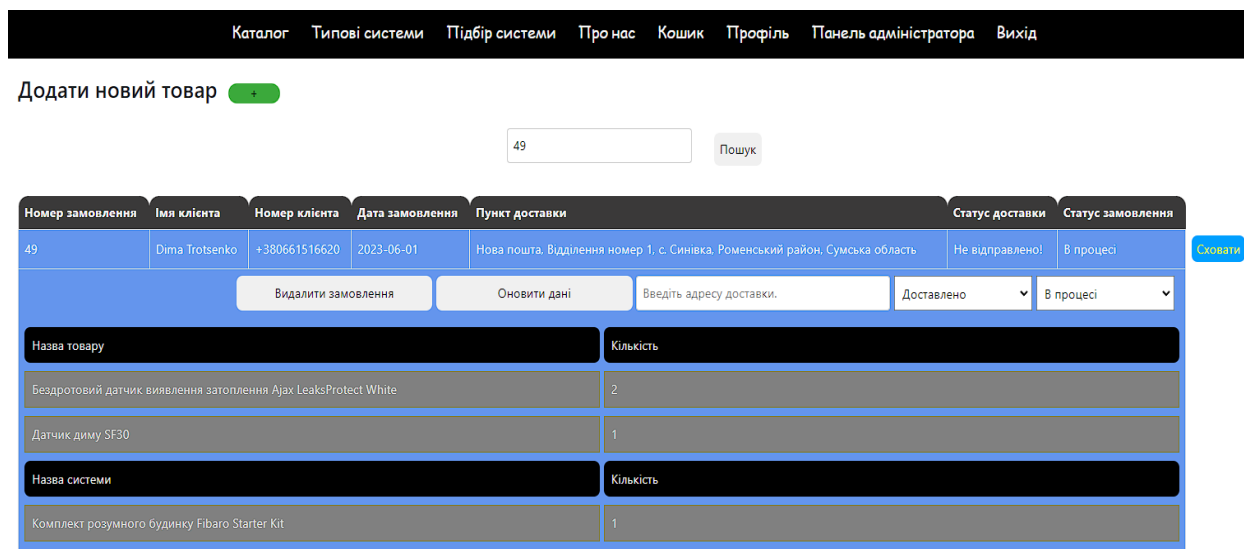


Рисунок 3.26 – Пошук замовлення за номером в панелі адміністратора

Адміністратор має можливість видалити замовлення. У такому разі буде виведено повідомлення про виконану дію, як на рисунку 3.27, а результат пошуку неіснуючого замовлення зображено на рисунку 3.28.

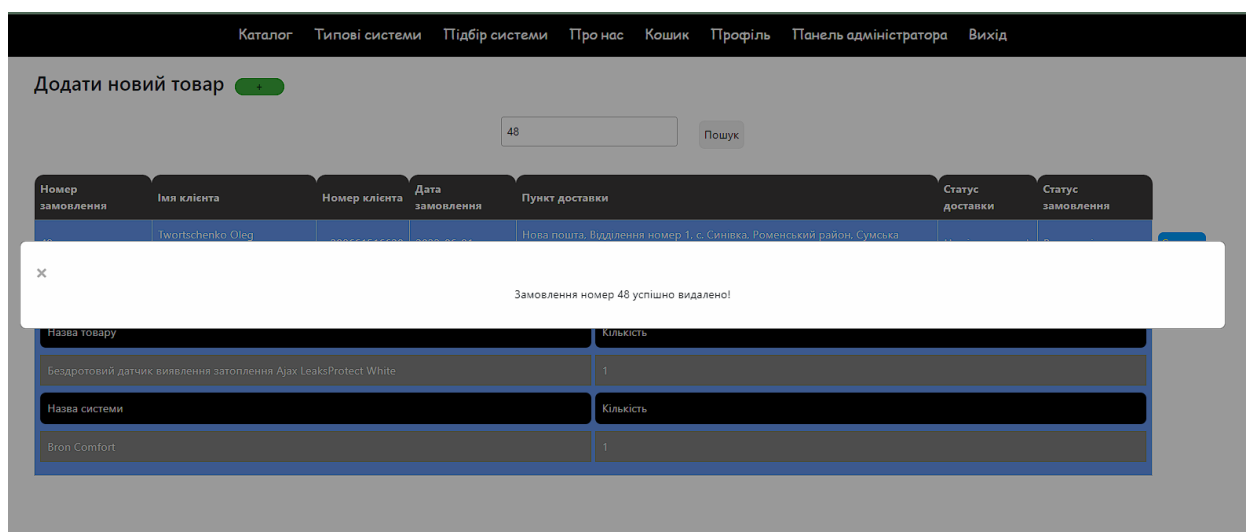


Рисунок 3.27 – Приклад видалення замовлення

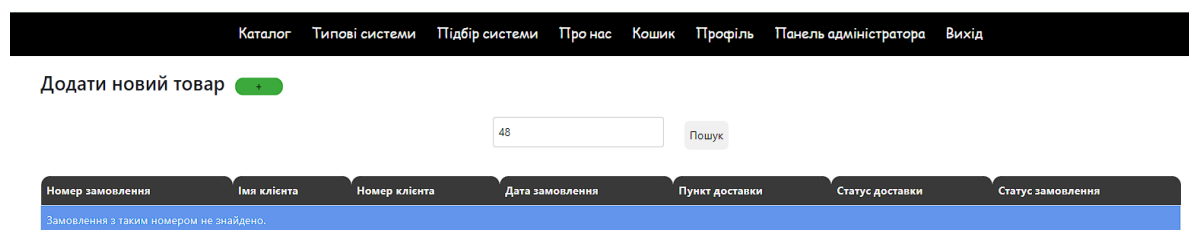


Рисунок 3.28 – Приклад результату пошуку замовлення, якого не існує

Якщо не використовувати поле пошуку, сторінка панелі адміністратора виглядатиме як на рисунку 3.29. Вона відображає коротку інформацію про 5 замовлень. Знизу є кнопки для завантаження більшої кількості записів. Можна ознайомитися детальніше з кожним із них. При цьому відкривається можливість для зміни інформації про статуси доставки та замовлення і місця відправки товарів. Після виконання необхідних операцій можна знову приховати дані, натиснувши кнопку «Сховати».

Вгорі розміщено блок для додавання товару. При натисканні на зелену кнопку біля напису відобразяться поля для додавання нового товару до каталогу розумних приладів. Дана форма представлена на рисунку 3.30.

Каталог Типові системи Підбір системи Про нас Кошик Профіль Панель адміністратора Вихід

Додати новий товар +

Введіть номер замовлення

Номер замовлення	Ім'я клієнта	Номер клієнта	Дата замовлення	Пункт доставки	Статус доставки	Статус замовлення	
2	Petrov Miha Olekseevich	+380661514412	2023-05-15	Ярослава Мудрого, ран. Пролетарська, 50. Суми, Сумська область, 40009	Доставлено	В процесі	Детальніше
3	Twortschenko Oleg Viktorovich	+380967891432	2023-05-16	Сумська область м.Суми проспект Михайла Лушпи, 13	Не відправлено	В процесі	Сховати

Видалити замовлення Введіть адресу доставки.

Доставлено

Доставлено
Відправлено
Не відправлено

Назва товару	Кількість
Датчик витoku води SW30	3
Бездротовий датчик виявлення затоплення Ajax LeaksProtect White	5

Назва системи	Кількість
Комлект незалежної системи охорони квартири	1
Система захисту від витoku газу та задимлення (ZigBee)	1

4	Twortschenko Oleg Viktorovich	+380966334516	2023-05-23	Сумська область с. Суха Грунь, Мегадева 8	Доставлено	Виконано	Детальніше
5	Twortschenko Oleg Viktorovich	+380661510372	2023-05-24	Ярослава Мудрого, ран. Пролетарська, 50. Суми, Сумська область, 40009	Не відправлено	В процесі	Детальніше
8	Twortschenko Oleg Viktorovich	+380954243455	2023-05-27	Ярослава Мудрого, ран. Пролетарська, 50. Суми, Сумська область, 40009	Не відправлено	В процесі	Детальніше

Завантажити попередні

Рисунок 3.29 – Сторінка панелі адміністратора

Каталог Типові системи Підбір системи Про нас Кошик Профіль Панель адміністратора Вихід

Додати новий товар +

Назва товару:

Опис:

Бренд:

Колір:

Тип з'єднання:

Кількість сценаріїв:

Специфікація:

Категорія:

Ціна:

Зображення: Файл не вибран

Рисунок 3.30 – Форма додавання товару до каталогу

На сторінці «Про нас» відображена загальна інформація про web-додаток та послуги, які він надає. Зміст відображено на рисунку 3.31. Контактні дані тут не дублюються, адже вони розміщені в самому нижньому блоці кожної сторінки.

Каталог
Типові системи
Підбір системи
Про нас
Кошик
Профіль
Панель адміністратора
Вихід

ПРО НАС

Ласкаво просимо до нашого каталогу розумних приладів та підбору систем розумного будинку! Ми - команда експертів з розумних технологій, які працюють з метою забезпечити вам максимальну зручність, безпеку та енергоефективність в вашому будинку.

Наша головна мета

Зробити ваш будинок більш безпечним, зручним та ефективним. Ми пропонуємо широкій вибір розумних пристроїв, від освітлення та термостатів до систем безпеки та розумних домоопікунів. Завдяки інтеграції цих пристроїв в одну систему, ви отримуєте повний контроль над вашим будинком, навіть коли ви знаходитесь далеко від нього.

Наші послуги

КАТАЛОГ РОЗУМНИХ ПРИЛАДІВ. У нас ви знайдете широкій вибір розумних пристроїв від провідних виробників. Від датчиків руху та води до автоматичних замків та відеодомофонів, у нас є все, що вам потрібно для створення розумного будинку за вашими вподобаннями.

ПІДБІР СИСТЕМ РОЗУМНОГО БУДИНКУ. Наш web-додаток може допомогти вам з підбором оптимальної системи розумного будинку для вашого дому. Враховуючи ваші потреби та бюджет, він запропонує вам рішення, які найкраще відповідають вашим вимогам.

Нвігація

- Головна
- Типові системи
- Підбрати систему
- Про нас

Місія

- Зробити вибір товарів простішим.
- Зробити розумний дім доступним.
- Зробити ваш простір комфортнішим.

Контакти

- +380661513310
- dragonhome@gmail.com
- вул. Римського-Корсакова,123
- м. Суми, Україна

Рисунок 3.31 – Сторінка «Про нас»

3.4 Тестування web-додатку

Тестування web-додатків є важливою складовою процесу розробки, що дозволяє перевірити функціональність та надійність програмних продуктів перед їх впровадженням. Одним з ключових аспектів такого тестування є перевірка посилань та кнопок на формах, оскільки ці елементи є важливими для взаємодії користувача з web-додатком.

Це необхідно для того, щоб переконатися, що всі гіперпосилання на сторінках працюють належним чином і ведуть до очікуваних ресурсів. Для цього буде застосовано підходи перелічені нижче:

- web-додаток повинен бути обстежений на наявність всіх url на сторінках, включаючи посилання на інші сторінки в рамках програмного продукту та зовнішні ресурси;
- кожне посилання повинно бути перевірене на правильність, тобто необхідно переконатися, що воно веде до очікуваного ресурсу для уникнення "потрібної сторінки не знайдено" або "помилкової сторінки".
- крім правильності посилань, важливо переконатися, що вони є активними, тобто можуть бути відкриті та завантажені користувачем, що включає перевірку на наявність HTTP-статусу 200 та відповідних обробників для посилань.

Перевірка кнопок на формах web-додатків є важливою, оскільки вони виконують дії користувача, сприяють обробці та надсиланню даних. Основні аспекти їх тестування включають перелічені нижче.

Перевірка наявності кнопок. Кожна форма повинна містити необхідні кнопки для взаємодії з користувачем, наприклад, кнопку "Відправити" для надсилання форми або кнопки "Скасувати" для відміни операції.

