ЗМІСТ

Вступ 4

1 Постановка задачі дослідження 6

1.1. Аналіз предметної області 6

1.2. Огляд існуючих рішень за обраною темою 7

1.2.1. Веб-сайт по продажу мебелі «Pufetto 7

1.2.2. Веб-сайт з продажу «Sofino» 8

1.2.3. Веб-сайт «Mebli.ars.ua»» 10

1.2.4. Сервіс онлайн купівлі «Dls.ua» 11

1.3. Постановка задачі 13

2 Спеціфікація вимог до програмеого забезпечення (Software requirement specification) 15

2.1 Загальна інформація щодо веб-застосунку 15

2.2.1 Огляд продукту 15

2.2.2 Мета 16

2.2.3 Припущення й залежності 16

2.2.4 Межі 16

2.2.5 Масштаб проекту 16

2.2.6 Обмеження реалізації 16

2.2.7 Перспективи продукту 17

2.2.8 Діаграми потоків даних 17

2.2 Вимоги до веб-застосунку 18

2.2.1 Функціональні вимоги 18

2.2.2 Вимоги до зовнішнього інтерфейсу 21

2.2.3 Нефункціональні вимоги 21

2.3 Опис обраних технологій для розробки веб-застосунку 22

2.3.3 Мова гіперрозмітки HTML5 22

2.3.4 Каскадні стилі CSS3 23

2.3.5 Мова програмування JavaScript 25

2.3.6 Серверна платформа Node.js 27

2.3.8 База даних MongoDB 29

2.3.9 Фреймворк Angular 30

2.4 Опис архітектури веб-застосунку 31

2.4.1 Діаграма розміщення 31

2.4.2 Діаграма компонентів 32

2.5 Модель бази даних 34

3 Опис веб-застосунку 34

3.1 Тестування програмного забезпечення 35

3.1.1 Валідація програмного продукту 35

3.1.2 Навантажене тестування 36

3.2 Керівництво користувача 38

3.3 Використання веб-застосунку 41

Висновки 44

Список джерел інформації 45

# **ВСТУП**

З розвитком конкуренції на ринку, перед підприємцями виникла необхідність шукати нові ніші на ринку для розвитку свого бізнесу, своїх проектів. Відмінним місцем для відкриття своєї справи, стала мережа Інтернет, яка об'єднує в собі щодня мільярди людей по всьому світу[1].

Сьогодні життя протікає дуже динамічно, і у багатьох людей просто немає часу на походи по магазинах. Зараз набагато простіше - не відходячи від своїх занять, завжди можна відкрити ноутбук або планшет, зайти на потрібний веб-сайт і придбати там всі необхідні товари.

Таким чином, для підприємців, Інтернет-магазин - це відмінна можливість, не витрачаючи безлічі засобів на відкриття, запустити свій проект.

Метою даної роботи є створення веб-сайту «Інтернет-магазин з продажу меблів». При створенні сайту, спочатку, буде проведено аналітичний етап, за результатами якого стане відомо, що робити і навіщо. На етапі проектування будуть описані методи і етапи написання веб-сайту. Веб-сайт буде створений за допомогою популярних технологій створення веб-сторінок, таких як HTML, CSS і JavaScript. У висновку буде проведено тестування і налагодження веб-сайту.

Результатом роботи стане веб-сайт «Інтернет-магазин з продажу меблів».

1. **ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ**
   1. **Аналіз предметної області**

Предметною областю обраного мною проекту є інтернет магазин з продажу мебелі. В якому представлені різна мебельна фурнітура для потенціального клієнта. Основними елементами предметної області є мебель, яка позначена наступними ознаками - назва, фірма опис, модель, ціна. Дана предметна область містить інформацію про покупців, замовленнях, переліком товарів.

Передбачувана система повинна забезпечувати роботу інтернет магазину з продажу мебельної фурнітури різних видів, дана система буде обслуговувати конкретний один магазин, а не систему магазинів в цілому. Тема покупок онлайн дуже популярна в наш час, так як сам предмет – меблі та облаштування свого дому користується великою популярністю у населення.

Інтернет ресурс спрощує процедуру пошуку необхідної інформації про товари і ціни на них. З його допомогою клієнт може легко дізнатися інформацію товара та замовити його.

* 1. **Огляд існуючих рішень за обраною темою**

Огляд існуючих інформаційної систем спрямованих на обслуговування автомобілів є дуже важливим пунктом в цій роботі. Мета цього – порівняльна таблиця всіх обраних віртуальних систем, виявлення їх недоліків і переваг.

Критерії оцінки інформаційної системи: дизайн, асортимент послуг, інформативність, консультація в реальному часі, реєстрація, простота і зрозумілість сайту.

* + 1. **Веб-сайт по продажу мебелі «Pufetto»**

Український російськомовний веб-сайт Pufetto.ua надає необхідну інформацію для купівлі меблів [2].

На сайті є наступні категорії: дивани, крісла, пуфи, матраси та інші аксесуари.

Також на даному сайті є багато підкатегорії для пошуку товарів.

Перейшовши на сайт, користувач потрапляє на головну сторінку, яка показана на рисунку 1.1.

Тут користувач може вибрати необхідний розділ, щоб отримати відповідну інформацію. Для замовлення потрібно авторизуватися або зареєструватися на даному сайті, після чого покупець може вибрати бажаний вид продукції. Оформлення заказу також можуть робити неавторизовані користувачі. Після замовлення товару, продавці магазину зв'язуються з клієнтом для підтвердження замовлення. Оплата замовлення здійснюється наступними способами: за допомогою готівки кур’єру, за допомогою готівки при отриманні в поштовому відділені, оплата замовлення онлайн за банківською карткою Visa і MasterCard.

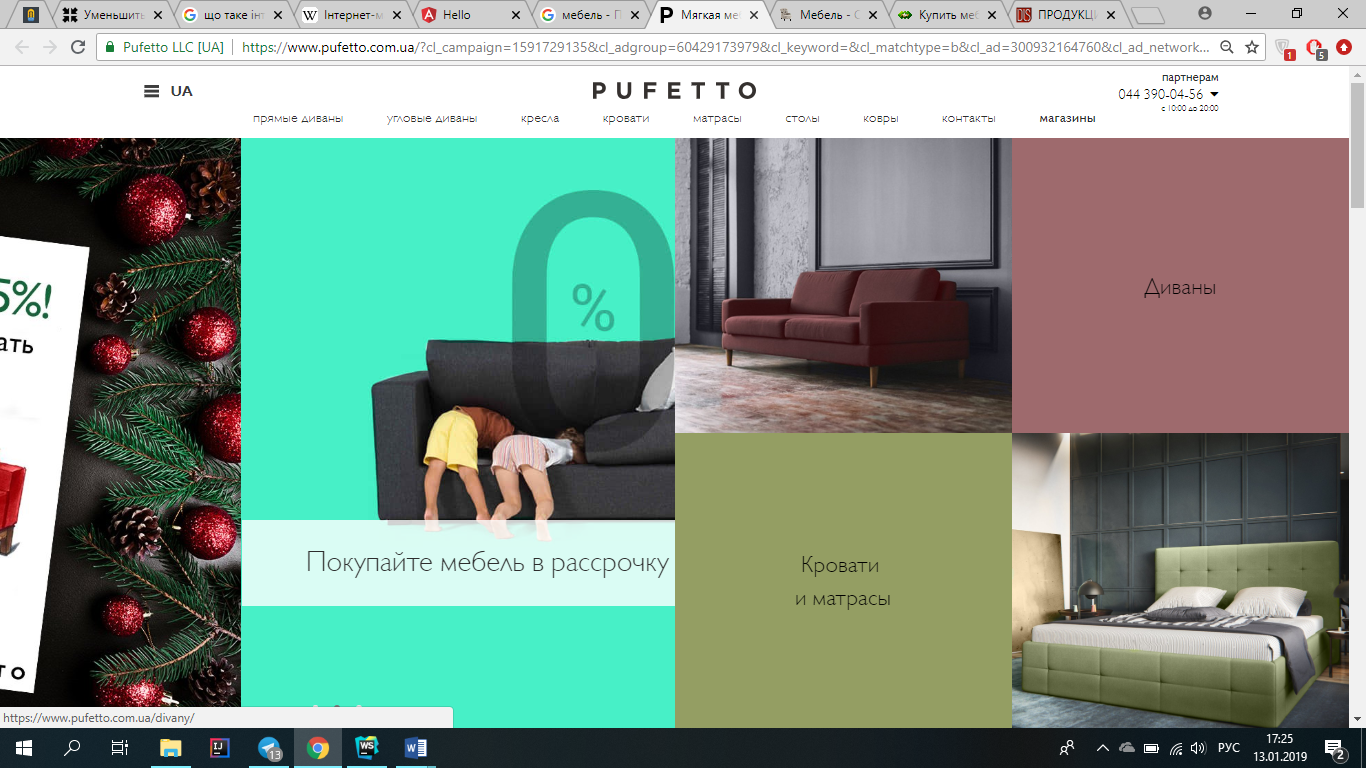


Рисунок 1.1 – Інтерфейс сервісу Pufetto.

* + 1. **Веб-сайт по продажу «Sofino.ua»**

Український веб-сайт Sofino.ua надає необхідну інформацію для покупки взуття [3].На сайті є наступні категорії : дивани, крісла, пуфи, матраси та інші аксесуари.

Також на даному сайті є багато підкатегорії для пошуку товарів.

Перейшовши на сайт користувач потрапляє на головну сторінку, яка показана на рисунку 1.2.

Сайт являє собою віртуальний магазин з продажу справжніх світових брендів мебельної фурнітури. Даний сайт є добре структурованим, усі товари розкладені, немов по поличках, що істотно полегшує роботу користувачам. Оформлення заказу також можуть робити неавторизовані користувачі. Після замовлення товару, продавці магазину зв'язуються з клієнтом для підтвердження замовлення. Оплата замовлення здійснюється наступними способами: за допомогою готівки кур’єру, за допомогою готівки при отриманні в поштовому відділені, оплата замовлення онлайн за банківською карткою Visa і MasterCard.

На сторінках магазину представлений великий вибір меблів. Є умови знижок.

