

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр
(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН СИСТЕМИ СКЛАДСЬКОГО
ОБЛІКУ
SOFTWARE IMPLEMENTATION OF AN ONLINE SYSTEM FOR TRACKING
THE WAREHOUSE INVENTORY

Виконав: студент(ка) 4 курсу, групи КНТ-227
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

Колташева А.О.

(прізвище та ініціали)

Керівник Колпакова Т.О.

(прізвище та ініціали)

Рецензент Голуб Т.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
 (код і найменування)
 Освітня програма (спеціалізація) Комп'ютерні науки
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Колташевої Анастасії Олегівни

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація онлайн системи складського обліку. Software Implementation of an Online System for Tracking the Warehouse Inventory.

керівник проєкту (роботи) Колпакова Тетяна Олексіївна, к.т.н., доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року №103

2. Строк подання студентом проєкту (роботи) 17 травня 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми складського обліку. 2. Основні рішення щодо реалізації компонентів системи. 3. Опис технології програми та її розробка 4. Експлуатація та тестування програми.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Колпакова Т.О., доцент		
Нормоконтроль	Липовець М.В., асистент		

7. Дата видачі завдання “15” лютого 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз проблеми складського обліку	2-3 тижні	Розділ 1
3	Основні рішення щодо компонентів системи	4 тиждень	Розділ 2
4	Опис технології програми та її розробка	5-6 тижні	Розділ 3
5	Оформлення пояснювальної записки та документів до неї.	7 тиждень	Розділ 4
6	Нормоконтроль та рецензування	8 тиждень	Додатки
7	Захист роботи	9 тиждень	

Студент(ка)

_____ Колташева А.О.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

_____ Колпакова Т.О.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 132 с., 39 рис., 5 табл., 3 дод., 18 джерел.

СКЛАД, ОНЛАЙН СИСТЕМА, АВТОМАТИЗАЦІЯ, ПРОГРАМНА РЕАЛІЗАЦІЯ, ЗАБЕЗПЕЧЕННЯ, СКЛАДСЬКИЙ ОБЛІК, SOFTWARE, DEVELOPMENT, WEB-TECHNOLOGIES.

Об'єкт дослідження – програмне забезпечення для обліку складу онлайн.

Предмет дослідження – онлайн програми або сайти для автоматизації складського обліку.

Мета роботи – розробка програмної онлайн системи для зниження навантаження на персонал через автоматизацію трудомісткої роботи та для підвищення прибутку підприємства.

Матеріали, методи та технічні засоби: веб-програмування, технологія мови програмування PHP, HTML та CSS, с базою даних PostgreSQL, персональний комп'ютер з процесором Intel Core i7 під управлінням операційної системи Microsoft Windows 7.

Результати. Створено веб-сайт, що дозволяє здійснювати складський облік товарів, зручний та ефективний як для покупців так і для персоналу.

Висновки. Розроблена та реалізована програмна веб-система для підвищення ефективності та автоматизації товарообігу складського обліку на прикладі онлайн магазину, для забезпечення спрощення роботи персоналу і зручності користування покупцям.

Галузь використання – подібну програму можна використовувати будь-яким підприємствам, супермаркетам, різноманітним заводам, меблевим магазинам, продуктовим крамницям та звичайним споживачам, клієнтам, користувачам.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 132 pages, 39 picture, 5 tables, 3 appendixes, 18 sources.

WAREHOUSE, ONLINE SYSTEM, AUTOMATION, SOFTWARE IMPLEMENTATION, PROCURING, STRUCTURE, DEVELOPMENT, ACCOUNTING, WEB-TECHNOLOGIES.

The object of research is online software for warehouse accounting.

Subject of research is online programs for automation os warehouse accounting.

The purpose of the work is to develop an online software system to reduce the workload on staff through the automation of time-consuming work and to increase the company's profits.

Materials, methods, and tools: web programming, PHP, HTML and CSS programming language technology, with PostgreSQL database, personal computer with Intel Core i7 processor running Microsoft Windows 7 operating system.

Results. The website has been created that provides inventory of goods and is convenient and efficient for by both customers and staff.

Conclusions. The software web system has been developed and implemented. It increases the efficiency and automation of warehouse accounting in the online store, to simplify the work of staff and to provide ease of use to customers.

Field of use. The such a program can be used by any business, supermarket, various factories, furniture stores, grocery stores and regular consumers, customers, users.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми систем складського обліку.....	10
1.1 Огляд проблеми складського обліку.....	10
1.2 Актуальність теми складських онлайн систем	12
1.3 Аналіз існуючих розробок	13
1.4 Висновки за розділом 1	19
2 Основні рішення щодо реалізації компонентів системи.....	20
2.1 Функціональність програми відповідно до запропонованих критеріїв	20
2.2 Розробка структурної схеми компонентів системи онлайн забезпечення складського обліку	21
2.3 Заходи та рекомендації щодо вдосконалення онлайн системи.....	22
2.4 SWOT-аналіз існуючого способу ведення складського обліку.....	23
2.5 Мета і призначення програмного забезпечення	27
2.6 Висновки за розділом 2	28
3 Опис технології програми та її розробка.....	29
3.1 Обґрунтування вибору середовища розробки програми	29
3.2 Вибір мови програмування для розробки системи	29
3.2.1 JavaScript	31
3.2.2 PHP	32
3.2.3 CGI.....	32
3.2.4 DHTML.....	35
3.2.5 ASP	36
3.2.6 CSS.....	37
3.2.7 SSI.....	38
3.3 Характеристика модулів реалізованої програми	40
3.4 Висновки за розділом 3	44

4 Експлуатація та тестування програми	46
4.1 Характеристика результативної інформації застосування	46
4.2 Виконання розробленої програми	46
4.3 Висновки за розділом 4	55
Висновки	54
Перелік джерел посилання	55
Додаток А Технічне завдання	57
Додаток Б Текст програми	61
Додаток В Слайди презентації	124

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ASP– Active Server Pages;

CGI– Common Gateway Interface;

CSS– Cascading Style Sheets;

DHTML– Dynamic HyperText Markup Language;

HTML– HyperText Markup Language;

PERL– Practical Extraction and Reporting Language;

PHP– Register Transfer Level;

SQL– Structured Query Language;

SSI– Server Side Includes;

SWOT– Strengths, Weaknesses, Opportunities, Threats;

WEB– World Wide Web;

WMS– Warehouse Management System;

БД– база даних;

ОС– операційна система;

ПЗ– програмне забезпечення;

СКБД– система керування базами даних.

ВСТУП

З розвитком комп'ютерної техніки розвивається і розширюється сфера застосування інформаційних технологій. Сьогодні вона зачіпає всі сфери нашої діяльності. Все це завдяки зростанню комп'ютерної техніки, яка з кожним роком удосконалюється. У зв'язку з цим вимоги до програмного забезпечення стають все більш загальними в сенсі їх застосування, що гарно як для виробників, як і для користувачів. Набагато зручніше, коли підприємство автоматизовано, облаштовано певними не застарілими технологіями.

Автоматизація вже давно стала невід'ємним процесом по ефективності використання трудових та матеріальних ресурсів. На даний момент систем складського обліку безліч, але не кожна з них дозволяє легко відслідковувати і вести підрахунок товарів. Автоматизація складу – одна з головних задач обліку на підприємстві. Складські програми допомагають контролювати запаси і товарообіг, мінімізують втрати, дозволяють більш ефективно використовувати обладнання для зберігання [1].

Таким чином, автоматизована система управління складом вигідна всюди, де потрібно облік і зберігання будь-яких одиниць товарів, будь то архіви, інтернет-магазини, поштові відправлення та інше. Практика показує, що система автоматизації складу користується попитом не тільки серед великих підприємств, а й на невеликих складах або архівах.

Щодня роль автоматизації в повсякденної і виробничої життєдіяльності людини тільки зростає, дозволяючи нам задовольняти власні потреби, кількість яких постійно збільшується. В останні роки у зв'язку з помітним зменшенням природних ресурсів, тривожними глобальними змінами клімату, роль автоматизації є дуже актуальною, через що вона не припиняє проникати в нові сфери людської діяльності.

Тому, я вважаю, що автоматизація, мабуть, єдине і найкраще рішення в поліпшенні якості та вирішенні питання продуктивності праці.

1 АНАЛІЗ ПРОБЛЕМИ СИСТЕМ СКЛАДСЬКОГО ОБЛІКУ

1.1 Огляд проблеми складського обліку

Склади в Україні почали розвиватися в 18-19-их століттях. Перші з них були одноповерховими, їх робоча висота становила 3-3,5 метра, а підлога зазвичай знаходилася на рівній землі. За радянських часів в країні було побудовано безліч складських комплексів, ступінь їх технічного оснащення залежала від галузевої приналежності. Найбільш передові технології впроваджувалися в складах оборонного комплексу, інші, зокрема для товарів народного споживання, оснащувалися за залишковим принципом. У 1980-х років з'явилися перші автоматизовані комплекси. У кінці 1990-х років – початку 2000-х на український ринок вийшли європейські та американські постачальники технологічного обладнання і програмних продуктів для оснащення складів. Почалося стрімке зростання складського господарства, що супроводжувалось впровадженням інноваційних технологічних і технічних рішень [2].

Сьогодні склади вже оснащуються сканерами штрих-кодів, терміналами збору даних, принтерами етикеток, електронними вагами та іншими комп'ютерними технологіями та обладнанням. А всіма процесами управляє єдина інформаційно-програмна система, яка автоматизує діяльність. Такий підхід дозволяє оптимізувати всі процеси, знижує втрати від впливу людського фактору.

Але у кожного підприємства є проблеми в роботі складського господарства, їх може бути безліч, і всі разом вони призводять до значних сукупних втрат, змушуючи керівництво задуматися про шляхи їх подолання. Автоматизація роботи складу починається з вивчення проблемних місць. Тільки після виявлення «вузьких місць» в роботі складу можна розробити результативну програму для складського обліку. За допомогою WMS (Warehouse Management System) систем ведення складу можна вдосконалити управління процесами складського обліку, а також підвищити прозорість та

ефективність складських процесів. В кінцевому підсумку, підприємство отримує інструменти для грамотного управління, аналізу, планування і контролю над роботою складу [3].

WMS-система – це програмне забезпечення, призначене для автоматизації управління процесами складу і роботи складського комплексу в цілому [4].

Розглянемо основні причини, які спонукають торгові підприємства впроваджувати у себе системи автоматизації складського обліку і управління складом. Найчастіше зустрічаються такі укрупнені групи проблем, які вирішуються за допомогою WMS-систем:

- організаційні (розподіл функцій, повноважень, зон відповідальності персоналу і підрозділів);
- технологічні (послідовність і методологія виконання операцій);
- інформаційні (інформаційні системи обліку та обробки даних, засоби комунікації);
- технічні (наявність, адекватність і ступінь зношеності технічних ресурсів).

Таким чином, завдяки автоматизації можна вирішити наступні задачі:

- систематизувати номенклатурний довідник;
- можливість отримувати у будь-якій обставині, у будь-якому місці інформацію про наявність товару на складі в режимі реального часу;
- управління складським простором;
- збільшення числа складських операцій;
- підвищення продуктивності складу в цілому.

Варіантів вирішення проблем складського обліку може бути безліч. В першу чергу, це може бути впровадження технологічного проекту складу, який дозволяє змінити принципи його роботи. Це може бути також реорганізація праці співробітників, перерозподіл обов'язків. Якщо на підприємстві вже функціонує корпоративна інформаційна система, можливо, її можна модернізувати або доповнити новими модулями для організації

складського обліку. Найбільш складним, але в той же час ефективним, є автоматизація складського господарства підприємства [5].

WMS-система дозволяє змінити принципи зонування складського приміщення, траєкторію переміщення запасів і рівень завантаженості персоналу. Дозволяє розрахувати кількість стелажів і вибрати їх оптимальний тип. Грамотна організація зберігання матеріальних запасів підприємства дозволяє оптимізувати роботу обладнання та перегрупувати різноманітні запаси для полегшення доступу до них [6].

Отже інформацію про реальну кількість товарів на складі, яку може надати складська система, критично важлива при розподілі товарів на замовлення та їх резервування за сотнями покупців. Знати місце розташування на складі конкретних партій і навіть одиниць товару необхідно, наприклад, для точності контролю термінів придатності харчових продуктів, відстеження серійних номерів товарів, що підлягають сервісному обслуговуванню.

Автоматизація складу торгової організації – завдання творче, що залежить від безлічі різноманітних і взаємопов'язаних факторів. Потреби самого підприємства, специфіка його роботи, наявність суміжних складських функцій, умови зовнішнього середовища – все це впливає на процес впровадження складської системи. Саме поєднання випробуваних проектних технологій автоматизації та обліку особливостей конкретного підприємства, робить можливим досягнення бажаного результату в заплановані терміни при оптимальному використанні ресурсів [7].

1.2 Актуальність теми складських онлайн систем

На сьогоднішній день сучасний склад – складне технічне спорудження, призначене для управління запасами на різних ділянках логістичного ланцюга і виконання конкретних функцій по зберіганню і перетворенню матеріального потоку в цілому [8].

Управління асортиментом, замовленнями і клієнтською базою, фінансовий і складський облік – обов’язкові умови для нормального розвитку інтернет-магазину, користування яким, є дуже актуальним на даний нелегкий період часу, котрий тяжко переживає наше людство. Більшість магазинів з одягом, побутовими товарами, товарами для дому перейшли на дистанційний формат, для того щоб сприяти врегулюванню цієї непростой ситуації, подбати про здоров’я населення, тим самим обмеживши робочий персонал від звичайного контакту в людях, накопичення великої кількості людей в сучасних маркетах і будь-яких магазинах. Це є досить ефективним рішенням, тому що ніхто так і не з’ясував, чи дійсно працюють рекомендації місцевих органів охорони здоров’я для попередження поширення COVID-19. Тому автоматизація торгівлі і грамотний підхід до даних процесів значно спрощують роботу компанії. Також це ще і гарна можливість, оперативного обслуговування клієнтів і доступу до актуальної аналітики.

Основні вимоги на вибір сервісу складського обліку для інтернет-магазину:

- простота освоєння. В ідеалі розібратися з функціоналом складської системи користувач або спеціаліст повинен освоїти за максимально короткий час;
- оперативність техпідтримки;
- стабільність роботи. Система працює без перебоїв, якщо це не вплив зовнішніх факторів.

1.3 Аналіз існуючих розробок

Кількість і функціональність програмних застосувань на 2021 рік становить широкий спектр діяльності.

Необхідно провести аналіз функцій існуючих програм, які можуть бути використані для вирішення поставлених завдань:

- можливість роботи в програмі без установки;

- можливість працювати в сервісі онлайн;
- можливість налаштування інтерфейсу;
- підтримка баз даних (БД) SQL;
- зображення до предмета обліку;
- пошук по базі даних;
- права і користувачі;
- облік робочого місця;
- облік ремонтів;
- облік техніки;
- облік витратних матеріалів.

Що повинна вміти хороша програма складського обліку. Вибираючи програмне забезпечення (ПЗ) товарної системи, слід звернути увагу на наступні моменти:

- сумісність з іншими програмами та обладнанням. Розумна програма повинна інтегруватися з онлайн-магазинами, онлайн-касами, онлайн-бухгалтеріями та іншими програмами;
- оновлення в режимі реального часу. Проданий, переміщений або списаний товар повинні відразу списувати з залишків;
- можливість завантаження і вивантаження бази в інші сервіси. Наприклад, якщо налаштувати спільну роботу програми з певною системою, то можна буде персоналізувати знижки і програми лояльності;
- використання хмарних технологій. Краще, якщо дані зберігаються на сервері постачальника послуги. Це дозволяє звертатися до інформації з будь-якого пристрою, підключено до мережі інтернет.

