

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ВОЙТЕНКО БОГДАН ОЛЕКСАНДРОВИЧ

Допускається до захисту:

В.о. завідувача кафедри
інформаційних технологій,
канд. техн. наук, доцент

Оксана ЗЕЛІНСЬКА

«_____» _____ 2024р.

**ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ПІДТРИМКИ ДІЯЛЬНОСТІ
МАГАЗИНУ ПОБУТОВИХ МАТЕРІАЛІВ**

Спеціальність 122 Комп'ютерні науки
Кваліфікаційна (магістерська) робота

Науковий керівник:

Павло РИМАР, старший викладач
кафедри інформаційних технологій

Науковий консультант:

Роман БАБАКОВ, д-р техн. наук, доцент,
професор кафедри інформаційних технологій

Оцінка: _____ / _____ / _____
(бали/ за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця 2024

АНОТАЦІЯ

Войтенко Б.О. Інформаційна система для підтримки діяльності магазину побутових матеріалів. Спеціальність 122 Комп'ютерні науки, освітня програма «Комп'ютерні технології обробки даних». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

Кваліфікаційна (магістерська) робота присвячена розробці інформаційної системи для підтримки діяльності магазину побутових матеріалів. В роботі досліджено актуальність та важливість використання інформаційних систем в бізнесі. Розроблено інформаційну систему, яка дозволяє вирішувати основні задачі роботи магазину побутових матеріалів.

52 стор., 10 рис., 28 джерел.

ABSTRACT

Voitenko B.O. Information system to Support the Activities of the Store of Household Materials. Specialty 122 "Computer science", educational program «Data Science». Vasyl' Stus Donetsk National University, Vinnytsia, 2024.

The qualification (master's) thesis is devoted to the development of an information system to support the activities of a household goods store. The paper investigates the relevance and importance of using information systems in business. An information system has been developed that allows solving the main tasks of a household goods store.

52 pages, 10 figures, 28 sources.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ПОНЯТТЯ ІНФОРМАЦІЙНИХ СИСТЕМ.....	7
1.1 Інформаційні системи: характеристика, етапи розвитку	7
1.2 Принципи створення і функціонування інформаційних систем	11
1.3 Проєктування інформаційних систем.....	13
Висновок до першого розділу	21
РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ. ОГЛЯД АНАЛОГІВ ТА ІНСТРУМЕНТІВ ДЛЯ РЕАЛІЗАЦІЇ.....	22
2.1 Постановка задачі дослідження.....	22
2.2 IBS Магазин Універсал.....	22
2.3 DNTrade.....	26
2.4 POS Vector	30
2.5 Використані інструменти та технології.....	32
Висновок до другого розділу	40
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	41
3.1 Структура бази даних	41
3.2 Програмна реалізація.....	41
Висновок до третього розділу	48
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ	50

ВСТУП

Торгівля побутовими товарами займає велику нішу в сучасному торгівельному світі товарів. Це в свою чергу вимагає вдосконалення методів управління за діяльністю магазинів. Його ефективне функціонування багато в чому залежить від того, наскільки чітко організовані робочі процеси, оперативного доступу до інформації та своєчасного прийняття управлінських рішень. Розробка інформаційної системи для підтримки діяльності магазину побутових матеріалів та налагодження роботи його персоналу є ключовим фактором для підвищення його конкурентоспроможності та розвитку.

Актуальність даного дослідження полягає в провадженні інформаційних технологій в магазинах побутових матеріалів. Конкуренція між магазинами кожного року тільки збільшується, підвищуються вимоги до розроблюваних програмних рішень для обліку та управління своєю діяльністю. Сучасні інформаційні технології дозволяють створювати необхідні рішення, що будуть включати в собі систему обліку товарів, облік фінансових операцій тощо.

1. Автоматизація бізнес-процесів.

Ефективне управління запасами. Система дозволяє контролювати рівень запасів, оптимізувати замовлення товарів та уникати ситуацій з дефіцитом або надлишком продукції.

Прискорення роботи. Використання інформаційної системи зменшує час на виконання рутинних операцій, таких як оформлення продажів, облік товарів чи складання звітності.

2. Покращення обслуговування клієнтів.

Швидкість і точність обробки замовлень. Система дозволяє швидко знайти потрібний товар, обробити покупку чи повернення.

Лояльність клієнтів. Впровадження програм лояльності, персоналізованих пропозицій та знижок через інформаційну систему сприяє підвищенню задоволеності клієнтів.

3. Аналітика та прийняття рішень.

Дані для аналізу. Інформаційна система збирає статистику щодо продажів, попиту на товари та поведінки клієнтів, що дає змогу приймати обґрунтовані управлінські рішення.

Прогнозування. Завдяки аналізу даних можна прогнозувати тренди та планувати розвиток магазину.

4. Конкурентоспроможність.

Цифровізація ринку. Конкуренція змушує магазини впроваджувати сучасні технології, щоб не відставати від конкурентів.

Інтеграція з онлайн-каналами. Інформаційна система може підтримувати інтеграцію з інтернет-магазинами чи платформами доставки, що розширює можливості продажів.

5. Економія ресурсів.

Зменшення витрат. Система допомагає зменшити витрати на облік, управління та підтримку операційної діяльності.

Мінімізація людського фактора. Автоматизація знижує ризики помилок, що виникають через людський фактор.

6. Юридичні та податкові вимоги.

Звітування. Інформаційна система спрощує створення та зберігання податкової звітності.

Відповідність регуляціям. Використання сучасного програмного забезпечення забезпечує дотримання законодавчих норм, наприклад, щодо фіскалізації чи обліку.

Розробка та впровадження такої системи дозволяє магазину побутових товарів бути ефективнішим, адаптуватися до змін у ринкових умовах і відповідати сучасним очікуванням клієнтів.

Метою даного дослідження є підвищення ефективності управління бізнес-процесами магазину, забезпечення якісного обслуговування клієнтів та оптимізація використання ресурсів за допомогою автоматизації основних операцій, аналітики та інтеграції сучасних цифрових технологій.

Об'єктом дослідження є інформаційна система магазину. Предметом дослідження є процес розробки інформаційної системи підтримки діяльності магазину побутових матеріалів.

Задачі дослідження:

1. Вивчення структури та особливостей бізнес-процесів магазину.
2. Аналіз вимог до інформаційної системи з боку управління, персоналу та клієнтів.
3. Визначення функціональних вимог до системи (облік, звітність, аналітика, клієнтський сервіс).
4. Проектування архітектури системи (база даних, програмні модулі, інтерфейси).
5. Розробка логіки бізнес-процесів у системі (прийом товарів, продаж, інвентаризація, звітність).
6. Створення схеми взаємодії між модулями системи.
7. Розробка програмного забезпечення:

Виконання цих завдань дозволить створити ефективну інформаційну систему, яка повністю відповідатиме потребам магазину побутових матеріалів, покращить управління бізнесом і підвищить рівень обслуговування клієнтів.

Робота складається зі вступу, трьох розділів, 10 рисунків, список літератури містить 22 найменування. Загальний об'єм роботи 52 сторінки.

РОЗДІЛ 1

ПОНЯТТЯ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Інформаційні системи: характеристика, етапи розвитку

Інформаційна система (ІС) – це набір (сукупність) елементів для збереження та обробки інформації. Кожна інформаційна система включає в себе користувачів, необхідне програмне забезпечення та технічні засоби для обробки інформації[1]. Інформаційна система містить наступні складові:

- структуру (з яких елементів вона складається);
- функції (як взаємодіють кожен елементи системи);
- входи та виходи для кожного елементу системи та самої системи загалом;
- мету та обмеження всієї системи, а також її окремих складових.

Інформаційні системи можуть не тільки показувати процес функціонування об'єктів, а також впливати на них через управління. ІС включає в себе сукупність інформаційних процесів для забезпечення необхідною інформацією під час прийняття рішень на кожному рівні. Мета її полягає у формуванні інформації, яка буде використовуватися під час прийняття управлінських рішень. У зв'язку з цим ІС забезпечує збереження, оброблення та узагальнення інформації за принципом «знизу доверху», а також її детальну конкретизацію за принципом «зверху до низу».

Мета ІС – це описати об'єкт з економічної точки зору, його стани та взаємодію через економічні показники. ІС призначена для своєчасної подачі необхідної та достатньої інформації для процесу прийняття рішень. Прийняття рішень на якісному рівні дозволить забезпечити ефективний процес роботи об'єкта управління та всіх підрозділів, які входять до його складу.

Основні завдання:

- виявити джерела необхідної інформації;
- зібрати, зареєструвати, обробити та надати інформацію, що характеризує стан на виробництві та його управлінні;

- розподілити інформацію між керівництвом, підрозділами та виконавцями у відповідності до їх участі в управлінні.

До ключових елементів кожної організації можна віднести структуру, робочі органи, працівників, культуру та політику. Кожна ІС також є важливим інструментом для здійснення управлінських функцій. Інформаційні системи дуже затребувані у бізнесі. Наприклад, у банківській сфері неможливо працювати без використання інформаційної системи, бо від неї залежить весь процес функціонування банків.

З розвитком комп'ютерних технологій нерозривно пов'язаний також і розвиток інформаційних систем. За їх допомоги можна автоматизувати процес розв'язку економічних задач. Для розв'язку довільної економічної задачі з використанням комп'ютера спочатку необхідно підготувати потрібні дані (інформаційне забезпечення) та створити математичну модель для розв'язку задачі. На підставі цієї моделі складається програма для розрахунку. Вхідна інформація, яка безпосередньо необхідна для розв'язку, може надходити в систему через канали вводу інформації. Особливістю системи є те, що вона забезпечує користувачів інформацією з різних джерел.

Системи, які використовуються для обробки інформації, містять головні компоненти: дані та обчислення. Переважна більшість ІС є інформаційними ресурсами в організаціях та містять в собі велику кількість компонентів як от функціональні вимоги, запити, звіти та інше. Всі вони також містять великі описи свого власного змісту в тій чи іншій формі. Вони необхідні для інтерпретації результатів, а також для коректного використання наданої інформації. Якщо система не містить повного опису, тоді розуміється, що користувачі можуть отримати інформацію з інших джерел.

Для основних компонентів інформації (дані та їх обчислення) важливе значення має така характеристика як їх надмірність. Вона суттєво залежить від одиниці інформації. Коли одиниця обрана, тоді надмірністю є процес дублювання однієї і тої самої одиниці в системі. Важливим у виборі одиниці інформації є її розмір. Якщо обрати занадто малу одиницю, то це призведе до

високого рівня незалежності блоків з інформацією. Проте в той самий час це також призведе до збільшення витрат на їх підтримку. Якщо обрати велику одиницю, тоді неможливо виключити численні дублювання підблоків з інформацією.

За час, який пройшов після виникнення та розвитку ІС організаційного типу, структура, надмірність даних та їх обчислень постійно та кардинально змінювалися, що в свою чергу впливало на визначення покоління даних систем.

Інформаційні системи першого покоління, які відомі під назвою Data Processing System – DPS («Системи обробки даних», синоніми – «Електронна обробка даних», «Системи електронної обробки даних», «Автоматизовані системи управління (АСУ)) – використовували позадачний підхід»: для кожної задачі окремо було сформовано набір даних та спроектована математична модель. Такий підхід обумовлював інформаційну надмірність даних (один і той самий набір даних міг бути використаний для розв'язку різних задач) та математичну надмірність (моделі розв'язування різних типів задач мали спільні блоки). До типових прикладів систем обробки даних можна віднести системи керування запасами, виписування рахунків, нарахування зарплати тощо.

Системи обробки даних були вузько прикладними (використовувалися для певної задачі) та орієнтованими на автоматизацію робіт з паперами за рахунок використання комп'ютерів для обрахунку великих масивів і потоків даних на операційному рівні. Ознакою, за якою можна розпізнати ці системи, є ефективна обробка запитів, використання інтегрованих файлів для пов'язування між собою задач і генерування зведених звітів для керівництва. Оскільки кожна система призначалася для конкретного застосування, опис її функцій (зазвичай у вигляді друкованих інструкцій за процедурою або у вигляді стандарту) було представлено по мінімуму, призначеному для фахівців в даній предметній області. Крім того, передбачалося, що користувачі мають достатній досвід як в прикладній області, так і в системі, що надає відповідну програму.

Друге покоління інформаційних систем відоме під назвою Management Information System – MIS («управлінські (адміністративні) інформаційні

системи» або «інформаційні системи в менеджменті»)[2]. Також відомий термін «АСУ – концепція баз даних». Такі системи призначені для забезпечення керівництва інформацією. Інформаційна система управлінського типу характеризується структурованим потоком інформації, інтеграцією задач обробки даних, генерування запитів та звітів.