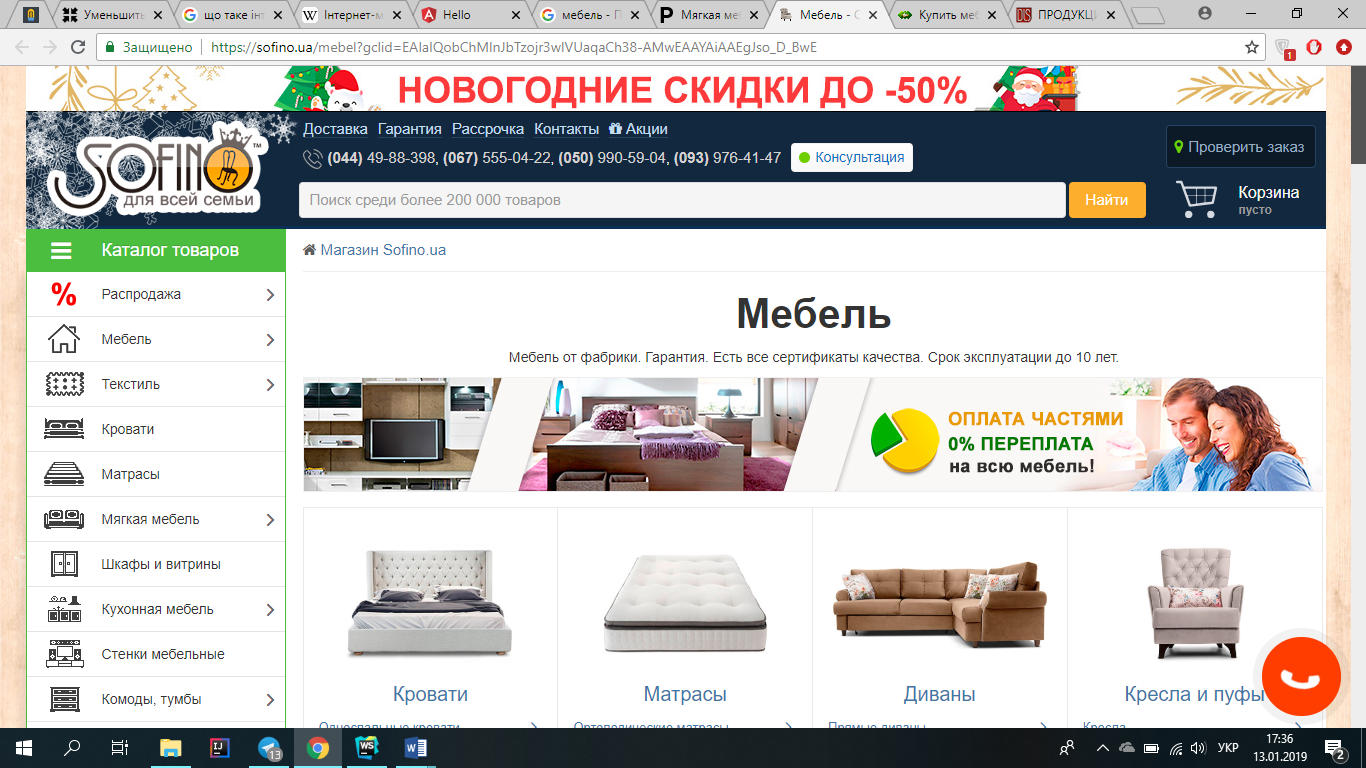


Рисунок 1.2 – Інтерфейс сервісу Sofino.ua.

* + 1. **Веб-сайт «Mebli.ars.ua»**

Офіційний веб-сайт «Mebli» – сайт на якому розташовано багато категорій меблів[4].

Перейшовши на сайт користувач потрапляє на головну сторінку, яка показана на рисунку 1.3.

Сайт являє собою віртуальний магазин з продажу меблів провідних брендів. Даний сайт є добре оформленим і має відмінну структуру.

На даному сайті розрізняються наступні категорії. Кожен товар має свій опис: фотознімки з декількох ракурсів, коротка характеристика моделі або марки, ціна, дані про наявність моделі на складі, вид товару, країна виробник, сезон. Для замовлення потрібно авторизуватися або зареєструватися на даному сайті, після чого покупець може вибрати бажаний вид продукції.

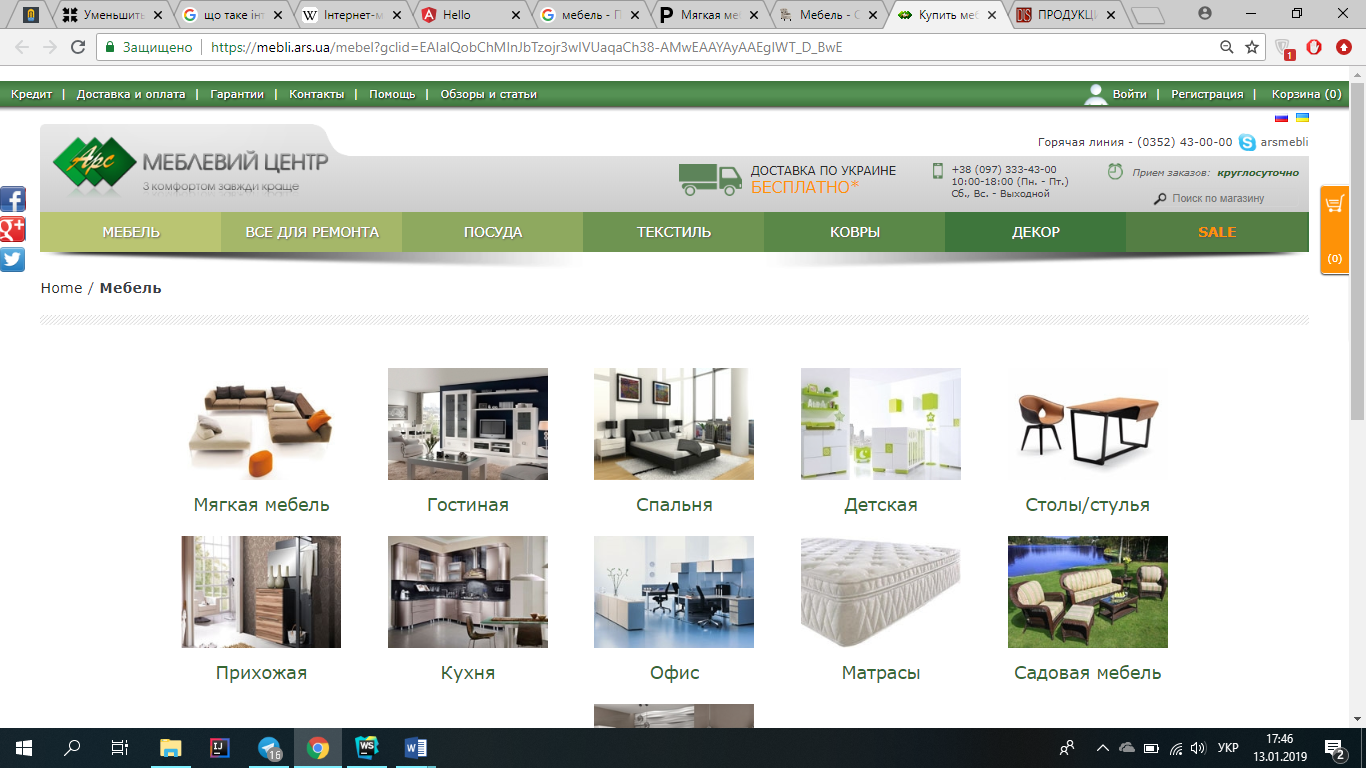


Рисунок 1.3 – Інтерфейс сервісу Mebli.

* + 1. **Сервіс онлайн запису на автосервіс «Dls.ua»**

Сайт являє собою віртуальний магазин продажу фірмової мебельної фурнітури провідних світових брендів. Даний сайт є добре оформленим і має відмінну структуру [5].

Інтерфейс інтернет-магазину зображено на рисунку 1.4.

На даному сайті розрізняються наступні категорії мебельної фурнітури: дивани, крісла, пуфи, матраси та інші аксесуари.

Спеціалізацією інтернет-магазину є виключно меблі..

Кожен товар має свій опис: фотознімки з декількох ракурсів, коротка характеристика моделі або марки, ціна, дані про наявність моделі на складі, вид товару, країна виробник, сезон. Для замовлення потрібно авторизуватися або зареєструватися на даному сайті, після чого покупець може вибрати бажаний вид продукції.

Оформлення заказу також можуть робити неавторизовані користувачі. Після замовлення товару, продавці магазину зв'язуються з клієнтом для підтвердження замовлення. Оплата замовлення здійснюється наступними способами: за допомогою готівки кур’єру, за допомогою готівки при отриманні в поштовому відділені, оплата замовлення онлайн за банківською карткою Visa і MasterCard.

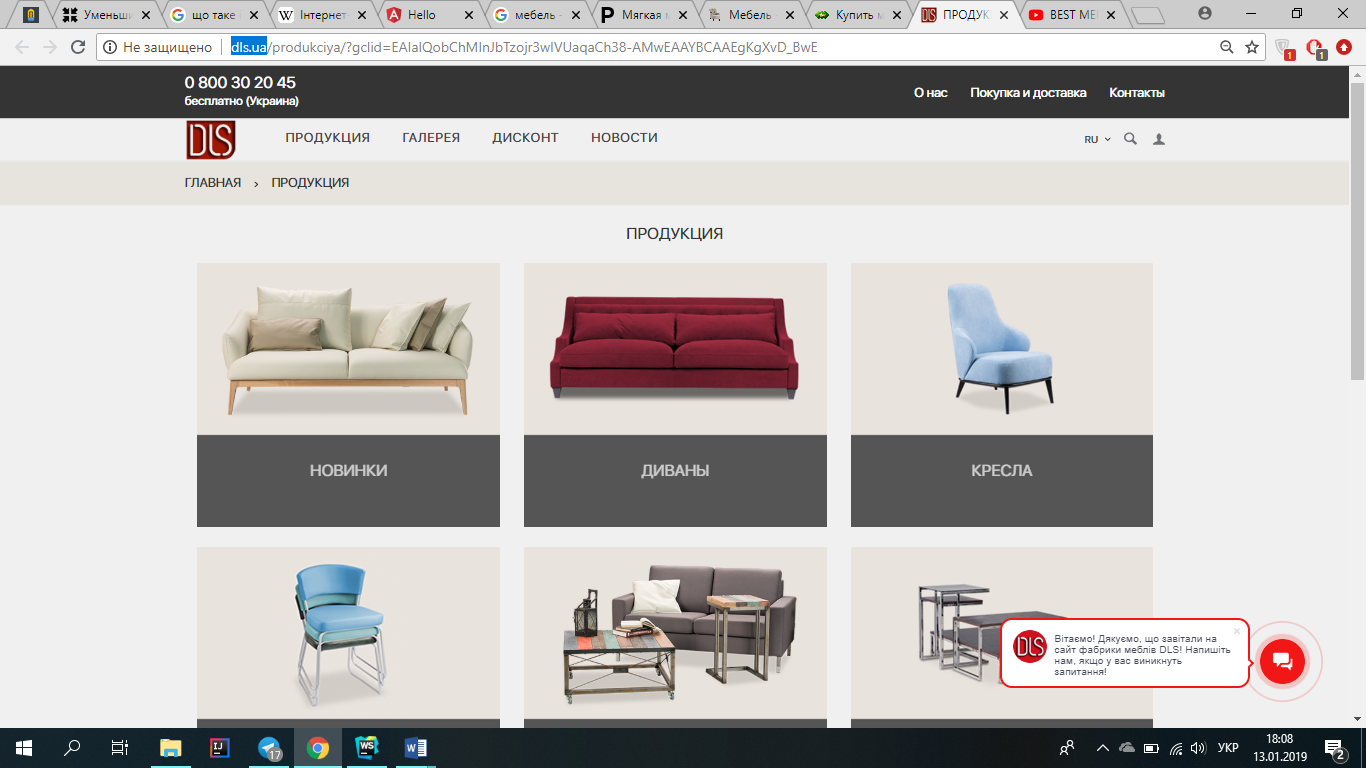


Рисунок 1.4 – Інтерфейс сервісу Dls.

Таблиця 1.1 – Порівняння інформаційних систем.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pufetto | Sofino | Mebli | Dls |
| Дизайн | Сучасний, зрозумілий | Зрозумілий, простий | Детальний, пророблений | Сучасний, якісний |
| Інформа-тивність | Велика | Середня | Велика | Мала |
| Реєстрація | Присутня | Присутня | Присутня | Присутня |
| Консультація | Присутня | Присутня | Присутня | Присутня |
| Недоліки | Ціна розробки | Малий каталог продукції. | Незручно реалізована навігація по сайту | Малий каталог продукції. |

Всі представлені сервіси задовольняють критеріям. При цьому вартість продукції на сайті «Pufetto» є досить дорогою, на звичайні послуги. Аналізуючи статистику відвідування сайтів з даною предметною областю, можна прийти до висновку, що купівля меблів в інтернеті використовуючи інформаційні системи наразі є досить актуальним і популярним. Дивлячись на розвиток розробки інформаційних систем ми бачимо, що розробка систем даного типу втратити актуальність найближчим часом не зможе.