Для порівняльного аналізу було обрано такі три сервіси складського обліку: Incrates, Контур.Маркет, IBS Торгівля і склад.

Назва: Incrates

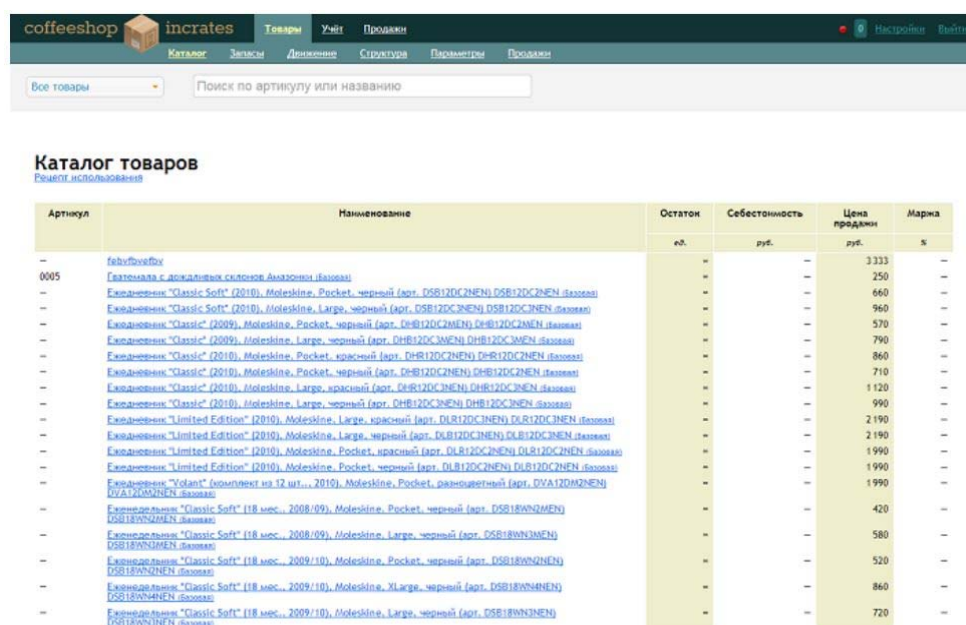
Вартість товарообігу системи Incrates становить 373 грн. в місяць.

Призначення: для інтернет-магазину InSales. Забезпечує повну сумісність в двох напрямках.

Можливості:

- прогнозування продажів (система вчасно нагадує про те, що товар закінчується);
- розрахунок собівартості при замовленнях в різний час у різних постачальників;
- можливість робити товарні набори і розраховувати їх вартість (система сама визначить прибуток від такого комплекту).

Інтерфейс програми «Incrates» наведений на рисунку 1.1.



The screenshot shows the 'coffeeshop incrates' interface. At the top, there are navigation tabs: 'Товари' (Goods), 'Увіт' (Exit), 'Продажи' (Sales), 'Каталог' (Catalog), 'Заказы' (Orders), 'Движения' (Movements), 'Структура' (Structure), 'Параметры' (Parameters), and 'Продажи' (Sales). Below the navigation bar is a search bar with the text 'Поиск по артикулу или названию' (Search by article or name). The main section is titled 'Каталог товаров' (Goods Catalog) and contains a table with the following columns: 'Артикул' (Article), 'Наименование' (Name), 'Остаток' (Stock), 'Себестоимость' (Cost), 'Цена продажи' (Price), and 'Маржа' (Margin). The table lists various coffee-related items, including 'Классический кофе' (Classic coffee) and 'Кофе' (Coffee), with their respective stock levels, costs, and prices.

Артикул	Наименование	Остаток	Себестоимость	Цена продажи	Маржа
		ед.	руб.	руб.	%
0005	Классический кофе	—	—	3333	—
—	Классический кофе	—	—	250	—
—	Классический кофе	—	—	660	—
—	Классический кофе	—	—	960	—
—	Классический кофе	—	—	570	—
—	Классический кофе	—	—	790	—
—	Классический кофе	—	—	860	—
—	Классический кофе	—	—	710	—
—	Классический кофе	—	—	1120	—
—	Классический кофе	—	—	990	—
—	Классический кофе	—	—	2190	—
—	Классический кофе	—	—	2190	—
—	Классический кофе	—	—	1990	—
—	Классический кофе	—	—	1990	—
—	Классический кофе	—	—	1990	—
—	Классический кофе	—	—	1990	—
—	Классический кофе	—	—	420	—
—	Классический кофе	—	—	580	—
—	Классический кофе	—	—	520	—
—	Классический кофе	—	—	860	—
—	Классический кофе	—	—	720	—

Рисунок 1.1 – Інтерфейс програми «Incrates»

Назва: Контур.Маркет

Вартість:

- тариф «Базовий» – від 4700 рублів за рік за одну точку продажів.

Цей тариф дозволяє вести товарний облік і відправляти дані на онлайн-касу, друкувати цінники і складати звіти про продажі. У послугу також входить інтеграція 1С та цілодобова технічна підтримка;

- тариф «Оптимальний» – від 9500 рублів за одну точку в рік. У плані є все те, що і в «Базовому» плюс можливість інвентаризації, розрахунок рентабельності за період, товарний звіт і доступ до переміщення товарів.

Призначення: зазвичай система працює в зв'язці з онлайн-касою. Але вона дуже багатофункціональна для будь-якого бізнесу.

Можливості:

- онлайн-облік товарних і складських залишків;
- інтеграція товарів з онлайн-касою (система зв'язується за касовим апаратом і в чеку пробивається найменування);
- можливість друку цінників і генерації штрих-кодів;
- онлайн-аналітика продажів по кожній торговій точці;
- сумісність з усіма популярними сервісами;
- просте завантаження накладних;
- зручний особистий кабінет керівника.
- створення великої кількості звітів.

Інтерфейс програми «Контур.Маркет» наведений на рисунку 1.2.

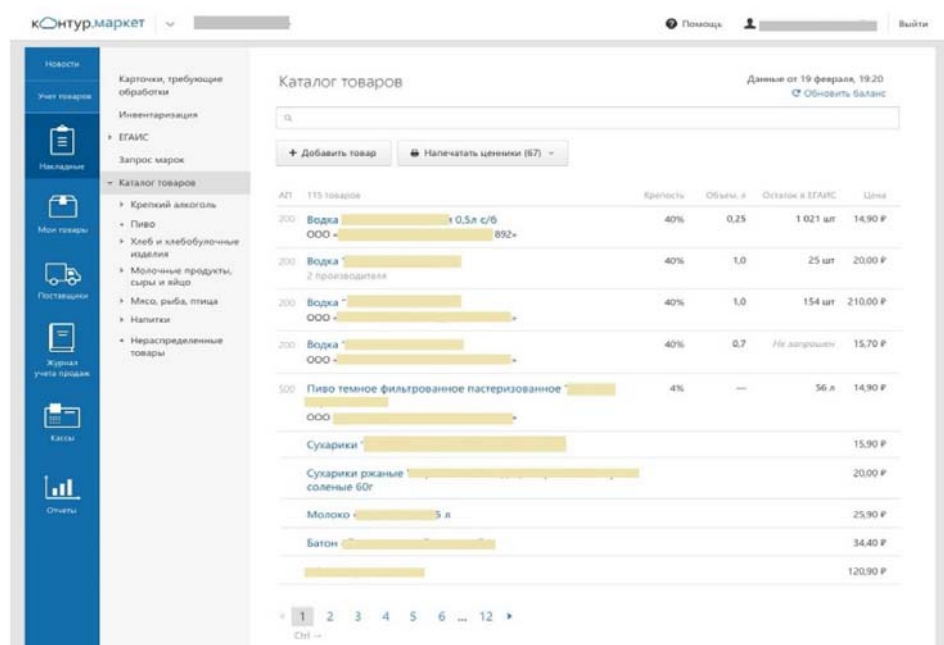


Рисунок 1.2 – Інтерфейс програми «Контур.Маркет»

Назва: IBS Торговля і склад

Вартість:

- оренда програми на місяць становить 750 грн., установка і навчання – включено у вартість;

– купівля постійної ліцензії «IBS Торгівля і Склад» становитиме 2950 грн, установка і навчання – включено у вартість, термін дії – без обмеження.

Призначення: програма підходить як для окремого складу, - продуктів, будматеріалів, авто-запчастин, одягу, - так і для розгалуженої складської мережі оптової компанії або складу виробничого підприємства.

Можливості:

- необмежена кількість підтримуваних внутрішніх фірм (юридичних осіб, підрозділів);
- необмежена кількість підтримуваних місць зберігання (складів);
- докладний довідник товарів і послуг з багатоповірхневою класифікацією і можливістю створювати нові атрибути;
- детальна класифікація контрагентів і їх торгових точок по видам (магазин, кіоск тощо).

Інтерфейс програми «IBS Торгівля і склад» наведений на рисунку 1.3.

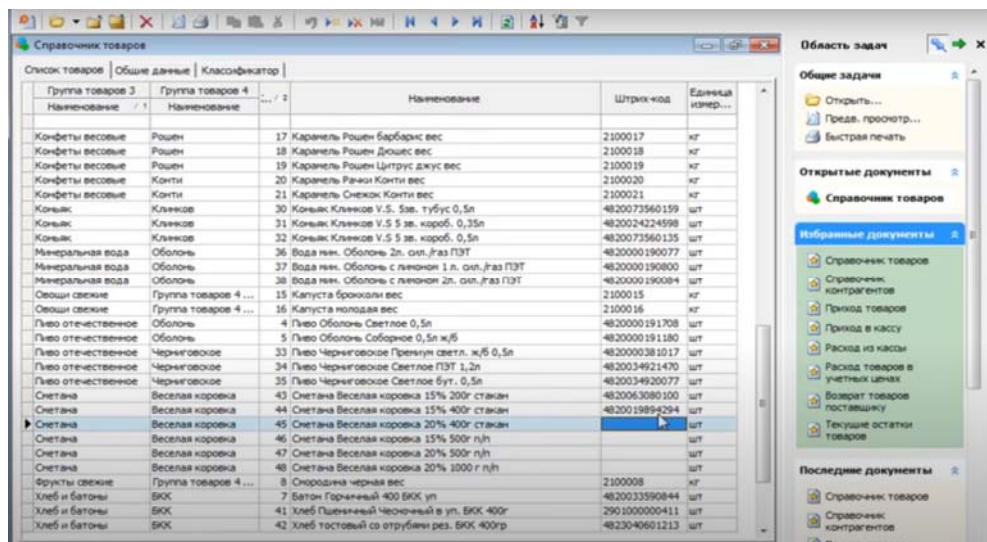


Рисунок 1.3 – Інтерфейс програми «IBS Торгівля і склад»

Порівняння розглянутих аналогів представлено в таблиці 1.1.

Таблиця 1.1 – Порівняння аналогів

Критерії порівняння	Incrates	Контур.М аркет	IBS Торгівля і склад
Загальні			
Робота в ОС Windows	+	+	+
Робота в ОС Linux	-	-	-
Вартість (USD)	14\$ в місяць	60-130\$ в рік	28\$ в місяць; 106\$ - безлімітне використання
Можливість працювати у програмі без її установки	-	+	+
Функціональні			
Облік техніки	+	+	+
Облік товарів	+	+	+
Можливість працювати на сервері онлайн	-	+	+
Підтримка баз даних SQL	+	+	+
Права та користувачі	+	+	+
Пошук по БД	-	+	+
Історія змін	-	+	+
Облік інвентарних номерів	+	+	+
Можливість налаштування інтерфейса	-	-	+
Створення власних шаблонів	+	-	+
Вбудовані друковані форми і звіти	+	+	-
Облік ремонтів	-	-	+
Облік витратних матеріалів	-	-	-
Всього:	8	10	13

В результаті порівняння був зроблений наступний висновок: на ринку існує багато аналогічних програм даного спектра діяльності, але вони не завжди надають усіх необхідних нам функцій. Потрібні функції або відсутні, або є тільки в платних версіях. Великі сумніви виникають при масштабованості зазначених платних програм, за додаткові робочі функції.

Спільні недоліки програм:

- складний інтерфейс, вимагає навчання персоналу;
- вартість цих програмних забезпечень;
- не всі програми можна доробляти, та змінювати якісь функції.

У зв'язку з тим, що при виборі потрібно враховувати, як функціонал так і вартість продукту, тому з порівняльного аналізу слід зробити вибір на користь сервісу «IBS Торгівля і склад».

1.4 Висновки за розділом 1

Розглянуті у першому розділі веб-застосунки та програми для реалізації складського обліку «Incrates», «Контур.Маркет», «IBS Торгівля і склад» мають небагато недоліків, але і недостатньо вагомих переваг. Єдиною перевагою є зручність у використанні в режимі онлайн та у відмінному функціонуванні на дистанційному режимі. Недоліки оглянуто у підрозділі 1.3 та наведено у таблиці 1.1.

2 ОСНОВНІ РІШЕННЯ ЩОДО КОМПОНЕНТІВ СИСТЕМИ

Науково-технічний прогрес і динаміка зовнішнього середовища змушують сучасні підприємства перетворюватися в більш складні системи, для яких необхідні нові методи забезпечення керованості. Можна стверджувати, що ефективний облік сучасного підприємства можливий тільки з централізованої онлайн системою обліку [9].

Вимоги, що пред'являються до систем обліку, які не залежать від форми власності та сфери діяльності підприємства, а її функціональні можливості повинні відповідати вимогам підприємства.

Для будь-якої інформаційної онлайн системи є необхідний перелік обов'язкових технічних вимог:

- швидкодія;
- надійний захист від несанкціонованого доступу до даних і реєстрації дій персоналу;
- зручний призначений для користувача інтерфейс;
- можливість конвертації даних, що використовувалися в минулих версіях в нову систему;
- висока надійність роботи.

2.1 Функціональність програми відповідно до запропонованих критеріїв

Впродовж огляду та аналізу в першому розділі функцій, які повинні відповідати вимогам систем для обліку інтернет-магазину або будь-якого підприємства було визначено 8 кращих показників якості онлайн застосунку:

- простота освоєння;
- оперативність техпідтримки;
- стабільність роботи;
- функціонал – чим він більший, тим краще;

- можливість роботи в програмі без установки;
- можливість працювати в сервісі онлайн;
- підтримка баз даних (БД) SQL;
- пошук по базі даних.

2.2 Розробка структурної схеми компонентів системи онлайн забезпечення складського обліку

Розроблюваний онлайн застосунок складається із декілька частин, які повинні взаємодіяти один з одним. Це вказує на його модульну структуру, що дозволить, при бажанні, у подальшому легко модернізувати його, або замінити будь-який модуль без глобальної перебудови всього забезпечення. Структурна схема розроблюваного онлайн системи з веденням складського обліку наведена на рисунку 2.1.



Рисунок 2.1 – Структурна схема онлайн забезпечення

Взагалі онлайн застосунок складається із трьох основних частин – клієнтська платформа, адмінська платформа та серверна частина. Серверна частина в свою чергу містить три головних модуля:

- модуль збору клієнтських заказів, який виконує обов’язок передавати дані про закази замовника з сайту;
- база даних, що зберігає як дані замовлень покупців, так і дані складу (список товарів);
- модуль сортування та обробки заказів, який створений і доступний тільки для адміна або керівника сайтом, який сортирує та відправляє закази клієнтам, займається онлайн складським обліком.