Перевірка функціональності кнопок. Кнопки повинні бути перевірені на правильність виконання функцій, які вони мають здійснювати. Наприклад, кнопка "Відправити" повинна відправляти дані форми на сервер, а кнопка "Скасувати" повинна скасовувати поточну операцію.

Перевірка стану кнопок. Кнопки можуть мати різні стани, такі як активна, неактивна або обрана. Ці стани повинні бути перевірені залежно від контексту використання. Наприклад, кнопка може бути неактивною, якщо не всі обов'язкові поля форми заповнені.

Також необхідно перевірити валідацію даних. Деякі сторінки передбачають введення користувачем певної інформації. Потрібно переконатися, що неправильні

дані не будуть додані, не викликатимуть помилку в роботі web-додатку, або не будуть доступні для введення взагалі.

Для тестування посилань було створено таблицю 3.1. Перший стовпчик містить загальну інформацію про посилання, другий про те, чи відбувається перенаправлення на певний ресурс при натисканні. Третій стовпчик вказує, чи відповідає нова сторінка очікуваній, а у четвертому буде відображено дані про помилки, якщо вони виникнуть.

Таблиця 3.1 – Тестування посилань

Посилання	Наявність посилання	Правильність посилання	Помилки при переході
Посилання «Головна» у футері	Є	Правильне	Відсутні
Посилання «Типові системи» у футері	Є	Правильне	Відсутні
Посилання «Підібрати систему» у футері	Є	Правильне	Відсутні
Посилання «Про нас» у футері	Є	Правильне	Відсутні
Посилання «Головна» в меню	Є	Правильне	Відсутні
Посилання «Типові системи» в меню	Є	Правильне	Відсутні
Посилання «Підібрати систему» в меню	Є	Правильне	Відсутні
Посилання «Кошик» в меню	Є	Правильне	Відсутні
Посилання «Профіль» в меню	Є	Правильне	Відсутні
Посилання «Панель Адміністратора» в меню	Є	Правильне	Відсутні
Посилання «Вихід» в меню	Є	Правильне	Відсутні
Посилання «Переглянути» в картці товару	Є	Правильне	Відсутні
Посилання «Переглянути» в картці системи	Є	Правильне	Відсутні
Посилання «Переглянути» в картці системи, отриманої з підбору системи	Є	Правильне	Відсутні
Посилання «Редагувати» в картці товару	Є	Правильне	Відсутні

Також створено таблицю тестування кнопок 3.2. Оскільки даних елементів у web-додатку досить багато, було прийнято рішення розглядати функціонування груп кнопок, в певних випадках.

Таблиця 3.2 – Тестування кнопок

Назва кнопки, або груп кнопок	Функціональність	Результат валідації
Кнопка «Увійти» на сторінці авторизації	Працює	Невірний логін або пароль!
Кнопка «Зареєструватися» на сторінці реєстрації	Працює	Даний логін вже використовується! Введіть ваше повне ім'я! Введіть правильну електронну адресу! Дана пошта вже використовується! Паролі не співпадають!
Кнопка «Вихід» на сторінці профілю	Працює	Не передбачається
Кнопки «Детальніше» та «Сховати» на сторінці профілю	Працюють	Не передбачається
Кнопка «Завантажити більше» на сторінці профілю	Працює	Не передбачається
Кнопка «Очистити корзину» на сторінці корзини	Працює	Не передбачається
Кнопка «Створити замовлення» на сторінці корзини	Працює	Невірний формат адреси проживання! (Довжина від 10 до 250 символів) Невірний формат номеру телефону!
Група кнопок вибору на сторінці підбору системи	Працює	Не передбачається
Кнопка «Далі» на сторінці підбору системи	Працює	Не передбачається
Кнопки «Наступна сторінка» та «Попередня сторінка» на сторінці типових систем	Працює	Не передбачається
Кнопка «Знайти» на головній сторінці	Працює	Не передбачається

Продовження таблиці 3.2 – Тестування кнопок

Назва кнопки, або груп кнопок	Функціональність	Результат валідації
Група кнопок вибору категорії на головній сторінці	Працює	Не передбачається
Група кнопок сортування на головній сторінці	Працює	Не передбачається
Кнопка «Пошук» блоку фільтрів на головній сторінці	Працює	Не передбачається
Кнопки «Наступна сторінка» та «Попередня сторінка» на головній сторінці	Працюють	Не передбачається
Кнопка «Додати» на сторінці панелі адміністратора	Працює	<p>Поле 'Назва товару' є обов'язковим</p> <p>Поле 'Опис' є обов'язковим</p> <p>Поле 'Бренд' є обов'язковим</p> <p>Поле 'Колір' є обов'язковим</p> <p>Поле 'Тип з'єднання' є обов'язковим</p> <p>Поле 'Кількість сценаріїв' є обов'язковим</p> <p>Поле 'Специфікація' є обов'язковим</p> <p>Поле 'Ціна' є обов'язковим</p> <p>Поле 'Зображення' є обов'язковим</p>
Кнопка «Оновити» на сторінці редагування товару	Працює	<p>Поле 'Опис' є обов'язковим</p> <p>Поле 'Бренд' є обов'язковим</p> <p>Поле 'Колір' є обов'язковим</p> <p>Поле 'Тип з'єднання' є обов'язковим</p> <p>Поле 'Кількість сценаріїв' має бути додатнім числом від 0 до 100</p> <p>Поле 'Специфікація' є обов'язковим</p> <p>Поле 'Ціна' є обов'язковим</p> <p>Поле 'Зображення' є обов'язковим</p>
Кнопки «Детальніше» та «Сховати », «Видалити замовлення» та «Оновити дані» на сторінці панелі адміністратора	Працює	Не передбачається
Кнопки «Пошук» на сторінці панелі адміністратора	Працює	Невірний формат номера замовлення!
Кнопки «Завантажити попередні» та «Завантажити наступні» на сторінці панелі адміністратора	Працює	Не передбачається

Можна зробити висновок, що тестування додатку на функціональність кнопок та посилань було виконано успішно. Всі елементи працюють, як передбачалося.

ВИСНОВОК

У результаті роботи над кваліфікаційною роботою бакалавра було створено web-додаток підтримки замовлення систем розумний будинок.

Перед початком розробки було проаналізовано предметну область та обґрунтовано її актуальність. Проаналізовано існуючі підходи до створення програмного продукту зі схожою направленістю. Розглянуто аналоги, які пропонують свої послуги у сфері підбору розумних товарів та їх систем. Після визначення вимог та постановки задачі сформовано основний перелік засобів реалізації.

У другому розділі виконано проектування web-додатку. В першому підрозділі представлено діаграми IDEF0 для головного процесу web-додатку з точки зору клієнта. Проаналізовано варіанти використання розробки та створено відповідні діаграми як для адміністратора, так і для клієнта. Також визначено фізичну модель бази даних та представлено дерево рішення, яким програмний продукт керується при підборі систем розумний будинок.

Третім етапом була практична реалізація web-додатку. Вона полягала у визначенні архітектури розробки та безпосередньому створенні продукту. В третьому підрозділі третього розділу представлено основні сторінки web-додатку, показано його використання. Тестування розробки було успішним, що говорить про її готовність до повноцінної роботи.

Планування робіт та опрацювання ризиків, що могли вплинути на розробку програмного продукту, містяться в додатку Б.

Результати роботи були апробовані на конференції ІМА 2023 в Сумському державному університеті (додаток В).

Лістинг основних модулів розробленого web-додатку представлено у додатку Г.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сосновська О. О., Вакофян В. Г. Індустрія 4.0: сутність і тенденції розвитку. *Бізнес Інформ.* 2032. Т. 14, № 30. С. 137–144. URL: <https://doi.org/10.32983/2222-4459-2022-1-137-144> (дата звернення: 19.04.2023).
2. Sovacool B. K., Furszyfer Del Rio D. D. Smart home technologies in Europe: a critical review of concepts, benefits, risks and policies. *Renewable and sustainable energy reviews.* 2020. Vol. 120. P. 109663. URL: <https://doi.org/10.1016/j.rser.2019.109663> (дата звернення: 10.05.2023).
3. Ноджак Л., Паращич М. Розвиток 4.0 індустрії в Україні: проблеми, перспективи. *Економіка та суспільство.* 2022. № 45. URL: <https://doi.org/10.32782/2524-0072/2022-45-29> (дата звернення: 10.05.2023).
4. Войтко В. В., Позур М. Ю., Денисюк А. В. Загальні підходи до розробки егр-систем, що інтегрують е-commerce системи. *Стан, досягнення та перспективи інформаційних систем і технологій* : Матеріали XXII Всеукр. науково-техн. конф. молодих вчен., аспірантів та студентів, м. Одеса, 21–22 квіт. 2022 р. Одеса, 2022. С. 113–115. URL: https://ontu.edu.ua/download/konfi/2022/Conference_abstract-IT-21-22-04-22.pdf#page=113 (дата звернення: 21.04.2023).
5. Application using MERN Stack / K. S. Santosh et al. *International journal for modern trends in science and technology.* 2022. Vol. 8, no. 6. P. 102–105.
6. Jeyaseelan S. Ecommerce platform for furniture store. *International journal of engineering and management research.* 2022. Vol. 12, no. 6. P. 1–15. URL: <https://ijemr.vandanapublications.com/index.php/ijemr/article/view/1017/832> (дата звернення: 26.04.2023).
7. Malshika M. D. J. Computerized system to manage business functions of e-commerce web application using MERN stack technology. *International journal of engineering and management research.* 22. Т. 12, № 5. С. 472–481.

URL: <https://ijemr.vandanapublications.com/index.php/ijemr/article/view/1003/821> (дата звернення: 26.04.2023).

8. The technical design and implementation of cross-platform industrial product order system / Y. Xue et al. *Journal of cyber security*. 2021. Vol. 3, no. 1. P. 1–10. URL: <https://doi.org/10.32604/jcs.2021.016371> дата звернення: 29.04.2023).

9. Sriram J., Akilan S. S. Online food ordering application for local vendors using PHP. *International journal of health sciences*. 2022. P. 4695–4702. URL: <https://doi.org/10.53730/ijhs.v6ns1.5895> (дата звернення: 19.04.2023).

10. Antony C. Canteen food ordering and managing system. *International journal of current science research and review*. 2022. Vol. 05, no. 06. P. 2194–2199. URL: <https://doi.org/10.47191/ijcsrr/v5-i6-51> (дата звернення: 21.04.2023).

11. Wijegunaratna K. P. Online supermarket web application. *International journal of engineering and management research*. 2022. Vol. 12, no. 6. P. 216–223. URL: <https://ijemr.vandanapublications.com/index.php/ijemr/article/view/1055/863> (дата звернення: 22.04.2023).

12. Журнал iLounge. *iLounge*. URL: <https://ilounge.ua/ua/journal> (дата звернення: 28.04.2023).

13. Обладнання для розумного будинку Розумний дім Domos. *Domos*. URL: <https://domos.ua> (дата звернення: 26.04.2023).

14. XTERM - Магазин розумного будинку та опалювального обладнання. URL: <https://xterm.com.ua/> (дата звернення: 28.04.2023).

15. Sriram J., Akilan S. S. Online food ordering application for local vendors using PHP. *International journal of health sciences*. 2022. P. 4695–4702. URL: <https://doi.org/10.53730/ijhs.v6ns1.5895> (дата звернення: 19.04.2023).

16. Вступ до JavaScript. *Сучасний підручник з JavaScript*. URL: <https://uk.javascript.info/intro> (дата звернення: 05.05.2023).

17. Гурчунова Є. Що таке Figma та як її використовувати digital-фахівцю - AboutMarketing. *AboutMarketing*. URL: <https://aboutmarketing.info/internet-marketynh/instrumenty/shcho-take-figma-ta-yak-yiyi-vykorystovuvaty-digital-fakhivtsyu/> (дата звернення: 07.05.2023).