* 1. **Постановка задачі**

Метою виконання курсової роботи є розробка веб-сервісу, який забезпечує автоматизацію рішення задач зберігання й обробки даних, що стосуються продажу меблів. До складу такого додатку обов'язково повинна входити база даних і прикладне програмне забезпечення, призначене для вирішення практичних завдань і орієнтоване на кінцевого користувача, тобто користувача, що не має спеціальної підготовки.

Розроблений веб застосунок повиннен зберігати інформацію про меблі.

Основні цілі, які повинні бути досягнуті в результаті виконання даної роботи, можна сформулювати наступним чином:

- дослідження і опис предметної області,

- вибір cистеми управління базами даних і засобів розробки прикладного програмного забезпечення,

- розробка прикладного програмного забезпечення,

- експериментальна експлуатація прикладного програмного забезпечення на контрольних даних і аналіз отриманих результатів,

- налагодження і тестування прикладного програмного забезпечення.

Передбачити на сайті можливість перегляду товарів, ознайомлення з описом товару, додавання товару, видалення товару, статус замовлення. Включити в функції сайту реєстрацію користувачів і оформлення замовлення на окремій сторінці, куди передається статус замовлень для оплати. Необхідно передбачити відповідний дизайн, інтерфейс повинен бути зрозумілий навіть не дуже досвідченому користувачеві.

Сайт повинен повністю давати уявлення користувачеві про інтернет-магазин. Веб-додаток має містити серверну частину і адміністраторську панель для роботи з наповненням сайту і різними сервісами. Безумовно, одним з найважливіших вимог є красивий і приємний дизайн сайту, який би гармонійно виглядав в сукупності з усіма наданими функціями.

Кожен вид меблів повинний мати зображення, головні характеристики, назву та ціну. Повинна бути присутня кнопка «Купити» для кожної номенклатури товару. Присутність консультації в реальному часі є обов’язковою – це може бути телефонний номер або електронна адреса.

Необхідно забезпечити інтернет-магазин кошиком покупок, куди клієнт може заносити обраний товар, після остаточного вибору, клієнт оформляє замовлення в даному пункті.

1. **СПЕЦИФІКАЦІЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (SRS – SOFTWARE REQUIREMENT SPECIFICATION)**
   1. **Загальна інформація щодо веб-застосунку**

Система інтернет-покупок для інтернет-магазину з продажу меблів призначена для надання повних рішень для працівників магазину, які в свою чергу є адміністраторами, а також для клієнтів через єдиний спосіб використання інтернету. Це дозволить клієнтам переглядати магазин і купувати речі в інтернеті без фізичного відвідування магазину.

* + 1. **Огляд продукту**

**«**Мир Мебели**»**– це інформаційна система, яка дозволить замінити поточні процеси замовлення товару у сервісі обслуговування **«**Мир Мебели**»**. Контекстна діаграма (рис 2.1) показує зовнішні об'єкти і системні інтерфейси для версії 1.0. Передбачається випустити кілька версій системи, щоб в кінцевому підсумку вдалося розширити систему покупок та використовувати служби авторизації кредитних і дебетових карт, і здійснювати оплату за меблі.

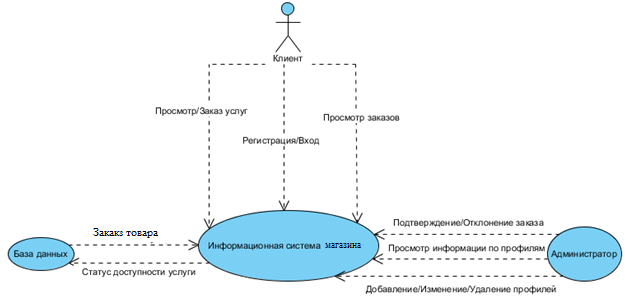


Рисунок 2.1 - Контекстна діаграма для розроблюваної інформаційної системи.

* + 1. **Мета**

Визначити товари, які найчастіше замовляють клієнти для оптимізації робити магазину.

Збільшити дохід магазину від замовлень.

Збільшити середній ефективний робочий час кожного співробітника протягом трьох місяців після першого випуску системи.

* + 1. **Припущення й залежності**

У інтернет-магазині будуть встановлені комп'ютери та принтери, підключені до інтернету, що дозволить співробітникам обробляти необхідний обсяг замовлень, не пропускаючи їх.

* + 1. **Межі**

Обслуговування клієнтів виконується в межах країни.

* + 1. **Масштаб проекту**

Інформаційна система дозволить співробітникам **«**Мир Мебели**»** замовляти товар в магазині через інтернет для обслуговування клієнтів.

Функції, реалізація яких запланована в цьому випуску:

1. Замовлення послуг з переліку.
2. Створення, перегляд, зміна і видалення замовлень у базі даних сервісу.
3. Забезпечення доступу до системи через мережу інтернет для авторизованих працівників.
4. Додання нових користувачів до бази даних.
5. Редагування інформації про користувача.
   * 1. Обмеження реалізації

Увесь код HTML повинен відповідати стандарту HTML 5.0.

Весь код CSS повинен відповідати рівню CSS3.

Всі сценарії повинні бути написані на TypeScript.

* + 1. **Перспективи продукту**

40% відвідувачів системи повинні почати використовувати інформаційну систему протягом 6 місяців після першого випуску системи.

Досягти збільшення середнього рейтингу по щоквартальному опитуванню про задоволення роботою магазину на 0,7 бали протягом 6 місяців після першого випуску системи і на 1,0 бали протягом 12 місяців після першого випуску системи.

* + 1. **Діаграми потоків даних**

Для опису присутньої функціональності були побудовані діаграми потоків даних (DFD – data flow diagram) (рис 2.2, 2.3).

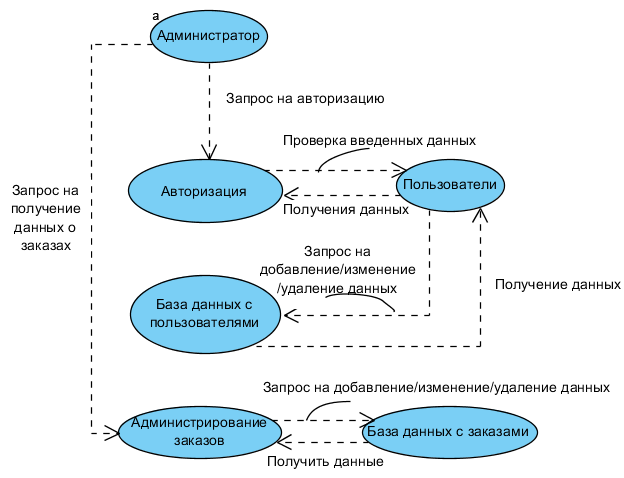


Рисунок 2.2 – Діаграма потоків даних (адміністратор).

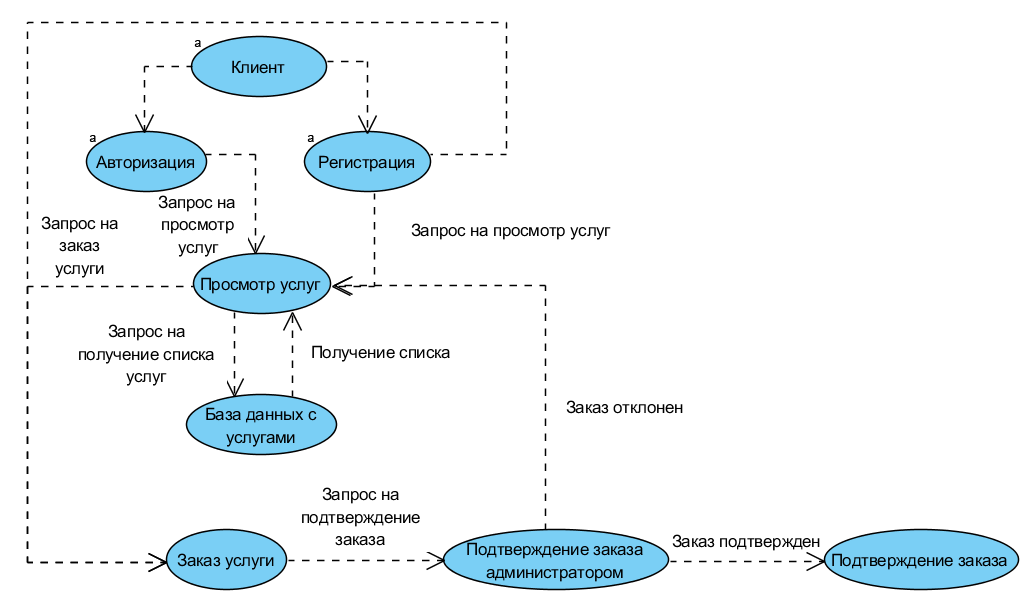


Рисунок 2.3 – Діаграма потоків даних (клієнт)

* 1. **Вимоги до веб-застосунку**
     1. **Функціональні вимоги**

Функціональні вимоги наведені в таблиці 2.1.

Таблиця 2.1 – Опис функціональних вимог до веб-застосунку.

|  |  |
| --- | --- |
| 1)Вимога | 2)Описання |
| Реєстрація.Клієнт | Якщо клієнт бажає зробити замовлення, то він має бути зареєстрованим. |
| Реєстрація.Клієнт.Дані | Для реєстрації клієнт має ввести необхідні дані в реєстраційну форму. |
| Реєстрація.Клієнт.Завершення | Після того як клієнт натисне кнопку «Зареєструватися», система робить наступну транзакцію: |

Продовження таблиці 2.1

|  |  |
| --- | --- |
| 1)Вимога | 2)Описання |
| Реєстрація.Клієнт.Перевірка | Після подання форми реєстрації клієнтом, система має провірити чи не існує даний клієнт в базі даних. |
| Реєстрація.Клієнт.Підтвердження | При невиявлені даного клієнта в базі даних, додати нового клієнта до бази даних з статусом «Реєстрація пройшла успішно». |
| Реєстрація.Клієнт.Відмова | При виявлені даного клієнта в базі даних, завершити транзакцію з статусом «Даний клієнт уже зареєстрований». |
| Вхід.Система | Для входу до системи, користувач має ввести логін і пароль в форму входу. |
| Вхід.Система.Підтвердження | При збігу даних завершити транзакцію з статусом «Вхід виконано успішно». |
| Вхід.Система.Відмова | При розбіжності даних завершити транзакцію з статусом «Невірні дані» |
| ОбліковийЗапис.Користувач | Клієнт має можливість вийти з власного облікового запису. |
| Замовлення.Оформлення | Система повинна дозволяти клієнту, зареєстрованому в системі, розміщувати замовлення на одну і більше послугу. |

Кінець таблиці 2.1

|  |  |
| --- | --- |
| 1) | 2) |
| Замовлення.Товара | Якщо на цей час замовити не можливо , клієнт повинен мати можливість змінити час замовлення. |
| Замовлення.Підтвердження | Клієнт підтверджує замовлення. |
| Замовлення.Підтвердження.Відмова | Якщо клієнт не підтверджує замовлення, він може або змінити, або скасувати його. |
| Замовлення.Завершення | Після того як клієнт підтвердив замовлення, система повинна зробити наступну як одну транзакцію: |
| Замовлення.Завершення.Збереження | призначити замовленню наступний доступний номер і зберегти замовлення. |
| Замовлення.Завершення.Скасування | Адміністратор скасував замовлення. |
| Замовлення.Завершення. Підтвердження | Адміністратор підтвердив замовлення. |
| Замовлення.Завершення.Оновлення | Оновлюється список послуг та доступний час за результатом проведених транзакцій. |

* + 1. **Вимоги до зовнішнього інтерфейсу**

Система буде взаємодіяти з базою даних інформаційної системи інтернет-магазину.