2.3 Заходи та рекомендації щодо вдосконалення онлайн системи

Основною метою створення нової системи є об’єднання розрізнених модулів компанії в одну для складання зручних онлайн звітів. Більшість компаній, бізнес продуктів і магазинів вручну заповнюють інформацію щодо складу, різноманітних накладних. Також малоприємним фактором є доступ до файлу, який зберігається на локальній машині і редагується однією людиною і окремо від самого онлайн підприємства.

Реалізація веб-сервісу автоматизації процесів ведення обліку, переміщення та складання звітів повинна забезпечувати наступні можливості:

- поділ авторизованих користувачів на 2 ролі: адміністратор і користувач;
- розмежування функціоналу для ролей;
- внесення, редагування і видалення даних в базу даних через форму на веб-сторінці;
- пошук необхідних даних;
- відстеження дати зміни в таблицях;
- функціонування при використанні браузерів;

– забезпечення безперервної та стабільної роботи в момент використання.

Застосунок має бути сумісний з будь-яким пристроєм, який забезпечує вихід в інтернет через браузер, але з урахуванням потужностей сервера, на якому запущено застосунок.

Програма повинна забезпечувати можливість виконання перерахованих функцій для адміністратора і користувача в представленій нижче таблиці.

Перелік основних функцій онлайн застосунка розглянуто у таблиці 2.1.

Таблиця 2.1 – Функції забезпечення для адміністратора і користувача

Функції системи	
Користувач	Адміністратор
Використання інтерфейсу	Додавання, редагування та видалення записів, інформації щодо товарів
Перегляд цікавих товарів	Перегляд сформованих таблиць та звітів організаційного характеру
Формування замовлення	Ведення зручного каталогу товарів
Заповнення особистих даних для оформлення замовлення	Формування звітів

2.4 SWOT-аналіз існуючого способу ведення складського обліку

SWOT-аналіз – це метод оцінки внутрішніх і зовнішніх факторів, які впливають на розвиток компанії або проекту. Ця методика допоможе оцінити сильні і слабкі сторони підприємства, знайти нові можливості і визначити можливі загрози [10].

SWOT-аналіз часто використовують при стратегічному плануванні зміни, але я вважаю цей метод зручний у використанні навіть незважаючи на не надто масштабні планування змін. Перші дві букви в аббревіатурі S (сильні

сторони) і W (слабкі сторони) відносяться до внутрішніх факторів, а значить, організація може їх легко оцінити.

Приклади областей, які, як правило, враховують:

- фінансові ресурси – це фінансування, можливості отримання доходу;
- фізичні ресурси – це обладнання компанії, будівлі, місцеположення;
- людські ресурси – співробітники, цільова аудиторія;
- доступ до природних ресурсів, авторські права, патенти.

Вважається, що компанії не можуть контролювати і оцінювати зовнішні фактори, поки внутрішні не оцінені об'єктивно. Вплив зовнішніх факторів на кожну організацію і окрему людину дуже сильний. Важливо знати і документувати найбільш значущі з них. Зовнішні чинники - це, як правило, ті обставини, котрі компанія не може контролювати.

Щоб знайти сильні сторони програми необхідно відповісти на наступні питання:

- які переваги має застосунок?
- що застосунок робить краще, ніж всі інші?
- які сильні сторони застосунка бачать клієнти?
- яка у застосунку унікальна торгова пропозиція?
- як застосунок може збільшити прибуток компанії?

Необхідно розглянути сильні сторони, як з внутрішньої точки зору компанії, так і з точки зору її клієнтів.

Недоліки:

- що може поліпшити застосунок?
- які слабкі сторони бачать клієнти програми?
- які фактори зменшують прибуток від програми?
- що конкуренти роблять краще?

Для того щоб виявити доцільність розробки веб-застосунку необхідно провести SWOT-аналіз наявного, звичного для людей, способу ведення

ручного обліку (таблиця 2.2) і порівняти його з аналізом ведення обліку засобами веб-застосунку (таблиця 2.3).

Аналіз позитивних та негативних факторів ручного обліку був представлений у таблиці 2.2.

Таблиця 2.2 – SWOT-аналіз ведення обліків обладнання ручного засобу

	Позитивні фактори	Негативні фактори
	Сильні сторони (внутрішній потенціал) (S)	Слабкі сторони (внутрішні недоліки) (W)
Внутрішнє середовище	Звичний, перевірений і надійний метод заповнення необхідної інформації вручну. Необхідність користування будь-якою програмою з таблицями окремо від сайту компанії. Немає потреби в потужному обладнанні.	Застарілий формат ведення документації (якщо це таблиці паперового формату). Необхідні базові вміння користування комп'ютером і ПЗ. Розрізнена структура файлів. Немає інтеграції з іншими системами. Немає вбудованих шаблонів звітності. Відсутній «розрахований на багато користувачів» режим. Відсутнє розмежування прав доступу.
Зовнішнє середовище	Можливість передачі інформації звітів на носіях або за коштами локальної та глобальної мережі.	Відсутність конфіденційності інформації. Безпека інформації.

Так само необхідно провести SWOT-аналіз ведення обліку онлайн засобами веб-застосунку. Результати даного аналізу представлені на таблиці 2.3.

Таблиця 2.3 – SWOT-аналіз ведення обліків засобами веб-застосунку

	Позитивні фактори	Негативні фактори
	Сильні сторони (внутрішній потенціал) (S)	Слабкі сторони (внутрішні недоліки) (W)
Внутрішнє середовище	Безкоштовно, доступ до системи онлайн. Простота поводження з програмою. Здатність швидко навчатися співробітникам. Немає потреби в потужному обладнанні. Можливість розмежування прав доступу. Вбудовані шаблони звітів, динамічні звіти. Інтеграція з іншими системами.	Базові вміння поводитися з комп'ютером і ПЗ. Досить незвичний вигляд програми.
	Потенційні можливості (O)	Наявні загрози (T)
Зовнішнє середовище	Скорочення часу на обробку даних. Можливість передачі інформації на носіях і за коштами локальної та глобальної мережі. Управління правами доступу до інформації. Робота в системі з будь-якої точки доступу до інтернету. Незалежність від більшості зовнішніх факторів.	Збій системи. Поломка устаткування. Відсутність інтернет з'єднання

Завдяки проведеним SWOT-аналізу існуючої і розроблюваної системи ведення обліку, виділяються ключові фактори, які вказують на доцільність розробки нового продукту.

Перший фактор – вартість. На даний момент компанії пропонують лише частину функціоналу безкоштовно або на певний проміжок часу, найчастіше зовсім ігноруючи безкоштовний період. Необхідно враховувати в увазі масштабованість будь-якого підприємства, і додаткова покупка функціоналу або робочих місць може сильно вдарити по витратам на ведення обліку.

Другий фактор – незалежність. Через можливість часткових або повних блокувань інтернет-ресурсів, варто врахувати фактор незалежності від даних інцидентів. При можливому виникненні конфліктів розробників програмного забезпечення з державою, будь-який сайт, а також їх продукт може бути заблокований. Маючи власний продукт на своїх серверах, компанія убезпечить себе від збоїв в роботі сервісу з обліку, що зменшить ризик втрати даних, які так само можуть привести до недостовірної і не актуальної інформації.

2.5 Мета і призначення програмного забезпечення

Веб-застосунок призначений для зручного способу придбання товарів користувачами, для автоматизації та полегшення ведення обліку певних матеріалів, а також складання певних видів звітів. При розробці програми акцент робиться на простоту інтерфейсу для швидкого освоєння співробітниками нової програми. Незважаючи на простий вигляд, в застосунку реалізовано весь необхідний функціонал, який на відміну від багатьох, дозволяє керувати певним бізнесом і вести облік товарів, зберігати дані накладних, а також зберігати архівні дані на одному і тому ж сайті компанії. Ніякого ручного заповнення обліків, ведення журналів і іншої «паперової тяганини». Застосунок забезпечує зберігання і обробку

необхідних даних, формування звітів, оперативне та актуальне надання інформації для клієнта про вигляд і стан потрібного їм товару.

Веб-застосунок дозволяє скоротити час на внесення запису про прихід обладнання, також прискорює пошук облікових одиниць за різними критеріями. З'являється можливість оперативно відстежувати переміщення облікових одиниць. Головною перевагою є онлайн доступ до потрібної інформації.

2.6 Висновки за розділом 2

У другому розділі було визначено та розглянуто такі характеристики і вимоги щодо створення онлайн застосунку:

- необхідний перелік обов'язкових технічних вимог для будь-якого інформаційного онлайн забезпечення;
- визначені кращі показники якості онлайн застосунку;
- розроблено структурну схему компонентів системи веб-забезпечення та охарактеризовано його функції для користувача та адміністратора;
- проведено SWOT-аналіз ведення обліків обладнання ручного засобу та засобами за допомогою веб-застосунків;
- також була визначена мета програмного забезпечення.

3 ОПИС ТЕХНОЛОГІЇ ПРОГРАМИ ТА ЇЇ РОЗРОБКА

3.1 Обґрунтування вибору середовища розробки програми

До етапу фізичного проектування переходять після обрання цільової системи керування веб-програмування, яке буде визначати особливості майбутнього програмного продукту. Визначитися із мовою веб-програмування необхідно до початку етапу фізичної розробки продукту.

Фізичне проектування – це уточнення рішень із урахуванням наявних розробників технологій, можливостей реалізації та необхідної продуктивності. З цією метою розробник повинен виконати наступні дії:

- розробка загального уявлення;
- реалізація бізнес-логіки баз даних;
- розробка внутрішніх даних сайту;
- створення таблиць та зв'язків між ними.

Застосунок, з точки зору бачення користувача, повинно здаватися якомога простіше. Для цього необхідно грамотно вибудувати внутрішню структуру програми на кожному етапі його проектування. Інформаційна система повинна бути легко супроводжуваною і керованою. Створення інформаційної системи являє собою достатньо складний і багатоступінчастий процес.

Інформаційне моделювання – процес опису предметної області або побудови моделі предметної області в тому вигляді або форматі, який, з одного боку, легко сприймається людиною, з іншого боку, легко може бути перетворений в набір елементів інформаційного сховища, програмних компонентів і інших складових прикладного програмного забезпечення [11].

3.2 Вибір мови програмування для розробки системи

Успіх функціонування програмного онлайн середовища найтіснішим чином пов'язаний із веб-програмуванням, завдяки якому буде реалізовано

проект. Отже, вибір відповідної платформи багато в чому визначає переваги та недоліки веб-системи. Існує широкий набір критеріїв оцінки веб-технологій, виконавши які проєктувальник зможе зупинити свій вибір на конкретному програмному продукті.

На першому кроці необхідно визначити тип моделі проєкту. Проаналізувавши в підрозділі 2.1 другого розділу всі необхідні функції і критерії, а також вивчивши порівняльні характеристики, наведені в табл. 2.2., і аналізи позитивних і негативних факторів, наведені в табл. 2.3, було прийнято рішення обрати програмне середовище розробки веб-застосунків для розробки інформаційної підсистеми зі складським обліком.

Значення веб-технології, як для розробників програмного забезпечення, так і для користувачів багато в чому залежить від того, що це перш за все – інтеграційна технологія. І важко знайти більш вдалий приклад того, як можна інтегрувати різні джерела інформації і різні її типи. Веб-технології дозволяють створювати простіші для освоєння легкодоступні, вкрай дешеві, швидко оновлювані, інформаційні, діалогові, довідкові системи.

На сьогоднішній день існує велика кількість різноманітних засобів розробки застосунків за допомогою веб-технологій. Функції та можливості кожної веб-технології можуть істотно відрізнятися. Різні структури у мережі Інтернет створюють власні інформаційні моделі завдяки веб-сайту, формуючи певні інформаційно-економічні простори засобами інтелектуалізованого програмного забезпечення (мультіагентні системи або програмні агенти як представники – агенти суб'єктів економічної діяльності) у глобальному електронному середовищі або об'єктно орієнтованого ПЗ (сайт, портал, електронна поштова скринька тощо) [12].

Існує декілька популярних веб-технологій, які можна рекомендувати для застосування в системі, яка передбачає не тільки ведення складського обліку товарів, але ще і необхідний інтерфейс для користувачів.

3.2.1 JavaScript

JavaScript – це мова програмування, що використовується в складі сторінок HTML для збільшення можливостей. Він був розроблений фірмою Netscape на базі мови Sun's Java корпорації Sun 10 років тому. Кажуть, що Java - мова програмування і платформа розробки № 1. Java знижує витрати, скорочує терміни розробки, стимулює інновації і покращує роботу застосунків.

JavaScript є надбудовою стандарту HTML і значно розширює можливості html-документа, створеного з використанням цієї технології. JavaScript інтегрується в файл HTML у вигляді кількох рядків коду (наприклад, це може бути функція, що викликається на виконання спеціальної командою). Вбудований в браузер інтерпретатор JavaScript сприймає і скрипт, і сам HTML-код як єдиний документ, обробляючи і ті, і інші дані одночасно.

За допомогою технології JavaScript можна надати своїй сторінці елементи інтерактивності, формувати, компоувати і повністю контролювати формат спливаючих вікон і вбудованих фреймів.

Використання цих технологій не вимагає установки і налаштування на сервері будь-яких додаткових модулів, оскільки скрипти і аплети виконуються безпосередньо на комп'ютері користувача.

Переваги:

- жодний сучасний браузер не обходиться без підтримки JavaScript;
- з використанням написаних на JavaScript плагінів і скриптів впорається навіть не фахівець;
- корисні функціональні настройки;
- постійно удосконалюється мова - зараз розробляється бета-варіація проекту, JavaScript2;

- взаємодія з застосунком може здійснюється навіть через текстові редактори - Microsoft Office і Open Office.

Недоліки:

- знижений рівень безпеки через повсюдного і вільного доступу до вихідного коду популярних скриптів;

- безліч дрібних дратівливих помилок на кожному етапі роботи. Велика частина з них легко виправляється, але їх наявність дозволяє вважати цю мову менш професійним, порівняно з іншими.

- повсюдне поширення. Своєрідним недоліком можна вважати той факт, що частина активно використовуваних програм, особливо програм, перестануть існувати при відсутності мови, оскільки цілком базуються на ньому.

3.2.2 PHP

PHP (Personal Home Page tools) – скриптова мова програмування з відкритими початковими кодами, яка призначена для додання веб-сторінок елементів інтерактивності. Код, написаний на мові PHP, вбудовується в документ HTML подібно підпрограмі: в ту ділянку документа, де необхідно розмістити інтерактивний елемент, просто вставляється сценарій PHP. Мнемоніка даного мови базується на синтаксисі PERL, Java і C, завдяки чому не викликає яких-небудь труднощів при вивченні. Методики, які дозволяють серверам коректно розпізнавати файли, що містять скрипти PHP, різні і залежать в першу чергу від типу конкретного сервера [13].

Технологія PHP дозволяє організовувати на веб-сторінці лічильник відвідувань, підраховувати статистику звернень до тих або інших розділів сайту, захистити доступ до будь-якого html-документу паролем і багато іншого.

Переваги:

- розробка за допомогою PHP дає багато можливостей. При належному рівні володіння можна створювати не тільки сценарії для веб-застосунків, але й повноцінні програми. Існують рішення, що дозволяють створювати мобільні застосунки на PHP;
- вивчення PHP не вимагає багато часу, але дуже докладне знання вимагає практики;
- кросплатформність. PHP може бути запущений в будь-якій операційній системі;
- підтримка веб-серверів, складно знайти той, який би не працював з PHP;
- безкоштовне розповсюдження. Можливо, PHP не був такий популярний для створення веб-застосунків, якби не був безкоштовним;
- безперервний розвиток. Те, що сьогодні про шаблонізатор знають так багато, означає лише одне: з недоліками, рано чи пізно, впораються.