18. *MAMP & MAMP PRO - Your local web development solution.* URL: <https://www.mamp.info/en/windows/> (дата звернення: 03.05.2023).
19. GitHub - PHPMailer/PHPMailer: The classic email sending library for PHP. *GitHub.* URL: <https://github.com/PHPMailer/PHPMailer> (дата звернення: 04.03.2023).
20. Getting started with functional modeling for system specifications. *Software Delivery Simulator.* URL: <https://softwaresim.com/blog/getting-started-with-functional-modeling-for-system-specifications/> (дата звернення: 22.01.2023).
21. Марголін О. Діаграми UML для моделювання процесів і архітектури проекту. *Evergreen - web розробка і діджиталізація бізнесу за допомогою AI продуктів.* URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення: 12.02.2023).
22. Застосування UML (частина 2). Діаграма послідовності - Sequence Diagram :: державний університет телекомунікацій. *Головна :: Державний університет телекомунікацій.* URL: https://dut.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram_kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy (дата звернення: 22.03.2023).
23. What is data modeling – conceptual, logical, physical models. *Software Testing Help.* URL: https://www.softwaretestinghelp.com/data-modeling-tutorial/#2Logical_Data_Model (дата звернення: 11.03.2023).

ДОДАТОК А**ТЕХНІЧНЕ ЗАВДАННЯ****на розробку****«Web-додаток підтримки замовлення систем розумного будинку»****ПОГОДЖЕНО**

Доцент кафедри інформаційних технологій

_____ Парфененко Ю.В.

Студент групи ІТ-92-1/2

_____ Троценко Д.С.

Суми 2023

1. Призначення й мета створення web-додатку

1.1 Призначення web-додатку

Web-додаток призначений для підтримки замовлення систем розумний будинок.

1.2 Мета створення web-додатку

Мета проекту – реалізація підтримки замовлень систем розумного будинку за рахунок використання розробленого web-додатку, що містить вбудовані функції підбору товарів та систем зі зв'язком між пристроями, розміщення та редагування асортименту адміністратором та замовлення виробів клієнтами. Впровадження додатку підвищить кількість клієнтів магазину та швидкість вибору товарів для оселі.

1.3 Цільова аудиторія

Цільовою аудиторією даного проекту є клієнти, зацікавлені у встановленні систем розумний будинок, оптимізації процесу їх підбору та скорочення витрат часу на вибір і замовлення конкретних розумних приладів. Адміністрація магазину, яка хоче отримати зручну систему управління асортиментом товарів інтелектуального будинку, також зацікавлена в продукті.

2 Вимоги до web-додатку

2.1 Вимоги до web-додатку в цілому

2.1.1 Вимоги до структури функціонування web-додатку

Кінцевий продукт даного проекту має бути представлений web-додатком підтримки замовлення систем розумний будинок, який міститиме каталог товарів, панель адміністратора, для керування асортиментом та інструмент для підбору системи розумний будинок. Також повинні бути реалізовані функція кошику, оформлення замовлення та сторінка з типовими наборами.

2.1.2 Вимоги до персоналу

Персонал закладу не повинен мати особливих технічних навичок для роботи з web-додатком та його підтримкою, за рахунок розробки графічної адміністративної панелі. Єдиною вимогою є наявність навичок користування персональним комп'ютером та web-браузером.

2.1.3 Вимоги до збереження інформації

Уся інформація надана в web-додатку повинна зберігатися у базі даних, що реалізована засобами системи управління базами даних MySQL.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним у мережі Інтернет. Права доступу до інформації розмежовані за групами користувачів: адміністратор, відвідувач та клієнт. Адміністратор має необмежений доступ до даних з правами перегляду, додавання, редагування та видалення товарів та категорій. Доступ до адміністративної панелі надається за спеціальним логіном та паролем.

Відвідувач додатку не матиме доступу до особистого кабінету. У користувача із роллю клієнт спектр доступу до інформації ширший за відвідувача, але менший за адміністратора. До переліку його можливостей входять всі варіанти взаємодії з web-додатком та доступ до особистого кабінету, але не доступна адміністративна панель та можливість змінювати інформаційне наповнення сайту.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

До структури web-додатку входять усі його web-сторінки, які є загальнодоступними, та адміністративна панель, що використовуватиметься персоналом магазину.

Перелік сторінок web-додатку наступний:

- «Головна» web-сторінка є основною, містить перелік товарів, систему фільтрів, категорій та пошук;
- web-сторінка товару, яка містить його детальний опис;
- web-сторінка «Типові набори», містить декілька типових наборів систем, які можна відсортувати за параметрами;
- web-сторінка набору, яка містить перелік пристроїв певного набору;

- web-сторінка «Підібрати систему», містить декілька змінюваних параметрів, на основі яких виконується індивідуальний підбір системи;
- web-сторінка «Про нас», містить інформацію про магазин;
- web-сторінка реєстрації містить форму реєстрації;
- web-сторінка авторизації слугує для авторизації користувача та адміністратора;
- web-сторінка профілю, на якій міститься інформація про обліковий запис;
- web-сторінка кошика, де знаходяться всі товари, що очікують на підтвердження замовлення та самі інструменти для його оформлення;
- web-сторінка адміністративної панелі слугує для управління товарами.

Кожна сторінка сайту матиме незмінювані верхній та нижній блоки. У верхній частині буде розміщено логотип, меню, значок кошику, кнопки для входу або реєстрації чи кнопка профілю користувача та кнопка виходу з облікового запису. Нижня частина міститиме контактну інформацію та навігацію по сторінкам web-додатку.

2.2.2 Навігація

Для зручної навігації повинно бути створене меню, що забезпечить швидке переміщення користувачів по всім доступним сторінкам web-додатку. Меню має бути закріплене і розташовуватися зверху та знизу на кожній сторінці.

2.2.3 Управління контентом

Управління товарами web-додатку має здійснюватися через адміністративну панель. Текстовий опис та зображення продукції повинні зберігатись у базі даних, як і повноцінні варіанти систем розумний будинок.

2.2.4 Дизайн web-додатку

Дизайн web-додатку має бути виконаний у мінімалістичному стилі. Під час розробки web-додатку треба використовувати переважно білий чорний та зелений кольори.

Види та розміри шрифтів повинні бути комфортними для перегляду. Інформаційні блоки, графічні матеріали та інші елементи сайту повинні мати зручне і логічне розташування. Орієнтовний шаблон майбутнього програмного продукту зображено на рисунку А.1

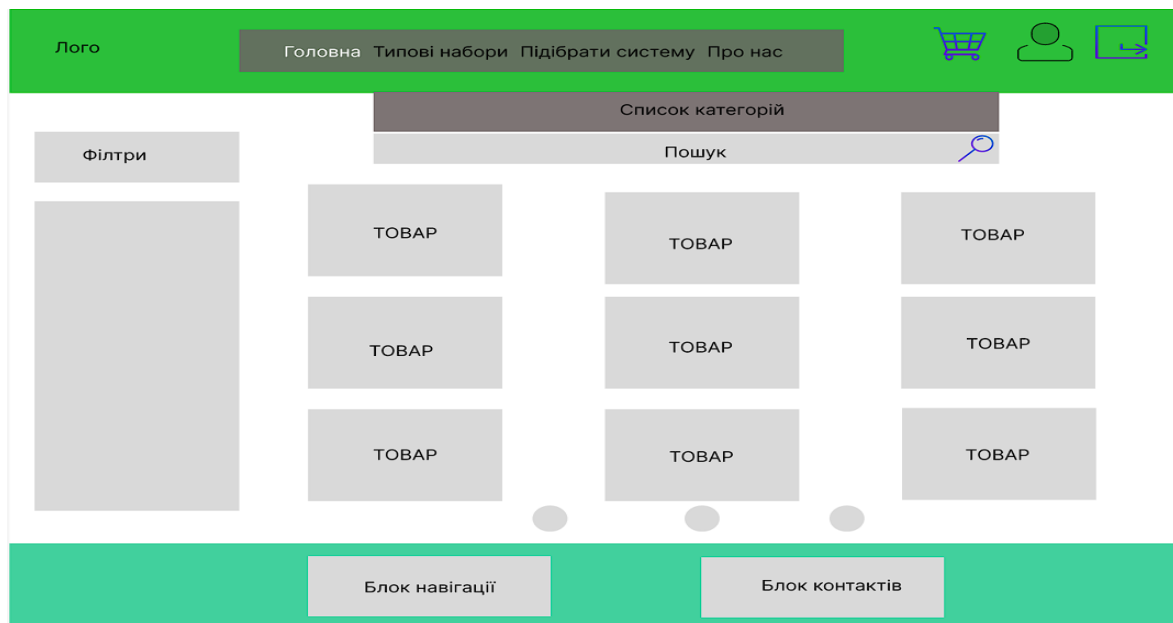


Рисунок А.1 – Схема головної сторінки

2.2.5 Система навігації (карта web-додатку)

Карта web-додатку зображена на рисунку А.2.

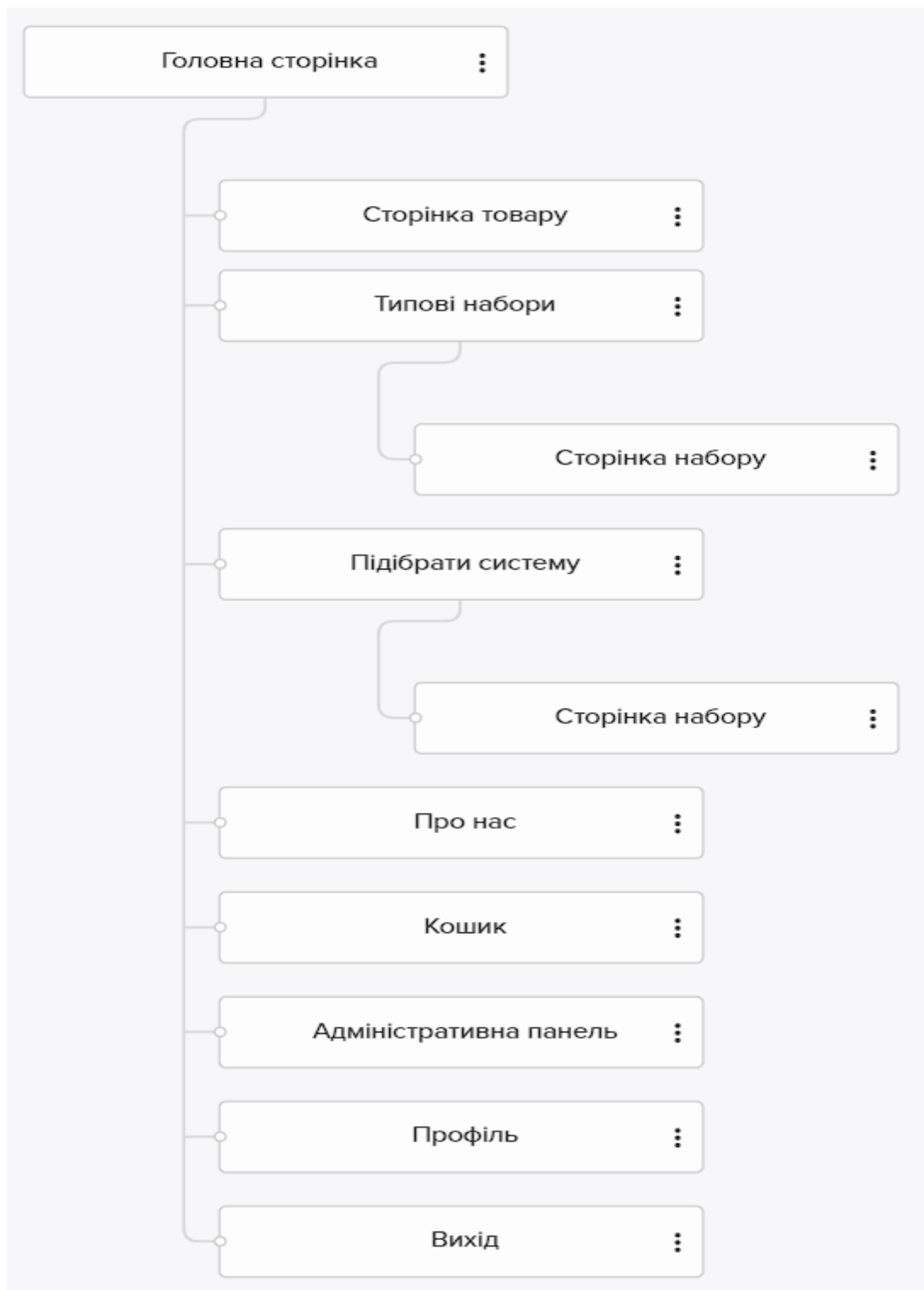


Рисунок А.2 – Карта web-додатку

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Реєстрація облікового запису	Відвідувач
UN-02	Вхід в обліковий запис	Клієнт
UN-03	Перегляд інформації про магазин	Клієнт, Відвідувач
UN-04	Перегляд інформації про товари	Клієнт, відвідувач
UN-05	Перегляд особистих даних	Клієнт
UN-06	Перегляд списку товарів на замовлення	Клієнт, відвідувач
UN-07	Перегляд типових наборів систем розумного будинку	Клієнт, відвідувач
UN-08	Індивідуальний підбір системи за введеними параметрами	Клієнт, відвідувач
UN-09	Замовлення продукції онлайн	Клієнт, відвідувач
UN-10	Введення та редагування даних	Адміністратор