Система забезпечить користувачам можливість доступу до інформаційної системи інтернет-магазину через мережу Інтернет. Розглядається два інтерфейси користувачів, які будуть супроводжувати даний веб-сайт: інтерфейс для користувачів та для адміністраторів.

Користувачі зможуть входити до системи після реєстрації, після чого будуть мати можливість переглядати послуги, а також робити замовлення.

Адміністратори, після входу до системи, будуть мати можливість підтверджувати/скасовувати замовлення, редагувати та видаляти користувачів.

Інтернет-сторінки повинні надавати повну можливість навігації і вибір послуг за допомогою пристрою-маніпулятора.

Інформаційна система повинна передавати замовлені послугі базі даних через програмний інтерфейс.

Інформаційна система повинна опитувати базу даних для визначення вільного часу для бронювання.

* + 1. **Нефункціональні вимоги**

Поняття надійності можна поясними тим. Якщо з'єднання між користувачем і системою розривається до того, як замовлення підтверджено або скасовано, інформаційна система повинна дозволяти користувачеві побачити незавершене замовлення.

Інформаційна система інтернет-магазину повинна бути доступною для всіх користувачів через мережу інтернет. Інформаційна система буде працювати 24 години на добу.

Користувачі повинні обов'язково реєструватися для входу в інформаційну систему для виконання операцій перегляду і замовлення послуг.

Клієнти повинні реєструватися для входу в систему згідно з політикою обмеженого доступу до комп'ютерних систем. Також система повинна дозволяти тільки співробітникам магазину, внесеним до списку адміністраторів, створювати, видаляти або змінювати замовлення.

Читабельність коду можна розглядати як важливу складову супроводжуваності. Код повинний бути зручним для аналізу, змін та тестування. Має бути присутня модифікація, яка здійснюється для усунення дефектів, удосконалення або адаптації інформаційної системи до змін в умовах функціонування, a також в складі і особливостях необхідних функцій є також одним із важливих факторів.

Додання, видалення, зміна характеристик послуг повинні бути зручними для адміністратора інформаційної системи.

Магазин інсталяції не потребує, так як він буде доступний користувачеві за допомогою мережі інтернет і тому буде мати відмінну швидкість доступу до бази даних та завантаження сторінок, не завантажувати процесор та оперативну пам'ять.

**2.3 Опис обраних технологій для розробки веб-застосунку**

**2.3.3 Мова гіперрозмітки HTML5**

HTML (від англ. HyperText Markup Language - «мова гіпертекстової розмітки») – стандартизована мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузерами; отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою[5].

Мова HTML до 5-ї версії визначалася як додаток SGML (стандартної узагальненої мови розмітки за стандартом ISO 8879). Специфікації HTML5 формулюються в термінах DOM (об'єктній моделі документа).

Мова XHTML є більш суворим варіантом HTML, вона є слідом синтаксису XML і є додатком мови XML в області розмітки гіпертексту.

У всесвітній павутині HTML-сторінки, як правило, передаються браузерам від сервера по протоколах HTTP або HTTPS, у вигляді простого тексту або з використанням шифрування.

Переваги HTML5:

* легкість створювати програмний код;
* усунення таблиці div;
* використання елементів article, footer, nav, header та ін.;
* у вікні браузера можна відтворювати Flash (анімації, відеоролики, музику) і для цього не потрібні драйвера і утиліти;
* процес обміну даними з сервером став поліпшеним;
* отримання оповіщень про помилки при роботі сайту;

Недоліки HTML:

* щоб створювати якісні проекти, доводиться серйозно попрацювати;
* при замовленні сайтів на HTML5, доведеться платити більше;
* деякі фахівці стверджують, що безпека після застосування нової технології стала нижче;
* розробники браузерних платформ намагаються не відставати від сучасних технологій, тим не менш, є деякі програми, в яких не вірно відображаються сайти на HTML5.

В цілому дана технологія є відмінною заміною застарілого HTML.

**2.3.4 Каскадні стилі CSS3**

CSS (англ. Cascading Style Sheets – каскадні таблиці стилів) – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки[6].

Переважно використовується як засіб опису, оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки HTML і XHTML, але може також застосовуватися до будь-яких XML-документах, наприклад, до SVG або XUL.

CSS використовується творцями веб-сторінок для задання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду цих веб-сторінок. Основною метою розробки CSS було розділення опису логічної структури веб-сторінки (яка проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (яка тепер проводиться за допомогою формального мови CSS). Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне уявлення, друковане подання, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля.

Переваги CSS:

1. Простота самої мови CSS разом з принципом відділення оформлення від змісту дає можливість скоротити час на розробку і підтримку сайту.
2. Є кілька варіантів дизайну сторінки для перегляду на різних пристроях. Наприклад, дизайн на екрані комп'ютера розрахований на одну ширину, і буде повністю виводитися на екран, а на мобільних пристроях він буде підлаштовуватися до розмірів екрану і деякі елементи будуть виключені від показу, також і при друці, буде друкуватися потрібний текст, без зайвого (наприклад, без шапки меню).
3. Зменшується час завантаження сторінок web-сайту за рахунок перенесення правил представлення даних в окремий CSS-файл. Завдяки цьому браузер завантажує тільки структуру документа, а також дані, що зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і може бути закешований, - завдяки цьому зменшується трафік, час завантаження, а також навантаження на сервер.
4. Простота зміни дизайну. Один CSS управляє відображенням безлічі HTML-сторінок. Коли виникає необхідність змінити дизайн сайту, то нема чого правити кожну сторінку. Для подальшої зміни дизайну всього лише потрібно змінити CSS-файл, і як результат, зміна дизайну робиться швидше.
5. CSS надає додаткові можливості форматування, про яких при використанні тільки самих атрибутів навіть і не мріяли.
6. Підвищення сумісності з різними платформами за рахунок використання web-стандартів.

Недоліки CSS:

1. Різне відображення верстки в різних браузерах. Якщо браузери застарілі, то можливо, що одні й ті ж дані CSS по-різному ними інтерпретуються.
2. Необхідність виправляти не тільки один CSS-файл, але і теги HTML. Часто зустрічається на практиці необхідність виправляти не тільки один CSS-файл, але і теги HTML, які пов'язані з селекторами CSS. Іноді це значно збільшує час редагування, а також і тестування.

Отже, CSS – це найсильніший інструмент і один з основних складових практично будь-якого web-сайту. Без CSS просто неможливо уявити сучасну діяльність web-розробників. Сьогодні CSS – це загальноприйнятий стандарт розробки, який приймається усіма без винятку компаніями-розробниками, що явно показує його значимість і необхідність.

### 2.3.5 Мова програмування JavaScript

JavaScript (JS) – це динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки[7].

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

* написання сценаріїв веб-сторінок для надання їм інтерактивності;
* створення односторінкових веб-застосунків (ReactJS, AngularJS, Vue.js);
* програмування на стороні сервера (Node.js);
* стаціонарних застосунків (Electron, NW.js);
* мобільних застосунків (React Native, Cordova);
* сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
* всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови С, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

Переваги JavaScript:

1. Жоден сучасний браузер не обходиться без підтримки JavaScript.
2. З використанням написаних на JavaScript плагінів і скриптів впорається навіть не фахівець.
3. Корисні функціональні настройки.
4. Постійно удосконалюється мова – зараз розробляється бета-варіація проекту, JavaScript2.
5. Взаємодія з додатком може здійснюється навіть через текстові редактори – Microsoft Office і Open Office.
6. Перспектива використання мови в процесі навчання програмуванню і інформатиці.

Недоліки JavaScript:

1. Знижений рівень безпеки через повсюдний і вільного доступу до вихідного коду популярних скриптів.
2. Безліч дрібних дратівливих помилок на кожному етапі роботи. Велика частина з них легко виправляється, але їх наявність дозволяє вважати цю мову менш професійною, порівняно з іншими.
3. Повсюдне поширення. Своєрідним недоліком можна вважати той факт, що частина активно використовуваних програм (особливо додатків) перестануть існувати при відсутності мови, оскільки цілком базуються на ньому.

### 2.3.6 Серверна платформа Node.js

Node.js – платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript[8]. Засновником платформи є Раян Дал (Ryan Dahl). Якщо раніше Javascript застосовувався для обробки даних в браузері на сторонні користувача, то node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їх виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

Node.js має наступні властивості:

* асинхронна однопотокова модель виконання запитів;
* не блокуючий ввід/вивід;
* система модулів CommonJS;
* рушій JavaScript Google V8.

Для керування модулями використовується пакетний менеджер npm (node package manager).

Платформа Node.js призначена для виконання високопродуктивних мережевих додатків, написаних мовою програмування JavaScript. Платформа окрім роботи із серверними скриптами для веб-запитів, також використовується для створення клієнтських та серверних програм.

В платформі використовується розроблений компанією Google рушій V8.

Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, заснована на обробці подій в неблокуючому режимі та визначенні обробників зворотніх викликів (callback). Як способи мультиплексування з'єднань підтримується epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується бібліотека libuv, для створення пулу потоків (thread pool) задіяна бібліотека libeio, для виконання DNS-запитів у неблокуючому режимі інтегрований c-ares. Всі системні виклики, що спричиняють блокування, виконуються всередині пула потоків і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (pipe).

За своєю суттю Node.js схожий на фреймворки Perl AnyEvent, Ruby Event Machine і Python Twisted, але цикл обробки подій (event loop) у Node.js прихований від розробника і нагадує обробку подій у веб-застосунку, що працює в браузері. При написанні програм для Node.js необхідно враховувати специфіку подієво-орієнтованого програмування.

Переваги Node.js:

* швидкість;
* використання мови JS;
* доступність;
* активна спільнота підтримки;
* асинхронність.

Недоліки Node.js:

* не підходить для завдань, інтенсивних для ЦП;
* неефективний при широкомасштабному застосуванні;
* невідповідність, проблеми в узгодженості;
* погано підтримується реляційними базами даних.

**2.3.7 Фреймворк Express**

Express Framework, або просто Express – програмний каркас розробки веб-застосунків для Node.js, реалізований як вільне і відкрите програмне забезпечення під ліцензією MIT[9]. Він спроектований для створення веб-застосунків і API. Де-факто є стандартним каркасом для Node.js. Автор фреймворка, TJ Holowaychuk, описує його як створений на основі написаного на мові Ruby каркаса Sinatra, маючи на увазі, що він мінімалістичний, але має велику кількість плаґінів, шо підключаються.

Express є бекендом для програмного стека MEAN, разом з базою даних MongoDB і каркасом AngularJS для фронтенду.

### 2.3.8 База даних MongoDB

MongoDB – документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів[10].

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії AGPLv3.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

Основні можливості MongoDB:

1. Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних).
2. Досить гнучка мова для формування запитів.
3. Динамічні запити.
4. Повна підтримка індексів.
5. Профілювання запитів.
6. Швидкі оновлення «на місці».
7. Ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео.
8. Журналювання операцій, що модифікують дані в БД.
9. Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг.
10. Може працювати відповідно до парадигми MapReduce.

Недоліком MongoDB є використання пам'яті.

