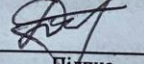


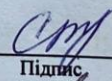
Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерних наук та інформаційних технологій

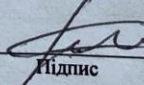
КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему АІС торгової точки з роздрібного продажу харчових товарів

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студентка 4 курсу, група КН-17-1  Д.Д. Гаврилюк
Курс, група виконавця Підпис Ініціали, прізвище

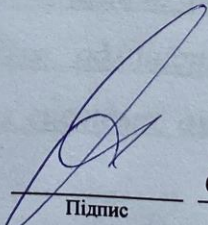
Керівник: старший викладач кафедри КНІТ  Т.К. Скрипник
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КНІТ  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор

09 червня 2021 р.

 О.В. Бармак
Підпис Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
 Факультет програмування та комп'ютерних і телекомунікаційних систем
 Кафедра комп'ютерних наук та інформаційних технологій
 Освітній ступінь бакалавр
 Галузь знань 12 – Інформаційні технології
 Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук та інформаційних технологій

(підпис)

д.т.н., професор О.В. Бармак

« 08 » лютого 2021 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «АІС торгової точки з роздрібного продажу харчових товарів»
2. Завдання видано студентці Гаврилюк Дар'ї Дмитрівні
(прізвище, ім'я, по батькові)
3. Керівник роботи ст. викладач кафедри КНІТ Скрипник Тетяна Казимирівна
(посада, прізвище, ім'я, по батькові)
4. Затверджено наказом університету від « 05 » лютого 2021 р. № 11
5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – розробка автоматизованої інформаційної системи торгової точки з роздрібного продажу харчових товарів. Для клієнта інформаційної системи необхідно автоматизувати: перегляд даних про торгові точки, перегляд каталогу товарів, перегляд статей. Для адміністратора торгової точки необхідно автоматизувати: авторизація, продажі, закупівлі, облік товарів, каталог товарів, персонал. Для адміністратора системи необхідно автоматизувати: доступ до всіх даних системи, активація облікових записів адміністраторів торгової точки.

Виконавець: студентка 4 курсу, група КН-17-1
Курс, група виконавця

Д.Д. Гаврилюк
Підпис

Д.Д. Гаврилюк
Ініціали, прізвище

Керівник: старший викладач кафедри КНІТ
Науковий ступінь, посада

Т.К. Скрипник
Підпис

Т.К. Скрипник
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: «АІС торгової точки з роздрібного продажу харчових товарів»

Виконавець кваліфікаційної роботи бакалавра: студентка групи КН-17-1 Гаврилюк Дар'я Дмитрівна

Керівник кваліфікаційної роботи бакалавра: старший викладач кафедри КНІТ Скрипник Тетяна Казимирівна

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
57	24	20	35	3

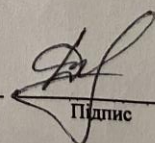
Метою кваліфікаційної роботи бакалавра є розробка АІС торгової точки з роздрібного продажу харчових товарів на платформі РНР. Для розробки інформаційної системи було використано мову програмування РНР, а також СКБД MySQL.

АІС торгової точки з роздрібного продажу харчових товарів призначена для автоматизації роботи трьох видів користувачів – клієнта, адміністратора торгової точки та адміністратора системи. Створений програмний додаток дозволяє спростити та автоматизувати їх роботу.

Розроблена АІС повністю готова до використання, тому в подальшому для неї можна придбати хостинг та налагодити обслуговування для реальних задач. Напрямами практичного використання розробленої інформаційної системи визначено автоматизовану реєстрацію та перегляд даних про продажі, закупівлі, ревізії, працівників.

Ключові слова: продаж, закупівля, інформаційна система, ревізія.

Виконавець: студентка 4 курсу, група КН-17-1
Курс, група виконавця


Підпис

Д.Д. Гаврилюк
Ініціали, прізвище

Зміст

Перелік скорочень	6
Вступ.....	7
Розділ 1	
Характеристика предметної області та постановка задачі	8
1.1 Аналіз предметної області	8
1.2 Аналіз існуючого програмного забезпечення предметної області	14
1.3 Аналіз сучасних засобів створення програмного забезпечення	18
1.4 Постановка задачі та вимоги до розробки інформаційної системи.....	20
Розділ 2	
Проектування інформаційної системи	21
2.1 Функціональна структура та бізнес-процеси системи	21
2.2 Інформаційна структура системи	24
2.3 Вибір засобів розробки інформаційної системи	32
2.3.1 Вибір мови програмування	32
2.3.2 Вибір фреймворку	33
2.3.3 Вибір СКБД	35
Розділ 3	
Програмна реалізація інформаційної системи	37
3.1 Структура та функціональне призначення складових системи	37
3.2 Особливості реалізації складових системи	39
3.3 Тестування інформаційної системи	46
3.4 Інструкція користувача.....	51
3.5 Вимоги до розгортання інформаційної системи.....	55
Висновки	56
Перелік посилань.....	57
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
PHP	Hypertext Preprocessor
ХТ	Харчовий товар
ТТ	Торгова точка
ЄСВ	Єдиний соціальний внесок
HTML	HyperText Markup Language
WML	Wireless Markup Language
XML	Extensible Markup Language
ФСС	Фонд соціального страхування
ПДВ	Податок на додану вартість
ІС	Інформаційна система
ІТ	Інформаційні технології
ПП	Програмний продукт
СКБД	Система керування базами даних
КРБ	Кваліфікаційна робота бакалавра
Кг	Кілограм
Уп.	Упаковка
Од.	Одиниця
MS	Microsoft

Вступ

У сьогоденному світі комерція відіграє дуже важливу роль для суспільства [1]. Майже кожна людина має можливість придбати будь-який продовольчий товар з будь-якої точки світу, при цьому не докладаючи зусиль.

У зв'язку з високим рівнем розвитку комерції, збільшенням кількості товарів на прилавках, збільшенням попиту виникло питання у автоматизації різних процесів продажів товарів за допомогою різних засобів ІТ, а саме у торгових точках з роздрібного продажу харчових товарів.

Таким чином, застосування ІТ для автоматизації процесів супроводу комерційної діяльності в області роздрібної торгівлі товарами харчового призначення є актуальною задачею на сучасному етапі. Особливо великою є потреба в автоматизації діяльності невеликих торгових точок, що займаються спеціалізованим продажем товарів певних категорій [2].

Використання електронного обліку дозволить автоматизувати багато процесів, що дозволить пришвидшити та спростити роботу магазину. Електронний облік полегшить процес документування процесів, дозволить з легкістю виконувати процедуру додавання нового продажу, ревізії, закупівлі, редагувати дані і т.д.. Також дозволить без додаткових економічних витрат організувати зберігання і використання існуючої інформації, а також автоматизувати дії по створенню нових даних. Дозволить легко отримувати потрібну інформацію.

Розділ 1

Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Комерція є різновидом підприємницької діяльності, дохід від якої утворюється за рахунок посередницької або торгівельної діяльності [1].

Роздрібною торгівлею являється продаж товарів і послуг, які доступні всім покупцям, в торгових центрах, салонах, через інтернет-сервіси. Здійснення комерційної діяльності є найважливішим моментом. Ще особливістю можна назвати наявність певного місця, в якому відбувається товарно-грошовий обмін, у більшості випадків це магазин, але при реалізації продукції через сайт або телефон – власні адреси замовників або приміщення, де проводиться видача замовлення [2].

Продаж ХТ – це продаж товарів, які являють собою харчові продукти в натуральному або переробленому вигляді та призначені для використання людиною в якості їжі.

ХТ поділяються за сировиною або за призначенням на 9 основних груп:

- зерно борошняні;
- плодоовочеві;
- цукор, крохмаль, мед і кондитерські вироби;
- смакові;
- харчові жири і концентрати;
- яєчні продукти;
- м'ясні;
- рибні;
- молочні.

Згідно до торгової класифікації розрізняють наступні групи товарів:

- хлібобулочні;
- кондитерські;
- плодоовочеві;

- чай;
- кава;
- соки;
- води;
- вино-горілчані;
- м'ясні;
- рибні;
- молочні;
- махорочні вироби.

Окрім того, у торгівлі ХТ можуть розподілятися на бакалійну і гастрономічну групу.

До складу бакалійної групи входять такі товари як: крупа, борошно, макаронні вироби, чай, кава, сіль, цукор, рослинна олія, прянощі та інші товари.

До гастрономічної групи входять продукти, які вже готові до вживання – ковбаси, вотчинні вироби (м'ясна гастрономія) також риба копчена, в'ялена, запечена, баликові вироби (рибна гастрономія), масло, сметана, сири (молочна гастрономія) і консерви. У межах групи, згідно до стандарту, товари в залежності від походження або виробництва розподіляються по видах, типах і різновидам і в залежності від якісних показників сортам [3].

Однонасінний, сухий сункарпний плід рослин, в основному чагарників і дерев називається горіхом [4]. До основних різновидів горіхів відносяться:

- з сімейства Горіхових: горіхи волоський, чорний, маньчжурський, карія, гіркий;
- з сімейства Березових: ліщина, фундук, турецька горіх;
- з сімейства Букових: каштан, чінарікі, жолудь [5].

Спеції використовуються як регулятор інтенсивності смаку і консистенції страви. Їх додають у їжу у процесі приготування для того, щоб посилити смак, зробити його солоним, солодким, гострим. Цукор, сіль, чорний мелений перець і перець чилі, сода, гірчиця, хрін, лимонна кислота – це також спеції.

Прянощами є будь-яка духмяна трава, яка росте. Крім апетитного запаху, прянощі також можуть надавати їжі терпкого, пекучого або гіркого смаку. Цибуля, часник, селера, фенхель, пастернак, їх також можна віднести до прянощів.

Приправа є вже більш широким поняттям. В складі приправи можуть бути і спеції, і прянощі, і ароматизатори, і емульгатори, і т.п. [6].

Сухофрукти – висушені фрукти або ягоди, з залишковою вологістю близько 20%. Вони сушаться природним шляхом (наприклад, на сонці), або промисловими методами (наприклад, за допомогою дегідратора) [7].

Асортимент сухофруктів можна порівняти з кількістю свіжих фруктів. Основні групи ознак, за якими умовно можна класифікувати всі сушені фрукти, характеризуються такими критеріями:

1. За приналежністю до регіону: місцеві і екзотичні.
2. За способом підготовки вихідної сировини: цільні, половинчасті (різані або рвані), подрібнені, а також висушені з кісточками або без них.
3. За сортовою приналежністю товару: екстра, вищий, перший і столовий.
Способом сушки: природна або штучна.
4. За способом приготування: оброблені попередньо і необроблені.

Також всі сухофрукти можна розділити на такі види: кісточкові, сім'янки і ягоди. Відповідно до класифікатора до сухофруктів також відносять натуральну пастилу, а також деякі різновиди турецького щербета і цукати [8].

Основними процесами у роботі торгової точки є продаж, ревізія та закупівля товарів. Продажем або реалізацією товарів називаються будь-які операції, які здійснюються згідно з договорами купівлі-продажу, обміну, поставки та іншими господарськими або цивільно-правовими договорами, які передбачають передачу прав власності на такі товари за плату або компенсацію незалежно від строків її надання, а також операції з безоплатного надання товарів [9].

Продаж здійснюється в різноманітних формах (див. рис. 1.1) в залежності від багатьох чинників: місце зустрічі, розмір торговельної площі та місце продажу, відстань між продавцем та покупцем, характер контакту тощо [10].

Місце зустрічі			
Покупець іде до продавця		Продавець іде до покупця	
Розмір торговельної площі та місце продажу		Відстань між продавцем і покупцем	
Мала площа	Середня та велика торговельна площа	На відстані	Близька, сам на сам
Контакт анонімний			
Продаж на базарі, виставці, ярмарковий продаж	Продаж у магазині самообслуговування, супер- та гіпер-маркеті, з торгів	Посилкова торгівля, «телепродаж»	Продаж колективний або груповий
Контакт особистий			
Продаж у роздрібному магазині	Продаж з прилавків	Угоди між приватними особами, продаж за телефоном	Прямий продаж за місцем перебування покупця, за місцем проживання або роботи
Тип клієнта			
Кінцевий споживач, купує для особистого некомерційного використання		Організований (професійний) споживач, купує для професійного використання або подальшого перепродажу	

Рисунок 1.1 – Форми продажу [10]

Ревізією називають форму документального контролю за фінансово-господарською діяльністю підприємства, дотриманням законодавства з фінансових питань, достовірністю обліку і звітності. Також це спосіб документального викриття нестач, розтрат, привласнень чи крадіжок коштів і матеріальних цінностей, попередження фінансових зловживань. Ревізія – як різностороння перевірка діяльності підприємства для встановлення законності, доцільності й ефективності дій [11].

Закупівлі (закупівля) – це процес придбання товарів замовником: спочатку визначення потреби, пошук і вибір постачальника, потім підписання контракту, доставка товару. Закупівлі є невід’ємною складовою частиною процесу логістики. Використовується ще термін постачання, який є ширшим терміном власне за закупки. Таким чином, закупівельна логістика охоплює весь рух товарів, які отримуємо від постачальників, тобто, складування та перевезення [12].

Визначена кількість продукції, що має однакову назву та властивості та вироблена за однакових умов з однієї сировини, утворює партію продукції.

Основою документації закупівель є накладна, документ, який використовується при роботі з вантажем, а саме при транспортуванні вантажів. Вона регулює відносини між перевізником, відправником та одержувачем вантажу; оформляє та засвідчує договір перевезення вантажу [13].

Залежно від способу, форми і види передачі товарно-матеріальних цінностей є такі різновиди накладних:

1. Товарна накладна – призначена для оформлення операцій з відпуску та прийому товарів зі складу.

2. Транспортна накладна – призначена для перевезення вантажів. Регулює відносини між учасниками перевезення вантажів: перевізником, відправником та одержувачем вантажу [14].

Будь-які дії з товаром такі як переміщення та списання товару називається рухом товару.

Основними посадами у торговій точці є системний адміністратор, продавець, адміністратор, керуючий, комірник і прибиральниця.

Адміністратор торгової точки контролює правильне оформлення торговельного залу, викладання товарів, наявність цінників, додержання правил торгівлі, консулює покупців з різноманітних питань, вживає заходів щодо прискорення обслуговування покупців і т.п. [15].

Системний адміністратор підтримує правильну роботу комп'ютерної техніки і програмного забезпечення, а також відповідає за інформацію безпеки організації.

Продавець зобов'язаний приймати товар, перевіряти заповнення тарних одиниць, їх кількість та вагу, розміщувати і викладати товар за групами, видами та сортами з дотриманням правил товарного сусідства, готувати робоче місце до роботи, а саме встановлювати і перевіряти справність ваговимірювальних приладів, реєстраторів розрахункових операцій, виробничого інвентарю, інструментів, також консулювати покупців про властивості, смакові особливості, кулінарне призначення та харчову цінність товарів, підраховувати гроші, здавати їх у встановленому порядку [16].

Керуючий організовує роботу ТТ, в тому числі встановлює графік, складання та нормування робочого розпорядку, визначення вихідних і святкових днів, контролює отримання законодавчих вимог до функціонування магазину, подача та оформлення необхідних документів, забезпечення наявності і функціонування необхідного торгового обладнання, засобів вимірювання, касових апаратів, терміналів і т.д., також він веде переговори з постачальникам, клієнтами, організовує і проводить ділові зустрічі, презентації, укладає договори купівлі-продажу і т.п. [17].

Комірник займається організацією, прийманням, зберіганням та видачою різних матеріальних цінностей на складі. А також веде постійний облік переміщення товару, стежить за правильністю оформлення заявок на ввезення і вивезення продукції. Ще комірник регулярно проводить інвентаризацію товару, виявляє його брак, визначає дефекти і списує його [18].

Використання електронного обліку дозволить без додаткових економічних витрат організувати зберігання й використання існуючої інформації, а також автоматизувати дії по створенню нових даних. Дозволить отримувати потрібну інформацію. Також дозволить автоматизувати ряд процесів, що дозволить пришвидшити і спростити роботу магазину. Електронний облік полегшить процес документування процесів, дозволить з легкістю виконувати процедуру додавання нового закупівлі, продажу, ревізії, редагувати дані тощо.

1.2 Аналіз існуючого програмного забезпечення предметної області

Для автоматизації діяльності магазинів з продажу харчових товарів використовуються різні види ІС, які можна розділити на наступні категорії:

- комплексні системи автоматизації діяльності;
- системи бухгалтерського обліку;
- спеціальні ПП;
- ПП для автоматизації діяльності невеликих торгівельних точок з продажу харчових товарів.

Система «Nutsbox» є характерний представником комплексних систем автоматизації діяльності. Інтернет-магазин з продажу горіхів цукатів і сухофруктів передбачає перегляд каталогу товарів (рис. 1.2), додавання товарів у кошик, замовлення товарів та перегляд контактних даних [19].

Основними перевагами даного інтернет-магазину є можливість перегляду наявних товарів, можливість придбати товар та можливість авторизації користувача. Недоліком є не зовсім зручний інтерфейс.

Система «Акцент» (рис. 1.3) є характерним представником систем бухгалтерського обліку [20].

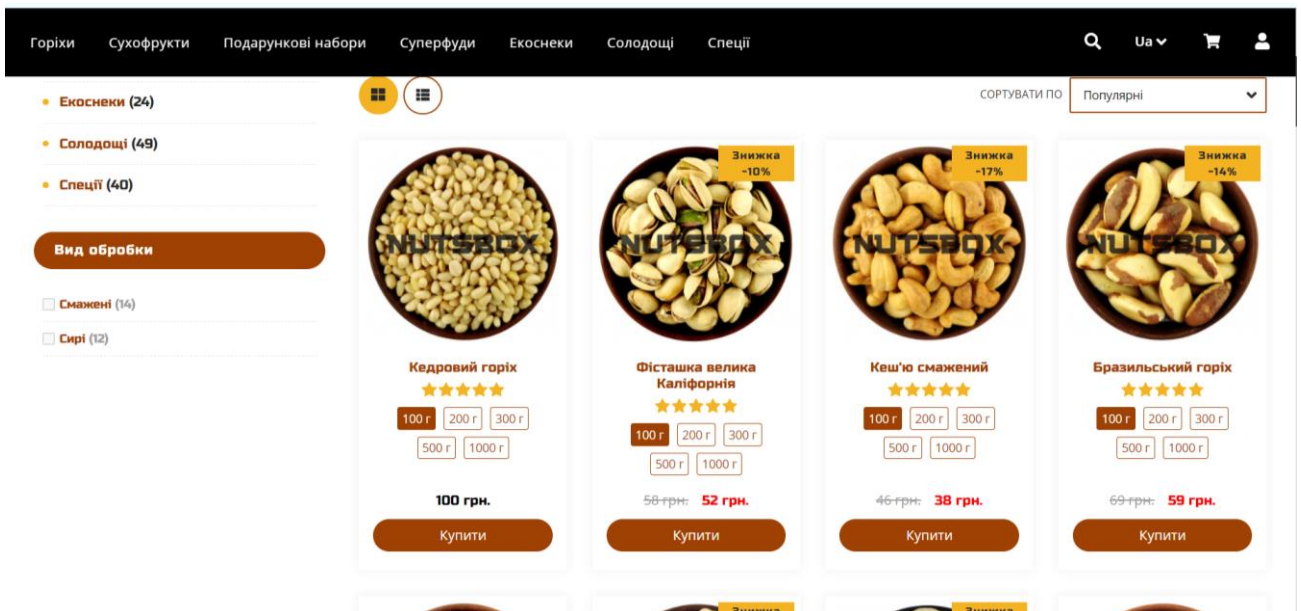


Рисунок 1.2 – Інтерфейс системи «Dream&Dress» [19].