2.3.2 Функціональні вимоги

Проаналізувавши потреби користувачів та персоналу закладу було визначено наступні вимоги:

- наявність реєстрації відвідувачів та авторизації клієнтів;

- поділ товарів на категорії;
- набір інструментів для фільтрування товарів;
- наявність переліку типових наборів;
- наявність інструменту для підбору індивідуальної системи;
- можливість клієнта здійснювати замовлення самостійно;
- наявність повної інформації про продукцію магазину;
- наявність сторінки з персональними даними клієнта;
- можливість додавати товари до кошика;
- наявність панелі адміністратора для роботи з товарами.

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Модуль авторизації	М	Надає можливість авторизуватися та отримати доступ до функціоналу Web-додатку
SR -02	База даних з інформацією про користувачів, товари та системи	М	Відповідає за зберігання інформації про зареєстрованих користувачів, товари та перелік готових систем
SR -03	Каталог товарів	М	Формує можливість перегляду переліку наявних товарів
SR -04	Каталог типових наборів	S	Формує перелік типових систем розумний будинок для перегляду

Продовження таблиці А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR -05	Модуль підбору систем розумний будинок	М	Надає можливість підібрати систему розумний будинок, відповідно до потреб клієнта
SR -06	Адміністративна панель	М	Відповідає за наповнення каталогів товарів та типових систем контентом та його редагування
SR -07	Модуль кошика	С	Надає можливість створювати список товарів для замовлення
SR -08	Модуль оформлення замовлення	М	Надає можливість оформлювати інформацію про замовлення

Умовні позначення в таблиці А.2:

- Must have (M) – вимоги, які повинні бути реалізовані в системі;
- Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;
- Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація web-додатку відбувається з використанням:

- Visual Studio Code;
- PHP 8.0.5;

– MySQL 8.0.

2.4.2 Вимоги до лінгвістичного забезпечення

Web-додатку має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Для забезпечення стабільної роботи web-додатку web-браузер має бути Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

3 Склад і зміст робіт зі створення web-додатку для підтримки замовлень систем розумного будинку

Детальний опис етапів створення web-додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення web-додатку

№	Склад і зміст робіт	Строк розробки
1	Розробка шаблону web-додатку	15 днів
2	Розробка Back End частини web-додатку	35 днів
3	Розробка Front End частини web-додатку	25 дні
5	Наповнення контентом web-додатку	9 дні
6	Beta-тестування	2 дні
7	Alpha-тестування	5 дні
8	Розміщення на хостингу	2 дні
9	Перевірка працездатності	2 дні
10	Написання супровідної документації	4 днів
11	Реліз web-додатку	1 день
	Загальна тривалість робіт	100 днів

4 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію

Web-додаток має бути затверджено та розміщено на web-хостингу. Щоб перенести продукт на web-хостинг потрібно перенести базу даних та всі файли web-додатку на фізичні пристрої зберігання даних, які надає хостинг-компанія. Для забезпечення коректного функціонування web-додатку, необхідно, щоб система на якій він розміщується мала 4 гігабайти оперативної пам'яті та 10 гігабайтів постійної пам'яті для зберігання даних.

ДОДАТОК Б

Планування робіт

Деталізація мети проекту методом SMART. Щоб проект був успішним та конкурентоспроможним треба на концептуальному етапі правильно визначити його мету за допомогою SMART-методу. Суть даного підходу полягає в тому, щоб розглянути поставлену ціль з різних точок зору.

Спочатку потрібно визначити ціль максимально конкретно - Specific. Це дуже допоможе при її досягненні. Адже розмита і не дуже зрозуміла мета буде витратити багато зайвих ресурсів. В процесі можна розгубитися у альтернативних шляхах і різноманітних додаткових завданнях, які будуть лише частково задовольняти поставлені вимоги, але при цьому вимагатимуть безліч зусиль. В такому випадку буде складно досягти того, що нам справді потрібно.

Наступним етапом є Measurable. Потрібно виміряти нашу ціль. Іншими словами, зазначити критерій завершення. Це повинно бути виконання якоїсь умови чи така відповідь на поставлене питання, отримавши яку, можна буде сказати, що мета проекту була досягнута.

Визначивши чого конкретно ми хочемо і скільки, слід задатися питанням, чи це взагалі можливо. Achievable – пункт SMART методу, який відповідальний саме за це. Адже витратити зусилля, у спробі зробити щось неможливе просто нераціонально, особливо з точки зору бізнесу. Якщо мета проекту здається недосяжною та нереалістичною, скоріше за все потрібно відмовитися від проекту або переглянути зміст пунктів Specific та Measurable.

Relevant. Тепер, коли ніщо не заважає нам досягти цілі, потрібно зрозуміти чи є вона релевантною. Тобто чи потрібно взагалі реалізувати даний проект. Потрібно подумати над цінністю, яку принесе досягнення даної мети. Можливо, все задумане зробити досить реально, проте це мало вплине на стратегічний хід справ. Проте час і

ресурси будуть затрачені. В такому випадку слід знову повернутися до перших двох пунктів деталізації мети.

Останнім елементом даного підходу є Time-framed – визначення обмеженості досягнення цілі у часі. В самому початку підходу до деталізації мети було вказано, що це потрібно зробити для того, щоб бути конкурентно спроможними. Адже зараз бізнес процеси доволі швидкі, а технології ще більше підсилюють даний тренд. Потрібно встигати за попитом користувачів та сервісом, який надають конкуренти. Тому потрібно чітко зазначити часові рамки реалізації. Якщо вони не можуть задовільнити потреби, слід знову повернутися до попередніх етапів. Результати деталізації процесу розробки web-додатку підтримки замовлення систем розумний будинок методом SMART розміщені у таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific	Підтримка замовлень систем розумного дому за рахунок створеного web-додатку.
Measurable	Створений додаток скоротить витрати часу на вибір та замовлення систем розумний будинок мінімум на 20%.
Achievable	Мета досяжна, є затверджене технічне завдання.
Relevant	Для підвищення рейтингу та конкурентоспроможності магазину.
Time-framed	Є конкретний термін – до дати здачі виконаної роботи на кафедрі (1 червня 2023 р.).

Планування змісту робіт. WBS (Work Breakdown Structure – Ієрархічна структура робіт) – це графічне представлення переліку робіт, виконання яких, є необхідним для реалізації проекту, та які мають рівні підпорядкування. Головною особливістю даного підходу є розбиття мети проекту на перелік більш конкретних дій, тобто виконання декомпозиції. Кожен наступний рівень є все більш конкретним і зрозумілим для виконавців завданням. Це не тільки дозволить робітникам чітко розуміти свою зону відповідальності, а ще й надасть повніше розуміння кінцевого

продукту проекту та всіх його процесів. Елементами декомпозиції можуть бути продукти, дані та послуги. WBS не містить інформації про терміни ресурси та вартість, але стане основою для всіх цих складових планування проекту.

На найвищому (першому) рівні розміщений продукт проекту. Основні напрями, що забезпечать його успішну реалізацію розміщені на другому рівні. Декомпозиція робіт виконується до тих пір, поки вони не стануть елементарними (простими).

Елементарні роботи – це дії, які мають однозначний чіткий результат, на які призначена відповідальному одна конкретна особа, для якої можна обчислити витрати праці і тривалість виконання. Зазвичай, це вузько направлений спеціаліст. Планування починаючи від цього рівня, якщо це потрібно, можна призначати саме йому. На рисунку Б.1 представлено WBS з розробки web-додатку підтримки замовлення систем розумний будинок.

Планування структури виконавців. Коли повноцінну ієрархію робіт, необхідних для реалізації проекту було створено, слід подбати про OBS. Вона являє собою графічне відображення всіх задіяних у виконанні осіб. Після цього етапу, за кожною дією закріплюється відповідальний за її реалізацію робітник, підрозділ чи організація. Елементи даної структури також знаходяться в ієрархічному відношенні. Так, особи що знаходяться вище на дереві є керівниками, а відносини співпраці відповідають горизонтальному розміщенню. OBS є найважливішим механізмом управління проектом, адже допомагає реалізувати всі пакети робіт, а також є основою формулюванні та реалізації діяльності команди проекту.

На рисунку Б.2 представлено організаційну структуру планування проекту. Список виконавців, що функціонують в проекті описано в таблиці Б.2.