### 2.3.9 Фреймворк Angular

Angular (зазвичай так називають фреймворк Angular 2 або Angular 2+, тобто вищі версії) – написаний на TypeScript front-end фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team у компанії Google, а також спільнотою приватних розробників та корпорацій. Angular – це AngularJS, який переосмислили та який був повністю переписаний тією ж командою розробників[11].

Angular – JavaScript-фреймворк з відкритим програмним кодом, який розробляє Google. Призначений для розробки односторінкових додатків, що складаються з одної HTML сторінки з CSS і JavaScript. Його мета – розширення браузерних застосунків на основі шаблону Модель-вид-контролер (MVC), а також спрощення їх тестування та розробки.

Фреймворк працює зі сторінкою HTML, що містить додаткові атрибути і пов'язує області вводу або виводу сторінки з моделлю, яка є звичайними змінними JavaScript. Значення цих змінних задаються вручну або отримуються зі статичних або динамічних JSON-даних.

Angular – це фронтенд частина стеку MEAN, що складається з бази даних MongoDB, програмного каркасу для розробки веб-додатків Express.js, самого Angular.js та платформи Node.js.

Переваги Angular:

1. Велике ком'юніті.
2. Декларативний стиль коду.
3. Використання директив.
4. Висока швидкість розробки.
5. Використання схеми MVC, яка розділяє логіку, уявлення і дані додатка.
6. Корисні можливості для SPA.
7. Наявність готових рішень.
8. Модульність.
9. Двостороннє зв'язування даних.
10. Простота тестування.

Недоліки Angular:

1. Складність освоєння.
2. Уповільнення роботи при використанні понад 2000 слухачів подій.
3. Відсутність зворотної сумісності з другою версією.

**2.4 Опис архітектури веб-застосунку**

**2.4.1 Діаграма розміщення**

Діаграми розгортання (рис. 2.4) представляють фізичне розташування системи, показуючи, на якому фізичному обладнанні запускається та чи інша складова програмного забезпечення.

Вузол (node) – це те, що може містити програмне забезпечення. Вузли бувають двох типів. Пристрій (device) – це фізичне обладнання: комп'ютер або пристрій, пов'язане з системою. Середовище виконання (execution environment) – це програмне забезпечення, яке саме може включати інше програмне забезпечення, наприклад операційну систему або процес-контейнер.

Вузли можуть містити артефакти (artifacts), які є фізичним уособленням програмного забезпечення; зазвичай це файли. Такими файлами можуть бути виконувані файли (такі як файли .eхе, виконавчі файли, файли DLL, файли JAR, складання або сценарії) або файли даних, конфігураційні файли, HTML-документи і т. д. Перелік артефактів всередині вузла вказує на те, що на даному вузлі артефакт розгортається в систему, яка запускається.

Артефакти можна зображувати у вигляді прямокутників класів або перераховувати їх імена всередині вузла.

Артефакти часто є реалізацією компонентів. Це можна показати, задавши значення-мітки всередині прямокутників артефактів.

Інформаційні шляхи між вузлами представляють обмін інформацією в системі. Можна супроводжувати ці шляхи інформацією про використання інформаційних протоколів.

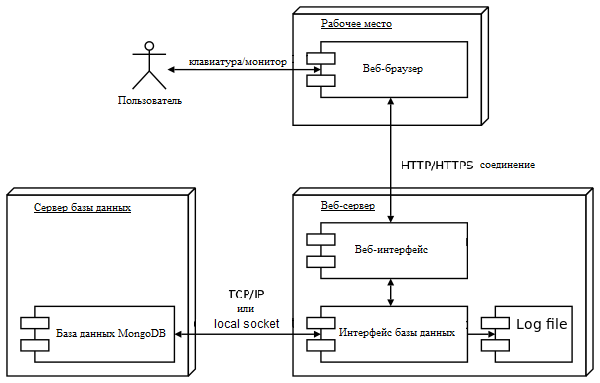


Рисунок 2.4 – Діаграма розміщення.

### 2.4.2 Діаграма компонентів

На діаграмі компонентів (рис. 2.5) показаний вид «білого ящика» внутрішньої структури двох пов'язаних підсистем – «Інформаційна система» та «Облік». В UML «підсистема» представляє собою стандартний компонентний стереотип для більших компонентів, зазвичай містить дрібніші компоненти.

Підсистема «Інформаційна система» містить три компоненти, пов'язані з інформаційною системою – «Пошукова система», «Замовлення» і «Авторизація». Компонент «Пошукова система» дозволяє шукати або переглядати елементи, виставляючи наданий інтерфейс «Пошук послуг» що надається компонентом «Товари». Компонент «Кошик» використовує інтерфейс «Управління замовленнями», що надається компонентом «Замовлення» під час перевірки. Компонент «Авторизація» дозволяє клієнтам створювати обліковий запис, вхід в систему або вийти з системи і пов'язувати клієнта з будь-яким обліковим записом.

Підсистема обліку надає два інтерфейси: «Управління замовленнями» і «Управління клієнтами». З'єднання делегацій пов'язують ці зовнішні контракти підсистеми з реалізацією контрактів компонент «Замовлення» і «Клієнти».

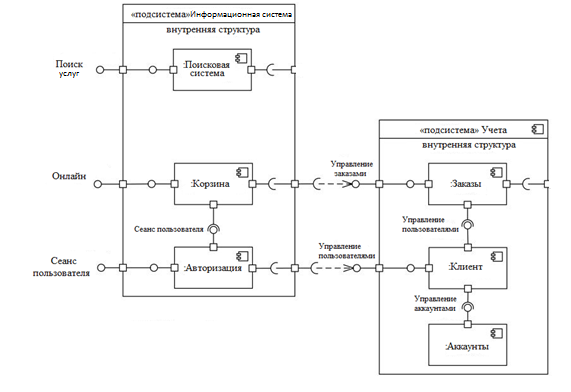


Рисунок 2.5 – Діаграма компонентів.

## **2.5 Модель бази даних**

База даних, представлена у форматі схеми, складається з двох схем (рис. 2.6), які мають наступну структуру:

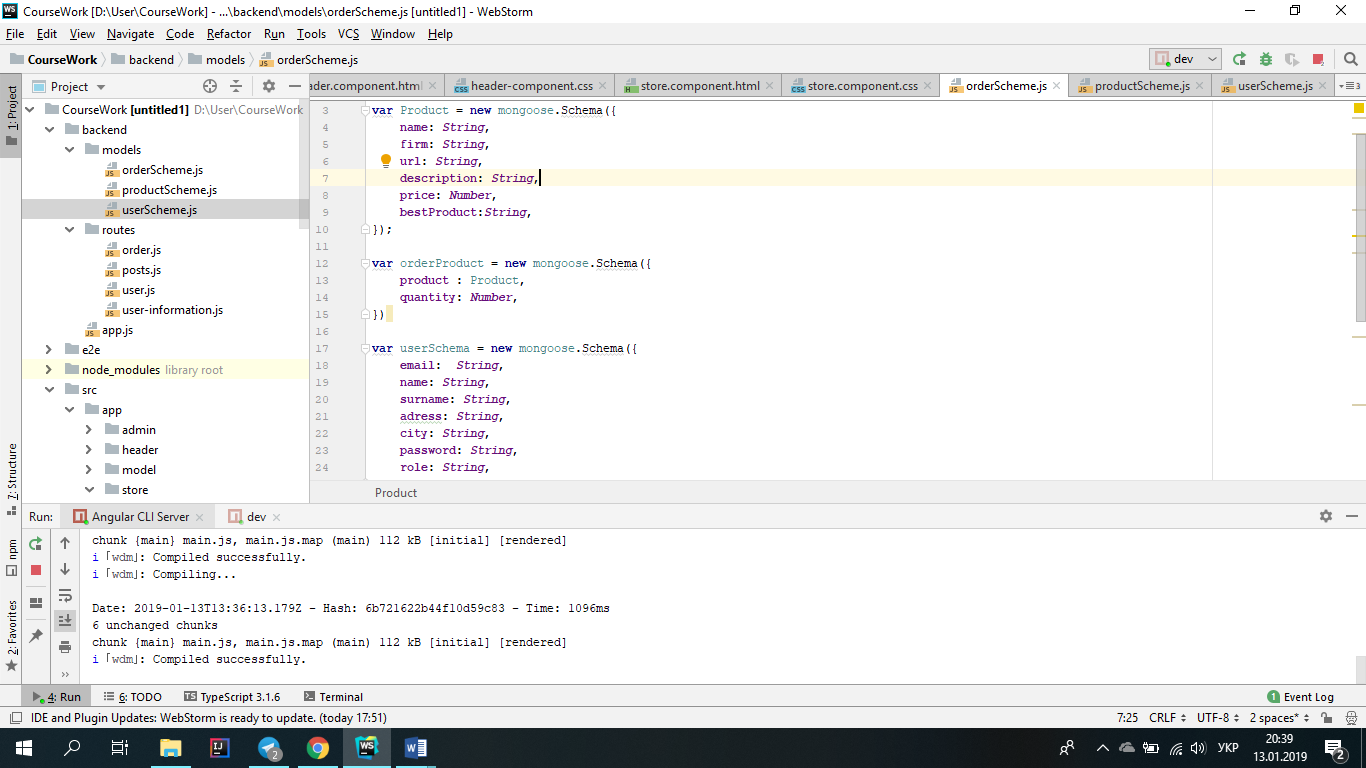


Рисунок 2.6 – Структура схеми бази даних.

Таблиця «Product» призначена для зберігання інформації про товар. Запис складається з наступних полів, опис яких наведений в таблиці 2.2.

Таблиця 2.2 ­­ – Опис структури таблиці «Product»

|  |  |  |
| --- | --- | --- |
| Ім’я поля | Тип даних | Опис |
| name | Текстовий | Назва продукту |
| firm | Текстовий | Назва фірми виробника |
| url | Текстовий | URL адреса зображення |
| description | Текстовий | Опис товара |
| price | Числовий | Ціна замовлення |
| bestProduct | Логічний | Чи є товар в карегорії «Найкращі» |

Таблиця «orderProduct» призначена для зберігання інформації про замовлення. Кожний запис складається з наступних полів, опис яких наведений в таблиці 2.3.

Таблиця 2.3 ­­ – Опис структури таблиці «orderProduct»

|  |  |  |
| --- | --- | --- |
| Ім’я поля | Тип даних | Опис |
| product | Текстовий | Id замовлення |
| quantity | Текстовий | Кількість продукції в замовленні |

Таблиця «userSchema» призначена для зберігання інформації про зареєстрованих клієнтів. Кожний запис складається з наступних полів, опис яких наведений в таблиці 2.4.

Таблиця 2.4 ­­ – Опис структури таблиці «userSchema»

|  |  |  |
| --- | --- | --- |
| Ім’я поля | Тип даних | Опис |
| email | Текстовий | Пошта користувача |
| name | Текстовий | Ім’я користувача |
| surname | Текстовий | Прізвище користувача |
| address | Текстовий | Адреса користувача |
| password | Текстовий | Пароль користувача |
| role | Логічний | Вибирає чи користувач є адміністратором |
| city | Текстовий | Місто користувача |

1. **ОПИС ВЕБ-ЗАСТОСУНКУ**

## **3.1 Тестування програмного забезпечення**

### 3.1.1 Валідація програмного продукту

Валідація програмного забезпечення — це процес визначення відповідності розроблюваного програмного забезпечення між очікуваннями і потребами користувача, вимогам до системи. Валідація є одним із основних етапів тестування програмного забезпечення.