Недоліки:

- вузькопрофільність. І хоча можливості розширені різними реалізаціями, все ж він «заточений» переважно під програмування для Інтернету;
- непослідовний синтаксис - при вивченні мови PHP, особливо старої частини, заснованої на функціях, можна помітити, що частина функцій має префікси `array_`, `str_`, частина не має;
- протиріччя в коді. Коли шаблонізатор був тільки створений, все програмне забезпечення розроблялося за допомогою C. Тому в мові було запозичено багато з його синтаксису.

3.2.3 CGI

Технологія CGI (Common Gateway Interface) має на увазі використання в складі ресурсу Інтернету інтерактивних елементів на базі застосунків, що забезпечують передачу потоку даних від об'єкта до об'єкта.

Спрощено принцип роботи CGI виглядає наступним чином: наприклад, користувач заповнює на веб-сторінці ту чи іншу форму і натискає на кнопку, після чого інформація з форми передається в CGI-скрипт, який запускається на виконання і обробляє отриману інформацію. Результат роботи CGI-скрипту може бути виданий (вбудований) в іншу html-сторінку, відправлений поштою або трансформований яким-небудь іншим способом. Як правило, скрипти CGI розміщуються на сервері в спеціально відведений для цієї мети директорії, якій призначається ім'я CGI-BIN.

Технологія CGI звичайно реалізується двома методами: або з використанням програм, написаних на мові PERL (Practical Extraction and Report Language), - як правило, такі файли мають розширення .pl, або за допомогою застосунків, створених зазвичай із застосуванням мови C і відкомпільованих безпосередньо на сервері, оскільки більшість UNIX-сумісних платформ включають в себе вбудований транслятор цієї мови. Подібні програми мають розширення .cgi. Необхідно відзначити, що PERL, як і HTML, є мовою, що інтерпретується, але він інтерпретується не браузером користувача, а сервером перед передачею даних користувачеві. Для того, щоб запустити подібний скрипт на виконання, на веб-сервері не потрібно додаткової компіляції, необхідна лише наявність встановленого на сервері інтерпретатора відповідної мови [14].

Переваги:

- процес CGI скрипту не залежить від веб-сервера і, в разі падіння, жодним чином не позначиться на роботі останнього;
- може бути написаний будь-якою мовою програмування;
- підтримується більшістю веб-серверів;
- незалежність від клієнтського програмного забезпечення.

Недоліки:

Головний недолік полягає в тому, що для встановлення та налаштування застосунків CGI на сервері потрібно володіти як мінімум правами адміністратора, оскільки ці програми при запуску здатні порушити нормальне функціонування серверного комп'ютера і дестабілізувати роботу

мережі, саме тому не всі служби хостингу надають можливість використовувати CGI-скрипти.

3.2.4 DHTML

DHTML (Dynamic Hyper Text Markup Language, динамічна мова розмітки гіпертексту) є розширенням стандарту HTML і дозволяє створювати веб-сторінки, що включають такі інтерактивні елементи, як фон, що рухається під статичним вмістом документа, рухомі об'єкти, меню, що випадають, кнопки, що підсвічуються при наведенні курсора миші, анімацію, титри, що біжать, і багато іншого. За великим рахунком DHTML є «середнім арифметичним» між технологіями HTML і JavaScript. Цей стандарт використовує прості сценарії, підготовлені за допомогою інтерпретованої макромови, що оброблюється браузером спільно з кодом HTML. Такі сценарії називають «скріплет». Для створення скріплетів використовуються стандартні розширення DHTML і будь-яка макромова, що підтримує директиви інтерфейсу ActiveX. DHTML розпізнається браузерами Microsoft Internet Explorer, починаючи з версії 4.0 і вище.

Переваги:

- dHTML завантажує контент на льоту, завантажується не вся сторінка, а тільки частина вмісту, яку необхідно змінити, що економить час користувачів і надає сайту гідний вигляд;
- модулі не потрібні – dHTML використовує більшість функцій, вже доступних в браузерах, тому немає необхідності завантажувати плагіни;
- відмінний інструмент – динамічні функції, якими володіє dHTML, допомагають веб-дизайнерам створювати веб-сайти з компактним дизайном, швидким завантаженням, графічними ефектами, розширеними функціональними можливостями і можуть зберігати набагато більше тексту або контенту одночасно.

Недоліки:

- дорогі інструменти редагування – dHTML забезпечує відмінну функціональність, але доступні на ринку редактори досить дорогі;
- довге і складне кодування – кодування dHTML є довгим і складним. Тільки досвідчені програмісти на JavaScript і HTML можуть писати і редагувати їх з хорошим рівнем функціональності;
- проблеми з підтримкою браузера – dHTML страждає від проблем з підтримкою браузера для різних браузерів. Наприклад, код, написаний для Netscape, може не працювати в Internet Explorer і навпаки. Проблема виникає через різницю функцій браузера.

3.2.5 ASP

ASP (Active Server Pages, активні сторінки сервера) – ще одна технологія, аналогічна JavaScript і PHP, але на цей раз розроблена всім відомою компанією Microsoft. Для того щоб зробити веб-сторінку інтерактивною із застосуванням технології ASP, необхідно вбудувати в її код відповідний скрипт, написаний на макромові, яка віддалено нагадує Java і C. Скрипт інтерпретується і виконується безпосередньо на сервері, після чого призначеному для користувача браузеру відправляється вже готовий html-документ з результатами роботи сценарію ASP. Звідси випливає цілком справедливий висновок про те, що для сторінок, що містять ASP, не має значення, яке програмне забезпечення встановлено на комп'ютері користувача. Зате принципове значення має тип сервера, на якому ви плануєте використовувати ASP, оскільки аж ніяк не всі вони підтримують дану технологію [15].

Переваги:

- ASP значно скорочує обсяг коду, необхідного для створення великих застосунків;
- завдяки вбудованій аутентифікації Windows налаштування для кожної програми захищені;

- він забезпечує більш високу продуктивність завдяки використанню переваг раннього зв'язування, своєчасної компіляції, вбудованої оптимізації і служб кешування прямо з коробки.

Недоліки:

- логіка тісно пов'язана з кодом, і, таким чином, їх складно розділити, тобто немає чіткої межі між рівнями застосунку;
- складно проводити модульне тестування;
- великий розмір сторінок.

3.2.6 CSS

Найчастіше у веб-дизайнера виникає необхідність застосувати в процесі створення html-документа складне форматування – від абзацу до абзацу міняти шрифт, розташування тексту, його колір, формувати різні таблиці даних. Можна вирішити цю проблему за допомогою стандартних засобів HTML: описувати кожен абзац окремим набором команд, але в цьому випадку підсумковий документ буде мати великий розмір, та й саме створення коду стає дуже трудомісткою роботою. Можна піти іншим шляхом: включити в сторінку опис CSS або підключити зовнішній файл, виконаний з стандарті CSS - Cascading Style Sheets (каскадні таблиці стилів), в якому за допомогою спеціального макромови один раз жорстко поставити форматування сторінки. Іншими словами, файл CSS виконує роль такого собі шаблону, який застосовується для форматування тексту, таблиць та інших елементів в документі HTML. Є можливість підключати один і той же фізичний файл CSS до різних веб-сторінок сайту. CSS можна використовувати на будь-якому сервері без будь-яких обмежень, оскільки команди CSS виконуються безпосередньо на комп'ютері користувача. До недоліків даної технології можна тільки віднести відсутність підтримки CSS старими браузером (Internet Explorer і Netscape Navigator нижче 4-ої версії) і трохи різний набір властивостей CSS, підтримуваний останніми версіями цих двох браузерів [16].

Переваги:

- кілька дизайнів сторінки для різних пристроїв перегляду;
- зменшення часу завантаження сторінок сайту за рахунок перенесення правил представлення даних в окремий CSS-файл;
- простота подальшої зміни дизайну - не потрібно правити кожену сторінку, а лише змінити CSS-файл;
- додаткові можливості оформлення;
- корисні для невеликої кількості стилів.

Недоліки:

- різне відображення верстки в різних браузерах (особливо застарілих), які по різному інтерпретують одні й ті ж дані CSS.
- найпоширеніша необхідність на практиці виправляти не тільки один CSS-файл, але і теги HTML, які складним і коханим способом пов'язані з селекторами CSS, що іноді зводить нанівець простоту застосування єдиних файлів стилів і значно подовжує час редагування та тестування.

3.2.7 SSI

SSI (Server Side Include) – технологія, яка виконується на веб-сервері, тісно переплетена зі згаданою вище CGI. На основі макромови, що дуже нагадує C ++, SSI дозволяє реалізувати такі можливості, як висновок в документі того чи іншого тексту в залежності від певних умов або згідно заданому алгоритму, формувати файл HTML з динамічно змінюються фрагментів або вбудовувати результат роботи CGI в будь-якої його ділянку.

Свого часу SSI був проривом, але зараз його майже витіснив PHP. Однак, незважаючи на це, деякі сайти досі використовують цю мову, наприклад, в зв'язці з CGI-скриптами.

Переваги:

- невелике навантаження на сервер;
- зручно реалізуються деякі функції;
- підтримка умов.

Недоліки:

- немає підтримки циклів;
- немає можливості роботи з файлами;
- необхідність використання файлів специфічного розширення.

Усі розглянуті особливості кожної технології веб-програмування впорядковано у звідну таблицю 3.1.

Таблиця 3.1 – Порівняльна характеристика параметрів веб-технологій

Веб-технологія	Розробник	Операційна система	Ліцензія	Стабільність	Популярність	Підтримка
JavaScript	Sun Microsystems и Oracle	Windows, Mac OS, Solaris	GPL, BCL	+	+	За гроші
PHP	PHP Group, Zend Technologies	Unix або Linux, OS X та Microsoft Windows	Вільне і відкрите програмне забезпечення, дозволена ліцензія	+	+	Безкоштовно
CGI	Weta Digital	Unix або Linux, Windows, MAC	Комерційна	+	-	Безкоштовно
DHTML	WHATWG	Windows, MAC	AGPL	+	-	За гроші
ASP	Microsoft	Microsoft Windows	Пропрієтарна (Власницьке програмне забезпечення)	+	-	За гроші
CSS	CSS Working Group	Microsoft Windows, Linux, Mac OS	GPL	+	+	Безкоштовно
SSI	Storage Solutions	Unix	Комерційна	+	-	За гроші

За результатами таблиці можна зробити висновок, що при проєктуванні веб-застосунку із складським обліком слід у першу чергу обернути увагу на скриптову мову PHP, яка є затребуваною вже близько 15 років і не втрачає своєї популярності і корисності для розробників.

Головна область застосування PHP – написання скриптів, що працюють на стороні сервера. На PHP роблять сайти і веб-застосунки – блоги, інтернет-магазини або соціальні мережі. Сьогодні на PHP працює майже 38 мільйонів сайтів - і серед них всесвітньо відомі Facebook, Wikipedia і «ВКонтакте».

PHP є рішенням для малих та середніх застосувань. Ця мова відмінно підходить для динамічного формування сторінок на основі інформації з бази даних. Також, на відміну від JavaScript, мова програмування PHP, виконується на стороні сервера і не залежить від програмного забезпечення клієнта. Тому програма написана на мові PHP буде виконана в будь-якому випадку.

PostgreSQL – вільна об'єктно-реляційна система керування базами даних (СКБД), тобто використовується, як середовище, з допомогою якого будується база даних.

3.3 Характеристика модулів реалізованої програми

Отже, для розробки веб-сервісу зі складським обліком, було обрано середовище розробки таких веб-застосунків, за допомогою якого вдалося збудувати проєкт:

- скриптова мова PHP 7.4;
- HTML (служив для побудови основи-скелету веб-сайту);
- CSS (займається безпосередньо оформленням сайту);
- база даних PostgreSQL 11 (дозволяє PHP робити запити в БД);
- використання хостингу, на якому зберігаються всі дані проєкту;

- домен, що зареєстрований на сайті freenom.tk, на три місяці безкоштовно);
- також в деяких випадках використовується JavaScript.

Обрана платформа для продукту, як і середовище його розробки є повністю безкоштовною для користування клієнтами та для ведення складського обліку адміністратором та співробітниками, в межах одного веб-сайту, внаслідок чого додаткові витрати на покупку ліцензій не буде потрібно. Дуже чітко простежується економічний фактор, який проявляється в тому, що весь готовий онлайн застосунок, в кінцевому рахунку, спрямований на поліпшення використання трудових і часових ресурсів співробітників та легкість і простоту у користуванні, не виходячи з дома, якщо ти клієнт компанії.

Структура файлів розроблюваної системи представлена на рисунку 3.1.

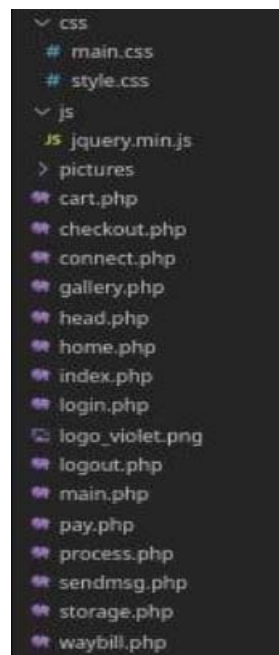


Рисунок 3.1 – Структура файлів проекту

На рисунку 3.2 наведено перелік таблиць в базі даних. Найменування кожного рядка вказує на призначення для функцій з загального меню сайту.



Рисунок 3.2 – Таблиця, що відповідає за бази даних

Узагальнена схема бази даних зі зв'язками наведена на рисунку 3.3.