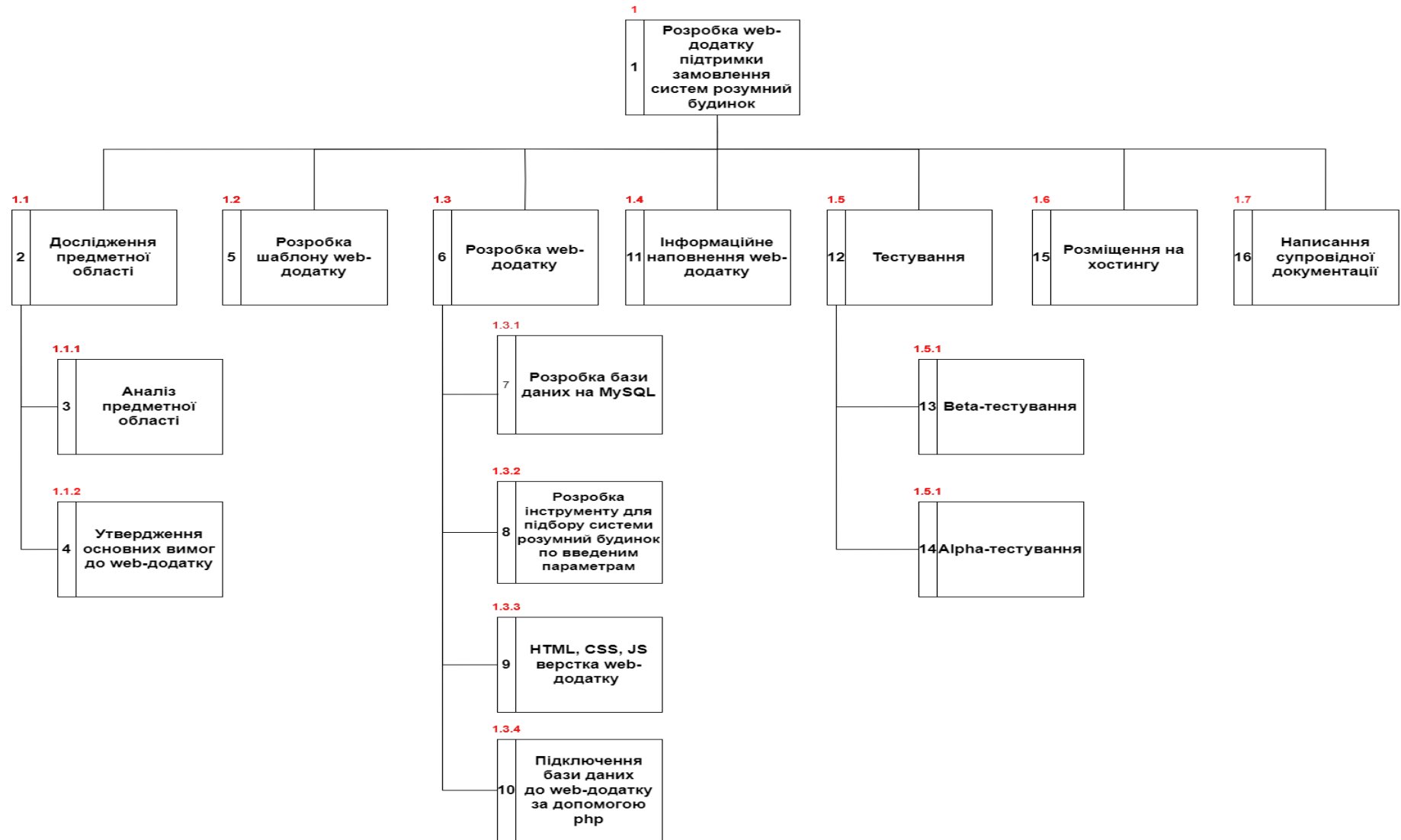


Рисунок Б.1 – WBS-структура робіт проекту

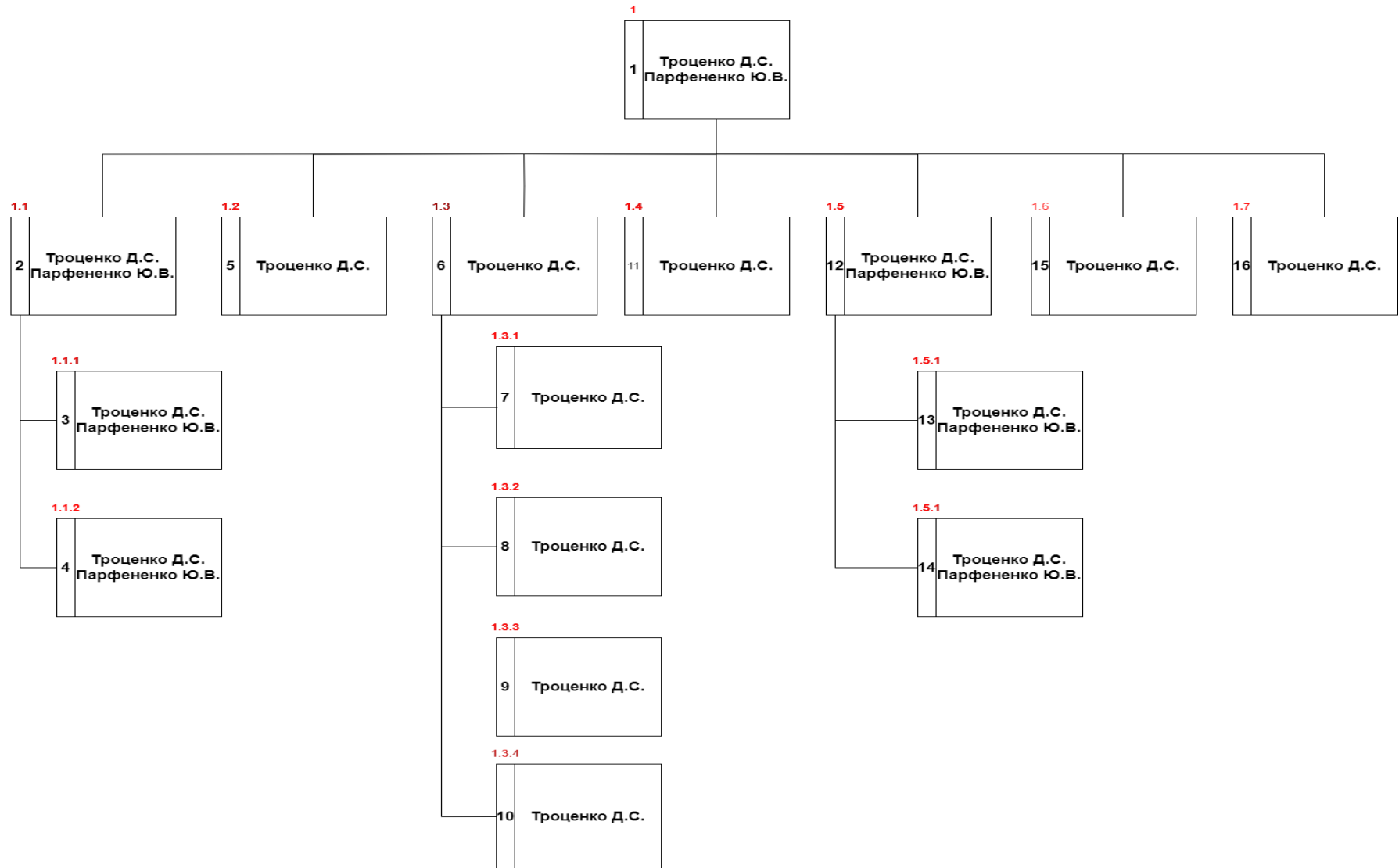


Рисунок Б.2 – OBS-структура робіт

Таблиця Б.2 – Виконавці проекту.

Роль	Ім'я	Проектна роль
Розробник	Троценко Д.С.	Виконує розробку web-додатку.
Дизайнер	Троценко Д.С.	Виконує розробку дизайну web-додатку.
Проектувальник	Троценко Д.С.	Виконує проектування бази даних та розробляє структуру web-додатку.
Тестувальник	Троценко Д.С. Парфененко Ю.В.	Відповідає за тестування функціоналу та дизайну web-додатку.
Керівник проекту	Парфененко Ю.В.	Формує завдання на розробку проекту.
Менеджер проекту	Троценко Д.С.	Відповідає за виконання термінів, розподіл ресурсів та завдань між учасниками. Виконує збір та аналіз даних.

Діаграма Ганта. Якщо WBS дає розуміння сукупності робіт, які необхідно виконати для завершення проекту, то завдяки календарному плану можна представити точні дати, коли необхідно виконати конкретну задачу. Даний підхід є дуже практичним, адже враховує свята, вихідні, тип та тривалість робочого графіку. Можна вказати, послідовність робіт, можливість їх паралельного виконання різними відділами команди. Окрім цього передбачено введення в систему затримок та резервів. Все це дає реалістичніше розуміння розміщення проекту у часі. Діаграми Ганта, по суті своїй, є графічним відображенням календарного плану. Всі задачі та їх параметри представлені у вигляді горизонтальних ліній, розташованих паралельно головній лінії часу.

Для управління даним проектом було застосовано програму MS Project. Вона підтримує всі перераховані вище механізми, а також має безліч інших, які будуть корисними при розробці масштабного продукту великою командою. Наприклад, можна відобразити на діаграмі відсоток виконання задачі та відповідального за неї робітника. Також показати резерви часу та критичні завдання, виконання яких до

вказаної дати, є обов'язковим. Календарний графік проекту представлено на рисунках Б.3 - Б.4.

	Р. з:	Ім'я завдання	Тривалі	Початок	Завершені	Поп	Імена ресурсів
1		Web-додаток підтримки замовлення систем розумний будинок	98 днів?	Пн 16.01.23	Ср 31.05.23		Троценко Д.С.; Парфененко Ю.В.
2		Дослідження предметної області	10 днів	Пн 16.01.23	Пт 27.01.23		Троценко Д.С.; Парфененко Ю.В.
3		Аналіз предметної області	6 днів	Пн 16.01.23	Пн 23.01.23		Троценко Д.С.; Парфененко Ю.В.
4		Утвердження основних вимог до web-додатку	4 днів	Вт 24.01.23	Пт 27.01.23	3	Троценко Д.С.; Парфененко Ю.В.
5		Розробка шаблону web-додатку	16 днів	Вт 31.01.23	Вт 21.02.23	2FS+1,	Троценко Д.С.
6		Розробка web-додатку	71 днів?	Вт 21.02.23	Вт 30.05.23	5	Троценко Д.С.
7		Розробка бази даних на MySQL	5 днів	Пн 06.03.23	Пт 10.03.23		Троценко Д.С.
8		Розробка інструменту для автоматичного підбору системи по введеним параметрам	15 днів	Пн 13.03.23	Пт 31.03.23	7	
9		HTML, CSS, JS верстка web-додатку	10 днів?	Пн 24.04.23	Пт 05.05.23	8;7	Троценко Д.С.
10		Підключення бази даних до web-додатку за допомогою php	6 днів	Ср 10.05.23	Ср 17.05.23	7FS+1,	Троценко Д.С.
11		Інформаційне наповнення web-додатку	9 днів	Чт 18.05.23	Вт 30.05.23		Троценко Д.С.
12		Тестування	10 днів	Ср 17.05.23	Вт 30.05.23		Троценко Д.С.; Парфененко Ю.В.
13		Beta-тестування	2 днів	Ср 17.05.23	Чт 18.05.23		Троценко Д.С.; Парфененко Ю.В.
14		Alpha-тестування	3 днів	Чт 25.05.23	Пн 29.05.23	13	Троценко Д.С.; Парфененко Ю.В.
15		Розміщення на хостингу	2 днів	Пт 19.05.23	Пн 22.05.23	13	Троценко Д.С.
16		Написання супровідної документації	4 днів	Пт 19.05.23	Ср 24.05.23		Троценко Д.С.

Рисунок Б.3 – Календарний графік проекту

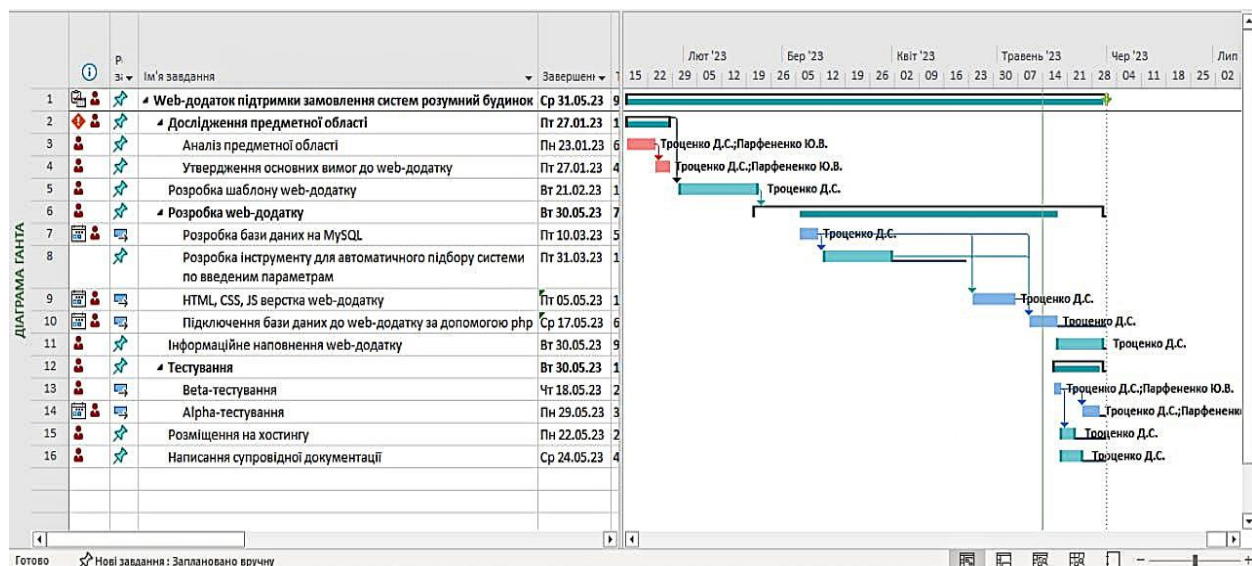


Рисунок Б.4 – Календарний графік проекту, діаграма Ганта

Управління ризиками проекту. Перед початком роботи над проектом потрібно розглянути негативні чинники, які можуть вплинути на хід та результати робіт. Це дуже важливо для того, щоб бізнес не втратив свої кошти, а також щоб проект зміг

дійти до фінальних стадій розвитку та приніс цінність людям. Перше, що потрібно зробити – ідентифікувати ризики. Тобто визначити всі фактори, що так чи інакше можуть вплинути на проект. Для даного проекту вони визначені у таблиці Б.3

Таблиця Б.3 — Ідентифікація ризиків проекту.