Процес валідації може проводитися самим виконавцем або іншою особою, наприклад, замовником, що здійснює дії з впровадженню і проведенню цього процесу за планом, у якому відбиті елементи і задачі перевірки. При цьому використовуються методи, інструментальні засоби і процедури виконання задач процесу для встановлення відповідності тестових вимог і особливостей використання програмних продуктів проекту на правильність реалізації вимог. Верифікація і валідація полягає в перевірці специфікацій і правильності виконання програм відповідно до заданих вимог і формального опису програми.

Валідація дозволяє встановити здійснимість заданих вимог шляхом їх перегляду, інспекції і оцінки результатів проектування на процесах життєвого циклу для підтвердження того, що здійснюється коректна реалізація вимог, дотримання заданих умов і обмежень до системи. Верифікація і валідація забезпечують перевірку повноти, несуперечності і однозначності специфікації і правильності виконання функцій системи.

Очікування користувача:

* Можливість зареєструвати особистий кабінет у магазині.
* Можливість купити продукцію магазину
* Можливість переглядати усі замовлення клієнта

Результати проведення валідації:

* Була створена сторінка з функціями реєстрації/авторізаціїї, яка зберігаю дані у базі даних на сервері.
* Була реалізована функція відновлення паролю через електронну пошту прив’язану до особистого кабінету.
* Була реалізована функція для адміністратора системи, яка дозволяє переглядати усі замовлення за обраними датами.

У результаті валідації програмний продукт є повнофункціональним, що пройшов верифікацію та валідацію.

**3.1.2 Навантажене тестування**

Навантажене тестування — підвид тестування продуктивності, збір показників і визначення продуктивності і часу відгуку програмно-технічної системи або пристрої у відповідь на зовнішній запит з метою встановлення відповідності вимогам, що висуваються до даної системи.

Для дослідження часу відгуку системи на високих або пікових навантаженнях проводиться стрес-тестування, при якому створюється на систему навантаження перевищує нормальні сценарії її використання. Не існує чіткої межі між навантажувальним і стрес-тестуванням, однак ці поняття не варто змішувати, так як ці види тестування відповідають на різні бізнес-питання і використовують різну методологію.

Проведемо навантажене тестування для програмного продукту, зробимо кілька сесій POST-запитів на сайт з різними параметрами часу та потоків (рис. 3.1-3.4).

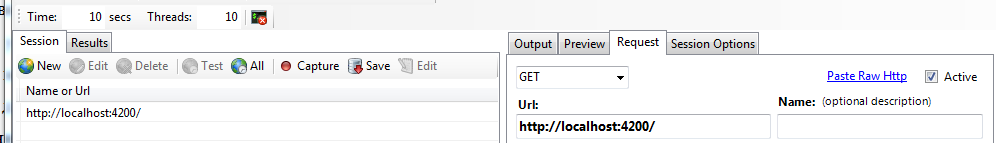


Рисунок 3.1 – Налаштування для тестування.

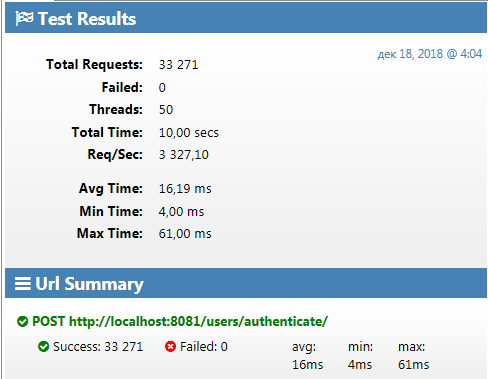


Рисунок 3.2 – Перший тест, 10 секунд, 50 потоків.

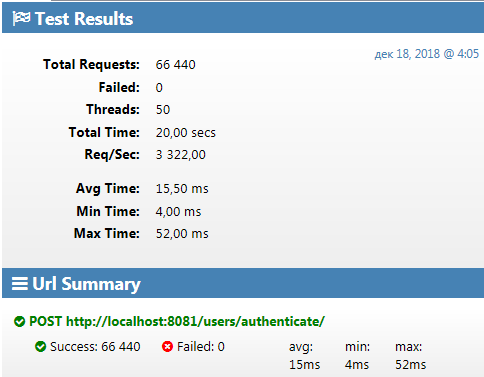


Рисунок 3.3 – Другий тест, 20 секунд, 50 потоків.



Рисунок 3.4 – Третій тест, 30 секунд, 50 потоків.

Як можемо побачити за результатами тестування сервер проходить навантажене тестування на відмінно. З трьох тестів не один не вивів сервер з ладу і також втратив жодного запиту.

**3.2 Керівництво користувача**

Розглянемо усю функціональність сервісу для адміністратору:

На наступному скріншоті ми бачимо головну сторінку магазину. (рис 3.5).



Рисунок 3.5 – Головна сторінка.

Форма реєстрації для створення особистого кабінета щоб увійти в систему (рис. 3.6).

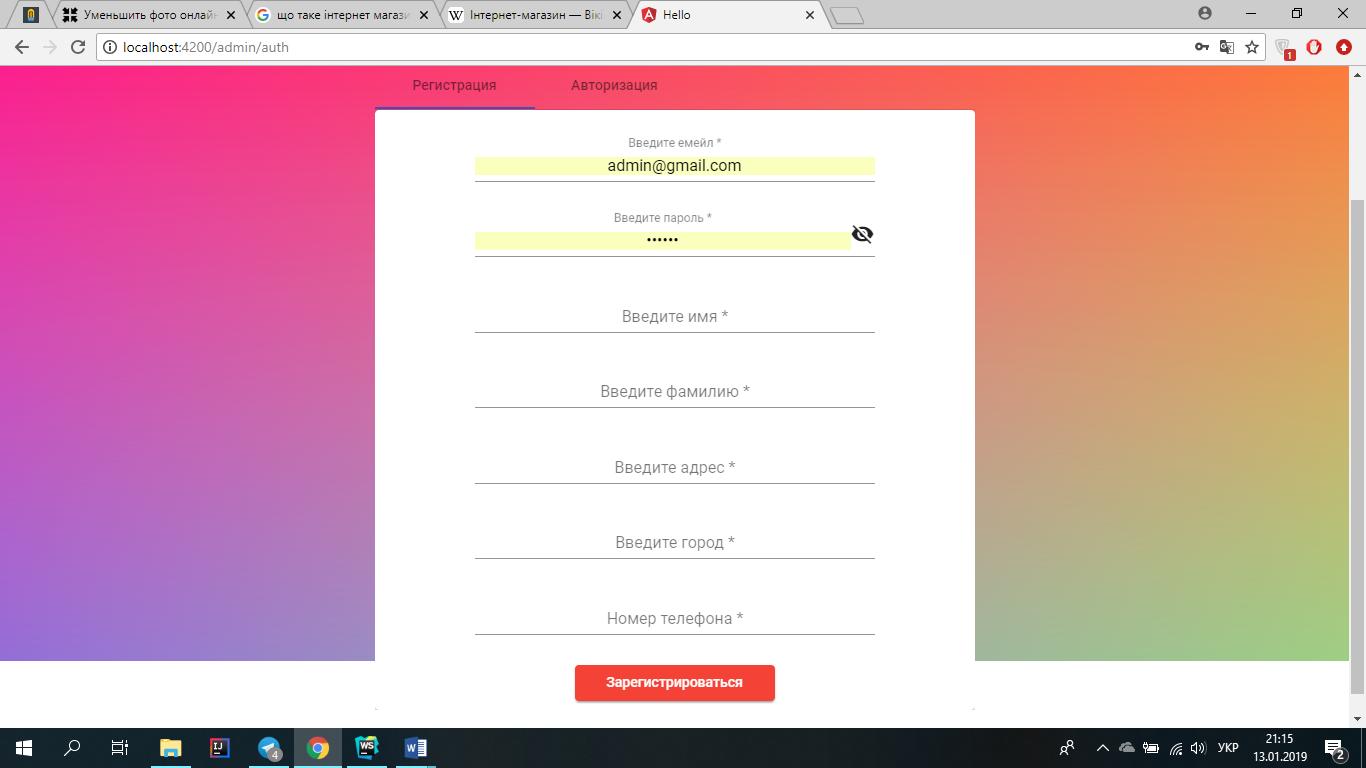


Рисунок 3.6 – Форма реєстрації.

Форма входу до системи (рис 3.7).

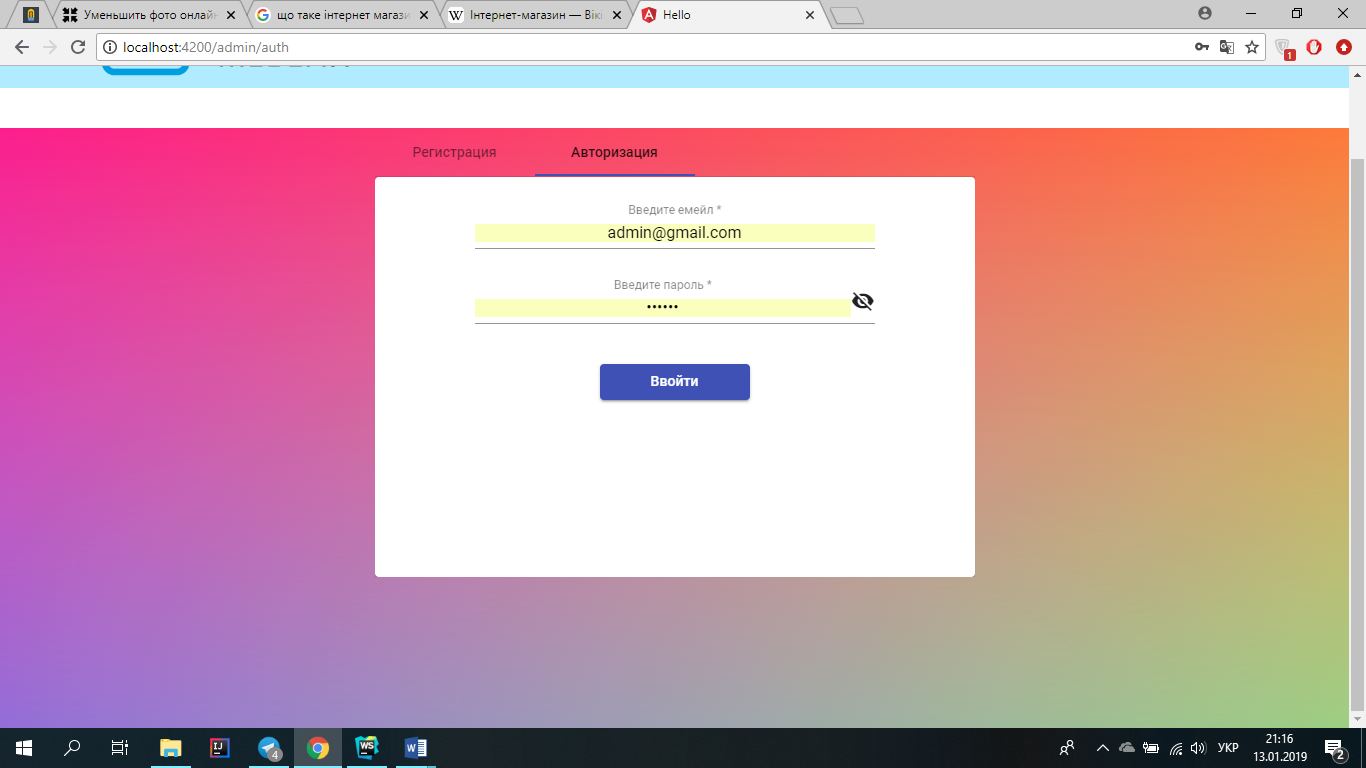


Рисунок 3.7 – Форма входу.