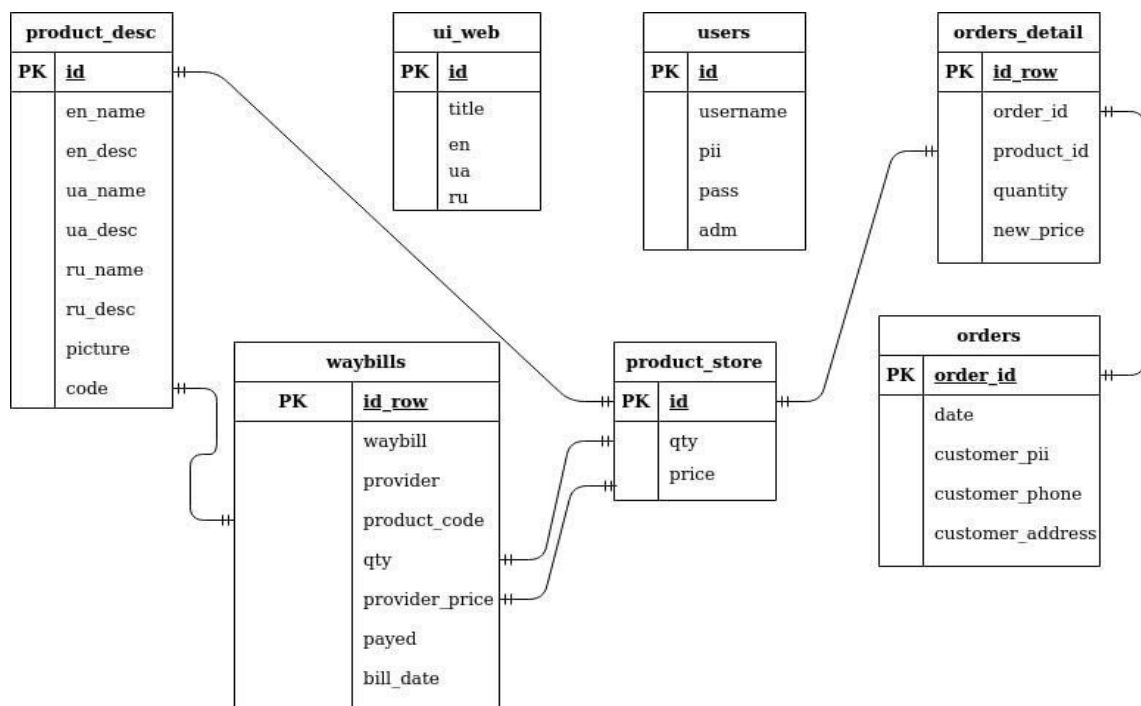


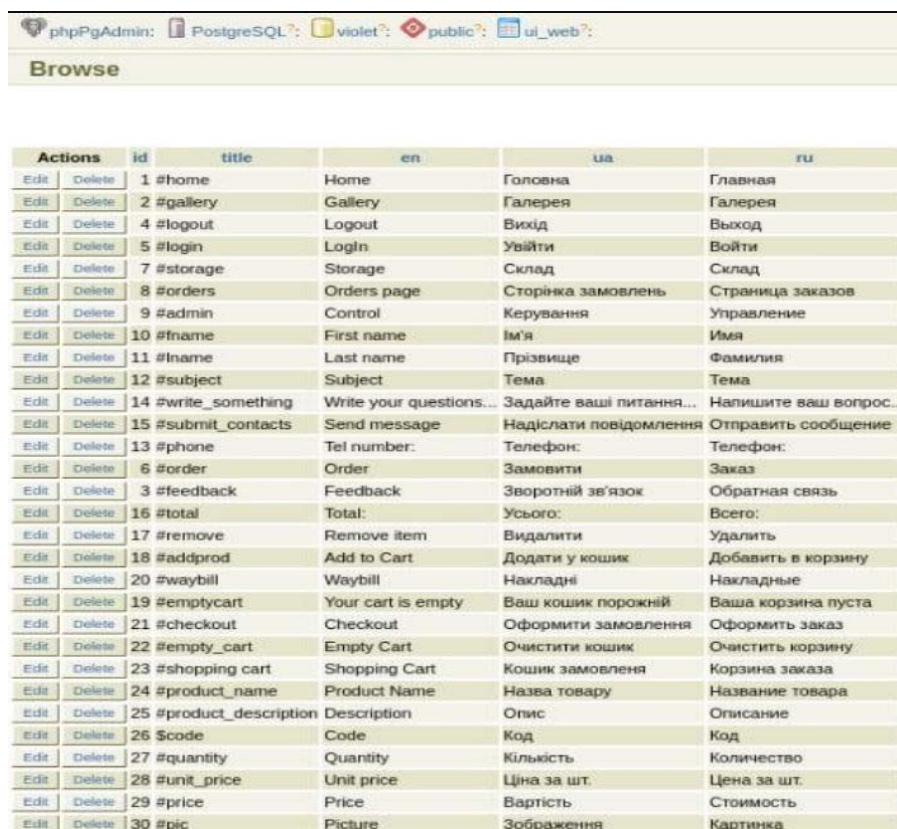
Рисунок 3.3 – Схема бази даних веб-застосунку

Розроблена база даних складається з наступних таблиць:

- таблиця «orders_details» відповідає за деталі замовлення, в ній є поле id – це ключ, який встановлює зв'язок цієї таблиці з таблицею «product_store». Ця таблиця розв'язує проблему зв'язку «багато до багатьох» між «product_store і «orders» і містить додаткові дані про замовлення, «quantity» та «new price»;

- таблиця замовлень «orders», зберігає дату замовлення, номер і адресу покупця, а також його id;
- таблиця продуктів «product store» містить ціну і кількість товарів;
- таблиця «product desk» містить всю інформацію про продукт, його назву на різних мовах, картинку і код;
- таблиця «waybills», у якій зберігаються накладні і необхідна інформація щодо накладних;
- таблиця користувачів «users» з логіном і паролем. Булеве поле адмін, визначає роль користувача, тобто адмін він, або звичайний користувач;
- таблиця «ui_web» містить усі сторінки сайту і служить для зберігання назв сторінок на різних мовах.

На рисунку 3.4 наведено структуру, що відповідає за локалізацію сторінок на різні користувацькі мови. Передбачено використання української, англійської, російської мов).



Actions	id	title	en	ua	ru
Edit Delete	1	#home	Home	Головна	Главная
Edit Delete	2	#gallery	Gallery	Галерея	Галерея
Edit Delete	4	#logout	Logout	Вихід	Выход
Edit Delete	5	#login	Login	Увійти	Войти
Edit Delete	7	#storage	Storage	Склад	Склад
Edit Delete	8	#orders	Orders page	Сторінка замовлень	Страница заказов
Edit Delete	9	#admin	Control	Керування	Управление
Edit Delete	10	#fname	First name	Ім'я	Имя
Edit Delete	11	#lname	Last name	Прізвище	Фамилия
Edit Delete	12	#subject	Subject	Тема	Тема
Edit Delete	14	#write_something	Write your questions...	Задайте ваші питання...	Напишите ваш вопрос...
Edit Delete	15	#submit_contacts	Send message	Надіслати повідомлення	Отправить сообщение
Edit Delete	13	#phone	Tel number:	Телефон:	Телефон:
Edit Delete	6	#order	Order	Замовити	Заказ
Edit Delete	3	#feedback	Feedback	Зворотній зв'язок	Обратная связь
Edit Delete	16	#total	Total:	Усього:	Всего:
Edit Delete	17	#remove	Remove item	Видалити	Удалить
Edit Delete	18	#addprod	Add to Cart	Додати у кошик	Добавить в корзину
Edit Delete	20	#waybill	Waybill	Накладні	Накладные
Edit Delete	19	#emptycart	Your cart is empty	Ваш кошик порожній	Ваша корзина пуста
Edit Delete	21	#checkout	Checkout	Оформити замовлення	Оформить заказ
Edit Delete	22	#empty_cart	Empty Cart	Очистити кошик	Очистить корзину
Edit Delete	23	#shopping cart	Shopping Cart	Кошик замовлення	Корзина заказа
Edit Delete	24	#product_name	Product Name	Назва товару	Название товара
Edit Delete	25	#product_description	Description	Опис	Описание
Edit Delete	26	#code	Code	Код	Код
Edit Delete	27	#quantity	Quantity	Кількість	Количество
Edit Delete	28	#unit_price	Unit price	Ціна за шт.	Цена за шт.
Edit Delete	29	#price	Price	Вартість	Стоимость
Edit Delete	30	#pic	Picture	Зображення	Картинка

Рисунок 3.4 – Загальний вигляд структури сайту

У наведеній нижче структурі зберігається довідник електронного кошика на різних мовах, назва, код і картинка до кожного товару. Це надає можливість зберігати і використовувати дані про замовлення на підприємстві з метою фіксування особистих кабінетів, контактів і телефонів, закріплених за клієнтом.

Структура збереження товарів зображена на рисунку 3.5.



Actions	id	en_name	en_desc	ua_name	ua_desc	ru_name	ru_desc	picture	code
Edit Delete	2	Violet Variant	Violet Variant test flower	Фіалка Варіативна	Фіалка Варіативна квітка д...	Фиалка Вариативная		pictures/3.jpg	v2
Edit Delete	3	Rum Punch	NULL	Ромовий пунш	NULL	Ромовый пунш	NULL	pictures/54.jpg	rm54
Edit Delete	4	Saintpaulia grotei	NULL	Гроте	NULL	Гроте	NULL	pictures/25.jpg	rm12
Edit Delete	5	Optimara MyLove	NULL	Фіалка Оптимара	NULL	Фиалка Оптимара	NULL	pictures/14.jpg	rmv14
Edit Delete	6	Melodie Kimi		Мелодії Кімі		Мелодии Кими		pictures/61.jpg	rmv61
Edit Delete	1	Taboo	Violet Regular Test flower	Фіалка Табуу	Фіалка звичайна тестова кв...	Фиалка Табуу		pictures/2.jpg	v1
Edit Delete	7	AV-Tango	NULL	Фіалка АВ-Танго	NULL	Фиалка АВ-Танго	NULL	pictures/46.jpg	jm46
Edit Delete	8	Rob's Dandy Lion	NULL	Фіалка Роб Денді	NULL	Фиалка Роб Денди	NULL	pictures/3.jpg	robs3
Edit Delete	9	Gillian		Фіалка Джиллиан		Фиалка Джиллиан		pictures/8.jpg	jul8
Edit Delete	10	Ramblin' Dots		Фіалка Рамблін Дотс		Фиалка Рамблин Дотс		pictures/28.jpg	sd12
Edit Delete	11	Boo Man		Буу Мен		Буу Мен		pictures/47.jpg	bo47
Edit Delete	12	Shining Bell		Сяючий Дзвіночок		Сияющий Колокольчик		pictures/49.jpg	sdf49
Edit Delete	13	Marshmallow		Зефір		Зефир		pictures/50.jpg	mall50
Edit Delete	14	Blue Chiffon		Синій шифон		Синий шифон		pictures/44.jpg	gtr44
Edit Delete	15	White Queen		Біла Королева		Белая Королева		pictures/43.jpg	que43

Рисунок 3.5 – Форма електронної таблиці з товарами

В структурі приватних кабінетів зберігаються ID адміна і користувача, призначені для авторизації. Ця структура формується напівавтоматично. Для входу в систему дані необхідно заповнювати вручну.

Форма кабінетів представлена на рисунку 3.5.

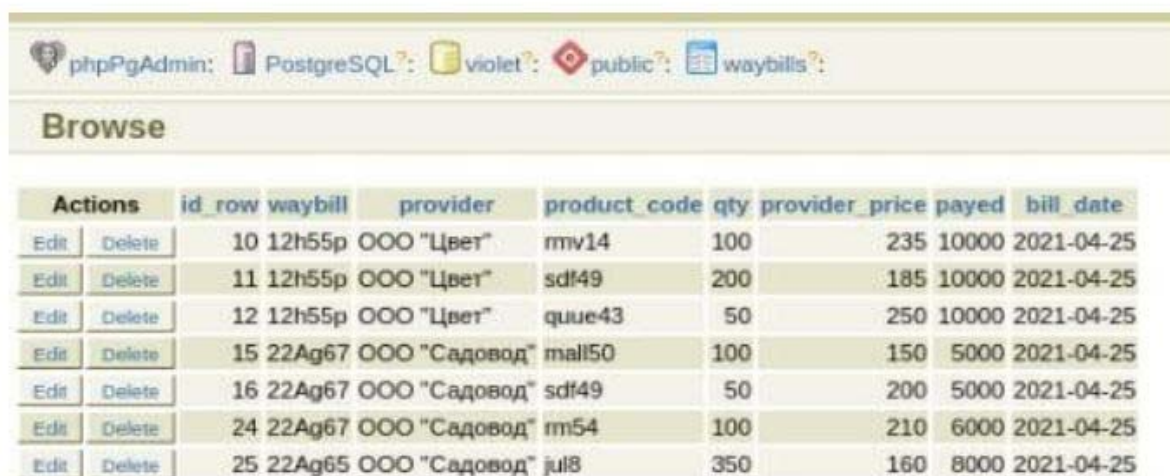


Actions	id	username	pii	pass	adm
Edit Delete	2	admin	Тестовий Адміністратор	pbkdf2_sha256\$12000\$+M5PZ2H0bMzPk/BMfEU1Pg==\$oKGN...	TRUE
Edit Delete	1	worker1	Тестовий Користувач	pbkdf2_sha256\$12000\$+M5PZ2H0bMzPk/BMfEU1Pg==\$oKGN...	FALSE

Рисунок 3.6 – Форма кабінетів для авторизації

Структура, що зберігає накладні, містить ID, шлях до файлу, назву організації, номер продукту, дату завантаження. Перед реєстрацією приходу обладнання або витратних матеріалів необхідно завантажити накладну, по якій прийшли дані облікові. Вона буде зберігатись на сервері.

Форма електронної таблиці накладних зображена на рисунку 3.7.



Actions	id_row	waybill	provider	product_code	qty	provider_price	payed	bill_date
Edit Delete	10	12h55p	ООО "Цвет"	rmv14	100	235	10000	2021-04-25
Edit Delete	11	12h55p	ООО "Цвет"	sdf49	200	185	10000	2021-04-25
Edit Delete	12	12h55p	ООО "Цвет"	queue43	50	250	10000	2021-04-25
Edit Delete	15	22Ag67	ООО "Садовод"	mall50	100	150	5000	2021-04-25
Edit Delete	16	22Ag67	ООО "Садовод"	sdf49	50	200	5000	2021-04-25
Edit Delete	24	22Ag67	ООО "Садовод"	rm54	100	210	6000	2021-04-25
Edit Delete	25	22Ag65	ООО "Садовод"	jul8	350	160	8000	2021-04-25

Рисунок 3.7 – Форма електронної таблиці накладних

3.4 Висновки за розділом 3

У третьому розділі було обґрунтовано вибір середовища розробки, шляхом грамотно побудованої необхідної структури для застосунку, також було обрано середовище розробки веб-застосунків, за допомогою якого вдалося збудувати проект. Була побудована детальна характеристика модулів реалізованої програми, за допомогою ілюстрацій, була покроково розібрана структура застосунку.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Характеристика результативної інформації розробленого застосунку

Вихідними даними є:

- головна сторінка;
- галерея з каталогом товарів;
- спосіб зворотного зв'язку з інтернет-магазином;
- електронний кошик для замовлення товарів;
- електронний кабінет працівника;
- звіт про наявність товарів на складі;
- спосіб поповнити запаси товару;
- спосіб реєстрації накладних;
- звіт про історію накладних;
- способи оплати клієнтом певного товару;
- інформація та дані щодо доставки.

4.2 Виконання розробленої програми

Вигляд головної сторінки зображений на рисунку 4.1.

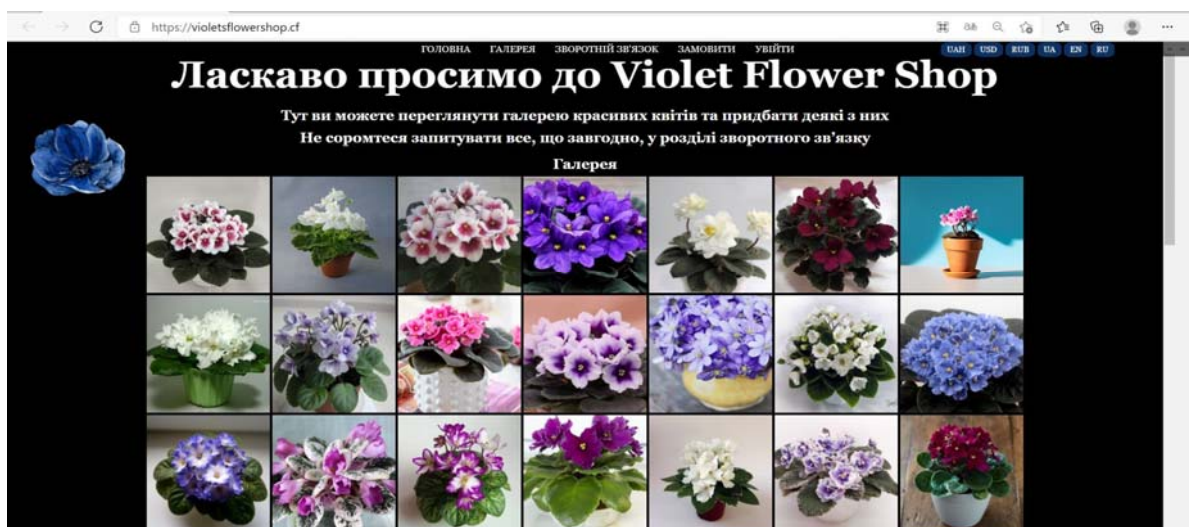


Рисунок 4.1 – Головна сторінка сайту

Для красивої фіксації та наглядного уявлення існуючого товару передбачена галерея (каталог товарів).