В інформаційних системах управлінського типу було визнано перевагу колективного використання даних. Також було помічено, що на одному підприємстві багато прикладних програм використовують один і той самий набір даних та відбувається процес дублювання роботи під час збирання, оброблення, зберігання та пошуку цих даних. У зв'язку зі збільшенням кількості прикладного програмного забезпечення, яке обслуговує всі рівні управління та оброблюють одну й ту саму інформацію, збільшувався обсяг дублювання, що в свою чергу гальмувало процес використання комп'ютерів в управлінні. Це дублювання інформації дуже часто було неефективним, бо призводило до несумісності прикладного програмного забезпечення. Для цього було розроблено концепцію створення єдиної централізованої бази даних, яка використовувала спеціальний програмний продукт – СКБД для обслуговування всього прикладного програмного забезпечення організацій.

Основною проблемою при створенні великої розподіленої бази даних є складність опису даних, метою якого є об'єктивне спрощення колективного використання даних різними додатками, незалежно від окремих прикладних програм. Для опису даних широко використовуються моделі даних і словники. Семантика даних, тобто вивчення їх змісту незалежно від окремих прикладних програм, стала самостійною галуззю досліджень.

До інформаційних систем третього покоління відносяться системи підтримки прийняття рішень – СППР (Decision Support Systems – DSS). СППР – інтерактивна комп'ютерна система, яка призначена для підтримки різних видів діяльності в разі прийняття рішень зі слабоструктурованих або неструктурованих проблем. Інтерес до СППР як до перспективної галузі комп'ютерних технологій та інструментом для підвищення ефективності праці в

сфері управління економікою постійно зростає. У багатьох країнах розробка та впровадження СППР стала швидко розвиненою сферою бізнесу.

СППР має не тільки загальну інформаційну підтримку, але й загальну математичну підтримку, засновану на моделях, тобто ідея розподілу обчислень реалізована так само, як і розподіл даних став вирішальним фактором у традиційних інформаційних системах.

Усвідомлення важливості розподілу обчислень за допомогою автоматичних обчислень сталося, коли багато додатків використовували подібні обчислення і зрозуміли, що в додаток були введені індивідуальні фактори, які допомагають певним користувачам внести невеликі зміни. Крім того, дії та процедури були значно дубльовані під час розробки, впровадження та тестування цих обчислювальних функцій.

1.2 Принципи створення і функціонування інформаційних систем

Під час створення інформаційних систем використовуються загальні вимоги, правила чи норми, яких слід дотримуватись.

Так, згідно з нормативними документами під час створення автоматизованих систем (АС) необхідно керуватися принципами системності, розвитку, сумісності, стандартизації та ефективності.

Принцип системності. Між конструктивними елементами системи необхідно встановити такі зв'язки, які забезпечують сумісність і взаємодію з іншими системами. Отже, всі ланки, елементи, функції та питання управління й діяльності повинні розглядатися як єдине ціле.

Принцип розвитку (відкритості). Автоматизована система повинна бути створена з урахуванням можливості оновлення та оновлення функцій і конфігурації АС без порушення їх функціональності.

Принцип сумісності. При створенні системи необхідно реалізувати інформаційний інтерфейс, оскільки ця система зможе взаємодіяти з іншими системами відповідно до встановлених правил.

Принцип стандартизації. При створенні системи необхідно враховувати стандарти, уніфіковані стандартизовані елементи, проектні рішення, пакети прикладного програмного забезпечення і т.д. Застосовувати його потрібно раціонально. Система та її елементи повинні бути стандартизовані, щоб можна було мінімізувати будь-які витрати, а також методи та інструкції, якими керується персонал.

Принцип економічної ефективності. Досягнення розумного співвідношення між витратами на створення автоматизованої системи управління і цільовим ефектом, включаючи кінцевий результат автоматизації, не обов'язково повинно мати грошову форму, це економія часу, певних зручностей, нових функцій і т.д.

Окрім розглянутих основних можна додатково визначити ще деякі принципи створення й функціонування автоматизованих ІС.

Принцип безпеки даних. Інформація повинна бути захищена як безпосередньо під час обробки і зберігання в системі, так і під час обміну між комп'ютерами. Необхідно виключити можливість несанкціонованого доступу до даних в системі. Всі транзакції в системі повинні реєструватися. Необхідно виявляти порушення системи безпеки.

Принцип надійності системи. І апаратне, і програмне забезпечення повинні бути однаково надійними. Інформація для клієнта повинна бути точною, доступною і надаватися йому без зволікання. У разі збою системи вам необхідно відновити дані і усунути пошкодження.

Принцип продуктивності системи. Необхідність дотримання цього обумовлена значною неоднорідністю потоку інформації, що підлягає обробці через певні умови, а також жорсткими вимогами до термінів її обробки.

Принцип пристосування (адаптації). Існуюча інформаційна система повинна мати право на зміну та розширення. Крім того, система може бути повністю перероблена, але інформація повинна бути збережена.

1.3 Проектування інформаційних систем

Метою автоматизації при розробці автоматизованих ІС є бізнес-процеси. А початок робіт з проектування інформаційних систем найчастіше супроводжується обов'язковим реінжинірингом бізнес-процесів. Отже, спочатку розглянемо концепцію бізнес-процесів та реінжинірингу.

Термін бізнес-процес (БП) набув популярності в останні роки серед керівників великих і середніх компаній. Бізнес-процес – це серія регулярно повторюваних дій, спрямованих на досягнення певних результатів, які є цінними для організації.

Бізнес-процес часто визначається як шлях, по якому важливий об'єкт або інформація потрапляє в компанію (наприклад, шлях, по якому замовлення клієнта надходить до корпоративного відділу, або література, яка в цьому відношенні прийняла визначення бізнес-процесу як набору завдань і процедур.

Наприклад, типовий бізнес-процес «контролю витрат» включає такі операції, як прийом замовлень, підписання контрактів на виробництво продукції, регулювання нормативних витрат на етапі технічної підготовки виробництва, аналіз собівартості і рентабельності певних видів продукції. У реалізації цього бізнес-процесу беруть участь відділ продажів, технічна підготовка виробництва, бухгалтерія та виробничий відділ.

Найбільш важливими характеристиками бізнес-процесів є:

входи і виходи бізнес-процесу – тобто необхідні для виконання процесу ресурси (матеріальні, інформаційні, трудові) і результати процесу;

власник процесу – посадова особа, що несе відповідальність за отримання результату процесу і має повноваження для розпорядження ресурсами, необхідними для виконання процесу;

початок, закінчення і тривалість;

позиціонування – процес протікає у взаємозв'язку з іншими процесами і (або) зовнішнім середовищем; досягнення результату можливе тільки за умови «синхронізації» цієї взаємодії;

показники ефективності – наприклад, час виконання процесу. Виділяють бізнес-процеси таких типів:

міжфункціональні (наскрізні), які проходять через все підприємство і перетинають межі функціональних (структурних) підрозділів;

функціональні, які виконуються в межах одного структурного підрозділу;

бізнес-функції – елемент бізнес-процесу, який виконується однією людиною для досягнення ділового результату.

Існує ряд програмних продуктів для бізнес-моделювання (тобто побудови моделей БП): ARIS; Microsoft Visio; Allfusion Process Modeler (паніше – BPWin); Інтальов Навігатор; Business Studio та ін[3].

Програмні продукти цього класу відносяться до CASE-засобів та дозволяють не тільки будувати візуальні моделі бізнес-процесів, але і проводити аналіз, і оптимізацію бізнес-процесів, тобто готувати БП до автоматизації.

Моделі бізнес-процесів підприємства є основою процесного управління і використовуються для розробки проектів автоматизації управління.

При цьому будують два типи моделей:

тип моделі «Як є» (As-Is) – це модель, що відображає поточний стан бізнесу (бізнес-процесу) аналізуючи цю модель можна планувати і реалізовувати майбутні зміни;

тип моделі «Як буде» (To-Be) – це модель, що описує ситуацію після автоматизації. Адже автоматизація приводить до видалення «паперових» технологій, виключення деяких бізнес-процесів з практики підприємства і появи нових комп'ютеризованих бізнес-процесів, тобто до істотних змін процесів.

Таким чином, нічого принципово нового термін «бізнес-процес» не несе. Своєю великою популярністю він зобов'язаний закладеному в ньому підходу до управління – процесно-орієнтованому підходу.

Процесно-орієнтований підхід до управління організаціями передбачає, що організація представляється не у вигляді набору функцій (збут, виробництво, закупівлі, інвестиції, фінанси), а у вигляді набору БП (прийом і виконання

замовлень, розробка нових видів продукції, укладання договорів). Управління такою організацією будується на основі управління БП.

Річ у тім, що до початку 80-х років в управлінні організаціями використовувався так званий функціональний підхід[3]. Його суть полягає в управлінні функціями. Вся діяльність організації підрозділяється на однорідні і спеціалізовані функції, які зосереджені в окремих підрозділах організації (наприклад, функція планування виконується відділом економіки та планування, а функція бухгалтерського обліку – бухгалтерією). Розділи згруповані в ієрархічну структуру. Управління такими структурами фактично становить управління функціями. Такий підхід до управління має багато переваг: чіткість роботи співробітників в структурному відділі і простота горизонтального і вертикального управління.

Недоліком функціонального підходу є те, що вам потрібно керувати функціями та відділами, а не процесами. Таким чином, якщо у виконанні процесу бере участь кілька відділів, практично ніхто не несе відповідальності за цей процес. Це означає, що результат процесу не відрізняється високою якістю.

Зі зростанням конкуренції організаціям довелося шукати способи радикального підвищення ефективності своєї діяльності. Це призвело до появи нових підходів до управління, в яких процесний підхід є найбільш популярним.

Його суть полягає в наступному. Уся діяльність організації складається з двох видів дій:

- а) повторюваних, які слід здійснювати періодично або в разі настання певних подій; цей вид дій називається процесами;
- б) разових, унікальних за складом, таких, що не повторюються в такому вигляді надалі; цей вид діяльності називається проектами, програмами, заходами і т. ін.

Виходячи з цього, управління діяльністю компанії включає: управління процесами (Process Management) та управління проектами (Project Management).

Процесний підхід означає управління саме діяльністю на відміну від управління структурою, як у функціональному підході.

При переході від функціонально-орієнтованої до процесно-орієнтованої організації виконують реінжиніринг БП (BPR – Business Process Reengineering) – тобто повне переосмислення, перегляд, оптимізацію БП відповідно до змін зовнішнього і внутрішнього середовища, і/або цілей бізнесу. У процесі реінжинірингу бізнес-процеси аналізуються та спрощуються.

Реструктуризації бізнес-процесів передують впровадження АІС на підприємствах. Реінжиніринг може бути виконаний компанією в майбутньому без урахування того, яке програмне забезпечення буде використовуватися, або без переналаштування обраного програмного продукту.

У першому випадку в результаті бізнес отримує опис стандартів і процесів управління, які в ідеалі відповідають цьому бізнесу, але може виявитися неможливим вибрати необхідне програмне забезпечення для реалізації. В другому випадку бізнес-процес в процесі оптимізації бізнесу адаптується до функцій, реалізованих в конкретному програмному пакеті. При такому підході можливе неповне задоволення потреб бізнесу, гарантується повна програмна реалізація змодельованого процесу і гарантується успіх реалізованих бізнес-додатків.

Суть розвитку інформаційних систем з плином часу відбивається в такій категорії, як «життєвий цикл». Як і вироблений продукт, інформаційна система має свій життєвий цикл з моменту створення до кінця своєї діяльності. Інформаційні системи – це спеціалізовані продукти. Без нього не може існувати жодна організація. Ми можемо говорити про припинення функціонування інформаційних систем цього покоління, їх окремих підсистем і елементів, а також про впровадження ІС наступного покоління, але про припинення його існування.

Життєвий цикл ІС, як правило, закінчується не в результаті фізичного зносу інформаційних систем, а в результаті їх старіння. Старіння означає, що в цьому випадку ІС не відповідає вимогам користувача. У той же час можливі зміни інформаційної системи економічно не вигідні або неможливі, що тягне за

собою необхідність розробки нових інформаційних систем. Для інформаційних технологій цілком природно, що вони застаріли і замінюються новими.