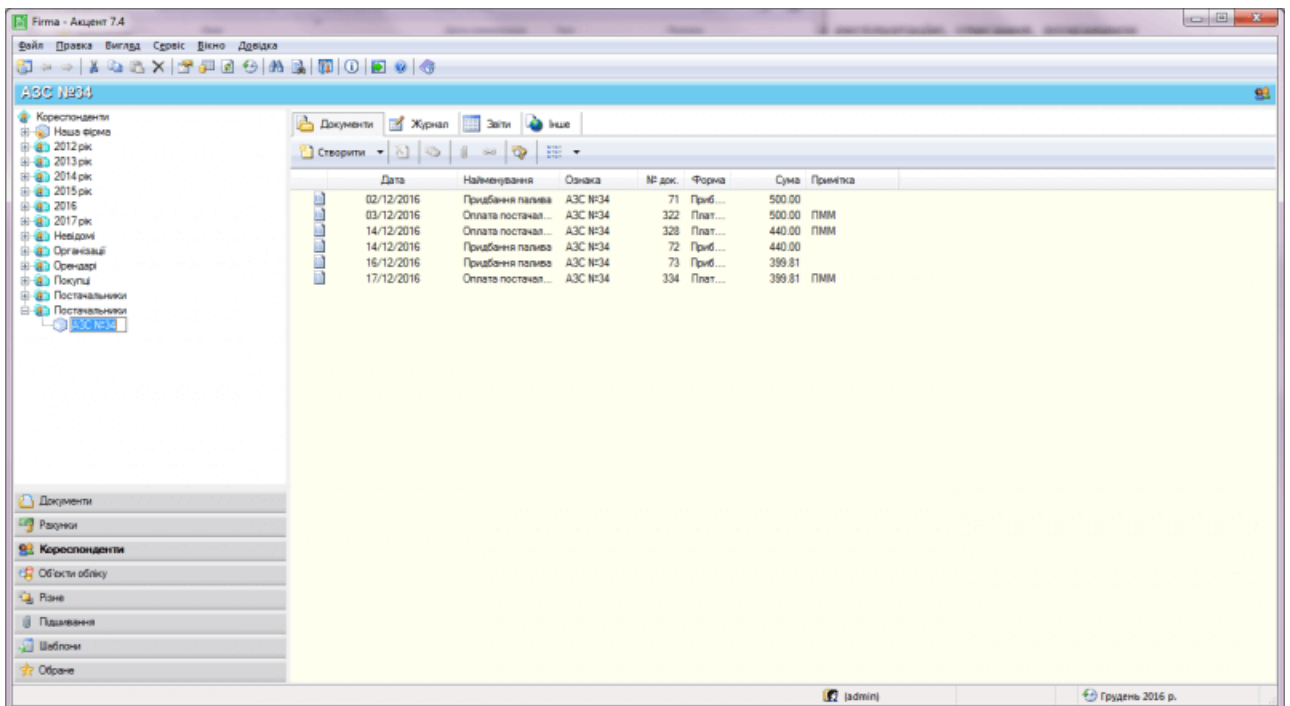


Рисунок 1.3 – Інтерфейс програми «Акцент» [21].

Можливості системи «Акцент»:

1. Облік грошових коштів: каса, розрахункові рахунки.
2. Рух необоротних активів: надходження, введення в експлуатацію, списання, розрахунок амортизації.
3. Облік ПДВ: реєстрація ПН/РК, формування декларації з ПДВ.

4. Фінансова звітність.

5. Розрахунок зарплати: таблиць, нарахування погодинної зарплати, надбавок та доплат, утримання, нарахування на ФОП, звітність ЄСВ, 1ДФ, ФСС.

6. Розрахунок за середньою заробітною платою: відпускні, лікарняні, компенсація за відпустку.

Переваги: зручний інтерфейс, легке освоєння програми, додавання довільної аналітики в шаблон господарських операцій, зручний механізм побудови власних звітів, вивантаження в Excel, можливість вести кілька підприємств в одній базі [21].

Недоліки: необхідність використання програми-«оболонки», тому необхідність додаткових витрат [22].

Система «ПАРУС – Бухгалтерія» (рис. 1.4) є характерний представником систем бухгалтерського обліку.

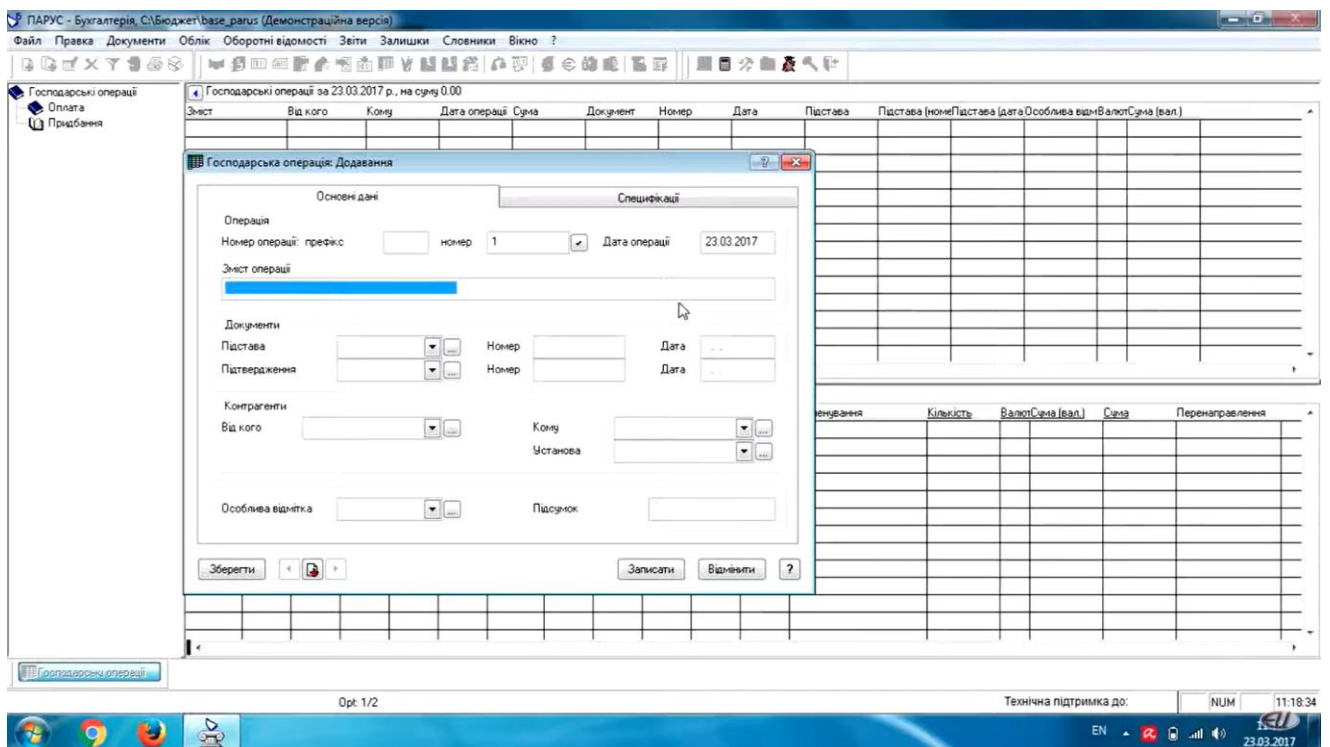


Рисунок 1.4 – Інтерфейс програми «ПАРУС – Бухгалтерія» [22]

Програма «ПАРУС – Бухгалтерія» призначена для автоматизації ведення бухгалтерського обліку в бюджетних установах будь-якого рівня. Вона забезпечує: формування повної і достовірної інформації про фінансово-

господарську діяльність, для потреб внутрішніх і зовнішніх користувачів та контрагентів; реєстрацію первинних документів та підготовку річної, квартальної, місячної звітності бюджетної установи.

Перевагами є доступність і можливість автоматизації підприємства поетапно у такій послідовності: склад, облік кадрів, нарахування заробітної плати, бухгалтерський облік, друк і зчитування штрих кодів тощо. Головним недоліком платформи «ПАРУС – Бухгалтерія» є те, що при веденні бухгалтерського обліку витрат на якість не має можливості здійснення аналітичних характеристик бухгалтерського рахунку не створюючи зайвих аналітичних рахунків, але даний елемент замінюється деталізацією синтетичного рахунку бухгалтерського обліку за допомогою аналітичних рахунків.

Система «Nutsking» (рис. 1.5) є характерним представником комплексних систем автоматизації діяльності. Інтернет-магазин з продажу горіхів і сухофруктів передбачає перегляд каталогу товарів [23].



Рисунок 1.5 – Інтерфейс системи «Nutsking» [23]

Дана система має доволі приємний інтерфейс, можливість авторизації користувача, можливість змінити мову сайту, а також оформлення замовлення товарів та можливість зв'язатися із консультантом.

Таким чином виявлені переваги та недоліки спонукають до розробки АІС, в якій буде приємний інтерфейс, автоматизація різних процесів для клієнта та адміністраторів системи та найголовніше те, що вона буде безкоштовна.

1.3 Аналіз сучасних засобів створення програмного забезпечення

Для розробки АІС доцільним буде проаналізувати сучасні засоби створення програмного забезпечення. Найбільш популярні програмні платформи:

- .NET Framework;
- PHP;
- Java.

Перевагами платформи .NET Framework є можливість розробляти ІС на найбільш підходящій мові програмування, компонентно-орієнтований підхід до проектування та реалізації ІС, багаторівнева, гнучка і надійна політика безпеки, інтеграція з технологією веб-сервісів, спрощення процедури розгортання і використання ІС. Недоліками даної платформи є високі вимоги до апаратного забезпечення, документація по ряду нових функцій ІС представлена не в зовсім повному обсязі, обмежена підтримка ряду теоретично цікавих і практично корисних мов програмування [24].

Платформа PHP має наступні переваги:

- Традиційність, синтаксис і конструкції PHP включають багато елементів, які є у мовах програмування C, Perl, PHP є мовою з універсальним синтаксисом і водночас пристосована до веб-програмування;
- Простота, PHP може бути вбудована безпосередньо в html-код сторінок, які коректно обробляються PHP-інтерпретатором;
- Засоби безпеки системного рівня;
- Засоби безпеки рівня програми;
- Гнучкість, PHP використовується не лише у поєднанні з HTML, але й із JavaScript, WML, XML та іншими мовами програмування. PHP-код може

передаватися будь-яким браузером і пристроям, в тому числі стільниковим телефонам, портативним комп'ютерам. PHP-код можна виконувати в режимі командного рядка;

– PHP працює на різних web-серверах (Apache, Netscape Enterprise Server, Microsoft IIS, Stronghold, Zeus) і платформах (UNIX, Solaris, FreeBSD, Windows 95/98/NT/2000/XP/2003) [25].

PHP має інтерфейси до багатьох баз даних:

– У PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, SQLite, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase.

– Завдяки стандарту відкритого інтерфейсу зв'язку з базами даних можна підключатися до всіх баз даних, до яких існує драйвер.

Ефективність є дуже важливим чинником у програмуванні для середовищ розрахованих на багато користувачів, до яких належить і web. Важливою перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програми, написані на Perl. Проте хоч би що робили розробники PHP, виконавчі файли, отримані за допомогою компіляції, працюватимуть значно швидше – в десятки, а іноді і в сотні разів. Але продуктивність PHP достатня для створення цілком серйозних веб-застосунків [26].

Недоліками є незручний дизайн мови, змінні з символом «\$», відсутність підтримки Unicode у версіях до 6.0 та непередбачуваність нових версій.

Перевагами платформи Java є об'єктно-орієнтоване програмування, мова – високорівнева із простим синтаксисом, безпека, кросплатформеність, автоматичне управління пам'яттю, багатопоточність і стабільність у суспільстві, але також є і недоліки: платне комерційне використання, низька продуктивність, багатослівний код та відсутність нативного дизайну [27].

Зважаючи на усі характеристики вище наведених платформ, для розробки АІС торгової точки з роздрібного продажу ХТ було обрано платформу PHP.

1.4 Постановка задачі та вимоги до розробки інформаційної системи

Метою КРБ є розробка АІС торгової точки з роздрібного продажу харчових товарів на платформі РНР, що призначена для автоматизації роботи трьох видів користувачів – клієнта торгової точки (в рекламних цілях), адміністратора торгової точки та адміністратора системи.

Для *клієнта* необхідно автоматизувати виконання наступних функцій:

- перегляд даних про торгові точки (місцезнаходження, контакти);
- перегляд каталогу товарів (прайсу);
- перегляд статей.

Для *адміністратора торгової точки* необхідно автоматизувати виконання наступних функцій:

- авторизація (реєстрація, авторизація);
- продажі (перегляд, додавання, формування чеку);
- закупівлі (перегляд, додавання, формування накладної);
- облік товарів (рух товару, ревізії);
- каталог товарів (перегляд, додавання, категорії);
- персонал (перегляд, реєстрація, редагування).

Для *адміністратора системи* необхідно автоматизувати виконання наступних функцій:

- доступ до всіх даних системи (перегляд, додавання, редагування);
- активація облікових записів адміністраторів торгової точки.

Розділ 2

Проектування інформаційної системи

2.1 Функціональна структура та бізнес-процеси системи

АІС торгової точки з роздрібного продажу харчових товарів призначена для автоматизації роботи трьох видів користувачів – клієнта торгової точки, адміністратора магазину та адміністратора системи.

Для клієнта торгової точки (в рекламних цілях) необхідно автоматизувати бізнес-процеси «Перегляд даних торгових точок клієнтом», «Перегляд каталогу товарів клієнтом», «Перегляд статей клієнтом», що вимагає автоматизації наступних функцій:

- перегляд даних про місцезнаходження торгових точок (адреси);
- перегляд даних про контакти торгових точок (електронна пошта, номер телефону);
- перегляд каталогу товарів (прайсу);
- перегляд статей.

Для адміністратора торгової точки необхідно автоматизувати наступні бізнес-процеси: «Авторизація адміністратора торгової точки», «Робота з продажами», «Робота із закупівлями», «Робота з обліком товарів», «Робота з каталогом товарів», «Робота з даними персоналу».

Для системного адміністратора торгової точки необхідно автоматизувати наступні бізнес-процеси: «Робота з даними системи», «Активація облікових записів».

Бізнес-процес «Перегляд даних торгових точок клієнтом» призначений для ознайомлення клієнтом з адресами та контактами торгових точок, що дозволяє автоматизувати виконання наступних функцій:

- перегляд даних про адреси торгових точок;
- перегляд даних про електронні пошти та контактні номери телефонів торгових точок.

Бізнес-процес «Перегляд каталогу товарів клієнтом», призначений для ознайомлення клієнтом з наявними товарами у торгових точках, що дозволяє автоматизувати виконання функції перегляду каталогу товарів.

Бізнес-процес «Перегляд статей клієнтом», призначений для прочитання клієнтом інформаційних статей, що дозволяє автоматизувати виконання функції перегляду статей.

Бізнес-процес «Авторизація адміністратора магазину», призначений для авторизації адміністратора торгової точки, що дозволяє автоматизувати виконання наступних функцій:

- реєстрація;
- авторизація.

Бізнес-процес «Робота з продажами», призначений для контролю продажів адміністратором ТТ, що дозволяє автоматизувати виконання наступних функцій:

- додавання продажу;
- перегляд продажів;
- формування чеку.

Бізнес-процес «Робота із закупівлями», призначений для контролю закупівель адміністратором торгової точки, що дозволяє автоматизувати виконання наступних функцій:

- додавання закупівлі;
- перегляд закупівель;
- формування накладної.

Бізнес-процес «Робота з обліком товарів», призначений для контролю обліку товарів адміністратором торгової точки, що дозволяє автоматизувати виконання наступних функцій:

- додавання ревізії;
- перегляд ревізій;
- додавання руху товарів;
- перегляд руху товарів.

Бізнес-процес «Робота з каталогом товарів», призначений для контролю каталогу товарів адміністратором торгової точки, що дозволяє автоматизувати виконання наступних функцій:

- додавання товару;
- перегляд товарів.

Бізнес-процес «Робота з даними персоналу», призначений для контролю даних персоналу адміністратором торгової точки, що дозволяє автоматизувати виконання наступних функцій:

- перегляд даних персоналу;
- реєстрацію звільнення працівника;
- редагування даних персоналу.

Бізнес-процес «Робота з даними системи», призначений для контролю даних системи системним адміністратором торгової точки, що дозволяє автоматизувати виконання наступних функцій:

- додавання даних;
- редагування даних;
- перегляд даних.

Бізнес-процес «Активація облікових записів», призначений для надання або заборони доступу до системи працівникам адміністратором торгової точки, що дозволяє автоматизувати виконання функції активації облікових записів працівників магазину.

Таким чином, для автоматизації роботи трьох видів користувачів АІС торгової точки з роздрібного продажу харчових товарів (клієнта, адміністратора магазину та адміністратора системи) було автоматизовано відповідні бізнес-процеси.

Таблиця «Products» (таблиця 2.1), призначена для збереження інформації про товари (назву, тип товару, тип одиниці, опис, ціна однієї одиниці, фото).

Таблиця 2.1 – Атрибути таблиці «Products»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Varchar(255)	Назва товару
3.	product_type_id	Int (255)	Тип товару
4.	units_type	Int (255)	Тип одиниці
5.	description	Varchar(255)	Опис
6.	one_unit_price	double	Ціна однієї одиниці
7.	image	Varchar(255)	Фото

Таблиця «Units_types» (таблиця 2.2), призначена для збереження інформації про типи одиниць товарів (назву).

Таблиця 2.2 – Атрибути таблиці «Units_types»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Varchar(255)	Назва типу одиниці товару

Таблиця «products_types» (таблиця 2.3), призначена для збереження інформації про типи товарів (назву, фото).

Таблиця 2.3 – Атрибути таблиці «products_types»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Varchar(255)	Назва типу товару
3.	image	Varchar(255)	Фото

Таблиця «sales_of_one_products» (таблиця 2.4), призначена для збереження інформації про продажі одного товару(продаж, партію, число одиниць, ціну).

Таблиця 2.4 – Атрибути таблиці «sales_of_one_products»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	sale_id	Int (255)	Продаж
3.	consignment_id	Int (255)	Партія
4.	number_of_units	double	Число одиниць
5.	price	double	Ціна

Таблиця «supplies» (таблиця 2.5), призначена для збереження інформації про поставки (дата й час, ім'я працівника, ТТ, постачальника, суму).

Таблиця 2.5 – Атрибути таблиці «supplies»

№ п/п	Назва атрибуту	Тип даних	Опис
6.	ID	Int (255)	Числовий ідентифікатор
7.	date_time	datetime	Дата й час
8.	storekeeper_id	Int (255)	Ім'я працівника
9.	shop_id	Int (255)	ТТ
10.	supplier_id	Int (255)	Постачальник
11.	sum	double	Сума

Таблиця «suppliers» (таблиця 2.6), призначена для збереження інформації про постачальників (назву, адресу, номер телефону).

Таблиця «cash_registers» (таблиця 2.7), призначена для збереження інформації про каси (назву, ТТ).

Таблиця 2.6 – Атрибути таблиці «suppliers»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Varchar(255)	Назву
3.	adress	Varchar(255)	Адресу
4.	phone_number	Varchar(255)	Номер телефону

Таблиця 2.7 – Атрибути таблиці «cash_registers»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Varchar(255)	Назва типу товару
3.	shop_id	Int (255)	ТТ

Таблиця «consignments» (таблиця 2.8), призначена для збереження інформації про партії (поставку, товар, кількість одиниць, ціну однієї одиниці, фото).

Таблиця 2.8 – Атрибути таблиці «consignments»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	suplies_id	Int (255)	Поставка
3.	product_id	Int (255)	Товар
4.	number_of_units	double	Кількість одиниць
5.	one_units_price	double	Ціна однієї одиниці
6.	image	Varchar(255)	Фото

Таблиця «revision_of_one_consignment» (таблиця 2.9), призначена для збереження інформації про ревізію однієї партії (ревізію, партію, кількість).

Таблиця 2.9 – Атрибути таблиці «revision_of_one_consignment»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	revision_id	Int (255)	Ревізія
3.	consignment_id	Int (255)	Партія
4.	number	double	Кількість

Таблиця «customer_types» (таблиця 2.10), призначена для збереження інформації про типи клієнтів (назву, знижку).

Таблиця 2.10 – Атрибути таблиці «customer_types»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Int (255)	Назва
3.	coefficient	Decimal(3,2)	Знижка

Таблиця «sales» (таблиця 2.11), призначена для збереження інформації про типи продажі (дату й час, працівника, касу, знижку, суму, фото).

Таблиця 2.11 – Атрибути таблиці «sales»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	date_time	Datetime	Дата і час
3.	cashier_id	Int (255)	Працівника
4.	cash_register_id	Int (255)	Каса
5.	customer_type_id	Int (255)	Знижка
6.	sum	Double	Сума
7.	image	Varchar(255)	Фото

Таблиця «shops» (таблиця 2.12), призначена для збереження інформації про магазини (назву, адресу, години роботи, номер телефону, фото).