Галерея товарів зображена на рисунку 4.2.



Рисунок 4.2 – Галерея товарів

Для більш детального перегляду можна відкрити потрібну картинку рисунок 4.3.

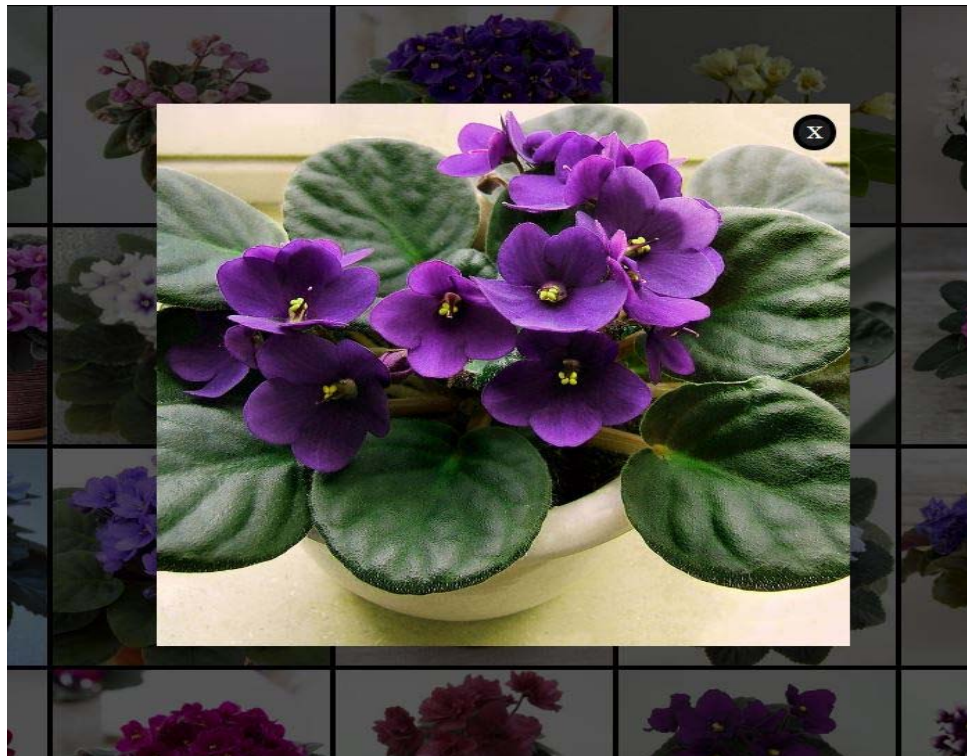


Рисунок 4.3 – Картинка потрібного для клієнта товару

Спосіб зворотного зв'язку для клієнтів зображено на рисунку 4.4.

Зворотній зв'язок

Ім'я:

Прізвище:

Телефон:

E-mail:

Тема:

[Надіслати повідомлення](#)

Рисунок 4.4 – Форма зворотного зв'язку для користувача

Для правильного оформлення замовлень передбачена система для закріплення товарів за допомогою електронної кошика. Для цього необхідно вибрати цікавлять товари в системі, щоб занести їх до списку замовлень.

Форма списку замовлень має такий вигляд, який зображено на рисунку 4.5.

Кошик замовлення

[Очистити кошик](#)

Назва товару	Опис	Кол	Єдиниць	Ціна за шт	Вартість	
Ромовий пунш		rm54	1	250 UAH	250.00 UAH	Вилучити
Гроте		rm12	1	325 UAH	325.00 UAH	Вилучити
Фіалка Оптімара		rmv14	1	315 UAH	315.00 UAH	Вилучити
Фіалка АВ-Танго		jm46	1	285 UAH	285.00 UAH	Вилучити
Сяючий Дзвіночок		sdf49	1	235 UAH	235.00 UAH	Вилучити
Усього:			5		1,410.00 UAH	

[Оформити замовлення](#)

Рисунок 4.5 – Список замовлень користувача

При авторизації в системі стають доступними такі категорії сайту як "Склад" і "Накладні", в яких зберігається інформація щодо обліку товарів та звіти про накладні.

Зовнішній вигляд вкладок, які з'явилися при авторизації, зображені на рисунку 4.6.



Рисунок 4.6 – Відображувані вкладки «Склад» та «Накладні»

Після реєстрації персоналу в доступному форматі відображається список обліку. Далі, при необхідності, можна скористатися двома функціями, які створені спеціально для проведення подібних складських операцій.

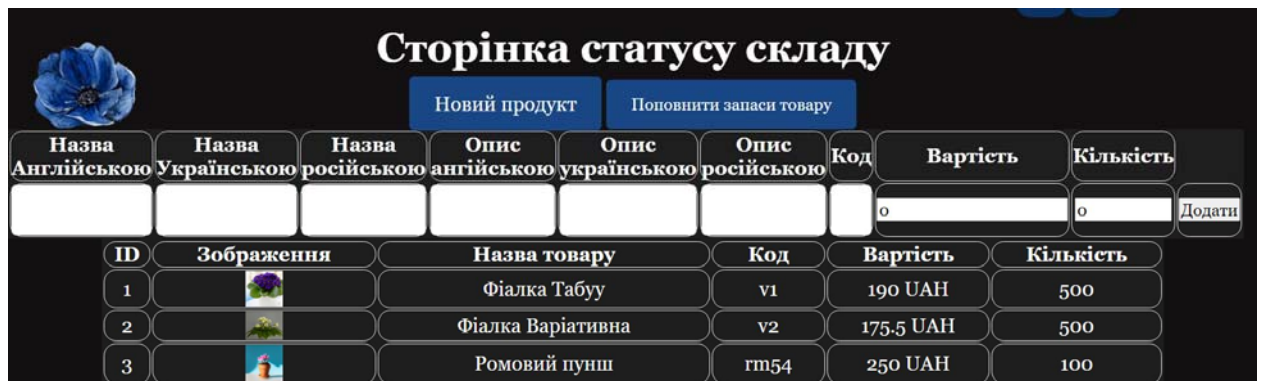
Загальний облік товарів зображено на рисунку 4.7.

	Зображення	Назва товару	Код	Вартість	Кількість
1		Фіалка Табуу	v1	190 UAH	500
2		Фіалка Варіативна	v2	175.5 UAH	500
3		Ромовий пуши	rm54	250 UAH	100
4		Гроте	gm12	325 UAH	100
5		Фіалка Олімпіара	rmv14	315 UAH	100
6		Мелодії Кіми	rmvk61	250 UAH	500
7		Фіалка АВ-Танго	jtm46	285 UAH	500
8		Фіалка Роб Денді	robs3	170 UAH	500
9		Фіалка Джилліан	jul8	210 UAH	500
10		Фіалка Рамблін Дотс	sd12	305 UAH	500
11		Буу Мен	bo47	195 UAH	500
12		Свяочий Дзвіночок	sdf49	235 UAH	500
13		Зефір	mall50	265 UAH	500
14		Синій шифон	gtr44	315.5 UAH	500
15		Біла Королева	quue43	185.5 UAH	100

Рисунок 4.7 – Статус товарів на складі

Для того, щоб додати новий товар до списку товарів, нам в допомогу працює функція «Новий продукт». Для додавання товару вводиться ім'я об'єкта потрібною мовою, код товару, а також його вартість і кількість.

Побачити подробиці використання даної функції можна на рисунку 4.8.



Назва	Назва	Назва	Опис	Опис	Опис	Код	Вартість	Кількість
Англійською	Українською	російською	англійською	українською	російською			
ID	Зображення	Назва товару	Код	Вартість	Кількість			
1		Фіалка Табуу	v1	190 UAH	500			
2		Фіалка Варіативна	v2	175.5 UAH	500			
3		Ромовий пунш	gm54	250 UAH	100			

Рисунок 4.8 – Функції використання застосування «Новий продукт»

Передбачена ще одна важлива функція в складському сховищі. Вона включає в себе поповнення запасів певного продукту. Функція фіксує введену кількість товару для замовлення постачальникам.

Робота функції виглядає таким чином, як зображено на рисунку 4.9.



Зображення	Назва товару	Код	Вартість	Кількість	
	Фіалка Табуу	v1	190 UAH	500	1
	Фіалка Варіативна	v2	175.5 UAH	500	1
	Ромовий пунш	gm54	250 UAH	100	1
	Гроте	gm12	325 UAH	100	1
	Фіалка Оптімара	gm14	315 UAH	100	1
	Мелодії Кіми	gmvk61	250 UAH	500	1
	Фіалка АВ-Танго	jtm46	285 UAH	500	1
	Фіалка Роб Денді	robs3	170 UAH	500	1
	Фіалка Джиллиан	jul8	210 UAH	500	1
	Фіалка Рамблин Дотс	sd12	305 UAH	500	1

Рисунок 4.9 – Функція поповнення запасів товару

Далі розглянемо функцію заповнення накладних і облік попередніх. Застосунок дозволяє враховувати попередні накладні підприємства і створювати нові. Фіксація створення накладних, як і всіх інших процесів в системі, відбувається з шляхом заповнення форми на веб-сторінці.

Робота даної функції для створення накладних зображено на рисунку 4.10.

Рисунок 4.10 – Форма заповнення для створення нової накладної

Робота функції, що зберігає інформацію попередніх накладних зображена на рисунку 4.11.



Попередні накладні			
Накладна №	22Ag67		
Організація-постачальник	ООО "Садовод"	Дата внесення	2021-04-25
Назва товару	Код	Кількість	Ціна постачальника
Зефір	mall50	100	150 UAH
Сяючий Дзвіночок	sdf49	50	200 UAH
Ромовий пунш	rm54	100	210 UAH
Сплачено за накладну		6000 UAH	

Накладна №	12h55p		
Організація-постачальник	ООО "Цвет"	Дата внесення	2021-04-25
Назва товару	Код	Кількість	Ціна постачальника
Фіалка Оптимара	rmv14	100	235 UAH
Сяючий Дзвіночок	sdf49	200	185 UAH
Біла Королева	quue43	50	250 UAH
Сплачено за накладну		10000 UAH	

Рисунок 4.11 – Облік попередніх накладних компанії

Функція «Доставка» призначена для користувача. Для виконання запиту користувача, йому потрібно надати свої дані через веб-форму на сторінці застосунка. Після чого його замовлення почне оброблятися.

Форма заповнення для даних щодо доставки зображено на рисунку 4.12.



The image shows a web form titled "Доставка" (Delivery) with a dark background. It contains seven input fields, each preceded by a label and an icon:

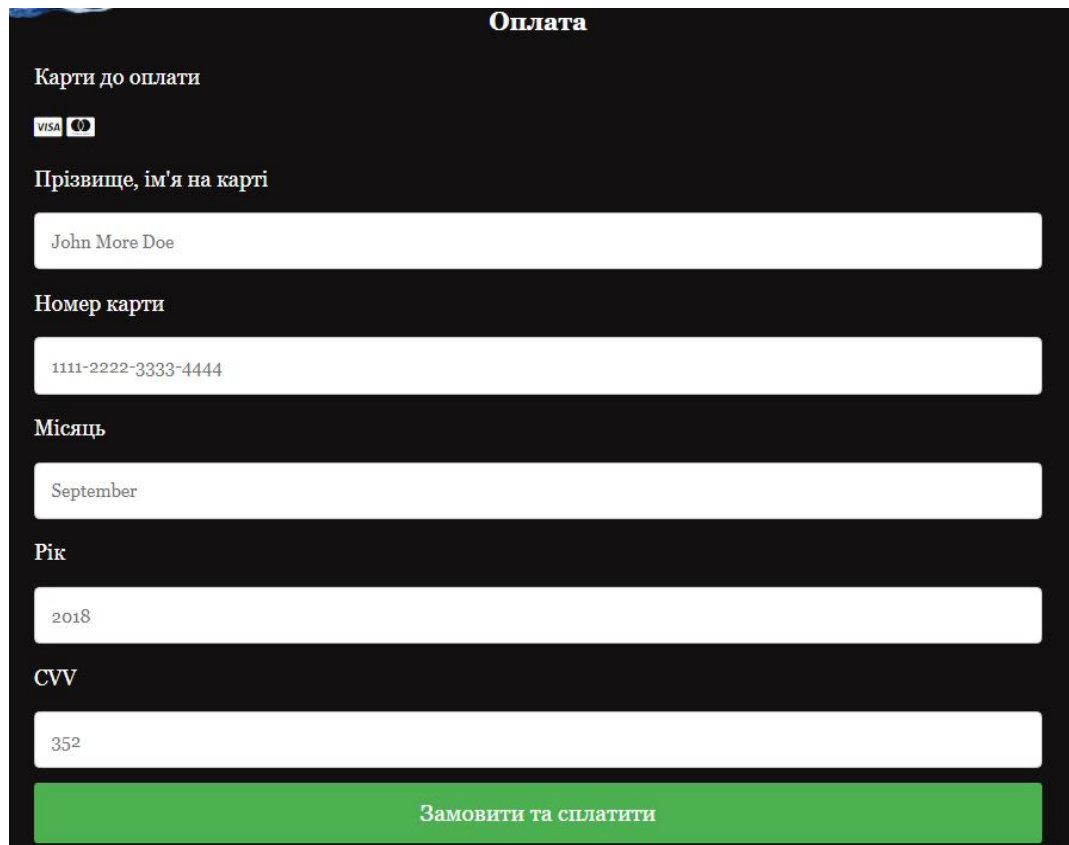
- Прізвище Ім'я** (Surname Name) with a person icon: John M. Doe
- Email** with an envelope icon: john@example.com
- Телефон:** (Phone) with a phone icon: +380123456789
- Поштовий індекс** (Postal Index) with a house icon: 10001
- Місто** (City) with a city icon: New York
- Область** (Region) with a region icon: NY
- Адреса** (Address) with a location pin icon: 542 W. 15th Street

Рисунок 4.12 – Форма заповнення даних для доставки

На тій же сторінці, поруч з функцією "Доставка", є можливість сплатити товар онлайн, вказавши персональну інформацію і дані банківської картки.

Створений веб-застосунок розрахований на багато користувачів і має різні інтерфейси. Користувачами виступають сам адміністратор застосунку, з можливістю запису і редагування інформації та користувач.

Також була створена форма заповнення даних для оплати товару, яка зображена на рисунку 4.13.



The image shows a payment form titled "Оплата" (Payment) in a dark-themed interface. The form contains several input fields for card information:

- Карти до оплати** (Cards to pay): Includes logos for VISA and Mastercard.
- Прізвище, ім'я на карті** (Surname, name on card): A text input field containing "John More Doe".
- Номер карти** (Card number): A text input field containing "1111-2222-3333-4444".
- Місяць** (Month): A text input field containing "September".
- Рік** (Year): A text input field containing "2018".
- CVV**: A text input field containing "352".

At the bottom of the form is a prominent green button with the text "Замовити та сплатити" (Order and pay).

Рисунок 4.13 – Форма для здійснення оплати

4.3 Висновки за розділом 4

В даному розділі було проведено огляд інтерфейсів розробленого застосунку, проведено тестування працездатності та функціональності створеного веб-сайту.

ВИСНОВКИ

Метою роботи була розробка веб-застосунку для обліку складського обладнання та витратних матеріалів, який буде за рахунок наявності системи автоматизації спрощувати роботу персоналу, економити час для роботи, і забезпечувати комфортне ведення обліку. Застосунок дозволяє формувати звіти, пов'язані з певною діяльністю підприємства, не перемикаючись на інші спеціалізовані програми.