При впровадженні в організацію нових інформаційних технологій необхідно враховувати ризик відставання від конкурентів в результаті неминучого старіння з плином часу, оскільки швидкість переходу інформаційних продуктів на нові версії дуже висока. Час випуску нової версії коливається від декількох місяців до 1 року. Якщо цьому фактору не приділяється достатньої уваги в процесі впровадження нових інформаційних технологій, можливо, до того часу, коли компанія завершить перехід на нові інформаційні технології, вона вже застаріє, і можуть знадобитися заходи щодо її модернізації.

Життєвий цикл інформаційної системи – це період створення і використання інформаційної системи, що охоплює її різні стани з моменту виникнення потреби в цій інформаційній системі до моменту її повного виведення з експлуатації.

Існує декілька загальновизнаних стандартів, що описують життєвий цикл програмного забезпечення (наприклад, стандарт ISO/IEC 15288).

Процес створення ІС включає ряд етапів (або стадій), що обмежені часовими рамками і закінчуються випуском конкретного продукту (моделей, програмних продуктів, документації і т. ін.). Склад етапів і стадій у різних стандартах відрізняється. Тому зазначимо типові етапи життєвого циклу ІС, що розроблюються на замовлення підприємств (тобто, продуктів на замовлення): визначення вимог до системи і їх аналіз; проектування; розробка; тестування; впровадження; функціонування; супроводження.

Розглянемо перелічені етапи життєвого циклу ІС.

Визначення вимог до системи і їх аналіз є першим етапом створення ІС, на якому вимоги замовника уточнюються, узгоджуються, формалізуються і документуються. Фактично на цьому етапі дається відповідь на питання: «Для чого призначена і що повинна робити інформаційна система?». Саме тут лежить ключ до успіху проекту.

Метою етапу є перетворення загальних, розпливчатих вимог замовника в точні визначення і специфікації для розробників, а також опис функціональності майбутньої системи.

На даному етапі виконується:

а) структурний аналіз підприємства – починається з дослідження того, як організована система управління підприємством, з обстеження функціональної й інформаційної структур системи управління, визначення існуючих і можливих споживачів інформації; за результатами обстеження аналітик буде функціональну модель «Як є» (As-Is);

б) аналіз моделі «Як є», виявлення її недоліків і вузьких місць, визначення шляхів вдосконалення бізнес-процесів, у цьому процесі обов'язково беруть участь представники компанії-замовника, а при необхідності і незалежні експерти;

в) побудова моделі «Як буде» (To-Be).

На цьому етапі визначаються і описуються: умови роботи майбутньої системи; функціональність системи; розподіл функцій між людиною і системою, інтерфейси; вимоги до технічних, інформаційних і програмних компонентів системи; вимоги до якості і безпеки; склад документації (технічної і призначеної для користувача), яка буде оформлюватися в процесі створення ІС; умови впровадження і експлуатації.

Основним документом, що відображає результати робіт першого етапу створення ІС, є технічне завдання на проект (розробку) або один з його аналогів (наприклад, SRS), що містить, окрім вище перелічених визначень і специфікацій, також відомості про черговість створення системи, відомості про ресурси, що виділяються, директивні терміни проведення окремих етапів роботи і так далі.

Проектування. Проектування починається з визначення мети проекту автоматизації та охоплює три основні області:

- проектування баз даних;
- проектування програм, екранних форм, форм звітів, які будуть формуватися в програмному комплексі;

- проектування конкретного середовища або технології, а саме: топології мережі, конфігурації апаратних засобів, використовуваної архітектури і т. д.

Результатом етапу є:

- діаграми потоків даних;
- функціональні блок-схеми програмного забезпечення; структури баз даних;
- мережні адреси, протоколи телекомунікацій і інші компоненти середовища обміну інформацією.

Розробка (програмування) – створення програмного забезпечення.

Тестування – перевірка відповідності системи вимогам, висуну- тим на етапі аналізу.

Командою розробників спільно з користувачами розробляється план тестування системи. Проведення тестування вимагає великих витрат часу: дані для тестування повинні бути ретельно підготовлені, результати перевірені, і на їх підставі до системи вносяться коректування. У деяких випадках, на підставі результатів тестування приймаються рішення про повторне проектування деяких частин ІС.

Впровадження – встановлення та введення системи в дію. Фактично, це процес переходу від використання старої системи до нової.

Можна виділити чотири основні типи переходу на використання нової системи:

- стратегія паралельного переходу – в організації одночасно функціонують і стара, і нова, до тих пір, поки кожен співробітник не переконається в тому, що нова система функціонує коректно;
- стратегія прямого переходу – у певний момент стара система повністю замінюється новою;
- стратегія пілотного переходу – доступ до нової системи надається обмеженій частині організації, наприклад окремому підрозділу; після того,

як пілотна версія впроваджена і працює коректно, вона встановлюється у всій організації, або одномоментно, або поетапно;

- пофазова стратегія – нова система вводиться поетапно за окремими функціями або за підрозділами організації.

У процесі переходу на нову систему готується «План переходу» у вигляді графіка робіт, а також за результатами експлуатації складається докладна документація, що описує роботу системи, як з технічної точки зору, так і з погляду кінцевих користувачів, для подальшого використання при навчанні і в щоденній роботі.

Функціонування (експлуатація) – процес експлуатації ІС відповідно до основних цілей і завдань ІС.

Супроводження – забезпечення процесу експлуатації системи на підприємстві замовника.

Етапи експлуатації та супроводження системи на підприємстві, що динамічно розвивається, є комплексним процесом, що включає: модернізацію програмно-апаратної частини, викликану фізичним і моральним старінням компонентів ІС; відслідковування змін у законодавстві та відображення їх у системі; доопрацювання системи відповідно до нових вимог користувачів; забезпечення захисту інформації і т. ін.

Тобто, на кожному етапі життєвого циклу ІС породжується певний набір технічних рішень і документів, при цьому для кожного етапу початковими є документи і рішення, прийняті на попередньому етапі.

Практично жоден серйозний проект зі створення ІС не здійснюється без використання CASE-засобів. CASE-засоби (Computer-Aided Software System Engineering) – програмні продукти, що використовуються для аналізу, проектування, розробки і супроводження складних програмних систем. Тобто, CASE – це інструментарій для системних аналітиків, розробників і програмістів, що замінює їм папір і олівець комп'ютером для автоматизації процесу проектування і розробки програмного забезпечення. Серед популярних CASE-

засобів можна назвати Rational Rose, Erwin, Allfusion Process Modeler (раніше – BPWin), ARIS.

Висновок до першого розділу

В даному розділі розглянуто поняття інформаційних систем. Наведено причини, які призвели до їх появи, технічні характеристики. Описано принципи створення та функціонування. Сформульовано поняття бізнес-процесу та подано етапи створення інформаційної системи. В наступному розділі буде сформульована постановка задачі, наведено аналоги та описано стек технологій для розробки інформаційної системи.

РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ. ОГЛЯД АНАЛОГІВ ТА ІНСТРУМЕНТІВ ДЛЯ РЕАЛІЗАЦІЇ

2.1 Постановка задачі дослідження

Необхідно розробити інформаційну систему, яка забезпечить автоматизацію ключових бізнес-процесів магазину побутових матеріалів, підвищить ефективність управління товарообігом, покращить обслуговування клієнтів і сприятиме прийняттю обґрунтованих управлінських рішень.

Інформаційна система магазину містить наступні характеристики:

- Асортимент продукції (побутові матеріали різних категорій).
- Типова кількість клієнтів і транзакцій у день.
- Місце розташування (один магазин чи мережа).
- Закупівля товарів у постачальників.
- Приймання, облік і зберігання товарів на складі.
- Продаж товарів через касу та обслуговування клієнтів.
- Формування звітів про продажі, залишки та прибутковість.
- Обробка повернень і обмінів.
- Звітність та аналітика:
- Генерація звітів про продажі, прибутковість та рівень запасів.
- Простота використання: інтуїтивно зрозумілий інтерфейс для персоналу магазину.
- Надійність: безперервна робота системи без втрати даних.

2.2 IBS Магазин Універсал

IBS Магазин Універсал – це програма для автоматизації магазину господарств, косметики, зоотоварів, квіткових салонів, універсальних і будівельних магазинів[4]. Програма призначена для обліку товарів в будь-яких непродовольчих магазинах. Вона включає в себе робочі місця продавця-касира,

оператора-товарознавця і керівника. Підтримується практично будь-яке торгове і фіскальне обладнання – програмні каси (РРО), сучасні фіскальні реєстратори, чекодрукуючі ваги, сканери штрих-коду та інше.



Рисунок 2.1 – Логотип програми IBS Магазин Універсал

Програма для обліку в магазині надасть ряд серйозних переваг для ефективного управління бізнесом і нові можливості для успішного ведення торгівлі.

Після встановлення програми і навчання Ви зможете:

1. Управляти закупівлями, автоматизувати переоблік товарів:

- вести довідник товарів, вести класифікацію товарів за групами і підгрупами, заводити штрих-коди для кожного товару;
- вести довідник постачальників з можливістю їх класифікації за групами;
- оформляти надходження товарів на склад або в торговий зал з реєстрацією закупівельної ціни для кожного товару;
- оформляти повернення товарів постачальнику;
- переміщати товари між місцями зберігання (склади, торговий зал);
- отримувати повну інформацію про залишки і рух товарів на будь-якому складі на будь-який момент часу, баланс по будь-якому товару;

- проводити інвентаризацію товарів у автоматизованому режимі з отриманням результатів у кількісному і сумовому вираженні.

2. Управляти ціновою політикою:

- вести будь-яку кількість прайс-листів з поділом їх на категорії (роздріб, великий опт, дрібний опт та ін.) і зберіганням історії змін цін;
- управляти націнками та цінами продажу в розрізі товарів і груп, встановлювати акційні ціни, знижки;
- друкувати різні види цінників, бирок.

3. Автоматизувати роздрібні продажі

- вести продаж товарів на автоматизованих робочих місцях касира з реєстрацією чеків в базі даних;
- використовувати для введення товарів в чек сканери штрих-коду;
- використовувати комбіновані форми оплати при закритті чека (готівковий, безготівковий розрахунок, бонуси);
- використовувати для друку чека фіскальні реєстратори або чекові принтери (для нефіскальних чеків);
- виконувати роздрібні повернення на робочому місці касира в автоматизованому режимі;
- оперативно отримувати інформацію по необхідному товару;
- працювати з декількома чеками одночасно (відкладені чеки).

4. Вести облік грошей і взаєморозрахунки з контрагентами:

- вести облік руху безготівкових коштів на розрахункових рахунках організації (надходження на розрахунковий рахунок, видатки з розрахункового рахунку, баланс по розрахунковому рахунку);
- вести інформацію про банківські реквізити кожного постачальника;
- вести облік руху готівкових коштів в касах організації (надходження в касу, видатки з каси, баланс по касі);
- контролювати стан взаєморозрахунків з постачальниками і покупцями.

5. Отримувати і аналізувати різні види звітності:

- вести облік і аналіз продажів в розрізі товарів, груп товарів, покупців, продавців і т.д.;
- отримувати інформацію про виручку, доходи, зміни товарних залишків та цін в режимі реального часу;
- формувати будь-які бізнес-звіти (з продажу, маркетингу, в цілях управління та ін.) довільної складності;
- оперативно отримувати консолідовані «звіти керівника» в розрізі необхідних аналітичних параметрів;
- друкувати стандартні друковані форми для будь-яких документів.

Автоматизація господарських, побутових та будівельних магазинів має свої специфічні особливості. Асортимент у таких магазинах змішаний та дуже різноманітний – від миючих засобів та косметики до сантехніки та великої побутової техніки. Формати магазинів також можуть бути різні – торгівля через прилавок, магазин самообслуговування. Ведення повноцінного обліку в магазині з таким широким асортиментом товарів без використання автоматизованої системи практично неможливо. Це обумовлено наступними моментами:

- великий асортимент товарів і обсяг продажів;
- як правило, декілька робочих місць касира;
- наявність вагового та штучного товару;
- кілька одиниць виміру (видів упаковок) у одного товару – штука, ящик і т.д.;
- кілька різних заводських штрих-кодів у однієї і тієї ж упаковки товару (один і той же товар може вироблятися на різних потужностях і мати різні штрих-коди виробника);
- облік залишків на кількох складах (місцях зберігання), переміщення між складами (склад зберігання, торговий зал і т.д.);
- необхідність контролю термінів зберігання, регулярні списання зіпсованих товарів (бою), проведення вибіркового або повних інвентаризацій;

- необхідність аналізу продажів в різних розрізах – по окремим товарним позиціям, товарним групам, підгрупам, касам;
- використання складних програм лояльності покупців – дисконтних і бонусних систем, маркетингових акцій;
- контроль взаєморозрахунків з великою кількістю постачальників.