Таблиця 2.12 – Атрибути таблиці «shops»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Varchar(255)	Дата і час
3.	adress	Varchar(255)	Працівника
4.	schedule	Varchar(255)	Каса
5.	phone_number	Varchar(255)	Знижка
6.	image	Varchar(255)	Фото

Таблиця «positions» (таблиця 2.13), призначена для збереження інформації про посади (назву)

Таблиця 2.13 – Атрибути таблиці «positions»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Varchar(255)	Назва

Таблиця «revisions» (таблиця 2.14), призначена для збереження інформації про ревізії (дату і час початку, дату і час закінчення, працівника, ТТ, коментар).

Таблиця 2.14 – Атрибути таблиці «revisions»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	start_date_time	Datetime	Дата і час початку
3.	end_date_time	Datetime	Дата і час закінчення
4.	inspector_id	Int (255)	Працівник
5.	shop_id	Int (255)	ТТ
6.	comment	Varchar(255)	Коментар

Таблиця «product_movements» (таблиця 2.15), призначена для збереження інформації про рух товарів (дату й час, партію, тип руху, ТТ, працівника, опис, суму, фото).

Таблиця 2.15 – Атрибути таблиці «revisions»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	date_time	Datetime	Дата і час
3.	consignment_id	Int (255)	Партія
4.	movement_type_id	Int (255)	Тип руху
5.	shop_id	Int (255)	ТТ
6.	worker_id	Int (255)	Працівника
7.	description	Varchar(255)	Опис
8.	sum	Double	Сума
9.	image	Varchar(255)	Фото

Таблиця «products_movements_types» (таблиця 2.16), призначена для збереження інформації про типи руху товарів (назву, напрям).

Таблиця 2.16 – Атрибути таблиці «products_movements_types»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	name	Int (255)	Назва
3.	destination	Varchar(255)	Напрямок

Таблиця «workers» (таблиця 2.17), призначена для збереження інформації про працівників (ім'я, прізвище, по-батькові, посаду, ТТ, дату народження, дату прийняття на роботу, дату звільнення, попередній стаж, номер телефону, логін, пароль, фото, доступ).

Таблиця 2.17 – Атрибути таблиці «workers»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	Int (255)	Числовий ідентифікатор
2.	first_name	Int (255)	Ім'я
3.	last_name	Varchar(255)	Прізвище
4.	father_name	Varchar(255)	По-батькові
5.	position_id	Int (255)	Посада
6.	shop_id	Int (255)	ТТ
7.	date_birth	date	Дата народження
8.	date_hiring	date	Дата прийняття на роботу
9.	date_firing	date	Дата звільнення
10.	previous_experience	Int (11)	Попередній стаж
11.	phone_number	Varchar(255)	Номер телефону
12.	login	Varchar(255)	Логін
13.	password	Varchar(255)	Пароль
14.	image	Varchar(255)	Фото
15.	access	tinyint(1)	Доступ

На рисунку 2.2 наведена діаграма активності роботи користувача із системою. Для клієнта інформаційної системи потрібне автоматизоване виконання функцій: перегляд даних про торгові точки, перегляд каталогу товарів, перегляд статей.

Процес роботи користувача із сайтом включає в себе підтвердження особистості користувача, при успішній авторизації він потрапляє до системи, де він має можливість працювати з різними даними, додавати нові, а також змінювати існуючі.

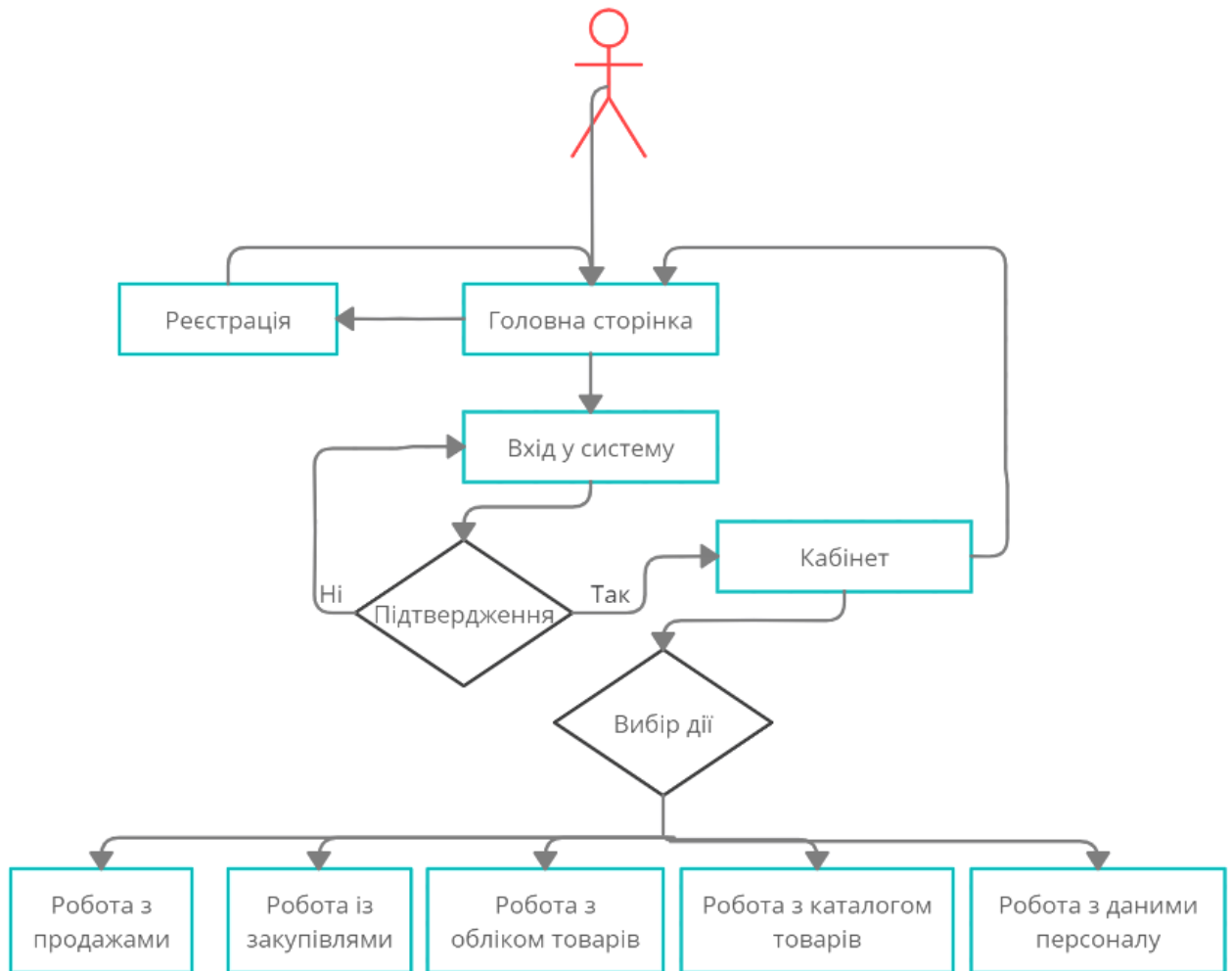


Рисунок 2.2 – Діаграма активності роботи користувача із системою

Отже, для вирішення поставленої задачі підходить наведена база даних для АІС торгової точки та діаграма активності користувача із системою. Наступним кроком було обрано засоби для розробки системи.

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

PHP (Hypertext Preprocessor) – це широко використовувана мова сценаріїв для загального призначення із відкритим вихідним кодом.

Синтаксис мови бере початок з таких мов як C, Java і Perl. PHP є досить простим для вивчення. Його перевагою також є надання веб-розробникам можливості швидкого створення динамічних веб-сторінок.

PHP відрізняється від якого-небудь коду, тим що виконується на стороні клієнта, наприклад, JavaScript, є те, що PHP-скрипти виконуються на стороні сервера. Розробник навіть може конфігурувати свій сервер таким чином, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл, чи результат виконання скрипта [27].

PHP можна використовувати при більшості операційних систем: Linux, багато варіантів Unix (включаючи HP-UX, Solaris та OpenBSD), Microsoft Windows, Mac OS X, RISC OS та, можливо, інших. Мова має також підтримку для більшості сучасних веб-серверів: Apache, IIS та багатьох інших. Окрім цього, підтримка є на будь-якому веб-сервері, що використовує бінарні коди FastCGI — клієнт-серверного протоколу взаємодії веб-сервера та програм [29].

2.3.2 Вибір фреймворку

Під час створення АІС торгової точки з роздрібного продажу харчових товарів, було проведено вибір фреймворку, який забезпечив створення необхідного веб-додатку. При виборі фреймворку пошук було зупинено на такому шаблоні як Laravel.

Laravel є фреймворком для веб-додатків, він має виразний і елегантний синтаксис. Він дозволяє спростити вирішення основних актуальних завдань, таких як аутентифікація, маршрутизація, сесії і кешування. Laravel – це спроба об'єднати все найкраще, що є в інших PHP фреймворках. Основні його переваги Laravel:

- велика екосистема з миттєвим розгортанням своєї платформи, офіційний сайт надає безліч мануалів і інформації для ознайомлення;
- документація Laravel близька до досконалості;
- у Laravel є свій движок для шаблонів Blade, «красивий» синтаксис мови, який сприяє вирішення всіх необхідних завдань, таких як аутентифікація, сесії, кешування і маршрутизація RESTful.

Незважаючи на свою молодість (перший реліз вийшов у 2011 році), це вже зовсім зрілий продукт, і, згідно з опитуванням, проведеним порталом SitePoint, цей фреймворк займає перше місце за популярністю серед розробників PHP [30].

Також перевагою Laravel є те, що сайти або веб-програми, розроблені з його використанням, не стають застарілими. Причина не є технічною, але полягає в тому, що він користується великим попитом. Це означає, що ринок завжди пропонує багато досвідчених розробників Laravel. Крім того, популярний інструмент програмування завжди матиме більшу підтримку та присутність на ринку.

Ще однією перевагою є те, що Laravel має можливості Object-Relational Mapping (ORM). Ця технологія дозволяє веб-розробникам створювати зв'язки з базами даних набагато легше. Крім того, Laravel полегшує використання бібліотек та моделей, оскільки на ньому містяться об'єктно-орієнтовані бібліотеки, що підтримуються функцією автодоповнення [31].

MVC (Model-view-controller) – це шаблон проектування додатків, при якому керуюча логіка поділена на три окремих компоненти таким чином, що модифікування одного з них дає мінімальний вплив на інші. Шаблон MVC добре застосовується при створенні складних проектів, де необхідно відокремити роботу php-програміста. MVC розділяє уявлення, дані та обробку дій користувача на три окремих компоненти:

- MVC Модель (Model). Вона надає дані (зазвичай для View), а також реагує на запити (зазвичай від контролера), змінюючи свій стан.

- MVC Вигляд (View). Відповідає за відображення інформації (призначений для користувача інтерфейс).

- MVC Контролер (Controller). Інтерпретує дані, які були введені користувачем, та інформує модель і уявлення про необхідність відповідної реакції [32].

2.3.3 Вибір СКБД

Microsoft SQL Server – система керування базами даних, яка розробляється корпорацією MS. Як сервер даних виконує головну функцію по збереженню та наданню даних у відповідь на запити інших застосунків, які можуть виконуватися як на тому ж самому сервері, так і у мережі.

Мова, що використовується для запитів – Transact-SQL, вона створена спільно MS та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів SQL із розширеннями, що використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими СКБД [33].

MySQL – вільна система керування реляційними базами даних. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця СКБД з відкритим кодом була створена як альтернатива комерційним системам. MySQL із самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL є одною із найпоширеніших СКБД. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, так як має чудову підтримку з боку різноманітних мов програмування [34].

Багато веб-додатків використовують саме MySQL в якості компонентів системи. Популярність використання веб-додатків тісно пов'язана з популярністю PHP і можливістю його роботи разом з MySQL.

Багато сайтів з високим трафіком (Flickr, Facebook, Wikipedia, Google, Nokia та YouTube) використовують MySQL для збереження даних і реєстрації користувачів.

Веб-інтерфейс для адміністрування управління базами даних MySQL написаний на мові PHP вільний веб-додаток з відкритим кодом phpMyAdmin. З допомогою phpMyAdmin через браузер користувача можна здійснювати адміністрацію сервера, запускати команди SQL та передивлятися вміст таблиці і даних [35].

Результатом виконання КРБ має бути застосунок, призначений для автоматизації роботи торгової точки з роздрібного продажу харчових товарів, що надає можливість зберігати та отримувати дані з бази даних, давати можливість здійснювати додаткові операції для зареєстрованих користувачів. Для написання ПП було обрано середовище розробки Laravel, СКБД – MySQL, мову програмування PHP, із застосуванням MVC5 шаблону. Застосування такої системи допоможе оптимізувати роботу ТТ.

Розділ 3

Програмна реалізація інформаційної системи

3.1 Структура та функціональне призначення складових системи

АІС торгової точки з роздрібного продажу харчових товарів використовує фреймворк Laravel, який реалізує шаблон проектування MVC. Суть цього шаблону полягає в розділенні програми на моделі, контролери та представлення. У моделі реалізується бізнес-логіка програми. У представленні знаходиться вивід даних системи. Контролер реалізує зв'язок між моделями та представленнями (рис. 3.1).

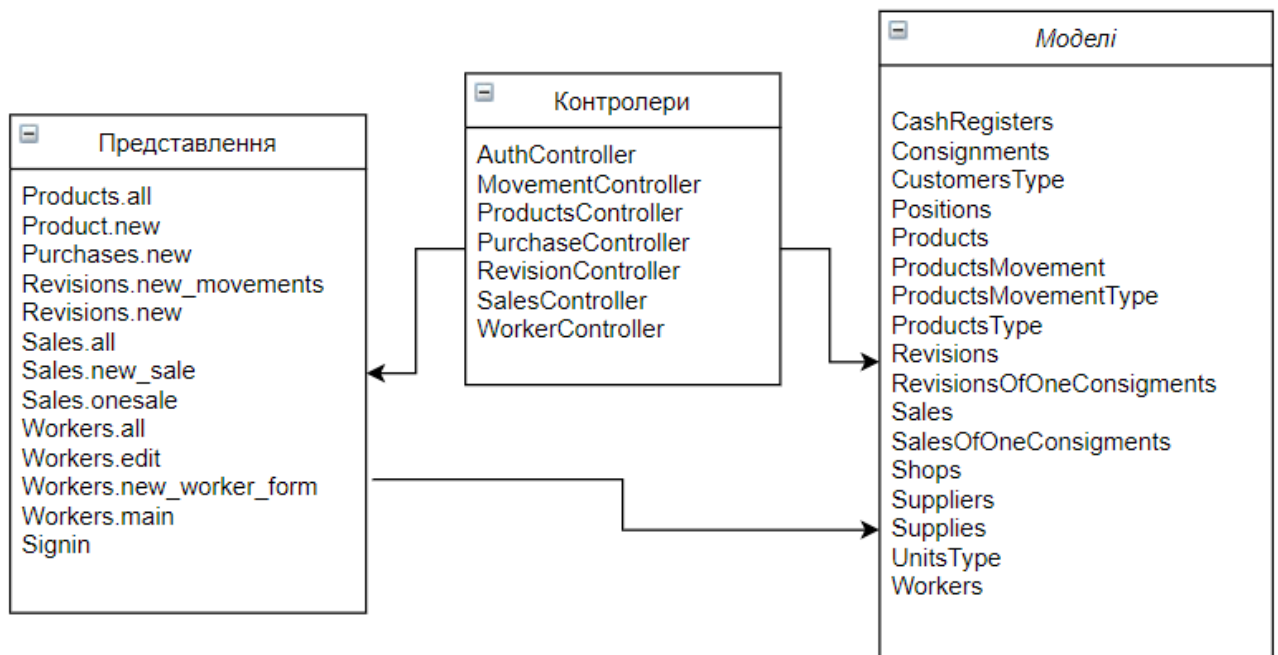


Рисунок 3.1 – Діаграма класів

Даний проєкт має наступні моделі:

- CashRegisters – є моделлю, яка відповідає за бізнес логіку, пов'язану з інформацією про каси;
- Consignments – є моделлю, яка відповідає за бізнес логіку, пов'язану з інформацією про каси;

- CustomersType – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- Positions – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- Products – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- ProductsMovement – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- ProductsMovementType – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- ProductsType – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- Revisions – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- RevisionsOfOneConsignments – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- Sales – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- SalesOfOneConsignments – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- Shops – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- Suppliers – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- Supplies – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- UnitsType – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси;
- Workers – є моделлю, яка відповідає за бізнес логіку, пов’язану з інформацією про каси.

В проєкті є наступні представлення:

- Products.all – Відповідає за вивід усіх товарів на екран;
- Product.new – Відповідає за додавання нового товару;
- Purchases.new – Відповідає за додавання нової закупівлі;
- Revisions.new_movements – Відповідає за додавання нового руху товару;
- Revisions.new – Відповідає за додавання нової ревізії;
- Sales.all – Відповідає за вивід усіх продажів на екран;
- Sales.new_sale – Відповідає за додавання нового продажу;
- Sales.onesale – Відповідає за вивід інформації про один продаж на екран;
- Workers.all – Відповідає за вивід інформації про усіх працівників на екран;
- Workers.edit – Відповідає за редагування даних працівника;
- Workers.new_worker_form – Відповідає за реєстрацію працівника;
- Workers.main – Відповідає за головну сторінку;
- Signin – Відповідає за вивід форми для входу в систему.

Також у системі є наступні функціональні контролери:

- AuthController – відповідає за автентифікацію користувачів;
- MovementController – відповідає за рух товарів;
- ProductsController – відповідає за товари;
- PurchaseController – відповідає за закупівлі;
- RevisionController – відповідає за ревізії;
- SalesController – відповідає за продажі;
- WorkerController – відповідає за дані працівників.

3.2 Особливості реалізації складових системи

Найбільш важливими частинами алгоритмів та програмного коду компонентів програмного продукту є функції з наступним призначенням:

- авторизація та аутентифікація користувача в системі;
- отримання даних таблиці з врахуванням зв'язаних таблиць, а також отримання даних для підстановки у зв'язані поля;
- додавання даних у таблиці та їх редагування.

Авторизація користувача проходить наступним чином. Спочатку користувач заповнює реєстраційну форму (рис. 3.2).

The screenshot shows a registration form for an employee. At the top, there is a green header with the text 'Реєстрація працівника'. Below the header, a note indicates that fields with an asterisk are mandatory. The form includes a profile picture selection area with a 'Browse' button. The registration details are as follows:

Field Name	Input Type
Логін*	Text input (with validation: лат. символи і цифри)
Пароль*	Text input (with validation: мін. 6 символів) and a 'Показати' button
Прізвище*	Text input
Ім'я*	Text input
По-батькові*	Text input
Дата народження*	Date picker (format: dd.mm.yyyy)
Номер телефону*	Text input (format: +000(00)0000000)
Попередній стаж	Text input (format: Років)
Дата прийняття на роботу*	Date picker (format: dd.mm.yyyy)
Дата звільнення	Date picker (format: dd.mm.yyyy)
Магазин*	Dropdown menu (option: Вибрати)
Посада*	Dropdown menu (option: Вибрати)

At the bottom of the form, there is a green button labeled 'Зареєструватись'.

Рисунок 3.2 – Форма реєстрації користувача

У випадку успішної реєстрації на екран виводиться повідомлення про успішну реєстрацію, користувач матиме доступ до системи лише після того як системний адміністратор надасть доступ.