Адміністраторська панель (рис 3.8), на якій можна переглядати усіх користувачів (рис 3.9), редагувати дані про користувачів (рис. 3.10), замовлення за обраними датами і підтверджувати/відхиляти замовлення (рис. 3.11).

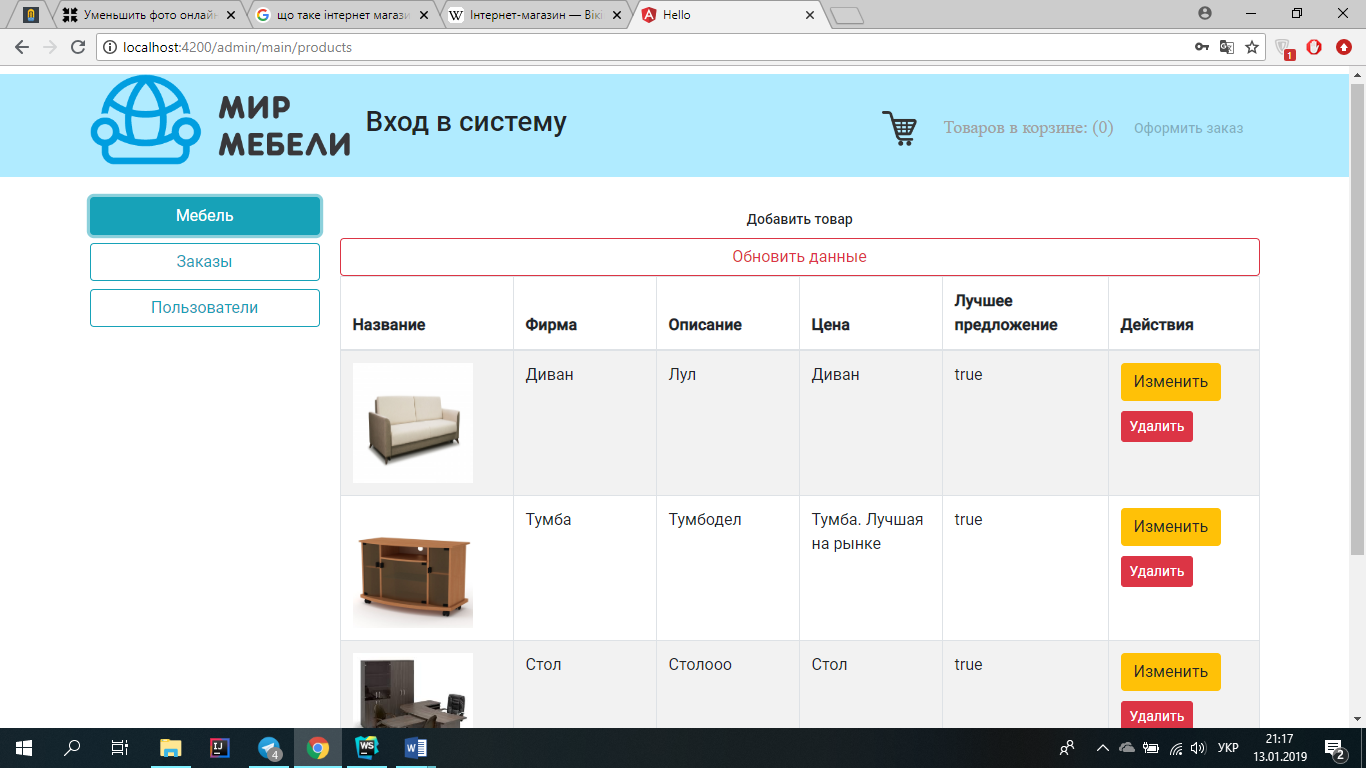


Рисунок 3.8 – Адміністраторська панель.

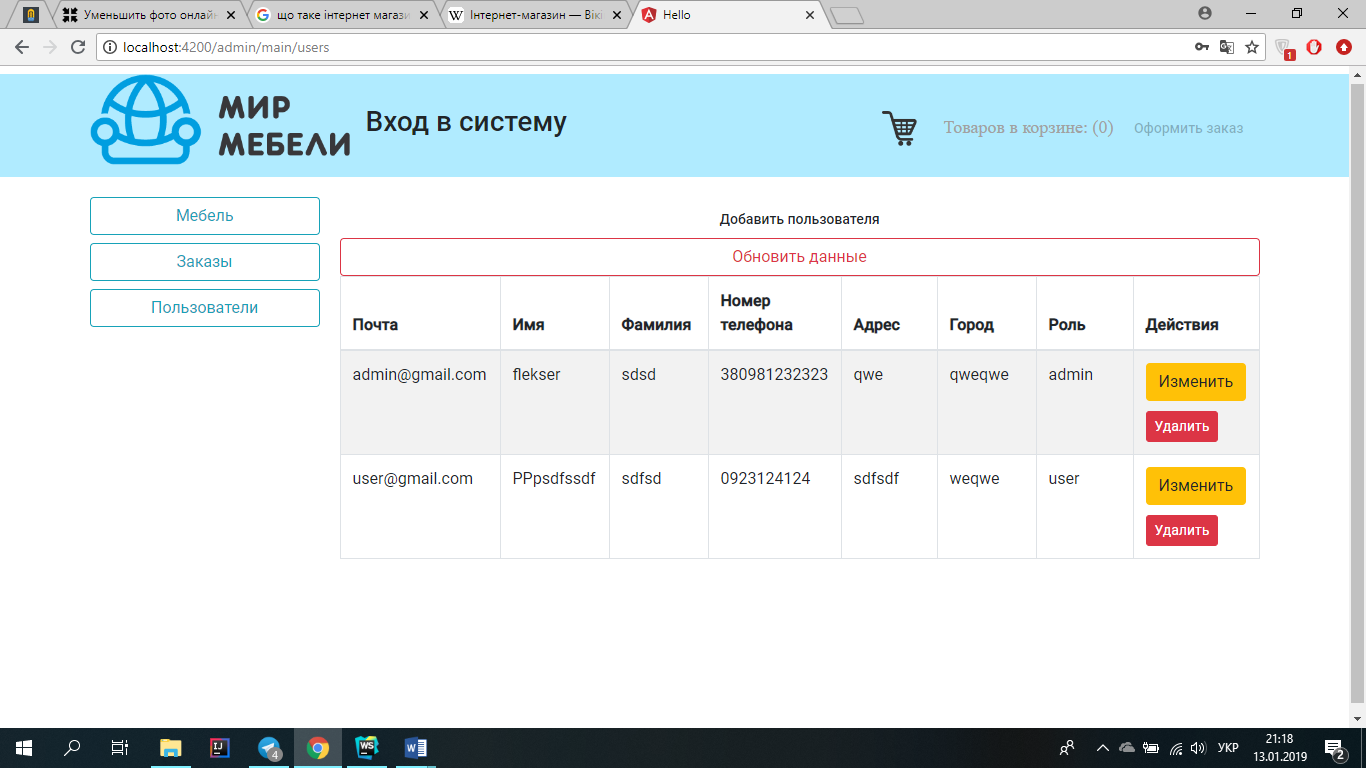


Рисунок 3.9 – Усі користувачі.

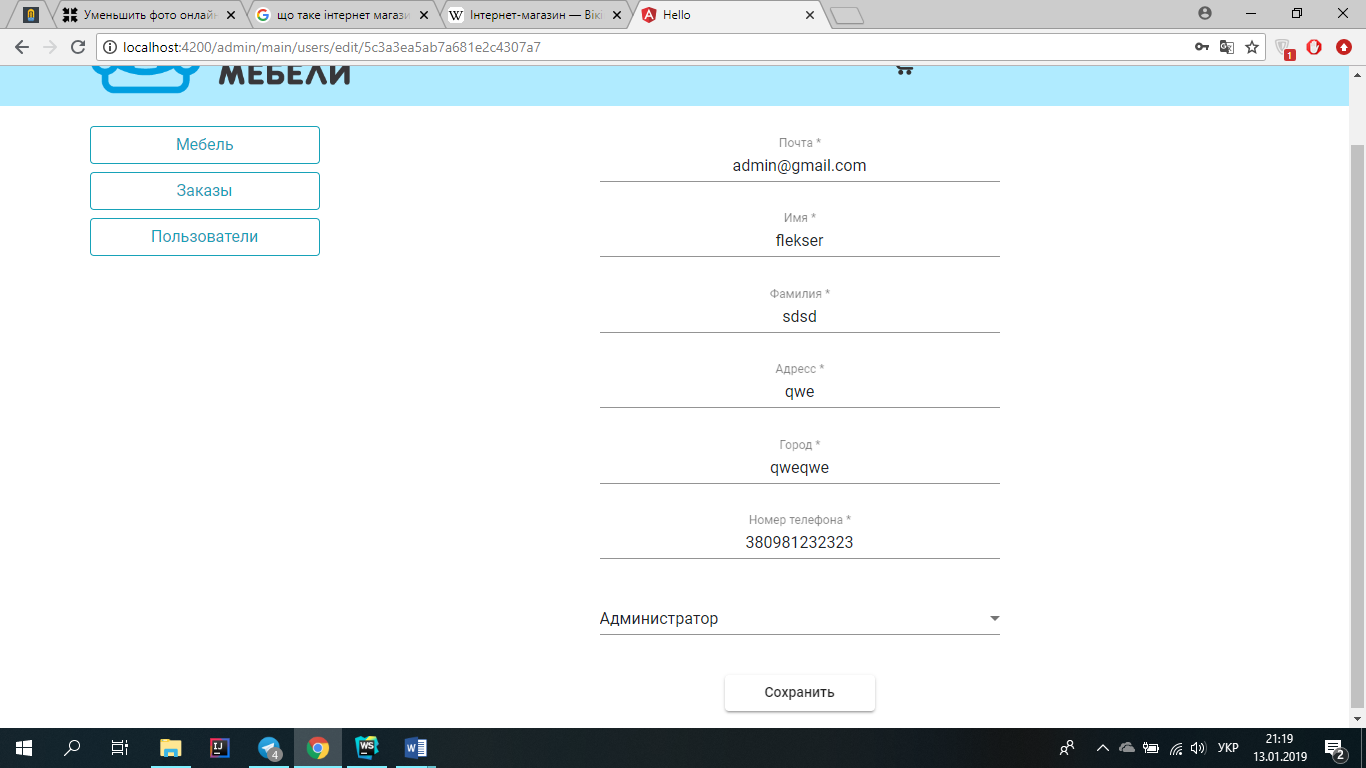


Рисунок 3.10 – Редагування користувача.