Програмне забезпечення для будівельних та господарських магазинів має забезпечувати зберігання та швидку обробку великих обсягів інформації, мінімізувати час обслуговування покупців, мати вбудовані засоби контролю для виключення помилок персоналу та відповідати всім зазначеним вище вимогам для таких торгових об'єктів.

Залежно від реалізованих груп товарів та особливостей бізнес-процесів, вимоги до програмного забезпечення для автоматизації обліку можуть суттєво відрізнятися.

2.3 DNTrade

DNTrade – хмарна система обліку товарів. Торгівля, склад, виробництво. Все в одній системі: продажі, закупки, склад, фінанси, клієнти та постачальники[5]. Що система пропонує:

Складський облік: DNTrade підтримує всі види складських операцій та звітів. Ведення безлічі складів з власними товарами та групами. Контроль залишків, резервування, планування закупок.

Цінова політика: встановлення власних типів цін. Контроль зміни ціни та автоматична переоцінка. Роздрук цінників, власні шаблони етикеток. Запуск періодичних акцій. Надання ручної знижки при продажі.

Розрахунки з контрагентами: контроль взаєморозрахунків з клієнтами та постачальниками. Налаштування необхідних форм для друку. Мультивалютність. Сегментація бази клієнтів, історія продаж, програми лояльності.

Обладнання: підключення сканерів, принтерів чеків та цінників, банківських терміналів, ваг з прямим підключенням та з друком етикетки, моніторів клієнта. Робота з терміналами збору даних.

Маркетплейси: виграшка товарів та цін. Отримання та обробка замовлень. Зміна статусів, смс розсилка, оформлення ТТН. Аналіз по каналам продаж та менеджерам.

Ведення витрат: ведення та аналіз витрат за підрозділами та видами. Автоматичний розрахунок валового та чистого прибутку. Графічна аналітика.



Рисунок 2.2 – Логотип системи

Переваги даної програми для магазину:

Зручність: простий та зрозумілий інтерфейс програми. Можливість працювати на будь-якому пристрої. Індивідуальні доступи та налаштування для користувачів.

Ефективність: ефективне управління однією або мережею торгових точок. Контроль за товарами, прибутками, виручкою та касирами навіть зі смартфона.

Підтримка: безоплатно відповідаємо на всі ваші запитання, допоможемо з опануванням програми, налаштуваннями, щоб користування було максимально легким та зрозумілим.

Створення та управління каталогом товарів, зручний довідник товарів: поділ товарів на категорії до десяти рівнів ієрархії. Створення товарів та категорій вручну, або ж за допомогою імпорту з Excel. Робота з властивостями

товару з використанням модифікацій або тег. Облік по серійним номерам. Швидке дублювання складу у випадку відкриття ще однієї точки продажу.

Швидке встановлення та управління цінами: встановлення різних націнок на категорії, автоматичний обрахунок ціни. Можливість зміни ціни за допомогою імпорту прайсу з файлу. Створення додаткових типів цін, закріплення їх за покупцями. Ручна знижка на чек або окрему позицію. Заборона знижок на вибрані позиції.

Акції та програми лояльності: запровадження періодичних акцій на вибрані товари, окремі категорії, або ж весь склад. Встановлюйте термін дії акції, або ж налаштовуйте акції вихідного дня. Друк акційних цінників. Персональні знижки клієнтам та бонусні програми лояльності.

Налаштуйте бізнес під себе: використовуйте гнучкі налаштування програми DNTrade під ваші задачі, обирайте дизайн та вкладки, якими найчастіше користуєтесь. Розділяйте права доступу користувачів до функціоналу програми та її елементів. Налаштовуйте обмеження до документів та права їх змінювати.

DNTrade підтримує всі типи складських операцій.

Можливість пошуку товару по різних параметрах. Швидке формування чеку за штрих-кодом. Індивідуальні налаштування вигляду вікна касира: вибір відображення товарів списком, або піктограмами, вимкнення зайвих колонок, винесення основних груп товарів у швидкий доступ, зміна розмірів вікна чеку та ін. Оплата готівкою, карткою, змішана, сертифікатом та бонусами.

Продаж поштучних товарів та закріплення їх кратності одиниці. Автоматичне зчитування у чек ваги товару з ваг, підключених до ПК або етикетки. Акцизні товари та налаштування всіх необхідних параметрів. Продаж складеного товару, модифікацій, товарів з серійними номерами.

Контроль виручки за зміну та залишку готівки в касі. Внесення та вилучення готівки. Звіти. Облік робочого часу касира по відкриттям та закриттям зміни. Можливість підключення програмного РРО. Встановлення заборони на продаж товару в мінус, а також нижче мінімальної ціни.

Створення документів приходу у два кліки

Здійснюйте прихід товару шляхом створення документу вручну, за допомогою імпорту накладної, або ж у два кліки із замовлення товарів у постачальника. З одного вікна роздрукуйте цінники на товари, в яких змінилася ціна, та здійснюйте завантаження їх у ваги. Слідкуйте за взаєморозрахунками з постачальниками.

Контроль товарів та їх переміщень

Переміщуйте товари між точками, або зі складу зберігання. Списуйте на витрати товари, які зіпсувалися, здійснюйте повернення. Будь-який документ швидко створюється за допомогою сканера, або ж на підставі іншого документа.

Швидке проведення інвентаризації

Здійснюйте інвентаризацію товарів зі смартфона або за допомогою ТЗД. Вкажіть лише фактичну наявність, все інше програма зробить сама. Проводьте інвентаризацію по категоріях, щоб не зупиняти торгівлю.

Вчасна закупка товарів

Слідкуйте в реальному часі за загальними залишками товару та в розрізі торгових точок. Система повідомить, якщо товару стало менше мінімального залишку. Формуйте замовлення у постачальника на підставі кількості продаж даного товару, або ж мінімального залишку.

Сучасна аналітика

Обраховуються і відображаються лише реальні цифри, які ви заробили.

Аналіз собівартості товару з урахуванням ціни постачальника та додатковими витратами.

Аналітика по товарах дозволяє детально проаналізувати попит на продукцію та зробити висновки щодо управління асортиментом.

Звіти по продажах, залишкам, коштам, прибуткам та витратам доступні у табличному та графічному форматі.

Формування звітів в розрізі точок, товарів, співробітників за будь-який період роботи.

2.4 POS Vector

Повний комплекс рішень для обліку продажів та аналітики ресторанного та торгового бізнесу[6]:

- POS-обладнання.
- Програмне забезпечення.
- Впровадження та підтримка.
- Автоматизація магазинів.
- POS-обладнання + програма обліку + впровадження у ваш бізнес.

Що входить у POS-систему

- POS-обладнання, що підходить саме вашій торговій точці.
- Програма для автоматизації ресторану Poster | Статистика.
- Встановлена та налаштована програма для обліку (з ПРРО).
- Впровадження автоматизації у ваш бізнес.
- Навчання роботі з програмою та обладнанням.
- Підтримка впродовж усього часу використання POS-системи.

Якщо система не підійде – без зайвих питань повертаються гроші. Фірма повністю беремо на себе автоматизацію у вашому магазині. Користувачам не доведеться самотужки розбиратись в налаштуваннях програми та підключати обладнання.

Підбирається POS-комплект: набір касового обладнання, що якнайкраще підходить магазину та відповідає вимогам



POS *Vector*

Рисунок 2.3 – Логотип системи

Є все необхідне для автоматизації роздрібного магазину будь-якого профілю.

Складський облік. Облік надходжень, переміщень, реалізації товарів.

Друк чеків. Чеки покупцю, у тому числі фіскальний чек.

Робота з вагами. Програма синхронізується із вагами.

ПРРО. Дані до ДФС передаються безпосередньо з облікової програми.

Штрих-коди. Введення товарів вручну в програмі і сканером.

Цінники/Етикетки. Друк цінників та етикеток на товари з облікової програми.

Модуль «Склад» дозволяє автоматизувати роботу з товарами: вести номенклатуру товарів, відстежувати складські запаси та багато іншого

Управління товарами. Необмежена кількість категорій та товарів. У картку товару додається вся потрібна вам інформація: постачальник, собівартість, націнка, дата приходу, термін придатності, мінімальний залишок тощо.

Рух товарів. Накладні по приходу, списанню, переміщенню, поверненню. Автоматична та ручна переоцінка. Інвентаризації, зокрема. за допомогою терміналу збору даних.

Робота з постачальниками. Програма для магазину дозволяє скласти базу постачальників з усіма необхідними реквізитами та вести облік взаєморозрахунків із ними.

Звітність. Звіти за будь-який період, на будь-яку дату: за продажами та чеками, за ТМЦ та залишками, за придатками, за накладними та рухами товарів, за порадами та скасуваннями, по застосуванню дисконтів, з продажу фіскальних товарів та багато інших. Усі дії у складській програмі фіксуються за користувачами.

Касовий модуль на Windows. Модуль для робочого місця касира з ПРРО. Інтерфейс касової програми гранично простий і зрозумілий – новий продавець опанує програму за півгодини.

Сумісність з обладнанням. Касова програма підтримує роботу з будь-яким POS-обладнанням: чековий та фіскальний принтер, сканер, ваги, банківський термінал та ін.

Будь-які форми оплати. Готівка, банківська картка, бонусний рахунок, борг, а також комбінована оплата.

Аналітика у роздрібній торгівлі. Модуль адміністрування та статистики дозволяє віддалено керувати торговою точкою в режимі реального часу, переглядати звіти та статистику за багатьма показниками.

Віддалене управління. Контроль роботи торгової точки, де б ви не перебували, з будь-якого пристрою з доступом до Інтернету.

Web доступ. Увійти до адмін-панелі для перегляду та редагування даних (товарів, дисконтних карток, користувачів) можна з браузера на ноутбучі або телефоні.

Контроль. Звіти в режимі реального часу та за будь-яку дату в розрізі продажів, прибутку, постачальників, персоналу та ін.

Статистика. Зведена інформація щодо товарів, прибутку, дій користувачів у вигляді наочних графіків та діаграм.

2.5 Використані інструменти та технології

Для практичної реалізації інформаційної системи був використаний наступний стек технологій.

Мова програмування C#

C# – це сучасна багатопарадигмова мова програмування, створена корпорацією Microsoft у 2000 році в рамках платформи .NET. Її основною метою є надання потужного інструменту для розробки різноманітного програмного забезпечення – від веб-додатків до ігор.

Підтримує основні принципи об'єктно-орієнтованого програмування: інкапсуляція, наслідування, поліморфізм. Забезпечує легкість створення та управління складними системами.

C# забезпечує суворий контроль типів, що зменшує кількість помилок під час виконання програм. Мова підтримує як явну, так і неявну типізацію. Використовується збирач сміття (Garbage Collector), який автоматично звільняє невикористовувану пам'ять.

Мова підтримує багатоплатформеність завдяки .NET Core (зараз .NET 5+) та Mono. Використовується для створення додатків для Windows, macOS, Linux, Android, iOS та інших платформ.

Підтримка асинхронного програмування через ключові слова `async` і `await`.

C# включає концепції функціонального програмування, такі як лямбда-вирази, функції вищого порядку, методи розширення та LINQ.

Мова постійно оновлюється з новими версіями, додаючи сучасні функції (наприклад, `nullable reference types`, `record types`).

Основні компоненти екосистеми:

- .NET Framework / .NET Core / .NET 6+: платформи виконання для програм, написаних на C#.
- Visual Studio: інтегроване середовище розробки (IDE), яке надає зручні інструменти для створення додатків.
- NuGet: система управління пакетами для розширення можливостей через бібліотеки.

Типи застосунків, які створюються на C#:

- Desktopні програми: Windows Forms, WPF, MAUI.
- Веб-додатки та сервіси: ASP.NET Core.
- Мобільні додатки: Xamarin, .NET MAUI.
- Ігри: Unity.
- Хмарні додатки: інтеграція з Azure.
- Мікросервіси: за допомогою .NET Core та Docker.

Переваги C#:

- Простота у вивченні та використанні.
- Висока продуктивність програм.

- Багатофункціональність.
- Сильна підтримка від Microsoft і великої спільноти.

Платформа .Net Core

.NET Core – це кросплатформена, високопродуктивна, відкрита платформа для розробки програмного забезпечення, створена компанією Microsoft. Вона є частиною екосистеми .NET і забезпечує гнучкість, масштабованість та ефективність для сучасних розробників.