Реєстрація користувача формується наступним програмним кодом:

```

public function postSignUp(Request $request){
    if(Auth::check()){
        $worker_name = Auth::user()->getName();
        $userimage = Workers::where('id',$worker_id)->first()->image;
    }
    else{
        $userimage = 'uploads.workers/default-user-image.png';
        $worker_name = 'Ім`я працівника';
    }
    $now_date = date("d.m.Y");

    $date = date("d.m.Y", mktime(0, 0, 0,date('d'), date('m'), date('Y')-18));
    $shop_id = Shops::where('name',$request->input('shop_id'))->first()->id;
    $position_id = Positions::where('name',$request->input('position_id'))->first()-
>id;

    $worker=new Workers();
    $worker->first_name=$request->input('first_name');
    $worker->last_name=$request->input('last_name');
    $worker->father name=$request->input('father name');
    $worker->position_id= $position_id;
    $worker->shop_id= $shop_id;
    $worker->date_birth=$request->input('date_birth');
    $worker->date_hiring=$request->input('date_hiring');
    $worker->date_firing=$request->input('date_firing');
    $worker->previous_experience=$request->input('previous_experience');
    $worker->phone_number=$request->input('phone_number');
    $worker->login=$request->input('login');
    $worker->password=bcrypt($request->input('password'));
    if ($request->has('image')) {
        $path = $request->file('image')-
>store('uploads.workers','public','worker_name');
        $worker->image = $path;}
    $worker->save();
    return redirect()
        ->route('success')
        ->with('info','Реєстрація працівника пройшла успішно');
}

```

Аутифікація користувача формується наступним програмним кодом:

```

public function postSignIn(Request $request){

    $validation = $request->validate([
        'login' => 'required|alpha-dash|max:20',
        'password' => 'required|min:6'
    ]);

    $login = $request->input('login');
    $password = $request->input('password');
    if(!Auth::attempt(['login' => $login, 'password' => $password, 'access' =
> 1])){

        return redirect()->back()-
>with('infolog','Неправильний логін або пароль');
    }
    return redirect()->route('newsale');
}

```

Наступним буде лістинг методу для отримання даних з таблиці з врахуванням зв'язаних таблиць, а також отримання даних для підстановки у зв'язані поля.

```

public function show()
{
    if(Auth::check()){
        $worker_name = Auth::user()->getName();
        $worker_id = Auth::user()->getWorkerId();
        $userimage = Workers::where('id',$worker_id)->first()->image;
    }
    else{
        $userimage = 'uploads.workers/default-user-image.png';
        $worker_name = 'Ім`я працівника';
    }
    $workers = Workers::all();
    $positions = Positions::all()->pluck('name');
    $shops = Shops::all()->pluck('name');
    return view('inc.workers.all', compact('workers','shops','positions','worker_name','userimage'));
}

```

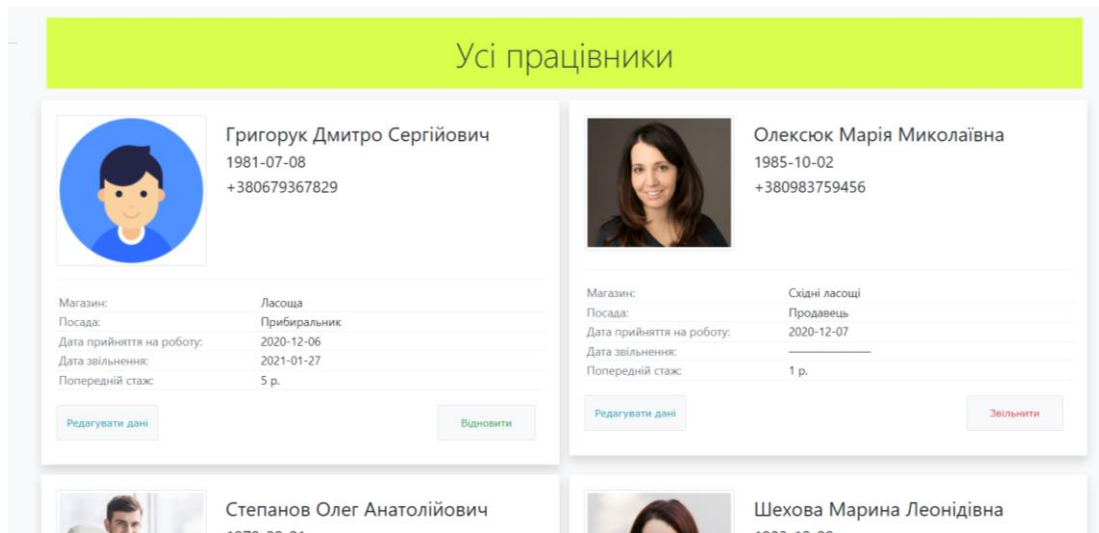


Рисунок 3.3 – Вивід даних усіх працівників

Результат виконання програмного коду зображений на рисунку 3.3.

Основними функціями програмного продукту є додавання даних у таблицю та їх редагування. Додавання даних формується наступним кодом:

```
public function postNew(Request $request)
{
    if(Auth::check()){
        $worker_name = Auth::user()->getName();
        $worker_id = Auth::user()->getWorkerId();
        $userimage = Workers::where('id',$worker_id)->first()->image;
    }
    else{
        $userimage = 'uploads.workers/default-user-image.png';
        $worker_name = 'Ім`я працівника';
    }
    $prodtype = ProductsType::all();
    $nitytype = UnitsType::all();
    $products = Products::all();
    $product = new Products();
    $prodtype_id = ProductsType::where('name',$request->input('prod_type'))-
    >pluck('id')->first();
    $nitytype_id = UnitsType::where('name',$request->input('unit_type'))-
    >pluck('id')->first();
```

```

$product->name = $request->input('name');
$product->description = $request->input('description');
if ($request->has('image')) {
    $path = $request->file('image')->store('uploads.products',
        'public',
        $request->input('name'));
    $product->image = $path;
}
$product->product_type_id = $prodtype_id;
$product->units_type = $unitttype_id;
$product->one_unit_price = $request->input('price');
$product->save();
return view('inc.products.new',compact('userimage','worker_name','prodtype
e','unitttype'))->with('info','Товар додано');
}

```

Редагування даних формується наступним кодом:

```

public function editpost($id, Request $request){
    if(Auth::check()){
        $worker_name = Auth::user()->getName();
        $worker_id = Auth::user()->getWorkerId();
        $userimage = Workers::where('id',$worker_id)->first()->image;
    }
    else{
        $userimage = 'uploads.workers/default-user-image.png';
        $worker_name = 'Ім`я працівника';
    }
    $shop_id = Shops::where('name',$request->input('shop_id'))->first()->id;
    $position_id = Positions::where('name',$request->input('position_id'))-
>first()->id;
    $worker = Workers::find($id);
    $worker->first_name = $request->input('first_name');
    $worker->last_name = $request->input('last_name');
    $worker->father_name = $request->input('father_name');
    $worker->position_id = $position_id;
    $worker->shop_id = $shop_id;
    $worker->date_birth=$request->input('date_birth');
}

```

```

$worker->date_hiring=$request->input('date_hiring');
$worker->date_firing=$request->input('date_firing');
$worker->previous_experience=$request->input('previous_experience');
$worker->phone_number=$request->input('phone_number');
if ($request->has('image')) {
    $path = $request->file('image')->store('uploads.workers','public');
    $worker->image = $path;}
$worker->save();
return redirect()-
>route('allworkers',compact('worker_name','userimage'))->with('info', 'Дані збережено');
}

```

Результат виконання наступного коду зображений на рисунку 3.4.

The screenshot shows a web interface for editing employee data. On the left, there is a profile picture of a woman with a button labeled 'Змінити фото'. The main area is titled 'Редагування даних працівника' and contains several input fields:

- Прізвище: Шехова
- Ім'я: Марина
- По-батькові: Леонідівна
- Дата народження: 29.12.1983
- Номер телефону: +380984679730
- Магазин: Ласоща
- Посада: Адміністратор
- Дата прийняття на роботу: 09.12.2020
- Дата звільнення: dd.mm.yyyy
- Попередній стаж (років): 6

A green button labeled 'Зберегти зміни' is located at the bottom right of the form.

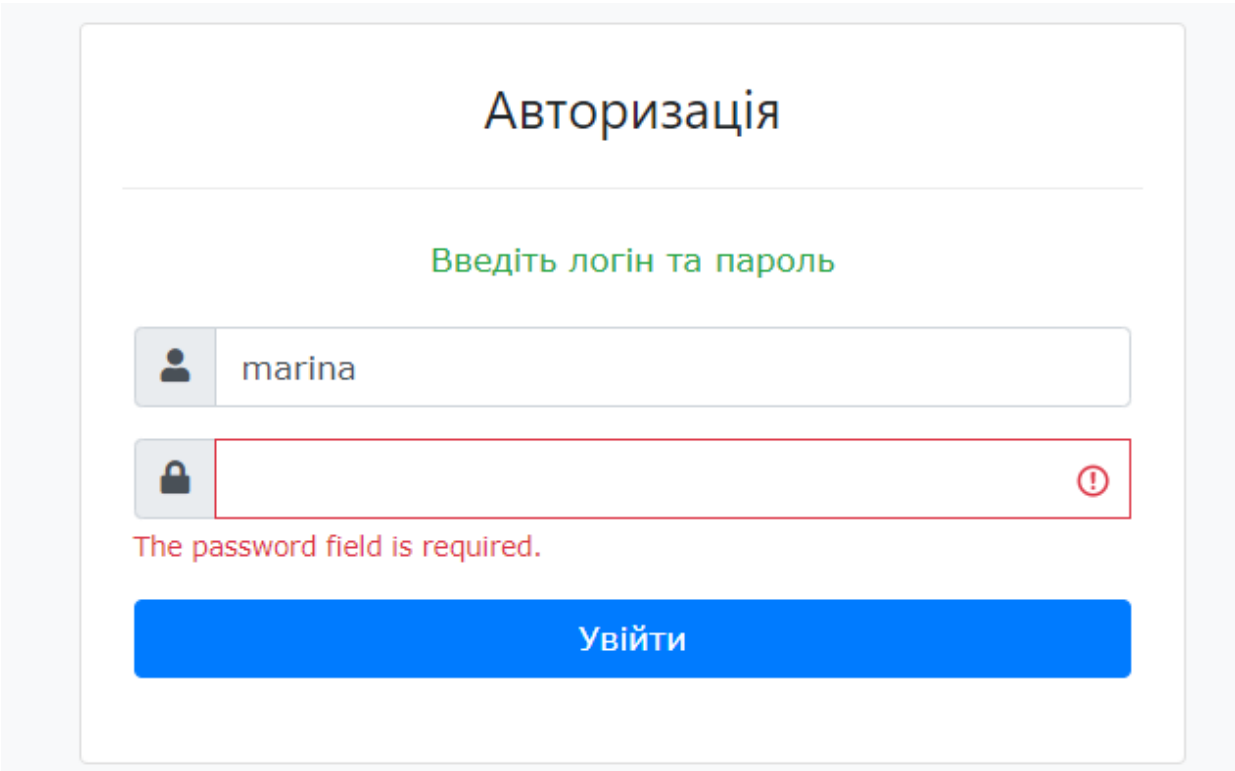
Рисунок 3.4 – Форма для редагування даних працівника

Таким чином в результаті створення АІС було реалізовано всі функції, необхідні для автоматизації роботи трьох видів користувачів – клієнта, адміністратора торгової точки та адміністратора системи.

3.3 Тестування інформаційної системи

Тестування ІС проводилось шляхом дослідження її функціональності в процесі перевірки відповідності вимог, які були заявлені до системи, до фактично реалізованої функціональності. Дослідження відбувається шляхом спостереження за процесами системи в штучно створених ситуаціях і на обмеженому наборі тестових випадків, обраних певним чином.

Для дослідження функціональності авторизації користувача був розроблений тестовий випадок (тест-кейс), де перевіряється процес входу у АІС. При вході у обліковий запис діє валідація даних. Тобто, якщо ввести значення у поле логіну, але не ввести пароль, з'явиться попередження із ввести пароль (рис. 3.5).



Авторизація

Введіть логін та пароль

marina

The password field is required.

Увійти

Рисунок 3.5 – Попередження у випадку, якщо поле не заповнено

У випадку, якщо було введено не правильне значення у полі логіну, при натисканні на кнопку «Увійти» з'явиться повідомлення із помилкою

«Неправильний логін або пароль» (рис. 3.6), при не правильному значенні паролю буде показано аналогічну помилку.

Таблиця 3.1 – Тест-кейс АТ0001

Тест-кейс ID: АТ0001	Пріоритет: 1	Створено: 23.05.2021, Д. Гаврилюк
Назва: Перевірка логіну у обліковий запис користувача Вхідні дані: Логін = «marina», Пароль = «111111»		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Запустити додаток 2. Ввести значення у поле логіну із вхідних даних 3. Натиснути кнопку «Увійти» 4. Порівняти фактичний результат з очікуваним 5. Ввести значення в поле логіну = «adm», а у поле паролю із вхідних даних 6. Натиснути кнопку «Увійти» 7. Порівняти фактичний результат з очікуваним 8. Ввести значення в поле логіну із вхідних даних, а у поле паролю = «adm» 9. Натиснути кнопку «Увійти» 10. Порівняти фактичний результат з очікуваним 11. Ввести значення в поля логіну та паролю із вхідних даних 12. Натиснути кнопку «Увійти» 13. Порівняти фактичний результат з очікуваним 		<p>З'являється попередження з вимогою заповнити пусте поле.</p> <p>Повідомлення із помилкою: «Неправильний логін або пароль».</p> <p>Повідомлення із помилкою: «Неправильний логін або пароль».</p> <p>Вікно входу у систему закрилось. Вікно для додавання нового продажу відкрито.</p>
Результат виконання тест-кейсу: пройдено успішно		


Якщо у поля логіну та паролю введено коректні дані, то при натисканні на кнопку «Увійти» вікно входу закриється та з'явиться вікно для додавання нового продажу (рис. 3.7).


Наступний тестовий випадок (таблиця 3.2) перевіряє функціонал реєстрації нового користувача. Після заповнення полей на формі та натисканні на кнопку «Зареєструватись», вікно закривається і з'являється повідомлення про успішну реєстрацію, а у базі даних з'являються дані зареєстрованого користувача без прав доступу до системи (рис. 3.8).

Авторизація

Неправильний логін або пароль

Введіть логін та пароль






Увійти

Рисунок 3.6 – Помилка у випадку, якщо логін та пароль не співпадають

Східні Ласощі

автоматизована інформаційна система торгової точки з роздрібного продажу харчових товарів



Марина Шехова
Вийти

Продажі
Закупівлі
Облік товарів
Каталог товарів
Персонал

Усі продажі

Новий продаж

Новий продаж

Торгова точка: Ласоца
Каса:
Працівник:  Марина Шехова

Назва товару	Партія	Кількість	Ціна за 1 од. (грн)
<input style="width: 90%; border: none;" type="text" value="Виберіть товар"/>	<input style="width: 90%; border: none;" type="text"/>	<input style="width: 90%; border: none;" type="text" value="0.00"/>	<input style="width: 90%; border: none;" type="text"/>
+			

Сума:
Знижка:
Сума зі знижкою:





Рисунок 3.7 – Вікно для додавання нового продажу

id	first_na	last_name	father_name	posi	shoj	date_birth	date_hiring	date_fi	previous	phone_number	login	password	image	access
2	Дмитро	Григорук	Сергійович	2	1	1981-07-08	2020-12-06	NULL	5	+380679367829	mykolagr	\$2y\$10\$bi/hNhC	uploads.work	1
3	Марія	Олексюк	Миколаївна	6	2	1985-10-02	2020-12-07	NULL	1	+380983759456	oleksiuk	\$2y\$10\$bi/hNhC	uploads.work	1
4	Олег	Степанов	Анатолійович	3	1	1979-08-01	2020-12-09	2020-1	10	+380953793748	stepanov	\$2y\$10\$Scf5v8!	uploads.work	1
5	Марина	Шехова	Леонідівна	1	1	1983-12-29	2020-12-09	NULL	6	+380984679730	marina	\$2y\$10\$RQkwx	uploads.work	1
6	Микола	Антонюк	Валерійович	6	2	1989-06-05	2020-12-14	NULL	3	+390974683758	kolia	\$2y\$10\$CGuRf	uploads.work	1
7	Олексій	Шуріпа	Володимирович	4	1	1999-03-10	2020-12-21	NULL	3	+380947836843	oleksi	\$2y\$10\$J9Nc/Q	uploads.work	0

Рисунок 3.8 – Оновленні дані у таблиці працівників

Таблиця 3.2 – Тест-кейс АТ0002

Тест-кейс ID: АТ0002	Пріоритет: 2	Створено: 23.05.2021, Д. Гаврилук
Назва: Перевірка функціоналу реєстрації користувача		
Кроки	Очікуваний результат	
<ol style="list-style-type: none"> Запустити додаток Натиснути кнопку «Зареєструватись» Ввести довільні дані про нового користувача у форму Натиснути кнопку «Зареєструватись» Порівняти фактичний результат з очікуваним 	<p>Відкрито вікно «Реєстрація працівника».</p> <p>Вікно «Реєстрація працівника» закрито. У таблиці із працівниками з'явився новий рядок, що містить дані введені на кроці 3.</p>	
Результат виконання тест-кейсу: пройдено успішно		



Змінити фото

Редагування даних працівника

Прізвище
Шехова

Ім'я
Марина

По-батькові
Леонідівна

Дата народження
29.12.1983

Номер телефону
+380984679730

Магазин:
Ласоща

Посада
Адміністратор

Дата прийняття на роботу:
09.12.2020

Дата звільнення
дд.мм.рррр

Попередній стаж (років)
6

Зберегти зміни

Рисунок 3.9 – Вікно «Редагування даних працівника»

Наступний тестовий випадок (таблиця 3.3) перевіряє функціонал редагування даних про працівника. При натисканні на кнопку «Редагувати дані», що знаходиться у вікні, де виводяться дані усіх працівників, відкривається вікно «Редагування даних працівника», форма якого містить дані про користувача обраного попередньо у таблиці (рисунок 3.9).

Таблиця 3.3 – Тест-кейс АТ0003

Тест-кейс ID: АТ0003	Пріоритет: 2	Створено: 23.05.2021, Д. Гаврилюк
Назва: Перевірка функціоналу редагування існуючого користувача Вхідні дані: Логін = «magina», Пароль = «111111»		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Запустити додаток 2. Ввести значення у поле логіну та пароллю із вхідних даних 3. Натиснути кнопку «Увійти» 4. Порівняти фактичний результат з очікуваним 5. Обрати пункт горизонтального меню «Персонал» 6. Порівняти фактичний результат з очікуваним 7. Натиснути кнопку «Редагувати дані» біля даних довільного працівника 8. Порівняти фактичний результат з очікуваним 9. Редагувати довільні дані про працівника 10. Натиснути кнопку «Зберегти зміни» 11. Порівняти фактичний результат з очікуваним 		<p>Вікно входу у систему закрито. Вікно для додавання нового продажу.</p> <p>Відкрито вікно «Усі працівники».</p> <p>Відкрито вікно «Редагування даних працівника».</p> <p>Вікно «Редагування даних працівника» закрито. У вікні з даними усіх працівників дані користувача, який редагувався змінено на дані введені на кроці 9.</p>
Результат виконання тест-кейсу: пройдено успішно		

Після редагування полей на формі та натисканні на кнопку «Зберегти зміни», вікно закриється, а у вікні з даними працівників, оновлюється дані працівника, який редагувався. Таким чином, розроблена АІС торгової точки з роздрібного продажу харчових товарів пройшла тестування успішно.

3.4 Інструкція користувача

Робота з АІС торгової точки з роздрібного продажу харчових товарів розпочинається з входу на головну сторінку, де користувач бачить інформацію про товари, які можна придбати у торгових точках (рис. 3.10). У нижній частині сайту знаходиться блок із посиланнями на різні сторінки, адресами торгових точок, контактними даними та посиланнями на сторінки у соціальних мережах (рис. 3.11).



Рисунок 3.10 – Головна сторінка сайту

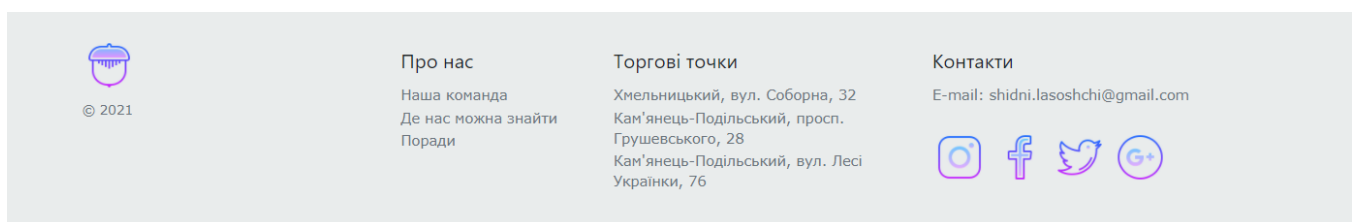


Рисунок 3.11 – Нижня частина сайту

Одним із посиланням у нижньому меню є посилання «Де нас можна знайти», перейшовши за ним користувачу виведеться список адрес торгових точок, графік роботи, контактні телефони і для зручності одразу позначення на карті до кожної адреси (рис. 3.12).