В результаті виконання бакалаврської роботи була досліджена предметна область: проаналізовано традиційний ручний процес обліку обладнання на підприємстві та розглянуто альтернативний, більш сучасний варіант. Проведено SWOT-аналіз позитивних та негативних факторів застарілого, ручного методу ведення обліку та обліку за допомогою веб-ресурсів. Було проведено аналіз існуючих аналогічних розробок. Визначено перелік важливих для застосунку функцій. Розроблено структурну та функціональну схеми онлайн застосунку. На основі порівняльного аналізу веб-технологій обрано технології для розробки, а саме мову PHP для розробки серверної частини застосунку, HTML та CSS для оформлення стилів клієнтської частини. Виконано проєктування та розробку бази даних системи. Розроблено веб-застосунок з урахуванням всіх визначених вимог.

Розроблений веб-застосунок дозволить виконувати такі функції:

- поділ авторизованих користувачів на 2 ролі: адміністратор і користувач;
- внесення, редагування і видалення даних в БД через форму на веб-сторінці та можливість відстеження дати зміни в таблицях;
- зберігання деяких архівних даних;
- забезпечення безперервної та стабільної роботи у момент використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Деркачева Т. Складской учет товарно-материальных ценностей учебное пособие / Т. Деркачева – Учебное пособие – Складской учет товарно-материальных ценностей, 2014. – 24 с.
2. История складских хозяйств и логистики [Электрон. ресурс]. – Режим доступа: <https://stolichnyi.kiev.ua/blogs/7-istoriya-skladskikh-khozyajstv-i-logistiki/>.
3. Яблочников Е. И. Реинжиниринг бизнеспроцессов проектирования и производства / Е. И. Яблочников, В. И. Молочник, Ю. Н. Фомина - Учебное пособие – СПб: СПбГУИТМО, 2008. – 152 с.
4. WMS – система [Электрон. ресурс]. – Режим доступа: <https://www.ant-tech.ru/fields/wms/>.
5. Тимашов А. Автоматизация и управление складом торговой компании [Электрон. ресурс]. — Режим доступа: https://www.axelot.ru/knowhow/press/detail_34062/.
6. Иванов Г. Г., Киреева Н. И. / Г. Г. Иванов, Н. И. Киреева – Складская логистика. 2016. – 56 с.
7. Котин М. 1С. Предприятие 8.2. Управление небольшой фирмой / М. Котин. — Санкт-Петербург: Питер, 2016. — 320 с.
8. Солодушкин С. И., Юманова И. Ф. / С. И. Солодушкин, И. Ф. Юманова – Web и DHTML – Учебное пособие/ УрФУ, 2015. — 128 с.
9. Годин, В. В. Информационное обеспечение управленческой деятельности: Учебник для студентов учреждений среднего профессионального образования / В. В. Годин, И. К. Корнеев — 2001. – 240 с.
10. SWOT-Анализ. 5 главных правил, которых стоит придерживаться [Электрон. ресурс]. – Режим доступа: <https://geniusmarketing.me/lab/swot-analiz-5-glavnyx-pravil-kotoryx-stoit-priderzhivatsya/>.
11. Федотова, Е. Л. Информационные технологии в профессиональной деятельности / Е. Л. Федотова – М.: Инфра, 2008. – 368 с.

12. Полат, Е. С. Современные педагогические и информационные технологии в системе образования : учеб. пособие для студ. высш. учеб. заведений / Е. С. Полат, М. Ю. Бухаркина – М. : Академия, 2007. – 368 с.
13. Котеров Д, Симдянов И. – Книга PHP 7 в подлиннике / Д. Котеров, И. Симдянов – БХВ-Петербург, 2016. – 1088 с.
14. Учебное пособие по CGI – программированию. Программирование с использованием CGI [Электрон. ресурс]. – Режим доступа: <https://lab-music.ru/uchebное-posobie-po-cgi-programmirovaniyu-programmirovaniye-s-ispolzovaniem/>.
15. Технология активных серверных страниц ASP [Электрон. ресурс]. – Режим доступа: <http://bourabai.kz/php/asp.htm>.
16. CSS – Преимущества стилей [Электрон. ресурс]. – Режим доступа: <http://htmlbook.ru/samcss/preimushchestva-stiley>.
17. Printstore — Учет техники и расходных материалов [Электрон. ресурс]. – Режим доступа: <http://printstore.ru/printstore/>.
18. Аникин Б. А. Коммерческая логистика / Б. А. Аникин, А. П. Тяпухин – М. : Проспект, 2012. – 432 с.

ДОДАТОК А
Технічне завдання

Вступ

Метою бакалаврської роботи є розробка онлайн забезпечення, яке дозволяє вести облік складського обладнання, який за рахунок наявності системи автоматизації спрощує роботу персоналу, економить час для роботи, і забезпечує комфортне ведення обліку. Застосунок дозволяє формувати звіти, пов'язані з певною діяльністю підприємства, не перемикаючись на інші спеціалізовані програми.

Розроблений веб-застосунок дозволить виконувати такі основні функції:

- внесення, редагування і видалення даних в БД через форму на веб-сторінці та можливість відстеження дати зміни в таблицях;
- зберігання деяких архівних даних;
- забезпечення безперервної та стабільної роботи у момент використання;
- можливість оформлювати замовлення користувачам;
- можливість оплачувати товари дистанційно.

А.1 Підстава для розробки

Підставою для розробки послужило завдання на диплому роботу бакалавра за темою: «Програмна реалізація онлайн системи складського обліку», затверджену наказом по Національному університету «Запорізька Політехніка» №103 від 30 березня 2021 року.

А.2 Призначення розробки

Веб-застосунок призначений для зручного способу придбання товарів користувачами, для автоматизації та полегшення ведення складського обліку, а також складання певних видів звітів. При розробці програми акцент робиться на простоту інтерфейсу для швидкого освоєння людиною.

A.3 Вимоги до програми

A.3.1 Вимоги до структури програми

Веб-застосунок повинен мати такі основні структурні елементи – клієнтська платформа, адмінська платформа та серверна частина. Серверна частина в свою чергу містить три головних модуля:

- модуль збору клієнтських заказів, який виконує обов’язок передавати дані про закази замовника з сайту;
- база даних, що зберігає як дані замовлень покупців, так і дані складу (список товарів);
- модуль сортування та обробки заказів, що доступний тільки для адміна або керівника сайтом, який сортирує та відправляє закази клієнтам, займається онлайн складським обліком.

A.3.2 Вимоги до функціональних характеристик

Відповідно до постановки задачі, були сформовані основні функціональні вимоги до програми.

Було визначено 8 кращих показників якості онлайн застосунку:

- простота освоєння;
- оперативність техпідтримки;
- стабільність роботи;
- функціонал – чим він більший, тим краще;
- можливість роботи в програмі без установки;
- можливість працювати в сервісі онлайн;
- підтримка баз даних (БД) SQL;
- пошук по базі даних.

A.3.3 Вимоги до інтерфейсу програми

Веб-забезпечення повинно мати простий та зручний для використання інтерфейс. Графічний інтерфейс має бути приємним та ергономічним для користувача та зі зручною стартовою сторінкою. Взаємодія застосунку із користувачем відбувається на постійній основі.

A.3.4 Умови експлуатації

Для роботи з програмою, користувач має володіти достатніми знаннями для коректного сприйняття інформації.

A.4 Обґрунтування вибору технічних засобів, операційної системи і мови програмування

Для забезпечення роботи веб-застосунку необхідно мати будь-який ПК, який під'єднано до всесвітньої мережі Інтернет, браузер з мінімальними вимогами для роботи під операційною системою Microsoft Windows.

Особливих вимог до експлуатації системи не пред'являється.

A.5 Порядок контролю й приймання

Дипломна кваліфікаційна робота допускається до захисту керівником, нормоконтролером, затверджується завідувачем кафедри і захищається в присутності членів екзаменаційної комісії.

ДОДАТОК Б
Текст програми

Б.1 Створення основи сайту

За допомогою цього коду я побудувала основу-скелет сайту, в якому знаходиться все, що показується вгорі сайту (логотип, посилання, перемикання мови, валюта).

```
<?php
ob_start();
session_start(); ///Початок сесії
error_reporting(0); // Рівень виводу помилок
ini_set('display_errors', 1); //Вивід помилок на сторінку сайту
require_once("connect.php"); //Підключення до бази даних
$page = (isset($_GET['page']) ? $_GET['page'] : 'main.php');
if (!isset($_SESSION['lang'])) {
    $_SESSION['lang'] = 'ua';
};
$lang = $_SESSION['lang'];
if (isset($_POST['lang'])) {
    $_SESSION['lang'] = $_POST['lang'];
    $lang = $_SESSION['lang'];
}
if (!isset($_SESSION['currency'])) {
    $_SESSION['currency'] = 'UAH';
};
$currency = $_SESSION['currency'];
if (isset($_POST['currency'])) {
    $_SESSION['currency'] = $_POST['currency'];
    $currency = $_SESSION['currency'];
};
switch ($currency) {
```

```

case "UAH":
$curr_kof = 1;
break;
case "USD":
$curr_kof = 0.036;
break;
case "RUB":
$curr_kof = 2.696;
break;
};

```

```
$translate = runQuery($con, "select * from ui_web order by id");
```

```
?>
```

```
<html>
```

```
<meta charset="utf-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link href="css/main.css" rel="stylesheet" />
```

```
<script src="js/jquery.min.js"></script>
```

```
<!--Таблиця стилів-->
```

```
<head>
```

```
<title>Онлайн магазин "Violet" </title>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<nav class="navbar">
```

```

<ul>
<li><a href="/"></a></li>
<li><a class="scroll" href="/#home"><?php echo $translate[0][$lang];
?></a></li>
<li><a class="scroll" href="/#gallery"><?php echo $translate[1][$lang];
?></a></li>
<li><a class="scroll" href="/#contacts"><?php echo $translate[2][$lang];
?></a></li>
<li><a class="scroll" href="cart.php"><?php echo $translate[5][$lang];
?></a></li>
<?php
if (!isset($_SESSION['use'])) {
echo '<li><a class="scroll" href="login.php">' . $translate[4][$lang] .
'</a></li>';
} else {
echo '<li><a class="scroll" href="storage.php">' . $translate[6][$lang] .
'</a></li>';
echo '<li><a class="scroll" href="waybill.php">' . $translate[19][$lang] .
'</a></li>';
// echo '<li><a class="scroll" href="orders.php">' . $translate[7][$lang] .
'</a></li>';
//echo '<li><a class="scroll" href="admin.php">' . $translate[8][$lang] .
'</a></li>';
echo '<li><a class="scroll" href="logout.php">' . $translate[3][$lang] .
'</a></li>';
}; ?>
</ul>
<div class="lang-form">
<form action="" method="POST">

```

```

<input name="currency" type="submit" value="UAH">
<input name="currency" type="submit" value="USD">
<input name="currency" type="submit" value="RUB">
<input name="lang" type="submit" value="ua">
<input name="lang" type="submit" value="en">
<input name="lang" type="submit" value="ru">
</form>

```

```

</div>

```

```

</nav>

```

```

<html>

```

```

<meta charset="utf-8" />

```

```

<meta name="viewport" content="width=device-width, initial-scale=1">

```

```

<link href="css/main.css" rel="stylesheet" />

```

```

<!--Таблиця стилів-->

```

```

<div class="container">

```

Б.2 Створення привітання сторінки сайту

```

<?php
switch ($lang) {
case "en":
echo '<div>
<h1>Welcome to Violet Flower Shop</h1>
<h2>Here you can look gallery of beautiful flowers and to purchase some
of them</h2>
<h2>Feel free to ask anything you want in feedback section</h2>
</div>';
break;

```

```

case "ua":
echo '<div>
<h1> Ласкаво просимо до Violet Flower Shop </h1>
<h2> Тут ви можете переглянути галерею красивих квітів та придбати
деякі з них </h2>
<h2> Не соромтеся запитувати все, що завгодно, у розділі зворотного
зв'язку </h2>
</div>';
break;
case "ru":
echo '<div>
<h1> Добро пожаловать в Violet Flower Shop </h1>
<h2> Здесь вы можете посмотреть галерею красивых цветов и купить
некоторые из них </h2>
<h2> Не стесняйтесь спрашивать все, что хотите, в разделе обратной
связи </h2>
</div>';
break;
default:
echo '<div>
<h1> Ласкаво просимо до Violet Flower Shop </h1>
<h2> Тут ви можете переглянути галерею красивих квітів та придбати
деякі з них </h2>
<h2> Не соромтеся запитувати все, що завгодно, у розділі зворотного
зв'язку </h2>
</div>';
break;
}
?>

```

Б.3 Відображення головної сторінки посекційно

```

<section id="home">
<?php include('home.php'); ?>

</section>

<section id="gallery">
<h2><?php echo $translate[1][$lang]; ?></h2>
<?php include("gallery.php"); ?>

</section>

<section id="contacts">
<h2><?php echo $translate[2][$lang]; ?></h2>
<div class="container-contact">
<form action="sendmsg.php">
<div class="row">
<div class="col-25">
<label for="fname"><?php echo $translate[9][$lang]; ?></label>
</div>
<div class="col-75">
<input      required      type="text"      id="fname"      name="firstname"
placeholder="<?php echo $translate[9][$lang]; ?>">
</div>
</div>
<div class="row">
<div class="col-25">
<label for="lname"><?php echo $translate[10][$lang]; ?></label>
</div>
<div class="col-75">

```

```

        <input type="text" id="lname" name="lastname" placeholder="<?php echo
$translate[10][$lang]; ?>..">
    </div>
</div>
<div class="row">
    <div class="col-25 ">
        <label for="phone"><?php echo $translate[12][$lang]; ?></label>
    </div>
    <div class="col-75 ">
        <input required type="text" id="phone" name="phone" pattern="\+380([0-
9]){9}" placeholder="+380123456789">
    </div>

</div>
<div class="row">
    <div class="col-25">
        <label for="email">E-mail: </label>
    </div>
    <div class="col-75">
        <input type="text" name="email" size="30">
    </div>
</div>
<div class="row">
    <div class="col-25">
        <label for="subject"><?php echo $translate[11][$lang]; ?></label>
    </div>
    <div class="col-75">
        <textarea id="subject" name="subject" placeholder="<?php echo
$translate[13][$lang]; ?>" style="height:200px"></textarea>
    </div>

```

```

</div>
<div class="row">
<input type="submit" value="<?php echo $translate[14][$lang]; ?>">
</div>
</form>
</div>
</section>

```

Б.4 Створення сторінки для переходу на головну

```

<?php include('head.php');?>

<body>
<header>
</header>
<!--<div class="front">

</div>-->
<?php include $page;
//print_r($_REQUEST['action']);?>
</body>

</html>

```

Б.5 Створення сторінки для з'єднання з базою даних

```

<?php
$DB_HOST="127.0.0.1";
$DB_USER="www";

```

```

$DB_PASS="post171";
$DB_DBNAME="violet";
$con = pg_connect("host=$DB_HOST port=5432
dbname=$DB_DBNAME user=$DB_USER password=$DB_PASS")
or die (" Не можу ініціалізувати з'єднання з базою\n");

function runQuery($a,$query) {

    $resultset=[];
    $result = pg_query($a,$query);
    while($row=pg_fetch_assoc($result,null)) {
        $resultset[] = $row;
    }
    if(!empty($resultset))
        return $resultset;
}

function numRows($a,$query) {
    $result = pg_query($a,$query);
    $rowcount = pg_num_rows($result);
    return $rowcount;
}

?>

```

Б.6 Сторінка для зберігання функцій логіна

```

<?php
//ini_set('display_errors', 'On');
//error_reporting(E_ALL);