№	Назва (опис ризику)
1	Нераціональний розподіл часу
2	Затримки у зв'язку з відключенням електроенергії
3	Затримки у зв'язку з військовими діями
4	Втрата напрацьованого через технічні поломки чи програмні помилки
5	Нечітке завдання на розробку
6	Поява альтернативного продукту
7	Непрацездатність розробника у зв'язку з непередбачуваними обставинами
8	Зниження економічних показників цільових користувачів
9	Затримки у зв'язку з технічними поломками
10	Нераціональний розподіл робіт

Після цього потрібно визначити ймовірність виникнення критичної ситуації та силу впливу на досягнення цілі. Дані два фактори сукупно визначають тип ризику. Цей показник на пряму вказує, як потрібно реагувати на відповідну можливість розвитку подій. Так, якщо ризик прийнятний, витратити на нього свою увагу не слід. Допустимий рівень загрози говорить сам за себе, а недопустимий ставить серйозний виклик на шляху виконання проекту. При виявленні ризику такого типу слід якнайшвидше відреагувати, виконавши дії, які зможуть перевести його хоча б на рівень нижче. У таблиці Б.4 представлено шкалу для класифікації ризиків за величиною впливу на проект та ймовірністю виникнення.

Таблиця Б.4 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу.

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Визначивши всі помітні загрози та спробувавши мінімізувати їх ранг ми не вирішили проблему цілком. Слід розуміти, що навіть допустимий ризик все ще може негативно вплинути на проект. Для поліпшення становища слід розробити план попередження та реагування на виникнення критичних ситуацій. До нього входить визначення ефективності розробки та оцінка наслідків впливу на проект. Оцінювання виконується за показниками, що описані в таблиці Б.4. У результаті планування реагування було отримано матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рисунку Б.5. Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

Таблиця Б.5 – Матриця ймовірності та впливу згідно проекту

Ймовірність ризику (Й)	Вплив загрози (ризика)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9					
0,7					
0,5			R8	R1	
0,3				R5	R3
0,1	R10			R2, R9, R7, R6	R4

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена у таблиці Б.6. У таблиці Б.7 описано ризики та стратегії реагування на кожен з них.

Таблиця Б.6 – Шкала оцінювання за рівнем ризику.

№	Назва	Межі	Ризики, які входять(номера) 1
1	Прийнятні	$0,005 \leq R \leq 0,05$	R2, R6, R7 R9, R10
2	Виправдані	$0,05 < R \leq 0,14$	R4, R5, R8
3	Недопустимі	$0,14 < R \leq 0,72$	R1, R3

Таблиця Б.7 – Ризики та стратегії реагування.

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS1	Відкритий	Нераціональний розподіл часу	0,5	0,4	0,2	Створити чіткий календарний план проекту та слідувати йому.	Попередження	Якщо не вдається дотримуватися плану, потрібно виявити причини цього та по можливості усунути їх або ж змінити план.
RS2	Відкритий	Затримки у зв'язку з відключенням електроенергії	0,1	0,4	0,04	1. Виконувати роботу на ноутбуках, або інших пристроях з автономними джерелами живлення. 2. Закласти в календарний план додаткові резерви. 3. Мати резервне джерело мобільного інтернету.	Прийняття	
RS3	Відкритий	Затримки у зв'язку з військовими діями	0,3	0,8	0,24	Попередження замовника про можливі ризики.	Прийняття	

Продовження таблиці Б.7 – Ризики та стратегії реагування.

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS4	Відкритий	Втрата напрацьованого через технічні поломки чи програмні помилки	0,1	0,8	0,08	Робити резервні копії проекту в хмарному сховищі, після виконання кожного значимого етапу.	Прийняття	
RS5	Відкритий	Нечітке завдання на розробку	0,3	0,4	0,12	1.Відповідально підійти до постановки технічного завдання проекту. 2.Скласти глосарій для запобігання розбіжностей у розумінні слів та термінів. 3.Періодичний контроль замовником етапів роботи. 4. Уникати нечітких, та невимірюваних характеристик продукту.	Попередження	При виявленні розбіжностей між характеристиками продукту та технічним завданням виконати виправлення.

Продовження таблиці Б.7 – Ризики та стратегії реагування.

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS6	Відкритий	Поява альтернативного продукту	0,1	0,4	0,04	1.Провести попереднє дослідження предметної області. 2. Вибрати унікальну стратегію створення проекту.	Прийняття	
RS7	Відкритий	Непрацездатність розробника у зв'язку з непередбачуваними обставинами	0,1	0,4	0,04	Забезпечити прийнятні резерви часу для завдань проекту.	Прийняття	
RS8	Відкритий	Зниження економічних показників цільових користувачів	0,5	0,2	0,1	Планувати проект з урахуванням можливості такого ризику.	Прийняття	

Продовження таблиці Б.7 – Ризики та стратегії реагування.

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS9	Відкритий	Затримки зв'язку технічними поломками	0,1	0,4	0,04	1.Регулярно проводити профілактичні обслуговування робочих машин. 2.Мати резервні пристрої для виконання роботи.	Прийняття	
RS 10	Відкритий	Нераціональний розподіл робіт	0,1	0,05	0,005	1.Чітко визначити перелік робіт та ресурси необхідні для їх виконання. 2.Раціонально розділити обов'язки між працівниками.	Прийняття	

ДОДАТОК В

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Web-додаток підтримки замовлення систем розумний будинок

Троценко Д.С., студент ІТ-92; Парфененко Ю.В., доцент
Сумський державний університет, м. Суми, Україна

Сьогодні все більшої популярності набувають складні електронні системи, які можуть зробити життя людини комфортнішим. Сюди можна віднести голосових асистентів чи розумні прилади. Такі системи зазвичай складаються з кількох пов'язаних між собою приладів.

Ефективність функціонування систем розумного будинку залежить в тому числі й від правильного підбору компонентів. Кожен елемент може мати аналоги, які будуть відрізнятися своїми характеристиками та набором функцій, але при цьому, повинні взаємодіяти один з одним. Щоб оптимізувати процес вибору таких систем для людей шляхом скорочення витрат часу та зусиль на підбір необхідної комплектації, було прийнято рішення створити web-додаток підтримки замовлення систем розумний будинок.

Розроблена інформаційна система реалізована у вигляді web-додатку. Для його створення використано HTML, CSS, JavaScript, які відповідають за візуальне представлення інтерфейсу користувача. Логіку додатку написано за допомогою мови програмування PHP. Для збереження даних про компоненти систем розумного будинку використовується СКБД MySQL.

Функціонально web-додаток представляє собою каталог, який містить типові для даного виду додатків функції, такі як сортування, фільтрування та пошук товарів. Web-додаток буде корисний тим, хто шукає конкретний пристрій у свою систему. Також в наявності є кошик, куди можна додати кілька товарів.

Окрім цього, на сторінках web-додатку представлено типові набори систем розумного будинку. Це вже укомплектована система певного типу, яка є універсальною та може бути запозичена відвідувачами для своїх цілей. Також можна підібрати більш персоналізовану систему за введеними параметрами. Для цього необхідно перейти до відповідного інструменту web-додатку.

Розроблений web-додаток зможе задовільнити різні потреби користувачів у сфері вибору систем розумний будинок, починаючи від пошуку одного пристрою до підбору комплексного набору.

ДОДАТОК Г

ЛІСТИНГ ПРОГРАМНОГО КОДУ ОСНОВНИХ МОДУЛІВ WEB-
ДОДАТКУ**Index.php**

```

<?php
    include_once 'header.php';
?>

<?php
    if (empty($_SESSION['user']['admin_status'])) {
        ?>
<style>
        .card-dell, .card-update{
            display: none;
        }
</style>

    <?php
        }
    ?>

    <div class="container">
        <div class="categories-search">
            <div class="categories">
                <button class="categori1">Всі</button>
                <button class="categori2">Хаби</button>
                <button class="categori3">Датчики</button>
                <button class="categori4">Розетки</button>
            </div>

<div class="filter-cards">
<div></div>
<div class="sort-search">
<button class="sortPriceABC">За зростанням ціни</button>
<button class="sortPriceDES">За зменшенням ціни</button>
<button class="sortNameABC">За назвою (А-Я)</button>
<button class="sortNameDES">За назвою (Я-А)</button>
<input type="text" class="searchInput" value="" placeholder="Пошук">
<button class="searchBTN"> Знайти </button>
</div>

        <div class="filter-cards">
            <div class="filter-plate">
                <div class="FilterHeader FilterMainHeader">
                    Фільтри
                </div>
            <div class="FilterHeader">
                Ціна
            </div>
            Від
            <input class="filterStartPrice" type="text" placeholder="0" phra>

```



```

var currentPage = 1;
var category = 0;
var selectedValuesScen = [];
var selectedValuesBrand = [];
var sortByPrice = 0;
var sortByName = 0;
var startPrice = 0;
var EndPrice = 100000;
var findByName = 0;

$(document).ready(function() {
    loadProducts(currentPage,category,sortByPrice,sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
});

// Кнопки сортування за іменем та ціною
$('#button.sortPriceABC').click(function() {
    currentPage=1;
    sortByPrice = 1;
    sortByName = 0;
    checkPage(currentPage);
    loadProducts(currentPage,category,sortByPrice,sortByName, startPrice,
EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
});

$('#button.sortPriceDES').click(function() {
    sortByPrice = 2;
    sortByName = 0;
    currentPage=1;
    checkPage(currentPage);
    loadProducts(currentPage,category,sortByPrice,sortByName, startPrice,
EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
});

$('#button.sortNameABC').click(function() {
    sortByPrice = 0;
    sortByName = 1;
    currentPage=1;
    checkPage(currentPage);
    loadProducts(currentPage,category,sortByPrice,sortByName, startPrice,
EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
});

$('#button.sortNameDES').click(function() {
    sortByPrice = 0;
    sortByName = 2;
    currentPage=1;
    checkPage(currentPage);
    loadProducts(currentPage,category,sortByPrice,sortByName, startPrice,
EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
});

// Кнопки введення значень брендів та кількості сценаріїв
$(document).ready(function() {
    $('#input.ScenarioBox').on('change', function() {
        var value = $(this).val();
        if ($(this).is(':checked')) {
            selectedValuesScen.push(value);
        } else {
            var index = selectedValuesScen.indexOf(value);
            if (index > -1) {
                selectedValuesScen.splice(index, 1);
            }
        }
    });
});

```

```

    }
  }
});
});

$(document).ready(function() {
  $('#input.BrandBox').on('change', function() {
    var value = $(this).val();

    if ($(this).is(':checked')) {
      selectedValuesBrand.push(value);
    } else {
      var index = selectedValuesBrand.indexOf(value);
      if (index > -1) {
        selectedValuesBrand.splice(index, 1);
      }
    }
  });
});

// Поля введення значень мінімальної та максимальної ціни
$('#input.filterStartPrice').on('input', function() {
  var value = $(this).val();
  if (!/^\d+$/.test(value)) {
    $(this).val('');
    startPrice = 0;
  }
  else{
    startPrice = value;
  }
});

$('#input.filterEndPrice').on('input', function() {
  var value = $(this).val();
  if (!/^\d+$/.test(value)) {
    $(this).val('');
    EndPrice = 100000;
  }
  else{
    EndPrice = value;
  }
});

// Кнопки перелистування сторінок
$('#button.next-page').addClass('invis');
checkPage(currentPage);
$('#button.next-page').click(function() {
  currentPage++;
  checkPage(currentPage);
  loadProducts(currentPage, category, sortByPrice, sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
});
$('#button.previous-page').click(function() {
  currentPage--;
  checkPage(currentPage);
  loadProducts(currentPage, category, sortByPrice, sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
});

$('#button.search-button').click(function() {
  currentPage = 1;
  checkPage(currentPage);
});