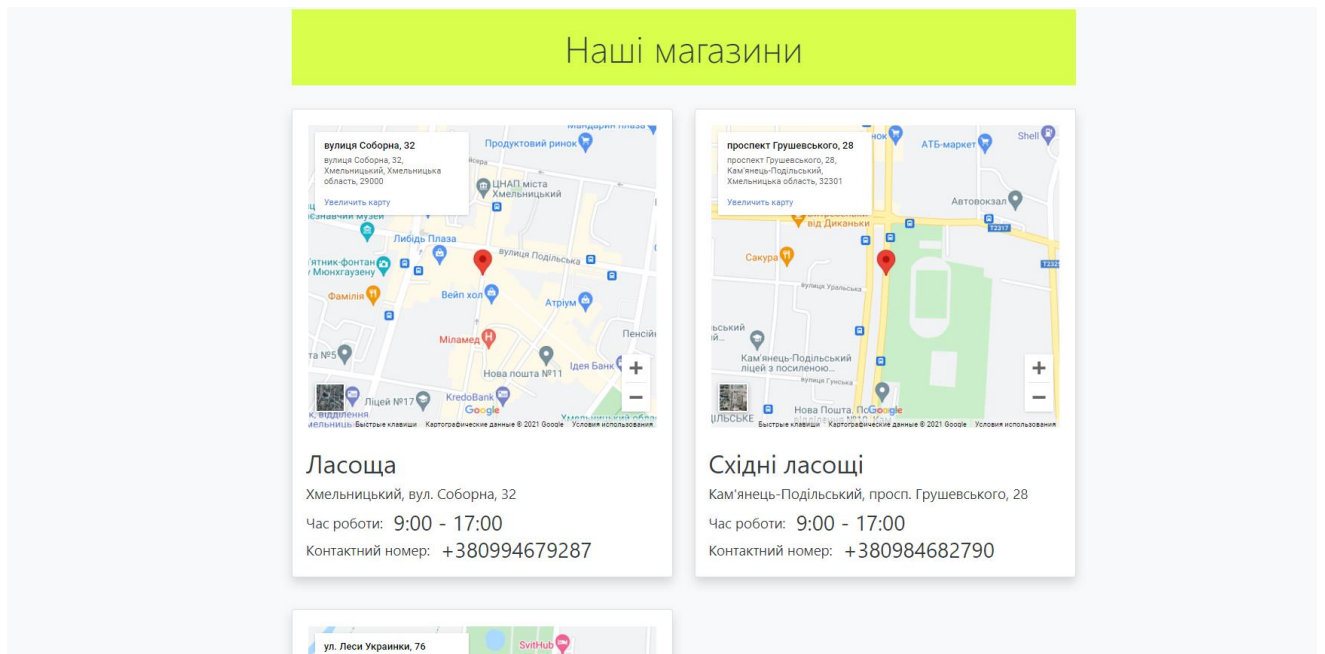


Рисунок 3.12 – Вивід даних про адреси, графік роботи та контактних телефонів

Реєстрація працівника

* – поля обов'язкові для заповнення

Логін*
Логін
лат. символи і цифри

Пароль*
Пароль
Показати
мін. 6 символів

Вибрати фото... Browse

Прізвище*
Прізвище

Ім'я*
Ім'я

По-батькові*
По-батькові

Дата народження*
дд.мм.рррр

Номер телефону*
+000(00)0000000

Попередній стаж
Років

Дата прийняття на роботу*
дд.мм.рррр

Дата звільнення
дд.мм.рррр

Магазин*
Вибрати

Посада*
Вибрати

Зареєструватись

Рисунок 3.13 – Форма реєстрації

Користувач також має змогу увійти у систему або зареєструватись. Для цього йому потрібно натиснути на відповідну кнопку у верхній частині сайту.

Якщо користувач обрав зареєструватись, він має заповнити усі дані на формі (рис. 3.13) та відправити її.

Після того як системний адміністратор надасть доступ користувачу змінивши поле «Access» у таблиці «Workers», користувач може увійти у систему. Після успішної аутентифікації користувач переходить на сторінку, де він має можливість додати новий продаж (рис.3.14).

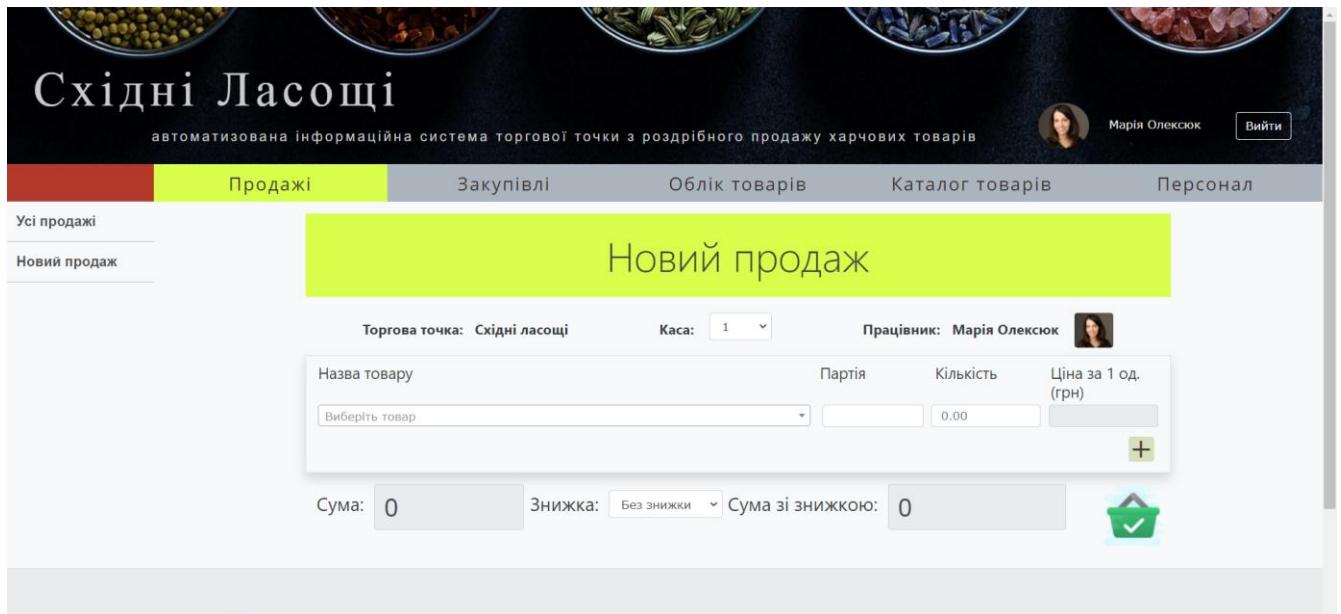


Рисунок 3.14 – Сторінка для додавання нового продажу

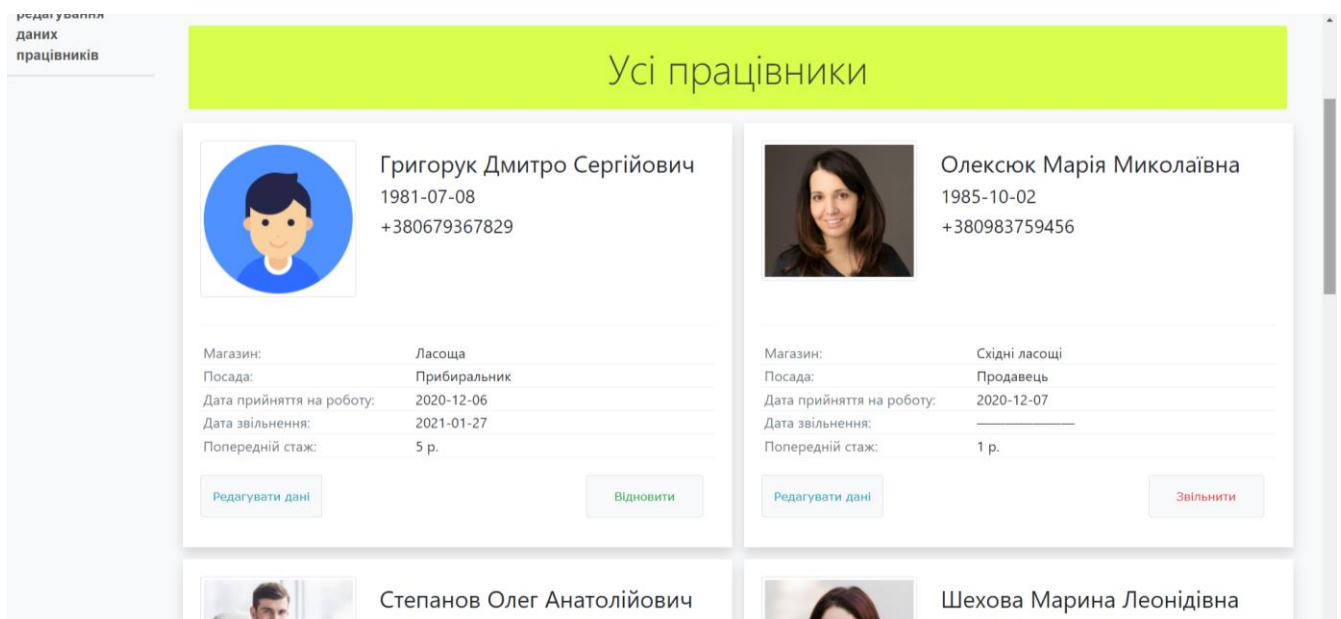


Рисунок 3.15 – Вивід даних усіх працівників

За допомогою горизонтального меню користувач має змогу змінювати вкладки, за допомогою вертикального меню користувач може вибрати функції для обраної ним раніше вкладки. Перегляд інформації про дані у таблиці зображений на рисунку 3.15.

Користувач має змогу редагувати дані, але у випадку якщо він не має доступу, на екран виведеться повідомлення зображене на рисунку 3.16.

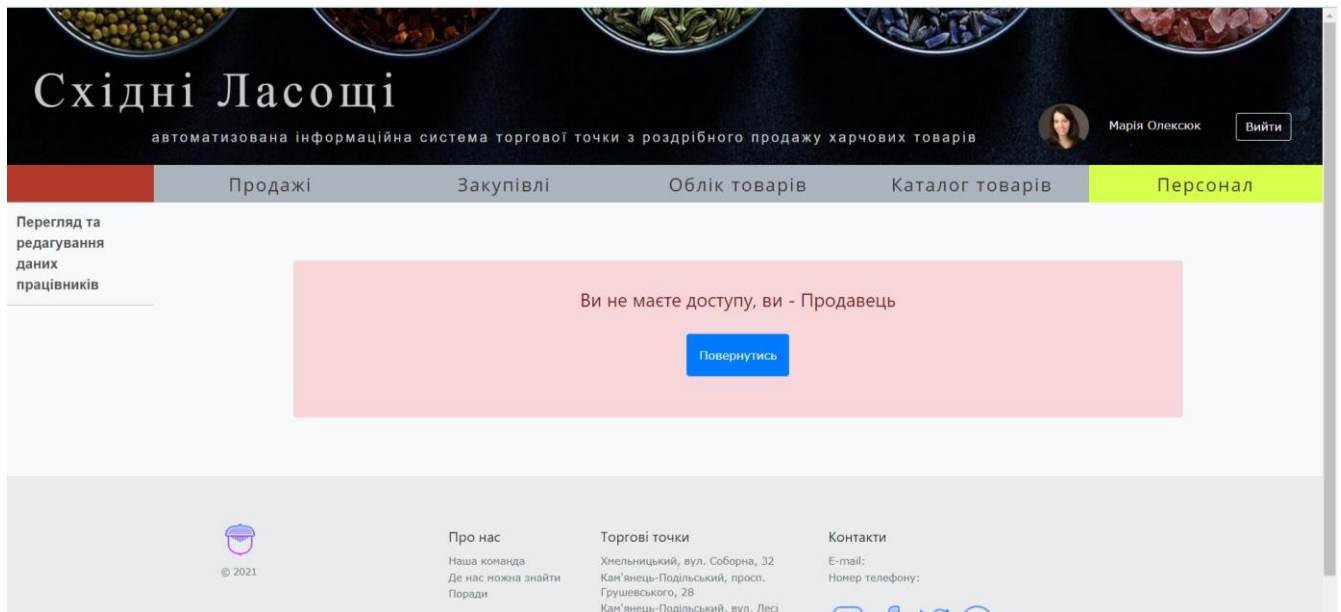


Рисунок 3.16 – Повідомлення про відмову в доступі

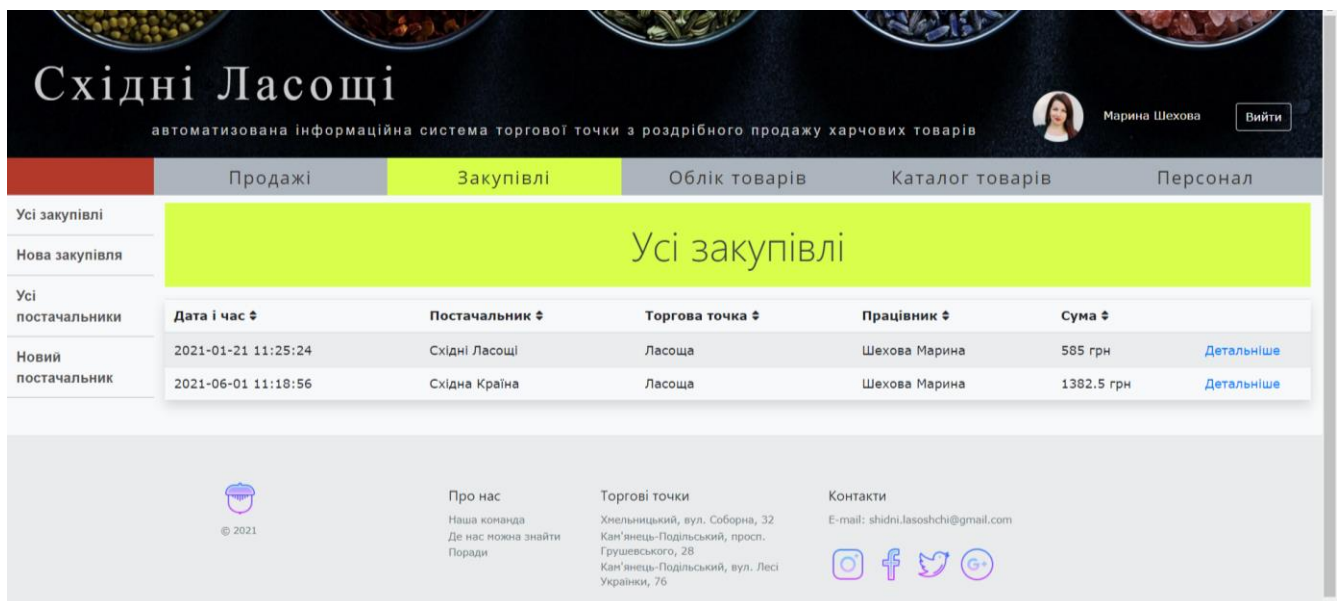


Рисунок 3.17 – Вивід списку усіх закупівель

Усі здійснені раніше продажі, ревізії та закупівлі виводяться списком (рис. 3.17) і кожної з них можна переглянути деталі натиснувши кнопку «Детальніше».

Дана інструкція допоможе користувача у користуванні системою, але загалом АІС розроблена таким чином, щоб користувач, навіть не маючи навиків користування нею, міг виконати необхідну йому функцію.

3.5 Вимоги до розгортання інформаційної системи

Рекомендовані браузерери:

- браузер Microsoft Internet Explorer версія 6.x і вище;
- браузер Microsoft Edge версія 20.10240;
- браузер Google Chrome версія 3.0.195 і вище;
- браузер Mozilla Firefox версія 3.0 і вище;
- браузер Safari версія 3.0 і вище.

Вимоги до сервера:

- сервер для роботи із базою даних MySQL;
- 130 Мб пам'яті на хмарному носії;
- підтримка CSS 3;
- підтримка HTML 5;
- підтримка Bootstrap 4.

Висновки

В результаті виконання кваліфікаційної роботи бакалавра було досягнуто мету – розробити на платформі PHP автоматизовану інформаційну систему торгової точки з роздрібного продажу харчових товарів, яка використовує базу даних відповідно до предметної області. Розробка проводилась за допомогою мови програмування PHP із використанням шаблону MVC 5, фреймворку Laravel та СКБД MySQL.

AIS торгової точки з роздрібного продажу харчових товарів призначена для автоматизації роботи трьох категорій користувачів – клієнта торгової точки, адміністратора торгової точки та адміністратора системи. Для клієнта інформаційної системи було автоматизовано виконання функцій: перегляд даних про торгові точки, перегляд каталогу товарів, перегляд статей. Для адміністратора торгової точки було автоматизовано виконання функцій: авторизація, продажі, закупівлі, облік товарів, каталог товарів, персонал. Для адміністратора системи було автоматизовано виконання функцій: доступ до всіх даних системи, активація облікових записів адміністраторів торгової точки. Відповідно, одержана AIS виконує функції додавання, редагування, перегляду даних про продажі, закупівлі, працівників, ревізії та товари для адміністраторів торгових точок, перегляд даних про товари та торгові точки – для клієнтів, а також надання доступу користувачів до AIS – для адміністратора системи. Таким чином, розроблена AIS повністю відповідає поставленому завданню.

Розроблена AIS повністю готова до використання, а в подальшому для неї можна придбати хостинг та налагодити обслуговування для реальних задач. У майбутньому систему можна вдосконалити, додавши додаткові функції для клієнтів, а саме можливість авторизації та оформлення замовлення. Створений веб-додаток дозволяє спростити та автоматизувати роботу працівників торгових точок з роздрібного продажу харчових продуктів.