Підтримує розробку і виконання програм на Windows, Linux, macOS. Дозволяє створювати програми, які можна розгортати в різних середовищах без змін у коді. Розроблений для створення швидких і масштабованих програм. Використовується в хмарних сервісах, таких як Microsoft Azure. Вихідний код доступний на GitHub. Спільнота активно бере участь у розробці платформи. Дозволяє використовувати тільки необхідні компоненти, що робить програми більш компактними. Управління залежностями здійснюється через NuGet. Можливість додавання нових бібліотек і функцій через сторонні пакети. Підтримує інтеграцію з іншими мовами та платформами. Сумісний із фреймворком .NET Framework через .NET Standard і бібліотеки. Постійні вдосконалення та регулярні оновлення забезпечують підтримку сучасних стандартів розробки.

ASP.NET Core:

Фреймворк для створення веб-додатків і API.

Типи програм, які можна створювати на .NET Core:

Веб-додатки: використовуючи ASP.NET Core для створення швидких та безпечних веб-сервісів і веб-сайтів.

Консольні програми: простий варіант програм для автоматизації або серверних задач.

Хмарні додатки: оптимізовані для роботи в Microsoft Azure або інших хмарних платформах.

Мікросервіси: побудова масштабованих і незалежних сервісів.

Інтернет речей (IoT): для роботи з апаратним забезпеченням через сенсори та контролери.

Кросплатформені десктопні додатки: завдяки інтеграції з .NET MAUI.

Переваги .NET Core:

- Гнучкість і масштабованість: підходить для невеликих програм та великих корпоративних рішень.
- Кросплатформеність: одна кодова база для різних платформ.
- Простота розгортання: підтримка контейнерів Docker та різних хмарних середовищ.
- Безкоштовна та відкрита: підтримка великої спільноти.

Фреймворк ASP.NET Core

ASP.NET Core – це кросплатформений, високопродуктивний фреймворк для розробки веб-додатків і веб-API, створений корпорацією Microsoft. Він є частиною екосистеми .NET Core (або .NET) і пропонує сучасні інструменти та архітектури для створення масштабованих і швидких веб-рішень.

Підтримує роботу на Windows, Linux, macOS. Дозволяє розробникам використовувати різні платформи для створення та хостингу додатків. Фреймворк оптимізований для створення високопродуктивних веб-додатків. Використовує легкий веб-сервер Kestrel для швидкого оброблення HTTP-запитів. Побудований на системі залежностей NuGet, що дозволяє використовувати тільки потрібні компоненти. Гнучкість у налаштуванні Middleware для обробки запитів. Код ASP.NET Core є відкритим та доступним на GitHub. Активно підтримується спільнотою розробників.

Запити обробляються через серію Middleware-компонентів, що дозволяє додавати, змінювати чи видаляти функціональність. Вбудована підтримка впровадження залежностей для побудови модульного та тестованого коду. Можливість створення власних Middleware та інтеграції сторонніх бібліотек.

Сучасний підхід для створення сторінок із вбудованим кодом С#. Розробка інтерактивних інтерфейсів за допомогою С# замість JavaScript.

Архітектура Model-View-Controller (MVC)

MVC – це популярна архітектурна парадигма, що використовується в розробці програмного забезпечення, зокрема веб-додатків. Вона розділяє програму на три основні компоненти: Model, View та Controller, що дозволяє спрощувати підтримку, тестування та розширення додатка.

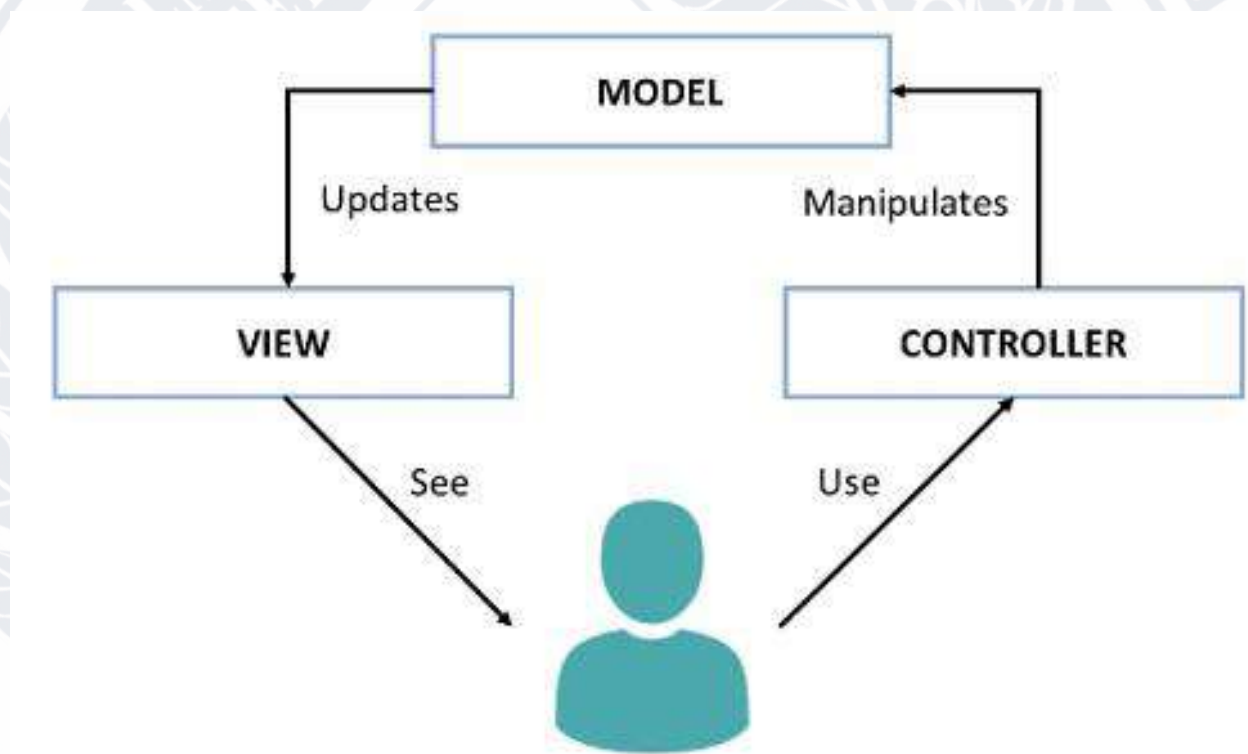


Рисунок 2.1 – Модель MVC

Основні компоненти MVC

Model (Модель). Представляє дані та логіку роботи з ними. Відповідає за управління станом програми: отримання, збереження та обробку даних. Приклади: класи для роботи з базою даних, бізнес-логіка.

Обов'язки:

- Отримання даних із бази даних або зовнішніх джерел.
- Зберігання стану додатка.

- Валідування даних перед обробкою.

View (Представлення). Відповідає за відображення інформації користувачу. Формує інтерфейс користувача (UI), відокремлений від бізнес-логіки. Використовує шаблони для рендерингу даних із Model.

Обов'язки:

- Форматування даних для представлення.
- Інтерактивність (зазвичай разом із JavaScript або іншими клієнтськими технологіями).

Controller (Контролер). Посередник між Model та View. Обробляє вхідні запити користувача (HTTP-запити). Викликає відповідні методи Model для отримання даних і передає їх до View для відображення.

Обов'язки:

- Управління маршрутизацією (який метод чи сторінку викликати).
- Обробка запитів користувачів і вибір правильного View або Model.
- Перетворення даних між Model та View.

Користувач надсилає запит (наприклад, натискає кнопку або вводить URL). Controller отримує цей запит і визначає, як його обробити. Викликає потрібний метод у Model, щоб отримати або обробити дані. Model повертає дані у відповідь на запит. Controller передає ці дані у View. View відображає дані користувачеві у вигляді HTML-сторінки, JSON-об'єкта тощо.

Переваги MVC

- Розділення відповідальностей: зменшує зв'язність між компонентами, полегшуючи зміну й масштабування програми.
- UI (View) відокремлений від бізнес-логіки (Model).
- Простота тестування: легше тестувати кожен компонент окремо.
- Повторне використання коду: один і той самий Model може бути використаний у кількох View.
- Гнучкість: можливість легко змінювати UI без втручання в Model або Controller.

- Розширюваність: зручна підтримка та додавання нових функцій.
- Недоліки MVC
- Складність для невеликих проєктів: для простих додатків структура MVC може бути надмірною.
 - Крута крива навчання: розробникам, які тільки починають працювати з MVC, може бути важко зрозуміти взаємодію між компонентами.
 - Більше файлів і коду: поділ на окремі компоненти збільшує кількість класів і файлів, що може ускладнити підтримку.

Entity Framework Core (EF Core)

Entity Framework Core – це сучасний, об'єктно-реляційний маппер (ORM) для платформи .NET, створений компанією Microsoft. EF Core дозволяє розробникам працювати з базами даних у термінах об'єктів C#, усуваючи необхідність написання більшості SQL-запитів вручну.

Працює на платформах Windows, Linux, macOS завдяки інтеграції з .NET Core. Підтримує SQL Server, PostgreSQL, MySQL, SQLite, Oracle, Cosmos DB та інші бази даних. Об'єктно-реляційне відображення (ORM): дозволяє працювати з даними у вигляді об'єктів, які відображаються на таблиці бази даних. Забезпечує можливість писати складні запити за допомогою мови запитів LINQ (Language Integrated Query). Інструменти для створення, оновлення та управління структурою бази даних у міру змін у коді (Code First). Підтримка Code First, Database First та Mixed Development. Оптимізований для створення високопродуктивних додатків.

Інтегроване середовище розробки Visual Studio

Visual Studio – це потужне інтегроване середовище розробки (IDE), створене компанією Microsoft. Воно забезпечує розробників усіма необхідними інструментами для створення, тестування та розгортання додатків для різних платформ, включаючи Windows, macOS, iOS, Android, веб, хмару тощо.

Підтримка розробки для .NET, ASP.NET, C++, Python, JavaScript, TypeScript, Node.js та інших мов. Інструменти для кросплатформової розробки, наприклад, через Xamarin або MAUI для мобільних додатків.

Підсвітка синтаксису, автодоповнення (IntelliSense), рефакторинг. Розширена підтримка різних мов програмування. Покрокове виконання коду (debugging). Аналіз змінних, стека викликів, логів. Підтримка віддаленого налагодження (Remote Debugging). Вбудовані інструменти для роботи з базами даних. Можливість підключення до баз даних (SQL Server, SQLite, MySQL). Інструменти для перегляду, редагування даних і виконання SQL-запитів. Швидке створення та розгортання додатків у хмарних середовищах, таких як Microsoft Azure. Інтеграція з Docker і Kubernetes. Розширення та кастомізація. Підтримка тисяч розширень через Visual Studio Marketplace. Інтеграція зі сторонніми інструментами (наприклад, Git, AWS Toolkit, Azure DevOps). Інструменти для створення та запуску модульних тестів (Unit Testing). Підтримка автоматизованого та інтеграційного тестування. Вбудована підтримка Git та Azure DevOps. Зручний інтерфейс для управління репозиторіями, гілками, злиттям.

Особливості Visual Studio

IntelliSense: потужне автодоповнення коду з підказками про методи, властивості та змінні. Допомагає уникнути синтаксичних помилок.

CodeLens: додатковий функціонал, який дозволяє переглядати зміни у файлах, історію комітів і результати тестів безпосередньо в редакторі.

Live Share: інструмент для спільної роботи в реальному часі, що дозволяє кільком розробникам редагувати та налагоджувати код одночасно.

Архітектурні діаграми: інструменти для створення UML-діаграм, аналізу залежностей та проектування архітектури програм.

Профілювання та аналіз продуктивності: вбудовані інструменти для моніторингу використання пам'яті, CPU, швидкості виконання запитів.

Міграції баз даних: інтеграція з Entity Framework Core для автоматизації міграцій баз даних.

Переваги Visual Studio

- Потужний інструментарій: універсальність для створення різних типів додатків.
- Зручний інтерфейс: інтуїтивно зрозумілий, легко налаштовується.
- Інтеграція з іншими продуктами Microsoft: тісна інтеграція з Azure, GitHub, Office 365.
- Спільна робота: інструменти для командної розробки (Live Share, Azure DevOps).

Недоліки Visual Studio

- Вимоги до ресурсів: IDE може працювати повільно на застарілих або малопотужних системах.
- Висока складність для початківців: розмаїття функцій може спантеличити новачків.
- Платність деяких редакцій: Professional і Enterprise потребують ліцензії.