```

```

        loadProducts (currentPage, category, sortByPrice, sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
        checkPage (currentPage);
    });

    $('button.categoril').click(function() {
        category = 0;
        currentPage = 1;
        checkPage (currentPage);
        loadProducts (currentPage, category, sortByPrice, sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
        checkPage (currentPage);
    });
    $('button.categori2').click(function() {
        category = 1;
        currentPage = 1;
        checkPage (currentPage);
        loadProducts (currentPage, category, sortByPrice, sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
        checkPage (currentPage);
    });
    $('button.categori3').click(function() {
        category = 2;
        currentPage = 1
        checkPage (currentPage);
        loadProducts (currentPage, category, sortByPrice, sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
        checkPage (currentPage);
    });
    $('button.categori4').click(function() {
        category = 3;
        currentPage = 1;
        checkPage (currentPage);
        loadProducts (currentPage, category, sortByPrice, sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
    });

    $('input.searchInput').change(function() {
        // Отримуємо нове значення
        findByName = $('input.searchInput').val();
    });
    $('button.searchBTN').click(function() {
        findByName = $('input.searchInput').val();
        currentPage = 1;;
        checkPage (currentPage);
        loadProducts (currentPage, category, sortByPrice, sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName)
    });

    function checkPage (current){
        if (currentPage == 1){
            $('button.previous-page').addClass('invis');
        }
        if (currentPage !== 1){
            $('button.previous-page').removeClass('invis');
        }
        if ($('input.hidden').val() <= 1) {
            $('button.next-page').addClass('invis');
        }
    }
}

```

```

function loadProducts(currentPage,category,sortByPrice,sortByName,
startPrice, EndPrice, selectedValuesBrand, selectedValuesScen, findByName) {
$.ajax({
type: 'POST',
url: 'vendor/load_products.php',
data: {
page: currentPage,
categoryFild: category,
sortByPriceFild: sortByPrice,
sortByNameFild: sortByName,
startPriceFild:startPrice,
EndPriceFild:EndPrice,
selectedValuesScenFild:selectedValuesScen,
selectedValuesBrandFild:selectedValuesBrand,
findByNameFild:findByName
},
success: function(data) {
$('.cardList').empty();
$('.cardList').append(data);
},
error: function() {
alert('Error occurred while loading products.');
```

searchSys.php

```

<?php
session_start();
require_once 'connect.php';
$minprice = $_POST['minpriceFild'];
$maxprice = $_POST['maxpriceFild'];
$set_type = $_POST['typeFild'];
$is_economy = $_POST['is_economyFild'];
$is_comfort = $_POST['is_comfortFid'];
$is_protect = $_POST['is_protectFild'];
if($set_type === 'true'){
$products = mysqli_query($connect, "SELECT * FROM `standardsets`
WHERE `is_multisystem` = '1' AND `price` <= '$maxprice'");
}
else{
$products = mysqli_query($connect, "SELECT * FROM `standardsets`
WHERE `is_multisystem` = '$set_type' AND `is_economy` = '$is_economy' AND
`is_protect` = '$is_protect' AND `is_comfort` = '$is_comfort' AND `price` <=
'$maxprice'");
}

$count = mysqli_num_rows($products);
echo $count;
$products = mysqli_fetch_all($products);

?>

<?php
if ($count){
```



```

?>

<div class="ask4">
    <div class="find-quest1">
        Результаты
    </div>
</div>
<div class="systemList">

    <?php
    foreach($products as $product){ ?>

        <div class="card">
            <div>
                <div class="card-img"> </div>
                </div>
                <div class="card-info">
                    <div class="card-info-name" ><?=$product[1] ?></div>
                    <div class="card-info-price"><?=$product[11] ?>
                        &#8372</div>
                    <a class="card-view" href="system.php?set_id=<?=$product[0]
?>">Переглянути </a>
                </div>
            </div>

            <?php
        }
        echo "</div>";
    }
    if(!$count){
        ?>
        <div class="ask4">
            <div class="find-quest1">
                Результаты
            </div>
        </div>
        <div class="ask" style="text-align: center;">
            Нажаль. системи, що відповідає ВАШИМ вимогам не знайдено.
        </div>
    }
    ?>

```

addProduct.php

```

<?php
require_once "connect.php";

$productName = $_POST['productName'];
$description = $_POST['description'];
$brand = $_POST['brand'];
$color = $_POST['color'];
$connection_type = $_POST['connection_type'];
$count_of_scenarious = $_POST['count_of_scenarious'];
$specification = $_POST['specification'];
$category_id = $_POST['category_id'];
$price = $_POST['price'];
$url = $_POST['url'];

$path = 'uploads/' . time() . $_FILES['url']['name'];

```

```

move_uploaded_file($_FILES['url']['tmp_name'], '../' . $path);

// Виконання запиту
$query = "SELECT category_id FROM `categories` WHERE `category_name` =
'$category_id'";
$result = mysqli_query($connect, $query);

// Отримання результату запиту
if ($result) {
    // Перевірка наявності результату
    if (mysqli_num_rows($result) > 0) {
        // Отримання першого рядка результату
        $row = mysqli_fetch_assoc($result);
        // Отримання значення category_id
        $category_id = $row['category_id'];
    } else {
        echo "Запис не знайдено";
    }
} else {
    echo "Помилка запиту: " . mysqli_error($connect);
}

// Звільнення ресурсів
mysqli_free_result($result);

$row = mysqli_query($connect, "INSERT INTO `products`
(`product_id`, `product_name`, `description`, `brand`, `color`,
`connection_type`, `count_of_scenarios`, `specification`, `category_id`,
`price`, `image_url`, `is_active`)
VALUES
(NULL, '$productName', '$description', '$brand', '$color',
'$connection_type', '$count_of_scenarios', '$specification', '$category_id',
'$price', '$path', '1')");

header('Location: ../index.php');
?>

```

buyList.php

```

<?php
    session_start();

    if (!isset($_SESSION['cart'])) {
        $_SESSION['cart'] = array(
            'type1' => array(),
            'type2' => array()
        );
    }

    $productId = $_POST['product'];
    $systemId = $_POST['system'];
    $quantity = 1;

    if (!isset($_SESSION['cart'])) {
        $_SESSION['cart'] = [];
    }

    if (isset($_SESSION['cart']['type1'][$productId])) {
        $_SESSION['cart']['type1'][$productId] += $quantity;
    } elseif($productId !== null) {
        $_SESSION['cart']['type1'][$productId] = $quantity;
    }

```

```

if (isset($_SESSION['cart']['type2'][$systemId])) {
    $_SESSION['cart']['type2'][$systemId] += $quantity;
} elseif($systemId !== null) {
    $_SESSION['cart']['type2'][$systemId] = $quantity;
}

?>

```

load_products.php

```

<?php
session_start();
require_once 'connect.php';

$page = $_POST['page'];
$category = $_POST['categoryFild'];

$sortPrice = $_POST['sortByPriceFild'];
$sortName = $_POST['sortByNameFild'];

$startPrice = $_POST['startPriceFild'];
$endPrice = $_POST['EndPriceFild'];

$brand = $_POST['selectedValuesBrandFild'];
$scenarious = $_POST['selectedValuesScenFild'];

$findByName = $_POST['findByNameFild'];

if(isset($_SESSION['user']["admin_status"])){
    $admin = 1;
}
else{
    $admin = 0;
}

$sort = " ";
$productsPerPage = 6;

if ($category == 0){
    $category = "WHERE 1";
}else{
    $category = "WHERE category_id = $category";
}

if ($sortName ==0 && $sortPrice ==0){
    $sort = "ORDER BY NULL";
}
else if($sortName ==0 && $sortPrice ==1){
    $sort = "ORDER BY `price`";
}
else if ($sortName ==0 && $sortPrice ==2){
    $sort = "ORDER BY `price` DESC";
}
else if ($sortName == 1 && $sortPrice ==0){
    $sort = "ORDER BY `product_name`";
}

```

```

else if ($sortName == 2 && $sortPrice == 0){
    $sort = "ORDER BY `product_name` DESC";
}

if ($admin === 1){
    $admin = " AND (is_active = 1 OR is_active = 0) ";
}
else{
    $admin = " AND (is_active = 1) ";
}

$totalProductsQuery = mysqli_query($connect, "SELECT COUNT(*) FROM
`products` $category $admin ");
$totalProducts = mysqli_fetch_row($totalProductsQuery)[0];

// Calculate the total number of pages
$totalPages = ceil($totalProducts / $productsPerPage);

$startIndex = ($page - 1) * $productsPerPage;

$query = "SELECT * FROM `products` $category $admin ";

if (!empty($brand)) {
    $query.=" AND ( ";
    foreach ($brand as $value) {
        $query .= "brand = '$value' OR ";
    }
    $query = rtrim($query, " OR ");
    $query .= ") ";
}

if (!empty($scenarious)) {
    $query.=" AND ( ";
    foreach ($scenarious as $value) {
        if($value == 0){
            $query .= "count_of_scenarious = '$value' OR ";
        }
        if($value == 3){
            $query .= "count_of_scenarious BETWEEN 0 AND 5 OR ";
        }
        if($value == 15){
            $query .= "count_of_scenarious BETWEEN 10 AND 20 OR ";
        }
        if($value == 35){
            $query .= "count_of_scenarious BETWEEN 20 AND 50 OR ";
        }
        if($value == 75){
            $query .= "count_of_scenarious > 50 OR ";
        }
    }
    $query = rtrim($query, " OR ");
    $query .= ") ";
}

if( $findByName !== NULL && $findByName != 0 ){
    $query .= "AND ( product_name LIKE '$findByName%' ) ";
}

// echo "$findByName";
$query .= " AND `price` BETWEEN $startPrice AND $endPrice $sort ";

```

```

$totalProductsResult = mysqli_query($connect, $query);
$totalProducts = mysqli_num_rows($totalProductsResult);

// Підрахунок загальної кількості сторінок
$totalPages = ceil($totalProducts / $productsPerPage);

$query .= " LIMIT $startIndex , $productsPerPage ";

$result = mysqli_query($connect, $query);
$products = mysqli_fetch_all($result);

foreach ($products as $product) {?>
    <div class="card">
        <div>
            <div class="card-img"> </div>
            </div>
            <div class="card-info">
                <div class="card-info-name"><?= $product[1] ?></div>
                <div class="card-info-price"><?= $product[9] ?>
&#8372</div>
                <div>
                    <a
href="product.php?product_id=<?= $product[0] ?>"
id="mainLink">Переглянути</a>
                    <a
class="card-view"
card-update"
href="updateProduct.php?product_id=<?= $product[0] ?>">Редагувати</a>
                </div>
            </div>
        </div>
    </div>
}

if ($totalPages == $page){ ?>
    <style>
        .next-page{
            visibility: hidden;
        }
    </style>
    <?php
}
elseif($totalPages == 1 || $totalPages == 0){?>
    <style>
        .next-page{
            visibility:hidden;
        }
    </style>
    <?php
}
else{?>
    <style>
        .next-page{