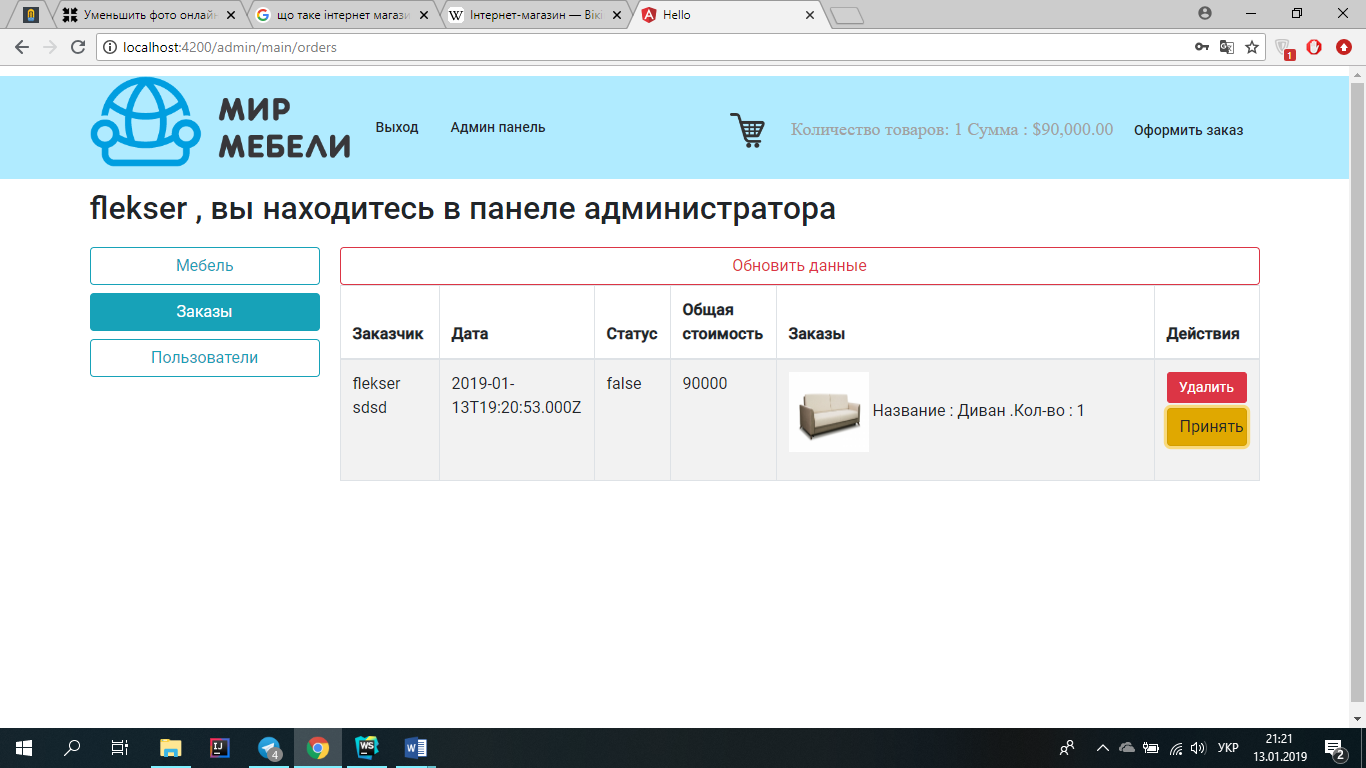


Рисунок 3.11 – Приймання замовлення.

**3.3 Використання веб-застосунку**

Розглянемо усю функціональність сервісу для адміністратору:

На наступному скріншоті ми бачимо головну сторінку сервісу (рис 3.5).



Рисунок 3.12 – Головна сторінка.

Форма реєстрації для створення особистого кабінета щоб увійти в систему (рис. 3.13).

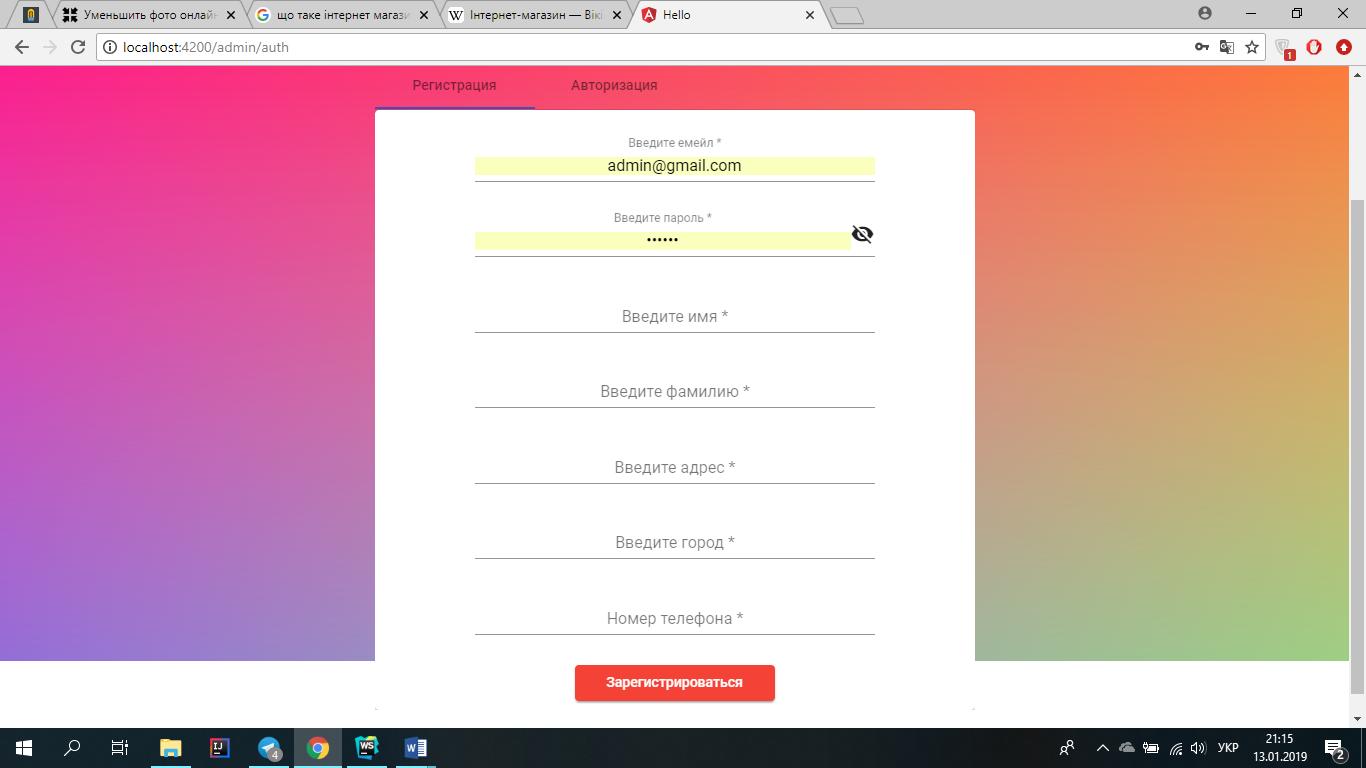


Рисунок 3.13 – Форма реєстрації.

Форма входу до системи (рис 3.14).

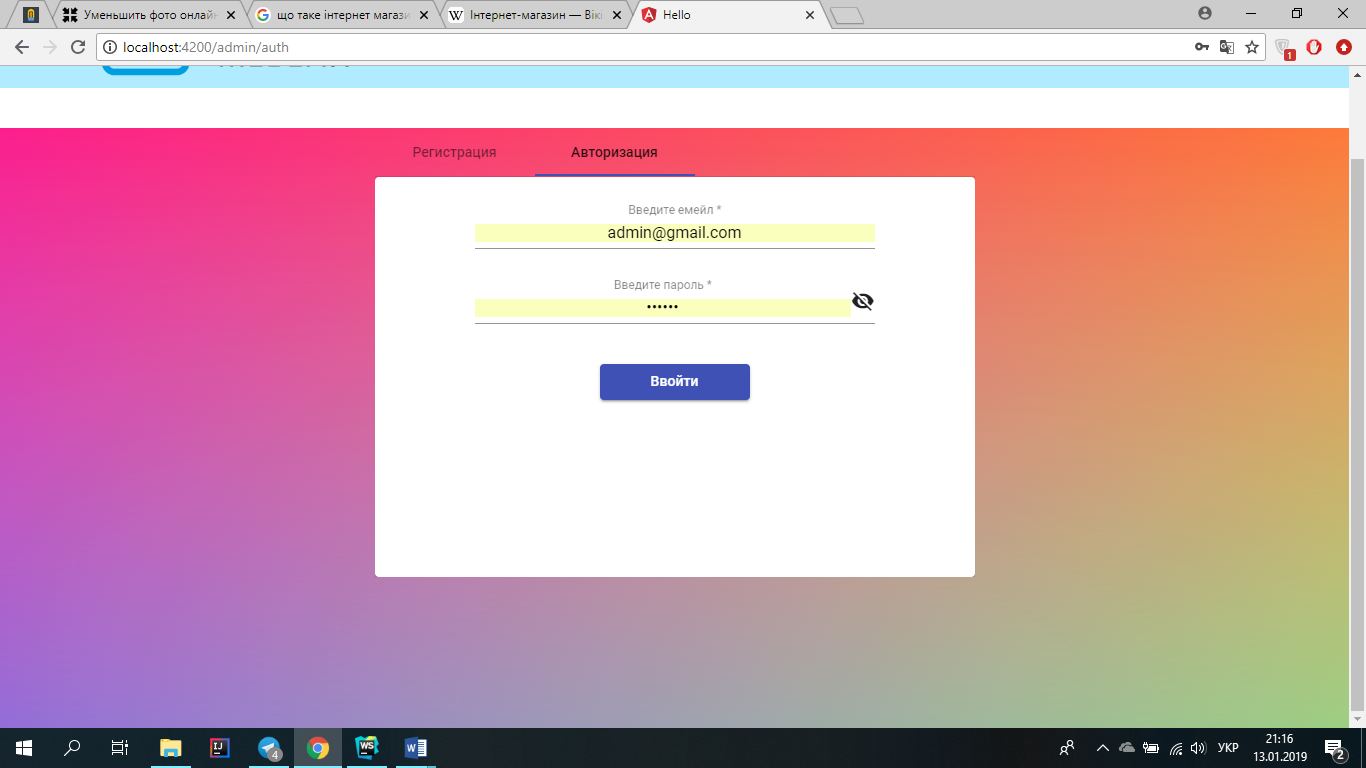


Рисунок 3.14 – Форма входу.

Профіль, де можна переглянути усі свої замовлення (рис. 3.15).



Рисунок 3.15 – Профіль.

**ВИСНОВКИ**

В ході виконання курсової роботи було виконано наступні етапи.

1. Було проаналізовано предметну область індивідуального завдання курсової роботи.
2. Розроблено модель даних у виді схем у документо-орієнтованій базі даних.
3. Проведено первинну розробку серверної архітектури додатку.
4. Проведено проектування архітектури інформаційної системи з використанням технології «клієнт-сервер».
5. Проведено програмну реалізацію та налагодження розробленої інформаційної системи.

Отже, в ході виконання курсової роботи, послідовно застосовуючи принципи об'єктно-орієнтованого аналізу, документо-орієнтованого моделювання даних, а також використовуючи програмний засіб СУБД MongoDB, фреймворк Angular і середу розробки IntelliJ IDEA Ultimate для програмування веб-додатку на серверній платформі NodeJS в операційній системі MS Windows 7, було розроблено, реалізовано і задокументовано інформаційну систему для інтернет-магазина з продажу меблів.

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Розробка сайтів https://inneti.com.ua - 11.01.2019
2. Веб-сайт з продажу меблів www.pufetto.com.ua/ – 11.01.2019
3. Веб-сайт з продажу меблів https://sofino.ua – 11.01.2019
4. Веб-сайт з продажу меблів mebli.ars.ua – 11.01.2019
5. Веб-сайт з продажу меблів <http://dls.ua> – 11.01.2019
6. Лоусон Б. Изучаем HTML 5. – К.: BHV, 2011. – 384 с.
7. Макфарланд Д Большая книга CSS3. – «Дейли Сенз», 2000. – С.256
8. Веб-сайт робочої платформи [https://nodejs.org/uk –11.01.2019](https://nodejs.org/uk%20–%20Введ.%2011.01.2019)
9. Дилеман П. Изучаем Angular 2. – «Sun Book», 2014. – С.198
10. Веб сайт з тестування <https://qalight.com.ua> – 11.01.2019