Висновок до другого розділу

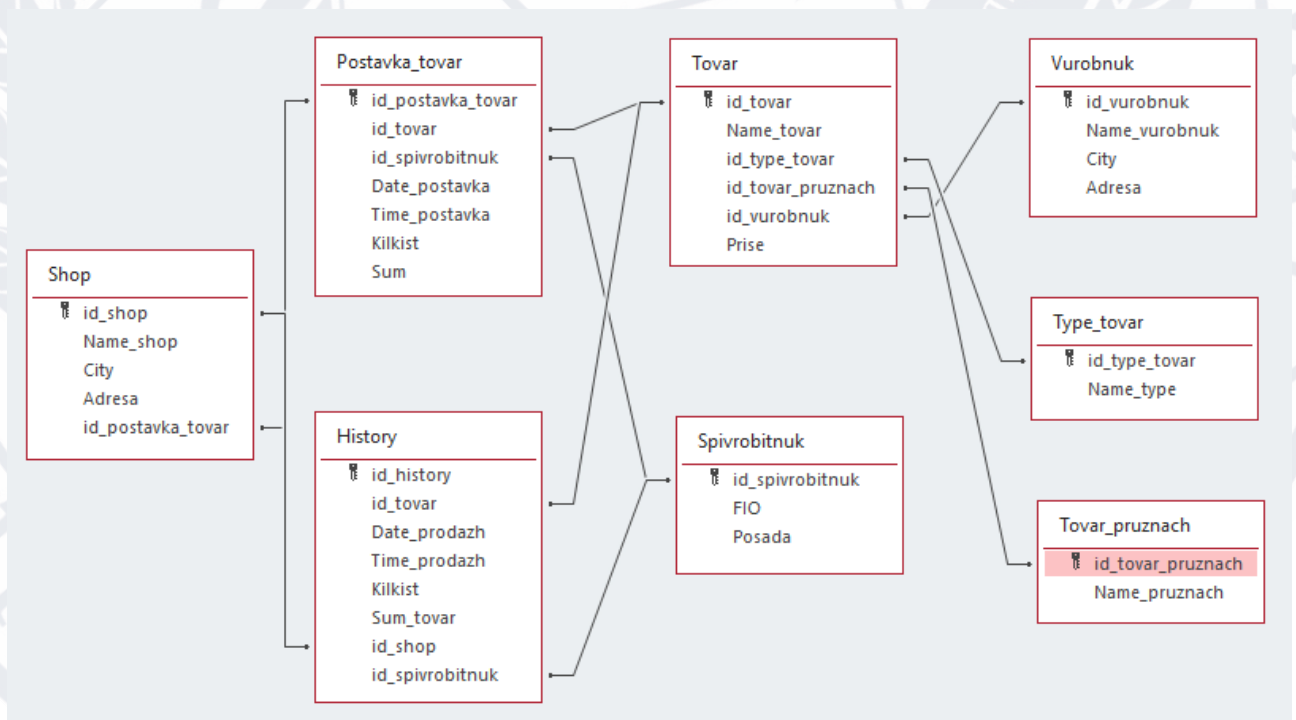
В даному розділі була сформульована постановка задачі з розробки інформаційної системи підтримки діяльності магазину побутових товарів. Розглянуті аналогічні системи, наведено їх короткий опис та характеристики. Також описано стек технологій, які використовувалися під час розробки інформаційної системи.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Структура бази даних

На рисунку 3.1 наведено схему бази даних інформаційної системи. Вона містить інформацію про магазин, який характеризується назвою, містом, адресою та поставкою товарів. Товаром, який характеризується назвою, типом, призначенням, виробником та ціною. Виробник товару характеризується назвою, містом та адресою, де він знаходиться. Співробітник магазину характеризується прізвищем, ім'ям, по батькові та посадою. Поставка товару характеризується товаром, співробітником, який його прийняв, датою та часом поставки, кількістю та вартістю. Також зберігається історія продажу товару, яка характеризується товаром, датою та часом продажу, кількістю, сумою, магазином та співробітником, який його продав.



3.2 Програмна реалізація

На рисунку 3.2 наведено головний екран розроблюваної системи. Тут можна побачити категорії всіх товарів, додавання нового товару в базу даних,

перевірка наявності товару, аналітика та кошик, за допомогою якого можна оформити покупку товару. Також в систему можна зайти під своїм логіном та паролем. Є окремий вхід для адміністратора та продавця.

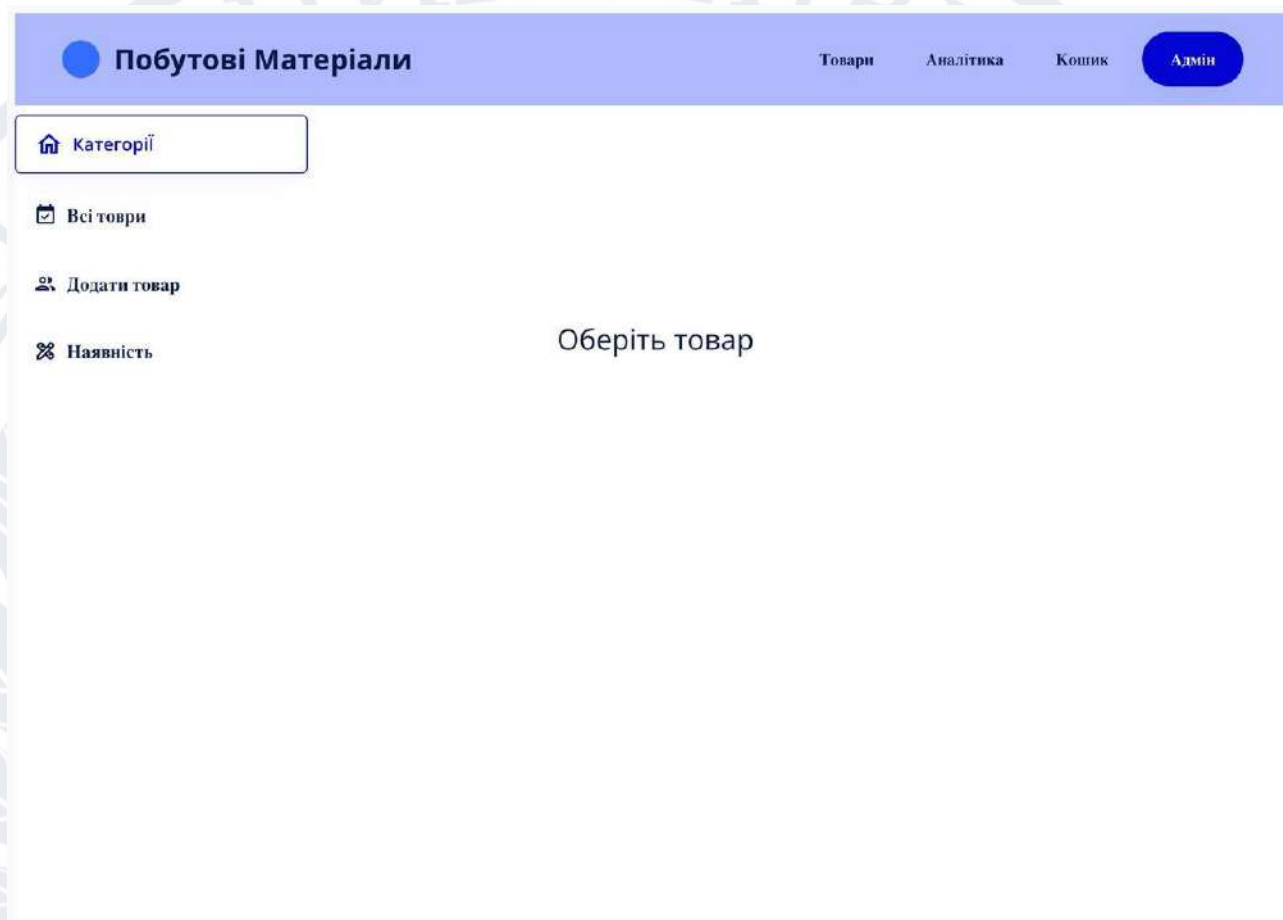


Рисунок 3.2 – Головний екран системи

На наступному рисунку наведені товари, які поділені по категоріям.

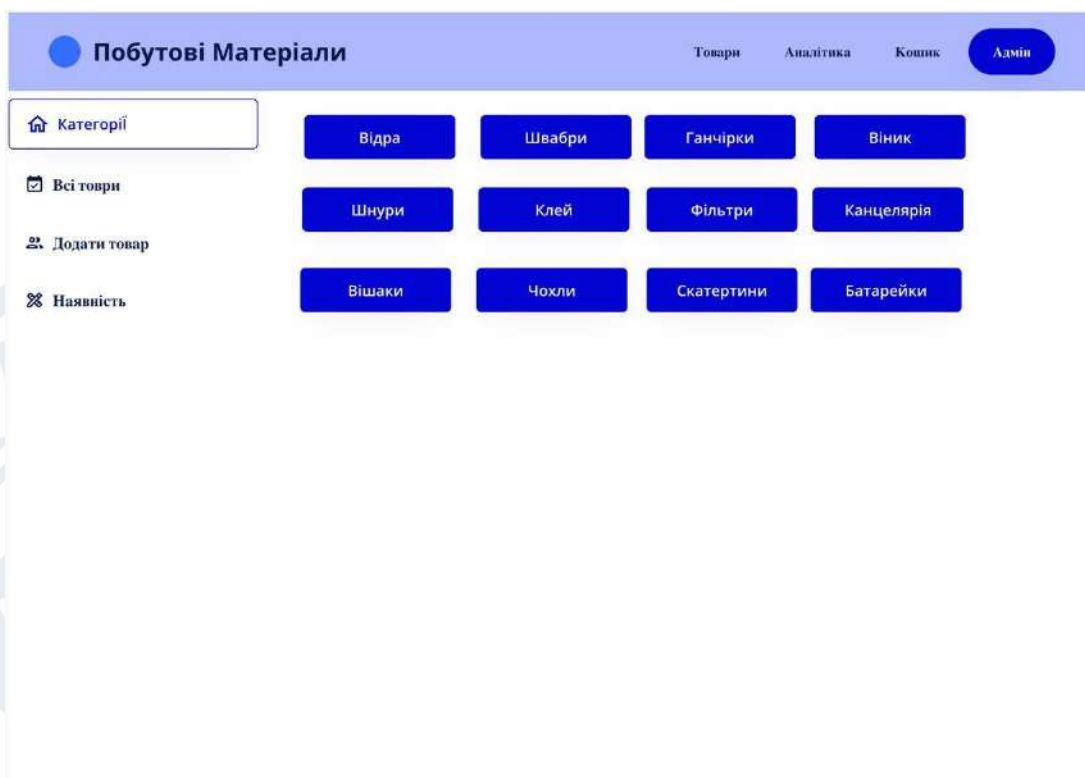


Рисунок 3.3 – Категорії товару

На рисунку 3.4 наведені товари для додавання їх у кошик під час покупки.

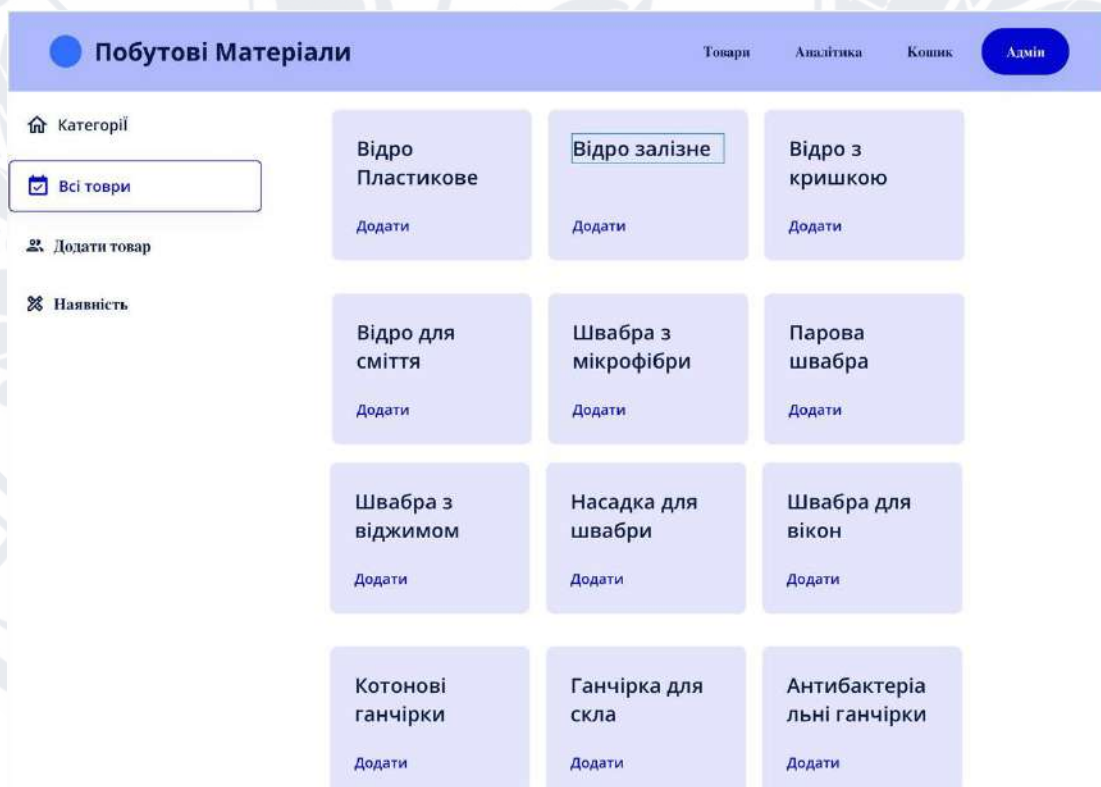


Рисунок 3.4 – Додавання товару до кошика

На рисунку 3.5 показано процес додавання нового товару в базу. Для цього необхідно вписати його назву, ціну за одиницю та вказати кількість одиниць. Ця кількість буде відображатися як кількість товару, що наявний в магазині.