Перелік посилань

1. Що таке комерція. URL: <https://ktonanovenkogo.ru/voprosy-i-otvety/kommerciya-chto-ehto-takoe.html>
2. Роздрібна торгівля: що це таке, правила, функції, як вести облік в магазині, цілі і завдання. URL: <https://aboutmarketing.info/biznes/rozdrubna-torhivlya/>
3. Класифікація продовольчих товарів. URL: https://studopedia.su/8_54848_klasifikatsiya-prodovolchih-tovariv.html
4. Плід горіх. URL: http://bearplanet.ru/kvytkovy_roslini/vidi_plodyv/plyd_goryh.html
5. 11 видів горіхів з назвами та описом. URL: <https://farmerstvo.net/4858806-11-types-of-nuts-with-names-anddescriptions>
6. Спеції, прянощі та приправи – у чому різниця. URL: <https://harchi.info/blogs/san-ayt-j/speciyi-pryanoshchi-ta-prypravu-u-chomu-riznytsya>
7. Wikipedia. Сухофрукти. URL: <https://uk.wikipedia.org/Сухофрукти>
8. Сухофрукти. URL: <https://healthapple.info/zdorovya-ta-organizm/sukhofrukty/#vidi-i-klasifikatsiya>
9. Продаж (реалізація) товарів. URL: https://kodeksy.com.ua/dictionary/p/prodazh_realizatsiya_tovariv.htm
10. Сутність, форми та відмінності персонального продажу. URL: <https://buklib.net/books/28181/>
11. Ревізія, її види та характеристики. URL: <http://www.ukr.vipreshebnik.ru/revizor/1170-reviziya-jiji-vidi-ta-kharakteristika.html>
12. Wikipedia. Закупівля. URL: <https://uk.wikipedia.org/wiki/Закупівля>
13. Накладна. URL: <https://www.pharmencyclopedia.com.ua/article/1165/nakladna>
14. Накладна. URL: <https://news.finance.ua/ua/news/-/297433/nakladna>
15. Адміністратор залу (в підприємствах торгівлі) Посадова інструкція. URL: https://www.borovik.com/index_instruction.php?Gins=dycy&lang_i=1

16. Продавець продовольчих товарів. URL:
http://proforientator.info/?page_id=5287
17. Керуючий магазином: обов'язки, посадова інструкція, функції, відповідальність. URL: <http://xn--80aimveh.pp.ua/nauka/10849-keruyuchiy-magazinom-obovyazki-posadova-nstrukcyia-funkcyi-vdpovdalnst.html>
18. Що робить комірник. URL: <https://www.work.ua/career-guide/storekeeper>
19. NUTSBOX. URL: <https://nutsbox.ua/ua/>
20. Акцент. URL: <https://www.accent.ua/>
21. Програма для ведення бухгалтерського обліку «Акцент». URL: <https://portfel.ua/programa-dlja-vedennja-obliku-akcent/>
22. Особливості ринку програмних засобів з комп'ютеризації бухгалтерського обліку в сільському господарстві. URL: <http://magazine.faaf.org.ua/osoblivosti-rinku-programnih-zasobiv-z-komp-yuterizacii-buhgalterskogo-obliku-v-silskomu-gospodarstvi.html>
23. Nutsking. URL: <https://nutsking.com.ua/>
24. Плюси і мінуси розробки .NET Framework. URL: <https://www.cisin.com/coffee-break/ru/enterprise/the-good-and-the-bad-of-net-framework-development.html>
25. PHP – найбільш популярна мова для веб програмування. URL: <https://chili-web.com.ua/php-5/>
26. PHP. URL: <https://uk.wikipedia.org/wiki/PHP>
27. Java переваги і недоліки. URL: <https://4systems.ru/inf/java-dostoinstva-i-nedostatki/>
28. PHP. URL: <http://programming.in.ua/web-design/allphp/30-about-php.html>
29. Розробка уроку. PHP. URL: http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/43_PHP/index.html
30. Laravel – лідер серед PHP фреймворків, схвалений розробниками. URL: <http://savelink.org.ua/laravel-lider-sered-php-frejmworkiv-shvalenij-rozrobnikami/>

31. Розробка PHP з Laravel - порівняння плюсів і мінусів. URL:
<https://solidbrain.com.ua/ua/news/rozrobka-php-s-laravel-porivniannia-plusiv-i-minysiv>

32. MVC PHP. URL: <https://sitear.ru/material/mvc-php>

33. Microsoft SQL Server. URL:
https://uk.wikipedia.org/wiki/Microsoft_SQL_Serve

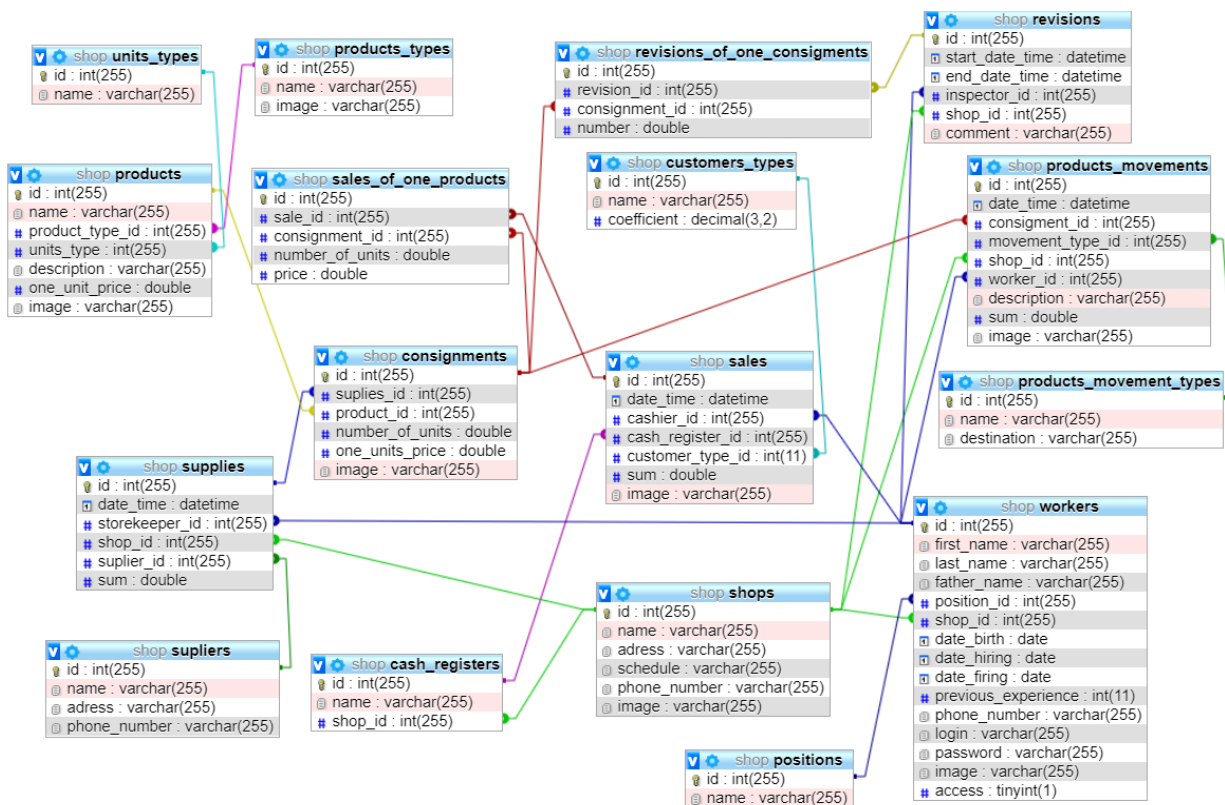
34. MySQL. URL: <https://uk.wikipedia.org/wiki/MySQL>

35. Що таке MySQL як і де використовують MySQL. URL:
<http://ruszura.in.ua/uncategorized/scho-take-mysql-yak-i-de-vykorystovuyut-mysql.html>

ДОДАТКИ

Додаток А

Структура бази даних автоматизованої інформаційної системи торгової точки з роздрібною продажу харчових товарів



Додаток Б

Програмні коди основних модулів автоматизованої інформаційної системи торгової точки з роздрібного продажу харчових товарів

Лістинг AuthController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Support\Facades\Auth;
use App\Models\Workers;
use App\Models\Positions;
use App\Models\Shops;
use Illuminate\Http\Request;

class AuthController extends Controller
{
    protected $redirectTo = '/main';

    public function getSignIn(){
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');
        return view('inc.signin',compact('data','shops'));
    }

    public function postSignIn(Request $request){

        $validation = $request->validate([
            'login' => 'required|alpha-dash|max:20',
            'password' => 'required|min:6'
        ]);

        $login = $request->input('login');
        $password = $request->input('password');
        if(!Auth::attempt(['login' => $login, 'password' => $password, 'access' => 1
    ]))){
            return redirect()->back()-
>with('infolog','Неправильний логін або пароль');
        }
        return redirect()->route('newsale');
    }

    public function getSignOut(){
        Auth::logout();
        return redirect()->route('start');
    }
}
}
```

Лістинг MovementController.php

```
<?php

namespace App\Http\Controllers;
```

```

use App\Models\ProductsMovement;
use App\Models\ProductsMovementTypes;
use App\Models\Consignments;
use App\Models\Shops;
use App\Models\Products;
use App\Models\Supplies;
use App\Models\Workers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class MovementController extends Controller
{
    public function index()
    {
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');
        $types = ProductsMovementTypes::all();
        $consignments = Consignments::all();
        return view('inc.revisions.new_movement', compact('consignments', 'types', 'data', 'shops' ));
    }

    public function postNew(Request $request){
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        $types = ProductsMovementTypes::all();
        $consignments = Consignments::all();
        $move = new ProductsMovement();
        $move->date_time = $request->input('date_time');
        $prod_name = substr($request->input('product'),0,-20);
        $prod_id = Products::where('name',$prod_name)->first()->id;

        $lenth = strlen($prod_name);
        $sup_date = substr($request->input('product'),$lenth+1);
        $sup_id =Supplies::where('date_time',$sup_date)->first()->id;
        $consignment_id = Consignments::where([
            ['suplies_id','=', $sup_id],
            ['product_id','=', $prod_id]
        ])->first()->id;
        $move->consignment_id = $consignment_id;
        $move->movement_type_id = $request->input('move_type');
        $move->shop_id = $data['shop_id'];
        $move->worker_id = $data['id'];
        $move->number = $request->input('number');
        $move->description = $request->input('description');
        $move->sum = $request->input('sum');
        if ($request->has('image')) {
            $path = $request->file('image')->store('uploads.movements',
            'public',
            $request->input('name'));
            $move->image = $path;
        }
        $move->save();
        return redirect()->route('newmovement')-
>with('info', 'Рух успішно зареєстровано');
    }
}

```

```

public function all_movements(){
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');
    $movements = ProductsMovement::sortable()->paginate(5);
    return view('inc.revisions.all_movements', compact('shops', 'data', 'movements'
));
}
}

```

Лістинг ProductsController.php

<?php

```

namespace App\Http\Controllers;

use App\Models\Products;
use App\Models\Shops;
use App\Models\ProductsType;
use App\Models\UnitsType;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use App\Models\Workers;

class ProductsController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        $prodtype = ProductsType::all();
        $unittype = UnitsType::all();
        $products = Products::all();
        return view('inc.products.new', compact('data', 'shops', 'prodtype', 'unittype')
    );
}

public function postNew(Request $request)
{
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');

    $prodtype = ProductsType::all();
    $unittype = UnitsType::all();
    $products = Products::all();

    $product = new Products();
    $prodtype_id = ProductsType::where('name', $request->input('prod_type'))-
>pluck('id')->first();

```

```

        $unittype_id = UnitsType::where('name',$request->input('unit_type'))->
>pluck('id')->first();
        $product->name = $request->input('name');
        $product->description = $request->input('description');
        if ($request->has('image')) {
            $path = $request->file('image')->store('uploads.products',
            'public',
            $request->input('name'));
            $product->image = $path;
        }
        $product->product_type_id = $prodtype_id;
        $product->units_type = $unittype_id;
        $product->one_unit_price = $request->input('price');
        $product->save();
        return view('inc.products.new',compact('userimage','worker_name','prodtype',
'unittype'))->with('info','Товар додано');
    }

    public function showall(){
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        $products = Products::all();
        foreach($products as $product){
            if($product->image == null)
                $product->image="no-image.png";
        }
        if(Auth::check())
            return view ('inc.products.all',compact('data','shops','products'));
        else
            return view ('inc.products.all_product',compact('data','shops','products
'));
    }

    public function edit($id){
        $product = Products::find($id);
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');
        $prodtype = ProductsType::all();
        $unittype = UnitsType::all();
        return view ('inc.products.edit',compact('data','shops','product','prodtype',
,'unittype'));
    }

    public function editpost($id,Request $request){
        $product = Products::find($id);
        $product->name = $request->input('name');
        $product->description = $request->input('description');
        $product->one_unit_price = $request->input('one_unit_price');
        $product->units_type =$request->input('units_type');
        $product->product_type_id = $request->input('prod_type');
        if ($request->has('image')) {
            $path = $request->file('image')->store('uploads.products','public');
            $product->image = $path;
        }
        $product->save();
        return redirect()->route('productsall')->
>with('info','Дані про товар "' . $product->name.'" змінено');
    }
}

```

}

Лістинг PurchaseController.php

<?php

```

namespace App\Http\Controllers;

use App\Models\Consignments;
use Illuminate\Support\Facades\Auth;
use App\Models\Shops;
use App\Models\Workers;
use App\Models\Products;
use App\Models\Suppliers;

use App\Models\Supplies;
use Illuminate\Http\Request;

class PurchaseController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');
        $products_name = Products::all()->pluck('name');
        $prod_type = Products::all()->pluck('units_type');
        $products = Products::all();
        $suppliers = Suppliers::all();
        return view('inc.purchases.new', compact('data', 'products', 'products_name', '
prod_type', 'suppliers', 'shops'));
    }

    public function postNewPurchase(Request $request){
        $method = new Workers();
        $data = $method->GetConstantData();
        if(Auth::check()){

            $supplier_id = Suppliers::where('name',$request->input('supplier'))-
>first()->id;

            $suplie = new Supplies();

            $suplie->date_time = $request->input('date_time');
            $suplie->storekeeper_id = $data['id'];
            $suplie->shop_id = $data['shop_id'];
            $suplie->supplier_id = $supplier_id;
            $suplie->sum = $request->input('sum');
            $suplie->save();
            $supplies_id = Supplies::where([
                ['date_time','=', $request->input('date_time')],
                ['supplier_id','=', $supplier_id]
            ]->first()->id;

```

```

    $i = $request->input('count_prod');

    for($x = 0; $x <= $i; $x++){
        $consignment = new Consignments();
        $prod = Products::where('name',$request->input('product'.$x))-
>pluck('id')->first();
        $consignment->suplies_id = $suplies_id;
        $consignment->product_id = $prod;
        $consignment->number_of_units = $request->input('number'.$x);
        $consignment->one_units_price = $request->input('price'.$x);
        if ($request->has('image')) {
            $path = $request->file('image')-
>store('uploads.consignments','public');
            $consignment->image = $path;
        }
        $consignment->save();
    }
    return redirect()
        ->route('newpurchase')
        ->with('info','Закупівля зареєстрована');

}else
{return redirect()
->route('newpurchase')
->with('infolog','Увійдіть у систему');}
}
public function allSupplies(){
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');
    $supplies = Supplies::sortable()->paginate(5);
    return view('inc.purchases.all',compact('data','shops','supplies'));
}
public function SupplyDetails($id){
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');
    $sup = Supplies::find($id);
    $consignment = Consignments::where('suplies_id',$id)->get();
    return view('inc.purchases.one_supplie',compact('data','shops','sup','consig
nment'));
}
}
}

```

Лістинг RevisionController.php

<?php

```

namespace App\Http\Controllers;

use App\Models\Revisions;
use App\Models\RevisionsOfOneConsignment;
use App\Models\Consignments;
use Illuminate\Support\Facades\Auth;
use App\Models\Shops;
use App\Models\Products;
use App\Models\Suppliers;
use App\Models\Workers;
use App\Models\Supplies;

```

```

use Illuminate\Http\Request;

class RevisionController extends Controller
{
    public function index()
    {
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');
        $consignments = Consignments::all();
        $products_name = Products::all()->pluck('name');
        $prod_type = Products::all()->pluck('units_type');
        $products = Products::all();
        return view('inc.revisions.new', compact('data', 'shops', 'consignments', 'products', 'products_name', 'prod_type'));
    }

    public function postNew(Request $request){
        $method = new Workers();
        $data = $method->GetConstantData();

        $revision = new Revisions();

        $revision->start_date_time = $request->input('start');
        $revision->end_date_time = date("Y-m-d H:i:s");
        $revision->inspector_id = $data['id'];
        $revision->shop_id = $data['shop_id'];
        $revision->comment = $request->input('comment');
        $revision->save();

        $rev_id = Revisions::where([
            ['start_date_time', '=', $request->input('start')],
            ['inspector_id', '=', $data['id']]
        ]->pluck('id')->first();
        $i = $request->input('count_prod');

        for($x = 0; $x <= $i; $x++){
            $prod_name = substr($request->input('product' . $x), 0, -20);
            $prod_id = Products::where('name', $prod_name)->first()->id;

            $lenth = strlen($prod_name);
            $sup_date = substr($request->input('product' . $x), $lenth+1);
            $sup_id = Supplies::where('date_time', $sup_date)->first()->id;
            $consignment_id = Consignments::where([
                ['suplies_id', '=', $sup_id],
                ['product_id', '=', $prod_id]
            ]->first()->id;
            $one_revision = new RevisionsOfOneConsignment();
            $one_revision->revision_id = $rev_id;
            $one_revision->consignment_id = $consignment_id;
            $one_revision->number = $request->input('number' . $x);
            $one_revision->save();
        }
        return redirect()->route('allrevisions');
    }

    public function all_rev(){
        $method = new Workers();
        $data = $method->GetConstantData();
    }
}

```

```

        $shops = Shops::all()->pluck('adress');

        $revisions = Revisions::sortable()->paginate(5);

        return view('inc.revisions.all_revisions', compact('data', 'shops', 'revisions'
    ));
    }
    public function one_rev($id){
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        $revision = Revisions::find($id);
        $sep_revisions = RevisionsOfOneConsignment::where('revision_id',$id)-
>sortable()->paginate(5);

        return view('inc.revisions.one_revision', compact('data', 'shops', 'revision', '
sep_revisions'));
    }
}

```

Лістинг SalesController.php

<?php

```

namespace App\Http\Controllers;

use App\Models\Sales;
use App\Models\SalesOfOneProduct;
use App\Models\Workers;
use App\Models\CashRegisters;
use App\Models\Products;
use App\Models\Shops;
use App\Models\UnitsType;
use App\Models\CustomersType;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;

class SalesController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */

    public function index()
    {
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        $registers = CashRegisters::where('shop_id', '=', Auth::user()-
>getWorkerShop()->get());
        $customers_typename = CustomersType::all()->pluck('name');
        $customers_typecoef = CustomersType::all()->pluck('coefficient');
        $products = Products::all();
        $products_name = Products::all()->pluck('name');
        $products_price = Products::all()->pluck('one_unit_price');
    }
}

```

```

$prod_type = Products::all()->pluck('units_type');

return view('inc.sales.new_sale', compact('products', 'registers',
    'products_name', 'products_price', 'data', 'shops',
    'customers_typename', 'customers_typecoef', 'prod_type'));
}

public function postNewSale(Request $request){
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');

    if(Auth::check()){
        $validation = $request->validate([
            'worker_id'=>'required'
        ]);

        $customer_type_id = CustomersType::where('name',$request-
>input('customer_type'))->first()->id;
        $cash_register = CashRegisters::where([[ 'name', '=', $request-
>input('cash_reg_id') ], [ 'shop_id', '=', $data['shop_id'] ]])->first()->id;
        $sale = new Sales();
        $sale->date_time = $request->input('date_time');
        $sale->cashier_id = $request->input('worker_id');
        $sale->cash_register_id = $cash_register;
        $sale->customer_type_id = $customer_type_id;
        $sale->sum = $request->input('sum_coef');
        $sale->save();

        $sale_id = Sales::where([
            ['date_time', '=', $request->input('date_time')],
            ['cash_register_id', '=', $request->input('cash_reg_id')]
        ])->first()->id;
        $i = $request->input('count_prod');

        for($x = 0; $x <= $i; $x++){
            $sale_prod = new SalesOfOneProduct();
            $sale_prod->sale_id = $sale_id;
            $sale_prod->consignment_id = $request->input('consigment'.$x);
            $sale_prod->number_of_units = $request->input('number'.$x);
            $sale_prod->price = floatval($request->input('price'.$x));
            $sale_prod->save();
        }

        return redirect()
            ->route('newsale', compact('data', 'shops'))
            ->with('info', 'Продаж зареєстровано');
    }else
    {return redirect()
        ->route('newsale', compact('data', 'shops'))
        ->with('infolog', 'Увійдіть у систему');}
}

public function all()
{
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');
}

```

```

    $sales = Sales::all();
    return view('inc.sales.all', compact('sales','shops','data'));
}

public function onesale($id){
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');

    $sale = Sales::find($id);
    $ct = new CustomersType();
    $shop_name = Shops::where('id',(CashRegisters::where('id',$sale-
>cash_register_id)->first()->id))->first()->name;
    $cash_register = CashRegisters::where('id',$sale->cash_register_id)-
>first()->name;
    $sales_one = SalesOfOneProduct::where('sale_id',$sale->id)->get();
    $cashier = $method->getNameById($sale->cashier_id);
    return view('inc.sales.onesale', compact('sale','shop_name','cashier','cash_
register','sales_one','data','shops' ));
}
}

```

Лістинг WorkersController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Support\Facades\Auth;
use App\Models\Workers;
use App\Models\Positions;
use App\Models\Shops;
use Illuminate\Http\Request;
```

```
class WorkersController extends Controller
```

```
{
    public function getSignUp(){
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        $positions = Positions::all();
        $shop = Shops::all();
        return view('inc.workers.signup', compact('positions','shop','data','sho
ps'));
    }
    public function postSignUp(Request $request){
        if(Auth::check()){
            $worker_name = Auth::user()->getName();
            $userimage = Workers::where('id',$worker_id)->first()->image;
        }
        else{
            $userimage = 'uploads.workers/default-user-image.png';
            $worker_name = 'Ім`я працівника';
        }
        $now_date = date("d.m.Y");
        $date = date("d.m.Y", mktime(0, 0, 0,date('d'), date('m'), date('Y')-18));
    }
}