```

```

        visibility:visible;
    }
</style>
<?php
}

```

```
?>
```

loadOrdersForAdmins.php

```

<?php

session_start();
require_once('connect.php');

$query1 = "SELECT o.order_id, c.customer_id, c.full_name, d.phone_number,
o.order_date, d.delivery_point, d.deliveries_status , o.order_status
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN deliveries d ON o.order_id = d.order_id";
$amount_orders = mysqli_query($connect, $query1);
$amount_orders = mysqli_fetch_all($amount_orders);
$amount_orders = count($amount_orders);

$limit = 5;
$offset = $_POST['per_pageFiled'];
?>



| Номер замовлення | <input &gt;&lt;="" <="" class="amount_orders" input&gt;="" style="display: none;" td="" value="&lt;?= \$amount_orders ?&gt;"/> <th>Ім'я клієнта</th> <th>Номер клієнта</th> <th>Дата замовлення</th> <th>Пункт доставки</th> <th>Статус доставки</th> <th>Статус замовлення</th> | Ім'я клієнта | Номер клієнта | Дата замовлення | Пункт доставки | Статус доставки | Статус замовлення |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|---------------|-----------------|----------------|-----------------|-------------------|
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|---------------|-----------------|----------------|-----------------|-------------------|



<?php

$query = "SELECT o.order_id, c.customer_id, c.full_name,
d.phone_number, o.order_date, d.delivery_point, d.deliveries_status ,
o.order_status
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN deliveries d ON o.order_id = d.order_id
LIMIT $offset , $limit";
$orders = mysqli_query($connect, $query);
$orders = mysqli_fetch_all($orders);

```

```

foreach ($orders as $order) {
    ?>
        <tr >
            <td><?= $order[0] ?></td>
            <td><?= $order[2] ?></td>
            <td><?= $order[3] ?></td>
            <td><?= $order[4] ?></td>
            <td><?= $order[5] ?></td>
            <td><?= $order[6] ?></td>
            <td><?= $order[7] ?></td>
            <td class="tdForExpandBtn" style="border:
0;">
                <button class="expandBTN"
id="expandButton_<?= $order[0] ?>" onclick="expandOrderDetails(<?= $order[0]
?>)">Детальніше</button>
                </td>
        </tr>
        <tr id="orderDetails_<?= $order[0] ?>"
style="display: none;">
            <td colspan="7">
                <div class="showColapse">
                    <div></div>
                    <button class="newdata deleteBtn" data-
id="<?= $order[0] ?>">Видалити замовлення</button>
                    <button class="newdata updateBtn" data-
id="<?= $order[0] ?>">Оновити дані</button>
                    <input class="newdata inputField" data-
id="<?= $order[0] ?>" type="text" placeholder="Введіть адресу доставки.">
                    <select class="selectCat" data-id="<?=
$order[0] ?>">
                        <option
value="Доставлено">Доставлено</option>
                        <option
value="Відправлено">Відправлено</option>
                        <option value="Не відправлено">Не
відправлено</option>
                    </select>
                    <select class="selectCat2" data-id="<?=
$order[0] ?>">
                        <option value="В процесі">В
процесі</option>
                        <option
value="Виконано">Виконано</option>
                    </select>
                </div>
            </td>
        </tr>
    <?php
    $query = "SELECT p.product_name, oil.item_count
FROM orderitemslist oil
JOIN orders o ON oil.order_id = o.order_id
JOIN products p ON p.product_id = oil.product_id
WHERE o.order_id = " . $order[0];

$orderItems = mysqli_query($connect, $query);
$orderItems = mysqli_fetch_all($orderItems);
if (!empty($orderItems))
{?>
        <div class="additionalInfo">

```

```

товару</div>
        <div class="productnameheader">Назва
        <div class="productamounthead">Кількість</div>
    </div>

        <?php
        foreach ($orderItems as $orderItem)
        {
            ?>
            <div class="additionalInfo">
                <div class="productname"><?=$orderItem[0]
?></div>
                <div class="productamount"><?=$orderItem[1]
?></div>
            </div>
            <?php
        }
    }
    ?>

    <?php
    $query = "SELECT s.set_name, osl.system_count
    FROM ordersystemslist osl
    JOIN orders o ON osl.order_id = o.order_id
    JOIN standardsets s ON s.set_id = osl.system_id
    WHERE o.order_id = " . $order[0];

    $orderSystems = mysqli_query($connect, $query);
    $orderSystems = mysqli_fetch_all($orderSystems);

    if (!empty($orderSystems))
    {?>
        <div class="additionalInfo">
            <div class="productnameheader">Назва системи</div>
            <div class="productamounthead">Кількість</div>
        </div>

        <?php
        foreach ($orderSystems as $orderSystem) {
            ?>
            <div class="additionalInfo">
                <div class="productname"><?=$orderSystem[0]
?></div>
                <div class="productamount"><?=$orderSystem[1]
?></div>
            </div>
            <?php
        }
    }
    ?>

    </td>
</tr>

    <?php }

```



```

echo "</table>";

?>

sendOrder.php

<?php

require_once "connect.php";

$order_id = $_POST['order_idField'];

$query = "SELECT o.order_id, d.phone_number, o.order_date,
d.delivery_point
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN deliveries d ON o.order_id = d.order_id
WHERE o.order_id = $order_id";
$order = mysqli_query($connect, $query);
$order = mysqli_fetch_assoc($order);

?>

<?php ob_start(); ?>

<body style="font-family: Arial, sans-serif; background-color:
#f1f1f1;margin: 0; padding: 20px;">

<h2 class="greet-msg" style="text-align: center; margin-bottom:
30px;">Дякуємо за замовлення в нашому web-додатку DragonHome!</h2>

<div class="additional-info" style="clear: both;">
  <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">
    <p>Номер замовлення:</p>
  </div>
  <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">
    <p>Номер телефону:</p>
  </div>
  <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">
    <p>Дата замовлення:</p>
  </div>
  <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">
    <p>Пункт доставки:</p>
  </div>
</div>

<div class="additional-info" style="clear: both;">
  <div class="info-value" style="font-weight: bold; display: inline-
block; width: 24%;">
    <p><?= $order['order_id'] ?></p>
  </div>
  <div class="info-value" style="font-weight: bold; display: inline-
block; width: 24%;">
    <p><?= $order['phone_number'] ?></p>
  </div>

```

```

        <div class="info-value" style="font-weight: bold; display: inline-
block; width: 24%;">
            <p><?= $order['order_date'] ?></p>
        </div>
        <div class="info-value" style="font-weight: bold; display: inline-
block; width: 24%;">
            <p><?= $order['delivery_point'] ?></p>
        </div>
    </div>

<h3>Замовлено:</h3>

<?php
$query = "SELECT p.product_name, oil.item_count, p.price
FROM orderitemslist oil
JOIN orders o ON oil.order_id = o.order_id
JOIN products p ON p.product_id = oil.product_id
WHERE o.order_id = $order_id";

$orderItems = mysqli_query($connect, $query);
$orderItems = mysqli_fetch_all($orderItems);
$totalPrice = 0;

if (!empty($orderItems)) {
    ?>
    <div class="additional-info" style="clear: both;">
        <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">Назва товару</div>
        <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">Кількість</div>
        <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">Вартість товару</div>
        <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">Загальна вартість</div>
    </div>
    <?php
    foreach ($orderItems as $orderItem) {
        $totalPrice += $orderItem[1] * $orderItem[2];
        ?>
        <div class="additional-info" style="clear: both;">
            <div class="info-value" style="display: inline-block; width:
24%;"><?= $orderItem[0] ?></div>
            <div class="info-value" style="display: inline-block; width:
24%;"><?= $orderItem[1] ?></div>
            <div class="info-value" style="display: inline-block; width:
24%;"><?= $orderItem[2] ?></div>
            <div class="info-value" style="display: inline-block; width:
24%;"><?= $orderItem[1] * $orderItem[2] ?></div>
        </div>
    <?php
    }
}
?>

<?php
$query = "SELECT s.set_name, osl.system_count, s.price
FROM ordersystemslist osl
JOIN orders o ON osl.order_id = o.order_id
JOIN standardsets s ON s.set_id = osl.system_id
WHERE o.order_id = $order_id";

$orderSystems = mysqli_query($connect, $query);
$orderSystems = mysqli_fetch_all($orderSystems);

```

```

    if (!empty($orderSystems)) {
        ?>
        <div class="additional-info" style="clear: both; padding-top: 15px">
            <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">Назва системи</div>
            <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">Кількість</div>
            <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">Вартість системи</div>
            <div class="info-header" style="font-weight: bold; display: inline-
block; width: 24%;">Загальна вартість</div>
        </div>
        <?php
        foreach ($orderSystems as $orderSystem) {
            $totalPrice += $orderSystem[1] * $orderSystem[2];
        ?>
        <div class="additional-info" style="clear: both;">
            <div class="info-value" style="display: inline-block; width:
24%;"><?=$orderSystem[0] ?></div>
            <div class="info-value" style="display: inline-block; width:
24%;"><?=$orderSystem[1] ?></div>
            <div class="info-value" style="display: inline-block; width:
24%;"><?=$orderSystem[2] ?></div>
            <div class="info-value" style="display: inline-block; width:
24%;"><?=$orderSystem[1] * $orderSystem[2] ?></div>
        </div>

        <?php
        }
        }
        ?>
        <div class="allPrice" style="margin-top: 20px;text-align: right;font-
size: 24px;">
            Загальна вартість замовлення: <?=$totalPrice ?> грн.
        </div>
    </body>
    <?php $body = ob_get_clean(); ?>

    <?php

    session_start();

    use PHPMailer\PHPMailer\PHPMailer;
    use PHPMailer\PHPMailer\Exception;

    // Включаємо необхідні файли для використання PHPMailer
    require '../PHPMailer/src/Exception.php';
    require '../PHPMailer/src/PHPMailer.php';
    require '../PHPMailer/src/SMTP.php';

    $phpmailer = new PHPMailer(); // Створюємо новий об'єкт PHPMailer
    $phpmailer->CharSet = 'UTF-8'; // Налаштовуємо кодування символів на UTF-
8

    $phpmailer->isHTML(true); // Встановлюємо формат повідомлення HTML
    $phpmailer->isSMTP(); // Використовуємо SMTP для відправки листів

```

```

    $phpmailer->Host = 'smtp.gmail.com'; // Вказуємо хост SMTP-сервера (у
цьому випадку використовується Gmail)
    $phpmailer->SMTPAuth = true; // Вимагаємо аутентифікацію на SMTP-сервері

    $phpmailer->Port = 587; // Вказуємо порт SMTP-сервера (у випадку Gmail -
587)
    $phpmailer->Username = 'dragonhomeua@gmail.com'; // Вказуємо логін
(адресу електронної пошти) для авторизації на SMTP-сервері
    $phpmailer->Password = 'vckqhrplbvvnurx'; // Вказуємо пароль для
авторизації на SMTP-сервері
    $phpmailer->SMTPSecure = 'tls'; // Встановлюємо тип безпеки з'єднання з
SMTP-сервером (у випадку Gmail - tls)
    $phpmailer->SMTPKeepAlive = true; // Залишаємо з'єднання з SMTP-сервером
активним після відправки кожного листа
    $phpmailer->setFrom('dragonhomeua@gmail.com', 'List manager'); //
Встановлюємо адресу та ім'я відправника
    $phpmailer->Subject = "Замовлення на сайті DragonHome"; // Встановлюємо
тему листа
    // Масив користувачів, яким буде надіслано листи
    $users = [
        ['email' => $_SESSION['user']['email'], 'name' =>
$_SESSION['user']['full_name']]
    ];
    // Ітеруємося по користувачам та надсилаємо листи
    foreach ($users as $user) {
        // Додаємо адресу отримувача
        $phpmailer->addAddress($user['email'], $user['name']);

        // Встановлюємо тіло повідомлення (HTML-версію)
        $phpmailer->Body = $body;

        // Встановлюємо альтернативний текст повідомлення (текстову версію)
        $phpmailer->AltBody = $body;

        try {
            // Надсилаємо лист
            $phpmailer->send();
            echo "Повідомлення надіслано на: ({$user['email']}) {$phpmailer-
>ErrorInfo}\n";
        } catch (Exception $e) {
            echo "Помилка відправки ({$user['email']}) {$phpmailer-
>ErrorInfo}\n";
        }
        $phpmailer->clearAddresses(); // Очищуємо адреси отримувачів для
наступного листа
    }

    $phpmailer->smtpClose(); // Закриваємо з'єднання з SMTP-сервером

```