The screenshot shows a web application interface for 'Побутові Матеріали' (Household Materials). The top navigation bar includes links for 'Товари' (Goods), 'Аналітика' (Analytics), 'Кошик' (Cart), and 'Адмін' (Admin). On the left sidebar, there are options: 'Категорії' (Categories), 'Всі товари' (All goods), 'Додати товар' (Add item), and 'Наявність' (Availability). The main content area is titled 'Додати товар' (Add item) and contains three input fields: 'Назва' (Name) with a placeholder 'Напишіть назву товару' (Write the item name), 'Ціна' (Price) with a placeholder 'Ціна(за од.)' (Price (per unit)), and 'Кількість' (Quantity) with a placeholder 'Вкажіть кількість' (Specify quantity). A blue button labeled 'Додати +' (Add +) is positioned below the input fields.

Рисунок 3.5 – Додавання нового товару в базу

На рисунку 3.6 показано процес оформлення покупки товару. Продавець-касир додає кожен товар до кошику та обирає кількість кожного товару. За потреби товар можна вилучити. Після натискання кнопки «Оплатити» на екрані з'явиться загальна вартість покупки. Оплата можлива двома способами: за готівку та картою. У випадку готівки касир вводить суму, яку отримав від покупця і калькулятор автоматично розрахує решту (якщо потрібно). У випадку оплати картою активується банківський термінал і відбувається оплата.

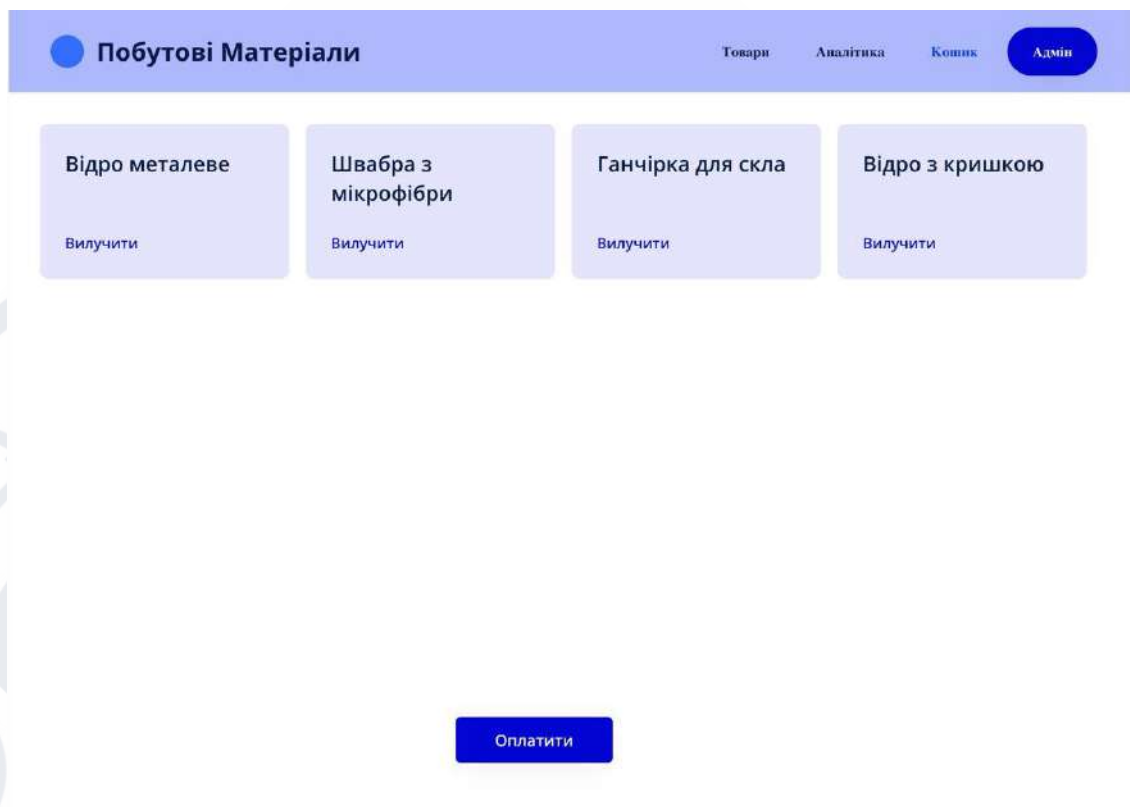


Рисунок 3.6 – Оформлення покупки

В системі налаштовані засоби для аналітики. За їх допомогою можна проаналізувати яка кількість товару по категоріях була продана за певний проміжок, як розподілена оплата у % (готівка/карта), скільки та якого товару лишилося на складі та інше. Це дає змогу вчасно замовляти той чи інший товар, щоб він не залежувався на складі, а з іншого боку, щоб не було випадків, що товар раптово закінчився.

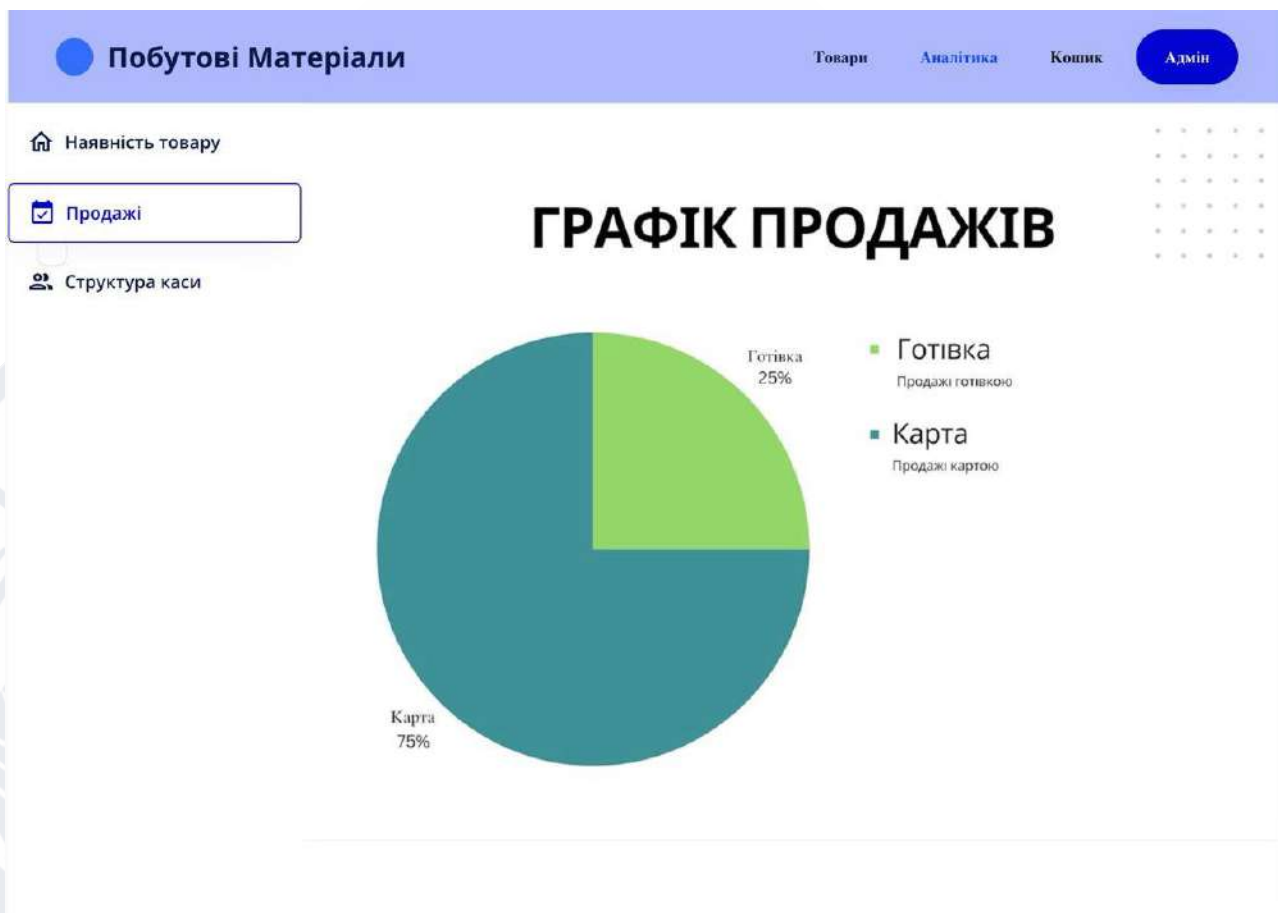


Рисунок 3.7 – Графік продажу товарів за готівку та карту

На рисунку 3.7. показано графік продажу товарів з оплатою за готівку та за карту за зміну. Також аналогічний графік можна сформувати за більший період ніж зміна.

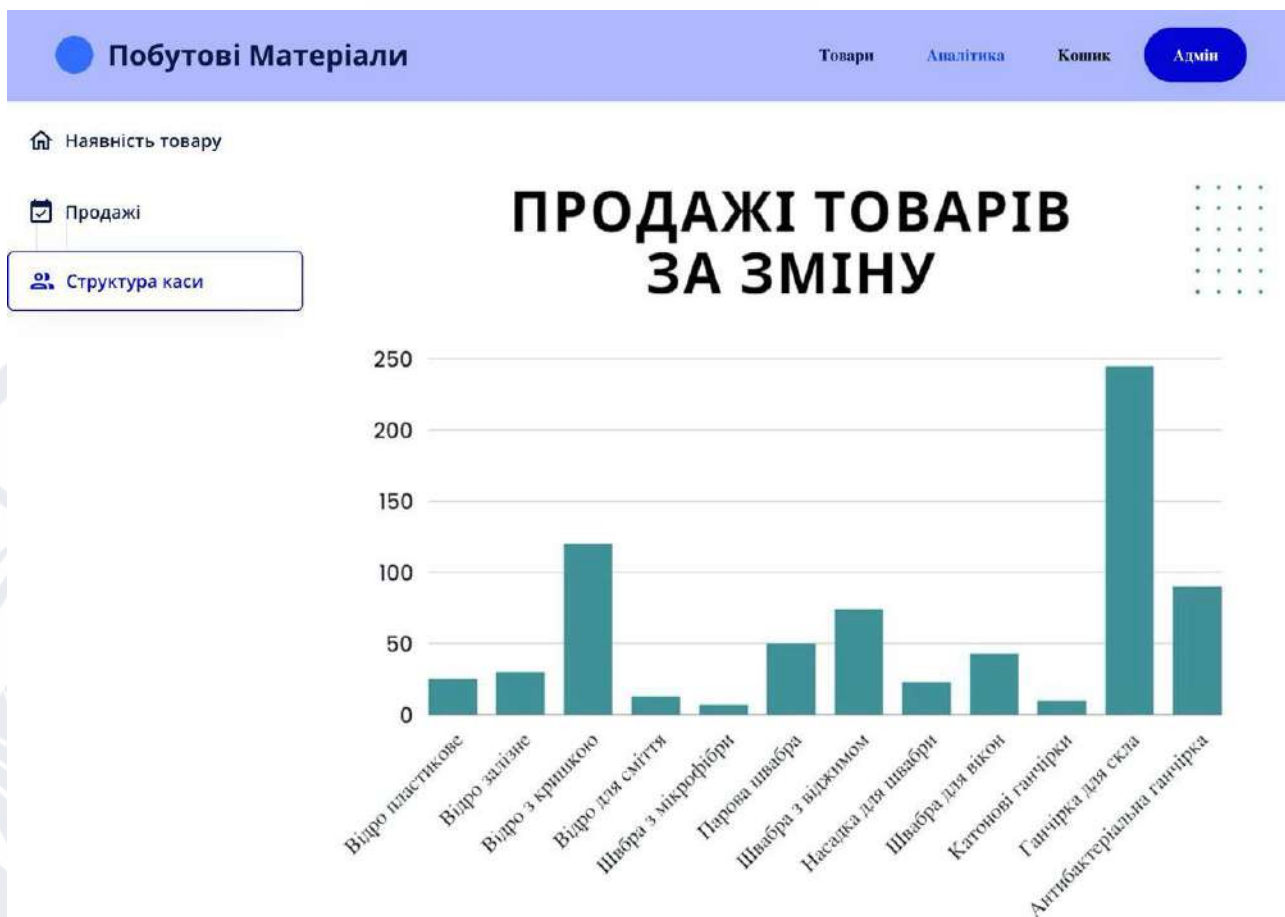


Рисунок 3.8 – Продаж товару за зміну

На рисунку 3.8 наведено графік продажу товарів (в одиницях) за зміну. На ньому показується скільки було продано того чи іншого товару за зміну роботи системи. В кожній зміні ці цифри будуть відрізнятися. В кінці місяця можна переглянути загальний графік продажу товару.