```

```

$validation = $request->validate([
    'login' => 'required|unique:workers|alpha-dash|max:20',
    'password' => 'required|min:6',
    'last_name' => 'required|min:6',
    'date_birth' => 'required',
    'first_name' => 'required',
    'phone_number' => 'required',
    'father_name' => 'required',
    'position_id'=>'required',
    'date_hiring'=>'required',

    'shop_id'=>'required',
    'image'=>'required'
]);

$shop_id = Shops::where('name',$request->input('shop_id'))->first()->id;
$position_id = Positions::where('name',$request->input('position_id'))-
>first()->id;
$worker=new Workers();
$worker->first_name=$request->input('first_name');
$worker->last_name=$request->input('last_name');
$worker->father_name=$request->input('father_name');
$worker->position_id= $position_id;
$worker->shop_id= $shop_id;
$worker->date_birth=$request->input('date_birth');
$worker->date_hiring=$request->input('date_hiring');
$worker->date_firing=$request->input('date_firing');
$worker->previous_experience=$request->input('previous_experience');
$worker->phone_number=$request->input('phone_number');
$worker->login=$request->input('login');
$worker->password=bcrypt($request->input('password'));

if ($request->has('image')) {
    $path = $request->file('image')-
>store('uploads.workers','public','worker_name');
    $worker->image = $path;
}
$worker->save();
return redirect()
    ->route('success')
    ->with('info','Реєстрація працівника пройшла успішно');
}
public function success(){
    if(Auth::check()){
        $worker_name = Auth::user()->getName();
        $worker_id = Auth::user()->getWorkerId();
        $userimage = Workers::where('id',$worker_id)->first()->image;
    }
    else{
        $userimage = 'uploads.workers/default-user-image.png';
        $worker_name = 'Ім'я працівника';
    }
    return view('inc.after_signup',compact('worker_name','userimage'));
}

public function main(){
    if(Auth::check()){
        $worker_name = Auth::user()->getName();
        $worker_id = Auth::user()->getWorkerId();

```

```

        $userimage = Workers::where('id',$worker_id)->first()->image;
    }
    else{
        $userimage = 'uploads.workers/default-user-image.png';
        $worker_name = 'Ім'я працівника';
    }
    return view('inc.workers.main',compact('worker_name','userimage'));
}

public function show()
{
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');

    $workers = Workers::all();
    $positions = Positions::all()->pluck('name');
    $shops = Shops::all()->pluck('name');
    return view('inc.workers.all_workers', compact('workers','shops','positions'
,'data'));
}

public function edit($id){
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');

    $positionsall = Positions::all();
    $shopsall = Shops::all();
    $worker = Workers::find($id);
    $positions = Positions::all()->pluck('name');
    $workername = $worker->last_name.' '.$worker->first_name;
    if(Auth::check()){
        $workerpos_id = Auth::user()->getWorkerPosition();
        $userposition = Positions::where('id',$workerpos_id)->first()->name;
        if($userposition=="Адміністратор" || $userposition=="Директор")
            return view('inc.workers.edit',compact('worker','workername','positi
ons','positionsall','shopsall','data','shops'));
        else
            return redirect()
                ->route('infoworker')
                ->with('infolog','Ви не маєте доступу, ви - '.$userposition);
    }else
        return redirect()
            ->route('infoworker')
            ->with('infolog','Ви не маєте доступу, ви - ');
}

public function info(){
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');

    return view('inc\workers\info',compact('data','shops'))-
>with('infolog','Ви не маєте доступу, ви - ');
}

public function editpost($id, Request $request){
    $method = new Workers();
    $data = $method->GetConstantData();

```

```

    $shops = Shops::all()->pluck('adress');

    $shop_id = Shops::where('name',$request->input('shop_id'))->first()->id;
    $position_id = Positions::where('name',$request->input('position_id'))-
>first()->id;
    $worker = Workers::find($id);
    $worker->first_name = $request->input('first_name');
    $worker->last_name = $request->input('last_name');
    $worker->father_name = $request->input('father_name');
    $worker->position_id = $position_id;
    $worker->shop_id = $shop_id;
    $worker->date_birth=$request->input('date_birth');
    $worker->date_hiring=$request->input('date_hiring');
    $worker->date_firing=$request->input('date_firing');
    $worker->previous_experience=$request->input('previous_experience');
    $worker->phone_number=$request->input('phone_number');
    if ($request->has('image')) {
        $path = $request->file('image')->store('uploads.workers','public');
        $worker->image = $path;
    }
    $worker->save();
    return redirect()->route('allworkers')->with('info', 'Дані збережено');
}

public function drop($id){

    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');
    if($data['id'] <> $id){
        $worker = Workers::find($id);
        $worker->date_firing = date("Y-m-d");
        $worker->save();
        return redirect()->route('allworkers')->with('info', $worker-
>last_name.' '.$worker->first_name.' - звільнено');
    }
    else{
        return redirect()->route('allworkers')-
>with('infolog', 'Ви не можете звільнити самого себе');
    }
}

public function restore($id){
    $method = new Workers();
    $data = $method->GetConstantData();
    $shops = Shops::all()->pluck('adress');

    $worker = Workers::find($id);
    $worker->date_firing = null;
    $worker->save();
    return redirect()->route('allworkers')->with('info', $worker-
>last_name.' '.$worker->first_name.' - відновлено');
}
}

```

Лістинг SuppliersController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use App\Models\Suppliers;
use Illuminate\Support\Facades\Auth;
use App\Models\Workers;
use App\Models\Shops;
use Illuminate\Http\Request;
use PDF;

class SuppliersController extends Controller
{
    public function All_Sup(){
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        $suppliers = Suppliers::all();
        return view('inc.purchases.all_suppliers', compact('suppliers','data','shops
    '));
    }
    public function createPDF()
    {
        $suppliers = Suppliers::all();

        $pdf = PDF::loadView('inc.purchases.supplier_pdf', compact('suppliers'));

        return $pdf->stream('itsolutionstuff.pdf');
    }

    public function newSup(){
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        return view('inc.purchases.newsup', compact('shops','data'));
    }
    public function postNewSup(Request $req){
        $method = new Workers();
        $data = $method->GetConstantData();
        $shops = Shops::all()->pluck('adress');

        $supplier = new Suppliers();
        $supplier->name = $req->input('name');
        $supplier->adress = $req->input('adress');
        $supplier->phone_number = $req->input('phone_number');
        $supplier->save();
        return redirect()
            ->route('newsup',compact('data','shops'))
            ->with('info','Постачальника додано');
    }
}
}

```

Лістинг Web.php

```
<?php
```

```

use Illuminate\Support\Facades\Route;
use Illuminate\Support\Facades\Auth;
//WORKERS

```

```

Route::get('workers/new-worker', 'App\Http\Controllers\WorkersController@getSignUp')->name('workernew');
Route::post('workers/new-worker', 'App\Http\Controllers\WorkersController@postSignUp');
Route::get('workers', 'App\Http\Controllers\WorkersController@main')->name('workers-main');
Route::get('workers/all', 'App\Http\Controllers\WorkersController@show')->name('allworkers');
Route::get('workers/all/{id}/edit', 'App\Http\Controllers\WorkersController@edit')->name('editworker');
Route::post('workers/all/{id}/edit', 'App\Http\Controllers\WorkersController@editpost')->name('editworkerpost');
Route::get('workers/all/{id}/drop', 'App\Http\Controllers\WorkersController@drop')->name('fire');
Route::get('workers/all/{id}/restore', 'App\Http\Controllers\WorkersController@restore')->name('restore');
Route::get('workers/all/info', 'App\Http\Controllers\WorkersController@info')->name('infoworker');

//LOG
Route::get('/signin', 'App\Http\Controllers\AuthController@getSignIn')->name('signin');
Route::post('/signin', 'App\Http\Controllers\AuthController@postSignIn');
Route::get('/logout', 'App\Http\Controllers\AuthController@getSignOut')->name('logout');
Route::get('/successful-registration', 'App\Http\Controllers\WorkersController@success')->name('success');

//SALES
Route::get('sales/new', 'App\Http\Controllers\SalesController@index')->name('newsale');
Route::post('sales/new', 'App\Http\Controllers\SalesController@postNewSale');
Route::get('sales/all', 'App\Http\Controllers\SalesController@all')->name('salesall');
Route::get('sales/all/{id}', 'App\Http\Controllers\SalesController@onesale')->name('onesale');

//PURCHASES
Route::get('purchases/new', 'App\Http\Controllers\PurchaseController@index')->name('newpurchase');
Route::post('purchases/new', 'App\Http\Controllers\PurchaseController@postNewPurchase');
Route::get('purchases/all-suppliers', 'App\Http\Controllers\SuppliersController@All_Sup')->name('allsup');
Route::get('purchases/new-supplier', 'App\Http\Controllers\SuppliersController@newSup')->name('newsup');
Route::post('purchases/new-supplier', 'App\Http\Controllers\SuppliersController@postNewSup');
Route::get('purchases/all-supplier/pdf', 'App\Http\Controllers\SuppliersController@createPDF')->name('pdfsup');
Route::get('purchases/all-supplies', 'App\Http\Controllers\PurchaseController@allSupplies')->name('allsupplies');
Route::get('purchases/all-supplies/{id}', 'App\Http\Controllers\PurchaseController@SupplierDetails')->name('onesupplie');

//REVISIONS

```



```

        </div>
    </div>
    <div class="col-5">
        <div class="form-group">
            <h5>Тип товару</h5>
            <select class="form-control" name="prod_type">
                @foreach($prodtype as $type)
                    @if($type-
>name == (\App\Models\ProductsType::find($product->product_type_id)->name))
                        <option selected value="{{ $type-
>id}}">{{ $type->name}}</option>
                    @else <option value="{{ $type-
>id}}">{{ $type->name}}</option>
                    @endif
                @endforeach
            </select>
        </div>
    </div>
    <div class="form-group">
        <h5>Опис</h5>
        <textarea name="description" id="textarea" rows="4" class="form-
control" >{{ $product->description}}</textarea>
        <small class="form-text text-
muted">Максимально 350 символів</small>
    </div>
    <div class="row">
        <div class="col-6">
            <div class="form-group">
                <h5>Ціна</h5>
                <input type="text" class="form-
control" name="one_unit_price" value="{{ $product->one_unit_price}}">
                <small class="form-text text-muted">Грн, 00.00</small>
            </div>
        </div>
        <div class="col-6">
            <form-group>
                <h5>Тип одиниці</h5>
                <select class="form-control" name="units_type" >
                    @foreach($unittype as $type)
                        @if($type-
>name == (\App\Models\ProductsType::find($product->units_type)->name))
                            <option selected value="{{ $type-
>id}}">{{ $type->name}}</option>
                        @else <option value="{{ $type->id}}">{{ $type-
>name}}</option>
                        @endif
                    @endforeach
                </select>
            </form-group>
        </div>
    </div>
    <div class="centering my-3">
        <button type="submit" class="btn btn-success px-
5"><h4>Зберегти зміни</h4></button></div>
    </div>
</div>
</form><script type = "text/javascript" src = "https://ajax.googleapis.com/ajax/libs
/jquery/1/jquery.min.js"></script>

```

```

<script>
function goBack() {
    window.history.back();
}

$('#file').change(function () {
    var input = $(this)[0];
    if (input.files && input.files[0]) {
        if (input.files[0].type.match('image.*')) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#img-preview').attr('src', e.target.result);
            }
            reader.readAsDataURL(input.files[0]);
        } else {
            console.log('ошибка, не изображение');
        }
    } else {
        console.log('хьюстон у нас проблема');
    }
});

$("textarea").keydown(function() {
    if (this.value.length > 350)
        this.value = this.value.substr(0, 350);
});

$('#form').bind('reset', function () {
    $('#img-preview').attr('src', 'default-preview.jpg');
});
</script>
@endsection
@include('layouts.footer')
@endsection

```

Лістинг Products/new.blade.php

```

@extends('layouts.app')
@section('title')Новий товар @endsection

@include('layouts.headers.header')
@section('content')
    @extends('inc.menus.products')
    @section('content in menu')
        <form method="POST" action="{{route('newproduct')}}" enctype="multipart/form-
        data" >
        {{ csrf_field() }}
        <div class="container flex-column d-flex justify-content-center my-0 mb-4">
            <h1 class="text-dark text-center display-4 p-4 heading-green">Новий товар</h1>
            @include('layouts.alerts')
            <div class="container-fluid border shadow d-flex flex-column p-5">
                <div class="row">
                    <div class="col-4">
                        
                        <input type="file" class="inputfile " name="image" id="file">
                        <label style="width:90%!important;" class="btn btn-outline-dark px-
                        5 shadow mt-1" for="file"><h4>Вибрати фото</h4></label>
                    </div>
                    <div class="col-8">
                        <h4>Назва товару</h4>
                    </div>
                </div>
            </div>
        </div>
    @endsection

```

```

        <input type="text" name="name" class="form-control">
        <h4>Опис</h4>
        <textarea name="description" class="form-
control" id="textarea" style="height:120px"></textarea>
        <small class="text-muted">Максимально 350 символів</small>
        <div class="d-flex justify-content-between">
            <div class="form-group">
                <h4>Ціна</h4>
                <input type="text" name="price" class="form-control">
                <small class="form-text text-muted">Грн, 00.00</small>
            </div>
            <div class="form-group">
                <h4>Тип одиниці</h4>
                <select class="form-control" style="min-
width:200px!important;" name="unit_type" >
                    @foreach($unittype as $type)
                        <option>{{ $type->name}}</option>
                    @endforeach
                </select>
            </div>
            <div class="form-group">
                <h4>Тип товару</h4>
                <select class="form-control" style="min-
width:200px!important;" name="prod_type">
                    @foreach($prodtype as $type)
                        <option>{{ $type->name}}</option>
                    @endforeach
                </select>
            </div>
        </div>
    </div>
</div>
<div class="d-flex justify-content-end">
    <button class="btn btn btn-success mt-5 px-
5 "><h4>Зберегти</h4></button>
</div>
</div>
</form>
<script type = "text/javascript" src = "https://ajax.googleapis.com/ajax/libs/jquery
/1/jquery.min.js"></script>
<script>
    $('#file').change(function () {
        var input = $(this)[0];
        if (input.files && input.files[0]) {
            if (input.files[0].type.match('image.*')) {
                var reader = new FileReader();
                reader.onload = function (e) {
                    $('#img-preview').attr('src', e.target.result);
                }
                reader.readAsDataURL(input.files[0]);
            } else {
                console.log('ошибка, не изображение');
            }
        } else {
            console.log('хьюстон у нас проблема');
        }
    });
    $("textarea").keydown(function() {

```

```

    if (this.value.length > 350)
        this.value = this.value.substr(0, 350);
});
$('#form').bind('reset', function () {
    $('#img-preview').attr('src', 'default-preview.jpg');
});
</script>
@endsection
@include('layouts.footer')
@endsection

```

Лістинг Purchases/all.blade.php

```

@extends('layouts.appi')
@section('title')Усі закупівлі @endsection
@include('layouts.headers.header')
@section('content')
    @extends('inc.menus.purchase')
    @section('content in menu')
    <div class="container-fluid mb-5">
        <h1 class="text-dark text-center display-4 p-4 heading-
green">Усі закупівлі</h1>
        <table class="table table-hover mt-3 table-striped shadow table-white ">
            <thead>
                <th>@sortablelink('date_time', 'Дата і час')</th>
                <th>@sortablelink('suplier_id', 'Постачальник')</th>
                <th>@sortablelink('shop_id', 'Торгова точка')</th>
                <th>@sortablelink('storekeeper_id', 'Працівник')</th>
                <th>@sortablelink('sum', 'Сума')</th>
                <th style="width:10%"></th>
            </thead>
            <tbody>
                @foreach($supplies as $sup)
                    <tr onclick="location.href='{{route('onesupplie', $sup-
>id)}}'" title="Детальніше" style="cursor: pointer;">
                        <td>{{ $sup->date_time }}</td>
                        <td>{{App\Models\Suppliers::getSupplierName($sup-
>suplier_id)}}</td>
                        <td>{{App\Models\Shops::getShopNameById($sup->shop_id)}}</td>
                        <td>{{App\Models\Workers::getNameById($sup-
>storekeeper_id)}}</td>
                        <td>{{ $sup->sum.' грн '}}</td>
                        <td><a class="text-primary" href="{{route('onesupplie', $sup-
>id)}}">Детальніше</a></td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
    @endsection
    @include('layouts.footer')
    @endsection

```

Лістинг Workers/new_worker_form.blade.php

```

@extends('layouts.appi')
@section('title')Реєстрація@endsection
@include('layouts.headers.header')
@section('content')

```

```

<h1 class="text-dark text-center display-4 p-4 heading-
green">Реєстрація працівника</h1>
<div class="signup bg-form">
<div class="container d-flex justify-content-center my-0 ">

<form method="POST" action="{{route('workernew')}}" enctype="multipart/form-data" >
{{ csrf_field() }}
    @include ('layouts.alerts')

    <div class="logp">
        <div class="row ">
            <div class="col-md-5 ">
                
                <input type="file" class="inputfile " name="image" id="file">
                <label class="@error('image') is-
invalid @enderror" for="file">Вибрати фото</label>
                @error('image')
                <div class="invalid-feedback">{{ $message }}</div>
                @enderror
            </div>
            <div class="col-md-7 p-4 mt-3">
                <h5>Логін</h5>
                <input type="text" name="login" class="form-
control @error('login') is-invalid @enderror"
                placeholder="лат. символи і цифри" id="login" value="{{Request::old(
'login')?:''}}">

                @error('login')
                <div class="invalid-feedback">{{ $message }}</div>
                @enderror
                <h5>Пароль</h5>
                <input type="password" id="password" class="form-
control @error('password') is-invalid @enderror" name="password"
                placeholder="мін. 6 символів" >

                @error('password')
                <div class="invalid-feedback">{{ $message }}</div>
                @enderror
            </div>
        </div>
    </div>
    <div class="row border-bottom border-top p-4">
        <div class="col-md-4 col-sm-4">
            <h5>Прізвище</h5>
            <input type="text" class="form-control @error('last_name') is-
invalid @enderror" name="last_name"
            placeholder="Прізвище" value="{{Request::old('last_name')?:''}}">

            @error('last_name')
            <div class="invalid-feedback">{{ $message }}</div>
            @enderror

            <h5>Дата народження</h5>
            <input type="date" class="form-control @error('date_birth') is-
invalid @enderror" name="date_birth"
            min="1950-01-01" max="2006-01-
01" value="{{Request::old('date_birth')?:''}}">

            @error('date_birth')

```

```

        <div class="invalid-feedback">{{ $message }}</div>
    @enderror
</div>
<div class="col-md-4 col-sm-4">
    <h5 >Ім'я</h5>
    <input type="text" class="form-control @error('first_name') is-
invalid @enderror" name="first_name"
placeholder="Ім'я" value="{{Request::old('first_name')}?:"}}">

    @error('first_name')
    <div class="invalid-feedback">{{ $message }}</div>
    @enderror

    <h5>Номер телефону</h5>
    <input type="text" class="form-control @error('phone_number') is-
invalid @enderror" name="phone_number"
placeholder="+000(00)0000000" value="{{Request::old('phone_number')}?:"}}
}">

    @error('phone_number')
    <div class="invalid-feedback">{{ $message }}</div>
    @enderror
</div>
<div class="col-md-4 col-sm-4">
    <h5 >По-батькові</h5>
    <input type="text" class="form-control @error('father_name') is-
invalid @enderror" name="father_name"
placeholder="По-батькові" value="{{Request::old('father_name')}?:"}}">

    @error('father_name')
    <div class="invalid-feedback">{{ $message }}</div>
    @enderror

    <h5 >Попередній стаж</h5>
    <input type="text" class="form-control @error('previous_experience') is-
invalid @enderror"
name="previous_experience" placeholder="Поків" value="{{Request::old('pr
vious_experience')}?:"}}">

    @error('previous_experience')
    <div class="invalid-feedback">{{ $message }}</div>
    @enderror
</div><br>
<div class="row justify-content-center">
    <div class="col-md-4">
        <h5>Посада</h5>
        <select name="position_id" class="form-control @error('position_id') is-
invalid @enderror" >
            <option disabled selected>Вибрати</option>
            @foreach($positions as $position)
                <option>{{ $position->name}}</option>
            @endforeach
        </select>

        @error('position_id')
        <div class="invalid-feedback">{{ $message }}</div>
        @enderror

        <h5 >Дата прийняття на роботу</h5>

```

```

        <input type="date" class="form-control @error('date_hiring') is-
invalid @enderror"
            name="date_hiring" min="2010-01-
01" max="today" value="{{Request::old('date_hiring')}?:"}}>
        @error('date_hiring')
        <div class="invalid-feedback">{{ $message }}</div>
        @enderror
    </div>
    <div class="col-md-4">
        <h5>Магазин</h5>
        <select name="shop_id" class="form-control @error('shop_id') is-
invalid @enderror">
            <option disabled selected>Вибрати</option>
            @foreach($shop as $shop)
                <option>{{ $shop->name}}</option>
            @endforeach
        </select>
        @error('shop_id')
        <div class="invalid-feedback">{{ $message }}</div>
        @enderror
        <h5>Дата звільнення</h5>
        <input type="date" class="form-control" name="date_firing" min="2010-01-
01" max="today"
            value="{{Request::old('date_firing')}?:"}}>
    </div>
</div>
<div class="container my-5 d-flex justify-content-center">
    <button type="submit" class="btn btn-primary">Зареєструватись</button>
</div>
</form>
</div>
</div>
<script type = "text/javascript" src = "https://ajax.googleapis.com/ajax/libs/jquery
/1/jquery.min.js"></script>
<script>
    $('#file').change(function () {
        var input = $(this)[0];
        if (input.files && input.files[0]) {
            if (input.files[0].type.match('image.*')) {
                var reader = new FileReader();
                reader.onload = function (e) {
                    $('#img-preview').attr('src', e.target.result);
                }
                reader.readAsDataURL(input.files[0]);
            } else {
                console.log('ошибка, не изображение');
            }
        } else {
            console.log('хьюстон у нас проблема');
        }
    });

    $('#form').bind('reset', function () {
        $('#img-preview').attr('src', 'default-preview.jpg');
    });
</script>
@include('layouts.footer')

```

```
@endsection
```

Лістинг Workers.php

```
<?php
```

```
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\Auth;
use App\Models\Shops;
use Illuminate\Foundation\Auth\User as Authenticatable;

class Workers extends Authenticatable
{
    use HasFactory;
    protected $fillable = [

];
    public function positions(){
        return $this->belongsTo('App\Models\Positions','position_id','id');
    }
    public $timestamps = false;

    public function getAuthPassword()
    {
        return $this->password;
    }
    public function getName()
    {
        return "{$this->first_name} {$this->last_name}";
    }
    public static function getNameById($id)
    {
        $name= Workers::where('id',$id)->pluck('last_name')-
>first().' '.Workers::where('id',$id)->pluck('first_name')->first();
        return $name;
    }
    public static function getWorkerImage($id){
        $image=Workers::where('id',$id)->pluck('image')->first();
        return $image;
    }
    public function getWorkerId()
    {
        return $this->id;
    }
    public function getWorkerShop()
    {
        return $this->shop_id;
    }
    public function getWorkerPosition()
    {
        return $this->position_id;
    }
    public function GetConstantData(){
        if(Auth::check()){
            $data = array(
                'id' => Auth::user()->id,
                'name' => Auth::user()->getName(),
```

```
        'image' => Auth::user()->image,  
        'users_shop_name' => Shops::where('id',Auth::user()->shop_id)-  
>pluck('name')->first(),  
        'shop_id' => Auth::user()->getWorkerShop()  
    );  
    return $data;  
}  
else  
    $data = array(  
        'id' => 0,  
        'name' => 'Працівник',  
        'image' => 'uploads.workers/default-user-image.png',  
        'users_shop_name' => 'Магазин',  
        'shop_id' => null  
    );  
}  
}
```

Додаток В

Презентаційний матеріал

Кваліфікаційна робота бакалавра Автоматизована інформаційна система торгової точки з роздрібного продажу харчових товарів

Виконала: студентка 4 курсу групи КН-17-1 Гаврилук Д.Д.
Керівник: старший викладач кафедри КНІТ Скрипник Т.К.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
2021

Актуальність

Сьогодні комерція відіграє важливу роль для суспільства. У зв'язку з високим рівнем розвитку комерції, збільшенням кількості товарів на прилавках, збільшенням попиту виникло питання у автоматизації різних процесів продажів товарів за допомогою різних засобів ІТ, а саме у торгових точках з роздрібного продажу харчових товарів.

Таким чином, застосування ІТ для автоматизації процесів супроводу комерційної діяльності в області роздрібної торгівлі товарами харчового призначення є актуальною задачею на сьогодні. Особливо великою є потреба в автоматизації діяльності невеликих торгових точок, що займаються спеціалізованим продажем товарів певних категорій.

Використання електронного обліку дозволить автоматизувати багато процесів. Електронний облік полегшить процес документування процесів, дозволить з легкістю виконувати процедуру додавання нового продажу, ревізії, закупівлі, редагувати дані і т.д.. Також дозволить без додаткових економічних витрат організувати зберігання і використання існуючої інформації, а також автоматизувати дії по створенню нових даних. Дозволить легко отримувати потрібну інформацію.

Завдання

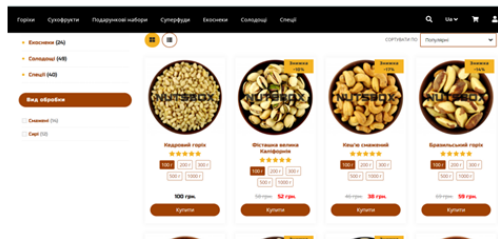
Мета роботи – розробка автоматизованої інформаційної системи торгової точки з роздрібного продажу харчових товарів.

- Для **клієнта торгової точки** необхідно автоматизувати: перегляд даних про торгові точки, перегляд каталогу товарів, перегляд статей.
- Для **адміністратора торгової точки** необхідно автоматизувати: авторизацію, продажі, закупівлі, облік товарів, каталог товарів, дані персоналу.
- Для **адміністратора системи** необхідно автоматизувати: доступ до всіх даних системи, активацію облікових записів адміністраторів торгової точки.

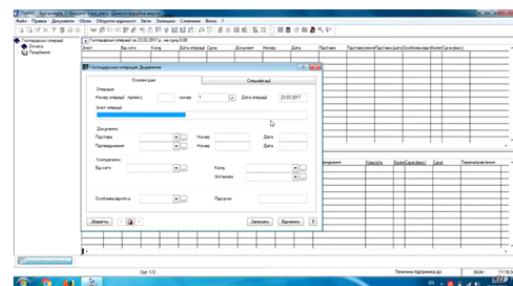
Відомі сучасні підходи до вирішення задачі

Для автоматизації діяльності магазинів з продажу харчових товарів використовуються різні види ІС, які можна розділити на наступні категорії:

- комплексні системи автоматизації діяльності;
- системи бухгалтерського обліку;
- спеціальні ПП;
- ПП для автоматизації діяльності невеликих торговельних точок з продажу харчових товарів.



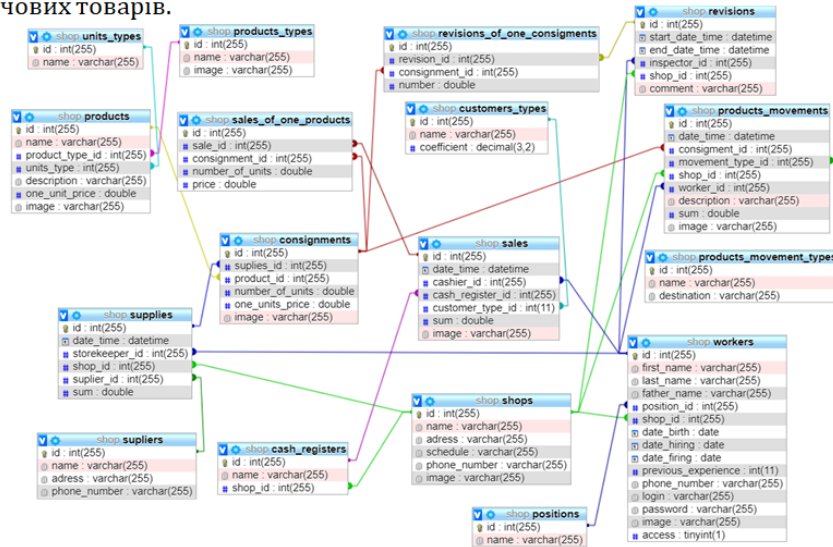
«Nutsbox»



«ПАРУС – Бухгалтерія»

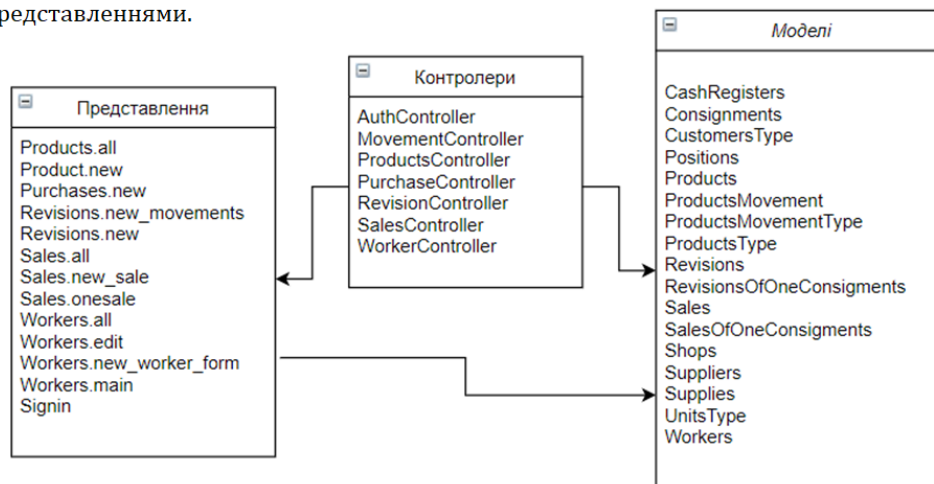
Схема бази даних

АІС торгової точки з роздрібною продажу харчових товарів є веб-додатком, що призначений для занесення до системи даних про продажі, закупівлі, облік товарів, товари та персонал. Тому було розроблено структуру бази даних, необхідну для функціонування АІС торгової точки з роздрібною продажу харчових товарів.



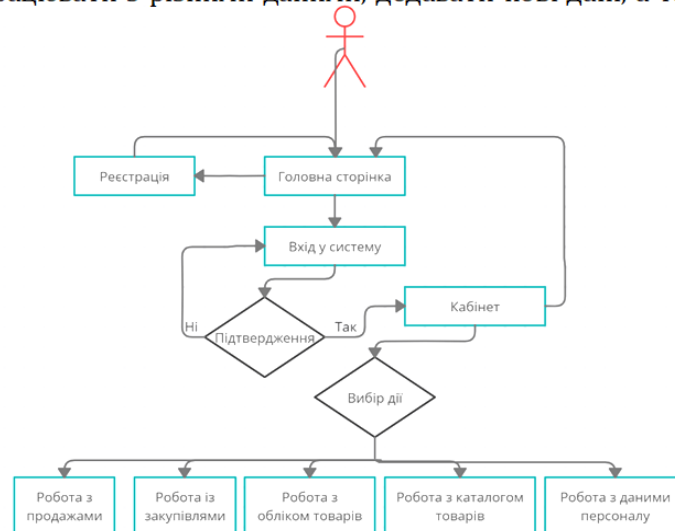
Діаграма класів

АІС використовує фреймворк Laravel, який реалізує шаблон проектування MVC. Суть цього шаблону полягає в розділенні програми на моделі, контролери та представлення. У моделі реалізується бізнес-логіка програми. У представленні знаходиться вивід даних системи. Контролер реалізує зв'язок між моделями та представленнями.



Діаграма активності роботи користувача із системою

Процес роботи користувача із сайтом включає в себе підтвердження особистості користувача, при успішній авторизації він потрапляє до системи, де він має можливість працювати з різними даними, додавати нові дані, а також змінювати існуючі.



Головна сторінка АІС

Східні Ласощі
автоматизована інформаційна система торгової точки з роздрібною продажу харчових товарів

Східні Ласощі
У нас ви можете придбати лише найсвіжіші та найбільш корисні горіхи, сухофрукти та прянощі. Завжди будемо раді бачити вас у нашому магазині.

Горіхи

Горіхи смачна і лісна їжа: у всіх горіхах міститься неповторний унікальний баланс вітамінів і мікроелементів, вони багаті складними білками, необхідними для тканин. Горіхи в 2,5-3 рази багатше фруктів по мінеральному складу вмісту калію, кальцію, магнію, фосфору, заліза та ін. В них багато білка (16-25%). Всі різновиди горіхів зберігають без всяких втрат свої вітаміни і поживні властивості не тільки протягом одного сезону, а набагато довше. У всіх горіхах міститься неповторний унікальний баланс вітамінів і мікроелементів. Вони багаті складними білками необхідними для тканин.

Форма реєстрації користувача у АІС та форма автентифікації

Реєстрація працівника

* - поля обов'язкові для заповнення

Вибрати фото... Вкласти

Логін*

має символ і цифри

Вибрати пароль... Показати

Пароль*

має 6 символів

Прізвище*

Ім'я*

По-батькові*

Дата народження*

Номер телефону*

Попередній стаж

Дата прийняття на роботу*

Дата звільнення

Магазин*

Посада*

Зареєструватися

Авторизація

Введіть логін та пароль

The password field is required.

Увійти

Перегляд та редагування даних усіх працівників

Усі працівники

Григорук Дмитро Сергійович
 1981-07-08
 +380679367829

Олексюк Марія Миколаївна
 1985-10-02
 +380983759456

Магазин:

Ласоща:

Магазин:

Сімей ласощі:

Посада:

Приймальник:

Посада:

Продавець:

Дата прийняття на роботу:

Дата прийняття на роботу:

Дата звільнення:

Дата звільнення:

Попередній стаж:

Попередній стаж:

показати

Редагувати дані

Закрити

Шехова Марина Леонідівна
 +3803 12 30

Змінити фото

Редагувати дані

Закрити

Редагування даних працівника

Прізвище

Ім'я

По-батькові

Дата народження

Номер телефону

Магазин

Посада

Дата прийняття на роботу:

Дата звільнення

Попередній стаж (років)

Зберегти зміни

Форма для додання нового продажу

The screenshot displays the 'Східні Ласощі' (Eastern Delicacies) POS system interface. The header includes the company name and a description: 'автоматизована інформаційна система торгової точки з роздрібного продажу харчових товарів'. The user is logged in as 'Марія Олексюк'. The main navigation bar contains 'Продажі', 'Закупівлі', 'Облік товарів', 'Каталог товарів', and 'Персонал'. The 'Продажі' section is active, showing 'Усі продажі' and 'Новий продаж' options. The 'Новий продаж' form is the central focus, featuring a green header and the following fields:

- Торгова точка: Східні ласощі
- Каса: 1
- Працівник: Марія Олексюк
- Назва товару: Вибірть товар (dropdown)
- Партія: (input field)
- Кількість: 0.00
- Ціна за 1 од. (грн): (input field)
- Сума: 0
- Знижка: Без знижки (dropdown)
- Сума зі знижкою: 0

A green checkmark icon is visible at the bottom right of the form.

Висновки

- В результаті виконання КРБ було розроблено АІС торгової точки з роздрібного продажу харчових товарів на платформі PHP, яка також має БД відповідно до вказаної предметної області.
- Розробка проводилась за допомогою мови програмування PHP із використанням шаблону MVC 5, фреймворку Laravel та СКБД MySQL.
- Для клієнта ІС було автоматизовано: перегляд даних про торгові точки, перегляд каталогу товарів, перегляд статей.
- Для адміністратора торгової точки було автоматизовано: авторизація, продажі, закупівлі, облік товарів, каталог товарів, персонал.
- Для адміністратора ІС було автоматизовано: доступ до всіх даних системи, активація облікових записів адміністраторів торгової точки.
- Розроблена АІС повністю готова до використання, тому в подальшому для неї можна придбати хостинг та налагодити обслуговування для реальних задач. У майбутньому систему можна вдосконалити додавши додаткові функції для клієнтів, а саме можливість авторизації та оформлення замовлення. Створений веб-додаток дозволяє спростити та автоматизувати роботу працівників торгових точок з роздрібного продажу харчових продуктів

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Гаврилюк Д. Д. на захист дипломного проекту (роботи)

(прізвище, ініціали)

за спеціальністю 122 - Комп'ютерні науки

На тему: АІС торгової точки з роздрібного продажу харчових товарів

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



САВЕНКО О. С.

(прізвище та ініціали)

ДОВІДКА УСПІШНОСТІ

Гаврилюк Д. Д. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки повністю виконав навчальний план спеціальності з такими розподілом оцінок за:

національною шкалою: відмінно 0,00 %, добре 18,75 %, задовільно 81,25 %.
шкалою ЄКТС: А 3,64 %, В 0,00 %, С 14,55 %, D 14,55 %, E 67,27 %.

Методист факультету

[Signature]

(прізвище та ініціали)

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент

Гаврилюк Д. Д. виконала кваліфікаційну роботу бакалавра на тему "АІС торгової точки з роздрібного продажу харчових товарів" у відповідності до завдання в повному обсязі

Оцінка дипломного проекту (роботи) добре

Керівник дипломного прокту (роботи)

[Signature]

(прізвище та ініціали)

" 08 " червня 2021 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Гаврилюк Д. Д. допускається до захисту цього

Завідувач кафедри

ЖМП

(назва)

[Signature]

(підпис, прізвище, ініціали)

" 09 " червня 2021 р.

08.06.2021

result_167483576292132457.html

Tue Jun 08 12:24:14 EEST 2021, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 19%

| | | | | |
|--|----------|---------|-------------------------------------|----------|
| ID: 92685
Название: АІС торгової точки з роздрібного продажу харчових товарів
Добавлено в БД: 2021-06-08
Авторы: Д.Д. Гаврилюк
Руководители: Т.К. Скрипник
Консультанты:
Опоненты: | Документ | | Суммарное совпадение по Базе Данных | |
| | Символы | Лексемы | Символы | Лексемы |
| | 51682 | 449 | 3711 (7%) | 45 (10%) |

Источник плагиата

| ID | Описание | Наличие плагиата в документе | |
|----|----------|------------------------------|---------|
| | | Символы | Лексемы |
| | | | |



Ім'я користувача:
Кафедра КН

Дата перевірки:
09.06.2021 12:12:56 EEST

Дата звіту:
09.06.2021 12:17:32 EEST

ID перевірки:
1008243352

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005671

Назва документа: 2021_КРБ_Гаврилюк_20210608_1_фінал_повний 02 Lite

Кількість сторінок: 54 Кількість слів: 7574 Кількість символів: 56763 Розмір файлу: 15.02 MB ID файлу: 1008315937

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.11%
Схожість

Найбільша схожість: 4.01% з джерелом з Бібліотеки (ID файлу: 1008302248)

3.72% Джерела з Інтернету 149

Сторінка 56

6.34% Джерела з Бібліотеки 45

Сторінка 57

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 18 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: АІС торгової точки з роздрібною продажу харчових товарів

Автор: Гаврилюк Дар'я Дмитрівна

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: Скрипник Тетяна Казимирівна, ст.викладач кафедри КНІТ

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що на мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає ____ і адресується до ____ першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Т.К. Скрипник

Гарант ОП

О. В. Мазурець

Завідувач кафедри КІСП

О. В. Бармак

Хмельницький національний університет
Кафедра комп'ютерних наук та інформаційних технологій

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студентки групи КН-17-1 Гаврилюк Дар'ї Дмитрівни

за темою «АІС торгової точки з роздрібного продажу харчових товарів».

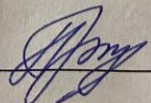
Актуальність і значення теми: розробка автоматизованої інформаційної системи торгової точки з роздрібного продажу харчових товарів дозволить забезпечити автоматизацію реєстрації та перегляду даних про продажі, закупівлі, ревізії, працівників.

1. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки: алгоритми оптимізації реєстрації та перегляду даних про продажі, закупівлі, ревізії, працівників використані доцільно, оскільки, одержана АІС виконує функції додавання, редагування, перегляду даних про продажі, закупівлі, працівників, ревізії та товари для адміністраторів торгових точок, перегляд даних про товари та торгові точки – для клієнтів, а також надання доступу користувачів до АІС – для адміністратора системи.

2. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність: розроблена «АІС торгової точки з роздрібного продажу харчових товарів» використовується для автоматизації реєстрації та перегляду даних про продажі, закупівлі, ревізії, працівників.

3. Загальний висновок та оцінка: вимоги поставленої задачі виконані в повному обсязі, була розроблена АІС торгової точки з роздрібного продажу харчових товарів на платформі РНР, яка використовує базу даних відповідно до предметної області, працює вірно, рекомендована оцінка «Відмінно».

Робота заслуговує на оцінку « добре »

Рецензент к. ер.-м. н., доц. Яценюк Н. О.  08.06.2021