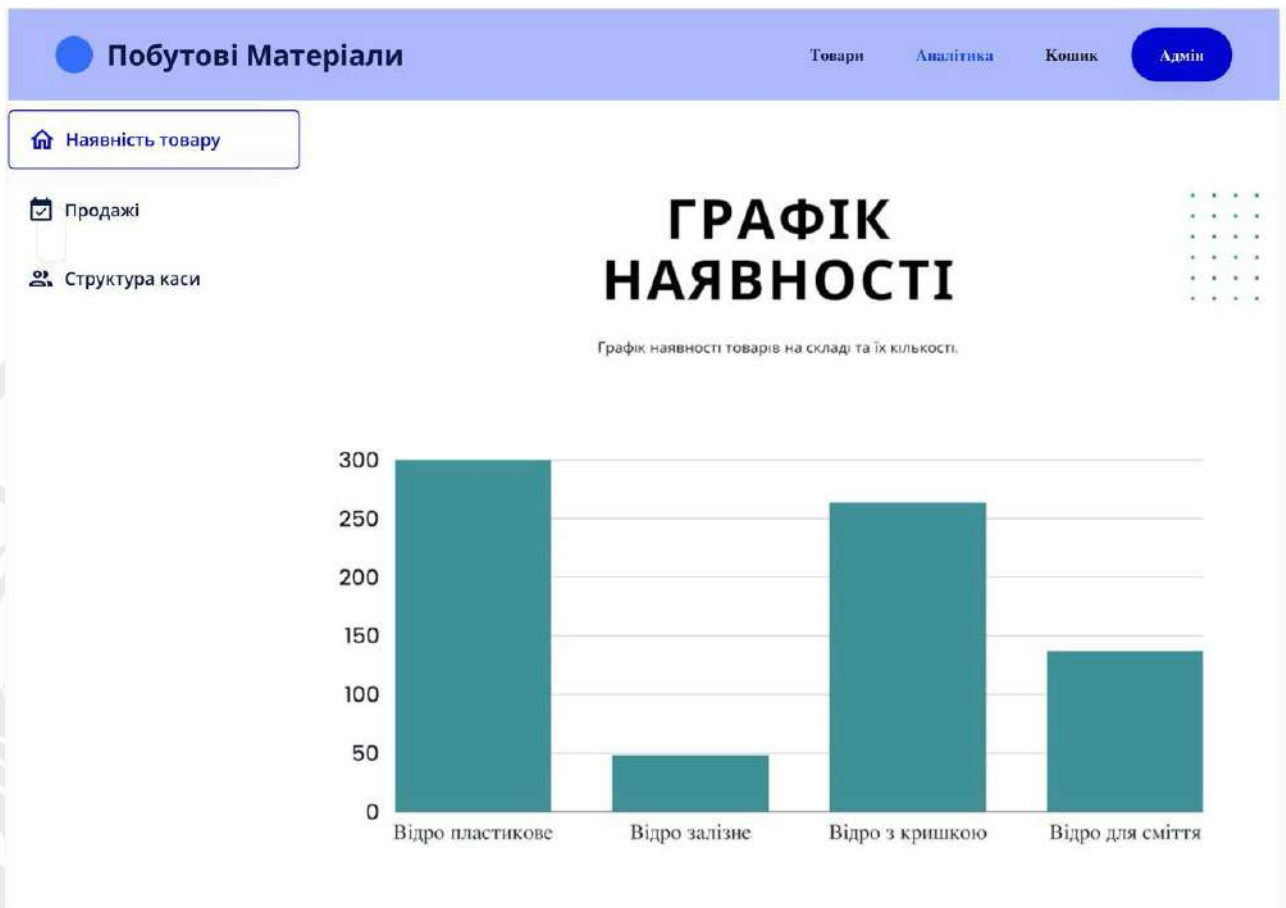


Рисунок 3.9 – Наявність товару на складі

На рисунку 3.9 наведено графік наявності товару (в одиницях) на складі. За його допомогою можна слідкувати за наявністю товару та вчасно робити замовлення нового.

Висновок до третього розділу

В даному розділі описано процес функціонування інформаційної системи. Наведено структуру бази даних, яка використовувалася для зберігання необхідної інформації.

ВИСНОВКИ

У результаті виконання кваліфікаційної (магістерської) роботи було розроблено інформаційну систему підтримки діяльності магазину побутових матеріалів. Розроблена інформаційна система дозволяє автоматизувати процеси управління та підтримки діяльності магазину побутових матеріалів, що сприяє підвищенню ефективності його функціонування.

Проведено детальне дослідження предметної області, що дозволило виявити основні потреби магазину побутових матеріалів, включаючи облік товарів, управління замовленнями, контроль залишків і підтримку взаємодії з клієнтами.

Проектування інформаційної системи виконано з урахуванням сучасних вимог до програмного забезпечення, таких як модульність, масштабованість та зручність інтеграції. Впроваджено функції обліку товарів, включаючи їх надходження, списання та інвентаризацію.

Реалізовано модуль обробки замовлень із можливістю автоматизованого формування чеків. Забезпечено інтеграцію з базами даних для зберігання та оперативного доступу до інформації. Розроблений інтуїтивно зрозумілий інтерфейс сприяє легкому освоєнню системи співробітниками магазину, зменшуючи витрати часу на навчання.

Розроблена система сприяє підвищенню точності обліку товарів за рахунок мінімізації людських помилок. Забезпечено використання функцій аналітики для прийняття управлінських рішень.

Розроблена інформаційна система підтвердила свою доцільність і ефективність у контексті підтримки діяльності магазину побутових матеріалів. Вона може бути рекомендована для впровадження у підприємствах подібного профілю. У перспективі систему можна розширити шляхом інтеграції з онлайн-магазином та додатковими аналітичними модулями, що дозволить ще більше підвищити конкурентоспроможність бізнесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Інформаційна система. URL: <https://naurok.com.ua/informaciyna-sistema-229642.html>
2. Конспект лекцій з інформаційних систем URL: https://www.dstu.dp.ua/Portal/Data/3/21/3-21-z_kl64.pdf
3. Моделі бізнес-процесів URL: <https://studfile.net/preview/1756224/page:5/>
4. IBS магазин Універсал. URL: <https://www.ibsystems.com.ua/ua/ibs-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0-%D0%B4%D0%BB%D1%8F-%D0%BC%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD%D1%83-%D0%BF%D1%80%D0%BE%D0%BC%D1%82%D0%BE%D0%B2%D0%B0%D1%80%D1%96%D0%B2/%D0%BE%D0%BF%D0%B8%D1%81>
5. Програма для магазину DNTrade. URL: <https://dntrade.com.ua/blog/avtomatizaciya-magazinu>
6. Програма для магазину PosVector. URL: <https://www.pos-vector.net/torhivlia/pz/>
7. OTT Streaming Market Global Briefing 2020-30: Covid 19 Growth And Change.
8. The Business Research Company, 2020. 18 с.
9. Introduction to recommender systems: вебсайт. URL: <https://towardsdatascience.com/introduction-to-recommender-systems6c66cf15ada>
10. Aggarwal C.. Recommender Systems: The Textbook. Springer, 2016. 518 с.
11. Загальна інформація ASP.NET Core MVC: вебсайт. URL: <https://docs.microsoft.com/aspnet/core/mvc/overview?view=aspnetcore-6.0>
12. Вступ до Razor Pages в ASP.NET Core: вебсайт. URL: <https://docs.microsoft.com/aspnet/core/razor-pages/?view=aspnetcore-6.0&tabs=visual-studio>

- 13.Климчук О. В. Сучасні аспекти використання інформаційних систем і технологій в управлінні / О. В. Климчук. // Бізнес, інновації, менеджмент: проблеми та перспективи. – 2021. – №2. – С. 170–171.
- 14.The Latest Information Systems in the Enterprise Management and Trends in their Development / [A. Asaul, M. Voynarenko, L. Dzhulii та ін.]. // 9th International Conference on Advanced Computer Information Technologies (ACIT). – 2019. – №9. – С. 409–412.
- 15.Струсинська І. В. Інформаційні технології організації бізнесу - імператив інноваційного розвитку бізнес-структур / Ірина Володимирівна Струсинська. // Економіка та управління національним господарством. – 2018. – №2. – С. 40–49.
- 16.Шевчук І. Б. Інформаційні технології в бізнесі / І. Б. Шевчук, А. І. Старух, О. М. Васьків. – Львів: ННБК «АТБ», 2020. – 455 с.
- 17.Онопко А. С. Застосування інформаційних технологій в управлінні підприємством / А. С. Онопко, Ж. М. Жигалкевич. // Актуальні проблеми економіки та управління: збірник наукових праць молодих вчених. – 2017. – №11. – С. 6.
- 18.Оляднічук Н. В. Інформаційні системи в бухгалтерському обліку/Н.В. Оляднічук. // Збірник наукових праць Уманського національного університету садівництва.. – 2016. – №2. – С. 198–206.
- 19.Joel M. Murach's MySQL / Murach Joel., 2019. – 646 с. – (Mike Murach and Associates, Incorporated). – (3).
- 20.Riaz A. Full Stack Web Development For Beginners: Learn Ecommerce Web Development Using HTML5, CSS3, Bootstrap, JavaScript, MySQL, and PHP / Ahmed Riaz., 2021. – 392 с. – (Independently Published).
- 21.Nixon R. Learning PHP, MySQL & JavaScript 5e : With jQuery, CSS & HTML5 / Robin Nixon. – Sebastopol, United States, 2018. – 800 с. – (O'Reilly Media, Inc, USA). – (5).
- 22.Deen D. Why WordPress is the best CMS (2021) [Електронний ресурс] / David Deen. – 2020. – Режим доступу до ресурсу:

<https://getflywheel.com/layout/best-cms-2020-wordpress/>

23. Шатілов О. В. Моделювання процесу управління стратегічною гнучкістю підприємства / О. В. Шатілов. // Ефективна економіка. – 2013. – №1.
24. Мінухін С. В. Методи і моделі проектування на основі сучасних CASE-засобів. Навчальний посібник / С. В. Мінухін, О. М. Беседовський, С. В. Беседовський. – Харків, 2008. – 272 с. – (Харківський національний економічний університет).
25. Алексенко О. В. Технології програмування та створення програмних продуктів : конспект лекцій / Ольга Василівна Алексенко. – Суми: СумДУ, 2017. – 161 с. – (СумДУ).
26. Гайна Г. А. Основи проектування бази даних / Г. А. Гайна. – Київ: КНУБА, 2005. – 204 с. – (КНУБА).
27. Змерзлий І. Клієнт-серверна архітектура та ролі серверів [Електронний ресурс] / Іван Змерзлий // Medium. – 2017.
28. Project Management Institute. A guide to the Project Management Body of Knowledge (PMBOK guide) and the Standard for project management / Project Management Institute. – Newton Square, PA, United States: Project Management Institute, 2021. – 250 с. – (Project Management Institute). – (7).