```

```

function make_password($password)
{
    $algorithm = "pbkdf2_sha256";
    $iterations = 120000;

    $newSalt = random_bytes(16);
    $newSalt = base64_encode($newSalt);

    $hash = hash_pbkdf2("SHA256", $password, $newSalt, $iterations, 0,
true);
    $toDBStr = $algorithm . "$" . $iterations . "$" . $newSalt . "$" .
base64_encode($hash);

    // This string is to be saved into DB, just like what Django generate.
    // echo $toDBStr;
    return $toDBStr;
}

function verify_Password($dbString, $password)
{
    $pieces = explode('$', "$dbString");

    $iterations = $pieces[1];
    $salt = $pieces[2];
    $old_hash = $pieces[3];

    $hash = hash_pbkdf2("SHA256", $password, $salt, $iterations, 0, true);
    $hash = base64_encode($hash);
    if ($hash == $old_hash) {
        // login ok.
    }
}

```

```

return true;
} else {
//login fail
return false;
}
}

?>

```

Б.7 Сторінка для форми зворотного зв'язку

```

<?php
include("head.php");

switch ($lang) {
case "en":
echo '<h1 class="send">Message sent to shop mail, wait for
responce</h1>';
break;
case "ua":
echo '<h1 class="send">Лист надіслано до поштової скриньки
магазину, очікуйте відповіді </h1>';
break;
case "ru":
echo '<h1 class="send">Письмо отправлено на почту магазина,
ожидайте ответа </h1>';
break;
}

```

```
//header('Location: ' . $_SERVER['HTTP_REFERER']);
header('Refresh:5; URL=/');
?>
```

Б.8 Сторінка, яка відкривається при натисканні «Замовити»

```
<?php
session_start();
require_once("connect.php");
//print_r($_SESSION);
$name=$_SESSION["lang"]."_name";
$desc=$_SESSION["lang"]."_desc";
require('head.php');
if(!empty($_GET["action"])) {
switch($_GET["action"]) {
case "add":
if(!empty($_POST["quantity"])) {
$productByCode = runQuery($con,"SELECT * FROM (select * from
product_store inner join product_desc using (id)) a WHERE code='" .
$_GET["code"] . "'");
$itemArray =
array($productByCode[0]["code"]=>array('name'=>$productByCode[0][$name],
'code'=>$productByCode[0]["code"], 'quantity'=>$_POST["quantity"],
'price'=>$productByCode[0]["price"], 'image'=>$productByCode[0]["picture"]));

if(!empty($_SESSION["cart_item"])) {
if(in_array($productByCode[0]["code"],array_keys($_SESSION["cart_ite
m"])))) {
foreach($_SESSION["cart_item"] as $k => $v) {
if($productByCode[0]["code"] == $k) {
```

```

if(empty($_SESSION["cart_item"][$k]["quantity"])) {
    $_SESSION["cart_item"][$k]["quantity"] = 0;
}
$_SESSION["cart_item"][$k]["quantity"] += $_POST["quantity"];
}
}
} else {
    $_SESSION["cart_item"]
array_merge($_SESSION["cart_item"],$itemArray);
}
} else {
    $_SESSION["cart_item"] = $itemArray;
}
}
break;
case "remove":
if(!empty($_SESSION["cart_item"])) {
foreach($_SESSION["cart_item"] as $k => $v) {
if($_GET["code"] == $k)
unset($_SESSION["cart_item"][$k]);
if(empty($_SESSION["cart_item"]))
unset($_SESSION["cart_item"]);
}
}
break;
case "empty":
unset($_SESSION["cart_item"]);
break;
}
}

```

?>

<div id="shopping-cart">

<HTML>

<HEAD>

<TITLE><?php echo \$translate[7][\$lang];?> </TITLE>

<link href="css/style.css" type="text/css" rel="stylesheet" />

</HEAD>

<BODY>

<div id="shopping-cart">

<div class="txt-heading"><?php echo \$translate[22][\$lang]; ?></div>

<?php echo \$translate[21][\$lang]; ?>

<?php

if(isset(\$_SESSION["cart_item"])){

 \$total_quantity = 0;

 \$total_price = 0;

?>

<table class="tbl-cart" cellpadding="10" cellspacing="1">

<tbody>

<tr>

<th width="30%"><?php echo \$translate[23][\$lang]; ?></th>

<th ><?php echo \$translate[24][\$lang]; ?></th>

<th width="5%"><?php echo \$translate[25][\$lang]; ?></th>

<th width="6%"><?php echo \$translate[26][\$lang]; ?></th>

<th width="10%"><?php echo \$translate[27][\$lang]; ?></th>

<th width="10%"><?php echo \$translate[28][\$lang]; ?></th>

```

<th width="5%"></th>

</tr>

<?php
    foreach ($_SESSION["cart_item"] as $item){
        $item_price = $item["quantity"]*$item["price"]*$curr_kof;
    ?>

<tr>

<td>" class="cart-item-image"
/><?php echo $item["name"]; ?></td>

<td><?php echo $item[$desc];?></td>

<td><?php echo $item["code"]; ?></td>

<td style="text-align:right;"><?php echo $item["quantity"]; ?></td>

<td style="text-align:right;"><?php echo $item["price"]*$curr_kof."
".$currency; ?></td>

<td style="text-align:right;"><?php echo number_format($item_price,2)."
".$currency; ?></td>

<td style="text-align:center;"><a
href="cart.php?action=remove&code=<?php echo $item["code"];
?>" class="btnRemoveAction">" /></a></td>

</tr>

<?php
$total_quantity += $item["quantity"];
$total_price += ($item["price"]*$item["quantity"]*$curr_kof);
}
?>

<tr>

<td colspan="3" style="font-size:larger;text:bold;" align="center"><?php
echo $translate[15][$lang]; ?></td>

```

```

<td align="right"><?php echo $total_quantity; ?></td>

<td align="right" colspan="2"><strong><?php echo
number_format($total_price, 2)." ".$currency; ?></strong></td>

</tr>
</tbody>
</table>
<a href="checkout.php" id="btnEmpty"><?php echo $translate[20][$lang];
?></a>

<?php
} else {
?>

<div class="no-records"><?php echo $translate[18][$lang]; ?></div>

<?php
};
?>

</div>

<div class="txt-heading"><br></div>

<div id="product-grid">

    <?php

$product_array = runQuery($con,"select * from product_store inner join
product_desc using (id) ORDER BY id ASC;");

    if (!empty($product_array)) {
        foreach($product_array as $key=>$value){
            ?>

            <div class="product-item">

```

```

        <form method="post" action="cart.php?action=add&code=<?php echo
$product_array[$key]["code"]; ?>">
            <div class="product-image">"></div>
            <div class="product-tile-footer">
                <div class="product-title"><?php echo $product_array[$key][$name];
?></div>
                <div class="product-price"><?php echo
$product_array[$key]["price"]*$curr_kof." ".$currency." "; ?></div>
                <div class="cart-action"><input type="text" class="product-quantity"
name="quantity" value="1" size="2" /><input type="submit" value="<?php echo
$translate[17][$lang]; ?>" class="btnAddAction" /></div>
            </div>
        </form>
    </div>
    <?php
    }
    }
    ?>
</div>
</body>
</html>

```

Б.9 Функція, яка з'являється після оформлення замовлення,

```

<?php
ob_start();
session_start();
include("head.php");

```

```
//$warehouse=runQuery($con,"SELECT * FROM (select * from
product_store inner join product_desc using (id));");
```

```
$cart = $_SESSION['cart_item'];
//unset($_SESSION['cart_item']);
//print_r($cart);
?>
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
```

```
<style>
```

```
.preview {
width: 50px;
object-fit: cover;
}
```

```
.btn {
background-color: #4CAF50;
color: white;
padding: 12px;
margin: 10px 0;
border: none;
width: 100%;
border-radius: 3px;
cursor: pointer;
font-size: 17px;
}
```

/* Responsive layout - when the screen is less than 800px wide, make the two columns stack on top of each other instead of next to each other (also change the direction - make the "cart" column go on top) */

```
@media (max-width: 800px) {
  .row {
    flex-direction: column-reverse;
  }
```

```
.col-25 {
  margin-bottom: 20px;
}
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="check-container">
```

```
<div class="row">
```

```
<div class="col-45">
```

```
<form action="pay.php">
```

```
<div class="col-50">
```

```
<h3 style="text-align:center"><?php echo $translate[58][$lang]; ?></h3>
```

```
<div class="row">
```

```
<div class="col-25">
```

```
<label for="fname"><i class="fa fa-user"></i> <?php echo
$translate[47][$lang]; ?></label>
```

```
</div>
```

```
<div class="col-75">
```

```
<input type="text" id="fname" name="firstname" placeholder="John M.
Doe">
```

```

</div>
</div>
<div class="row">
<div class="col-25">
<label for="email"><i class="fa fa-envelope"></i> Email</label>
</div>
<div class="col-75">
<input          type="text"          id="email"          name="email"
placeholder="john@example.com">
</div>
</div>
<div class="row">
<div class="col-25">
<label    for="tel"><i    class="fa    fa-user"></i>    <?php    echo
$translate[12][$lang]; ?></label>
</div>
<div class="col-75">
<input          type="text"          id="tel"          name="phone"
placeholder="+380123456789">
</div>
</div>
<div class="row">
<div class="col-25">

<label for="zip"><?php echo $translate[50][$lang]; ?></label>
</div>
<div class="col-75">
<input type="text" id="zip" name="zip" placeholder="10001">
</div>
</div>

```

```

<div class="row">
  <div class="col-25">
    <label for="city"><i class="fa fa-institution"></i> <?php echo
$translate[48][$lang]; ?></label>
  </div>
  <div class="col-75">
    <input type="text" id="city" name="city" placeholder="New York">
  </div>
</div>
<div class="row">
  <div class="col-25">

    <label for="state"><?php echo $translate[49][$lang]; ?></label>
  </div>
  <div class="col-75">
    <input type="text" id="state" name="state" placeholder="NY">
  </div>
</div>
<div class="row">
  <div class="col-25">
    <label for="adr"><i class="fa fa-address-card-o"></i> <?php echo
$translate[46][$lang]; ?></label>
  </div>
  <div class="col-75">
    <input type="text" id="adr" name="address" placeholder="542 W. 15th
Street">
  </div>
</div>

```

```

</div>

<br>

<br>

<div class="col-50">
<h3 style="text-align:center"><?php echo $translate[51][$lang]; ?></h3>
<label for="fname"><?php echo $translate[52][$lang]; ?></label>
<div class="icon-container">
<i class="fa fa-cc-visa"></i>
<i class="fa fa-cc-mastercard"></i>
</div>
<label for="cname"><?php echo $translate[53][$lang]; ?></label>
<input type="text" id="cname" name="cardname" placeholder="John More
Doe">
<label for="ccnum"><?php echo $translate[54][$lang]; ?></label>
<input type="text" id="ccnum" name="cardnumber" placeholder="1111-
2222-3333-4444">
<label for="expmonth"><?php echo $translate[55][$lang]; ?></label>
<input type="text" id="expmonth" name="expmonth"
placeholder="September">
<div class="row">
<div class="col-50">
<label for="expyear"><?php echo $translate[56][$lang]; ?></label>
<input type="text" id="expyear" name="expyear" placeholder="2018">
</div>
<div class="col-50">
<label for="cvv">CVV</label>
<input type="text" id="cvv" name="cvv" placeholder="352">
</div>
</div>
</div>

```

```

        <input type="submit" value="<?php echo $translate[57][$lang]; ?>"
class="btn">

    </form>

</div>

<div class="col-452">

<table cellpadding="10" cellspacing="1">

<tbody>

<tr>

<th width="30%"><?php echo $translate[23][$lang]; ?></th>

<th><?php echo $translate[24][$lang]; ?></th>

<th width="5%"><?php echo $translate[25][$lang]; ?></th>

<th width="6%"><?php echo $translate[26][$lang]; ?></th>

<th width="10%"><?php echo $translate[27][$lang]; ?></th>

<th width="10%"><?php echo $translate[28][$lang]; ?></th>

<th width="5%"></th>

</tr>

<?php
foreach ($cart as $item) {
    $item_price = $item["quantity"] * $item["price"] * $curr_kof;
    ?>

<tr>

<td>" </td>

<td><?php echo $item["name"]; ?></td>

<td><?php echo $item["code"]; ?></td>

<td style="text-align:right;"><?php echo $item["quantity"]; ?></td>

<td style="text-align:right;"><?php echo $item["price"] * $curr_kof . " " .
$currency; ?></td>

    <td style="text-align:right;"><?php echo number_format($item_price, 2) .
" " . $currency; ?></td>

</tr>

```

```

<?php
$total_quantity += $item["quantity"];
$total_price += ($item["price"] * $item["quantity"] * $curr_kof);
}
?>

<tr>

<td colspan="3" align="right"><?php echo $translate[15][$lang]; ?></td>
<td align="right"><?php echo $total_quantity; ?></td>
<td align="right" colspan="2"><strong><?php echo
number_format($total_price, 2) . " " . $currency; ?></strong></td>
<td></td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
</body>

</html>

```

Б.10 Функція, яка показує номер замовлення після його оформлення

```

<?php
ob_start();
session_start();
include("head.php");

unset($_SESSION['cart_item']);

```

```

$order_id = date('YmdHms').'00';
//echo $order_id;
//$order_id+=1;
//echo $order_id;
//print_r($_SESSION['cart_item']);

?>
<div style="margin-top:15%; margin-left:5%">
<?php switch($lang) {
case"en":
echo "Your order is in progress, order number is ".$order_id;
break;
case"ua":
echo "Ваше замовлення надано у обробку, ваш номер:".$order_id;
break;
case"ru":
echo "Ваш заказ обрабатывается, ваш номер заказа:".$order_id;
break;
}?>
</div>

```

Б.11 Сторінка входу для працівника

```

<?php
session_start();
require_once "connect.php";
require_once "process.php";
?>
<html>

```

```

<meta charset="utf-8" />

<!--<link href="css/main.css" rel="stylesheet" />-->

<head>
<title>Вхід працівника</title>
</head>
<body>
<div class="login-page">
<div class="form">
<form action="" method="post" class="login-form">
<input type="text" name="user" placeholder="Логін" />
<input type="password" name="pass" placeholder="Пароль" />
<input type="submit" name="login" value="Увійти" />
</form>
<?php
$user = $pass = $result = $row = "";
if (isset($_POST['login'])) {
    $user = $_POST['user'];
    $pass = $_POST['pass'];
    $query_user = "select id,pass from users where username='$user'";
    $result = pg_query($con, $query_user);
    $count = pg_num_rows($result);
    $row = pg_fetch_row($result);
    $db_user_id = $row[0];
    $db_user_pass = $row[1];
    if ($user != NULL && $count == '1' && $pass != "" && $pass != NULL
&& verify_Password($db_user_pass, $pass)) {
        $_SESSION['use'] = $user;
        $_SESSION['id'] = $db_user_id;
        echo 'Доброго дня, вас переадресує на головну через 2 секунди';
    }
}

```

```

header('Location: /');
} else {
echo '<p style="color:red; font-size:larger">Не вірний пароль чи
логін</p>';
}
}
?>
</div>

</div>

<style>
.login-page {
width: 360px;
padding: 8% 0 0;
margin: auto;
}

.form {
position: relative;
z-index: 1;
background: #FFFFFF;
max-width: 360px;
margin: 0 auto 100px;
padding: 45px;
text-align: center;
box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}

.form input {
font-family: sans-serif;

```

```
outline: 0;
background: #f2f2f2;
width: 100%;
border: 0;
margin: 0 0 15px;
padding: 15px;
box-sizing: border-box;
font-size: 14px;
}
```

```
.form input[type=submit] {
font-family: sans-serif;
text-transform: uppercase;
outline: 0;
background: #194785;;
width: 100%;
border: 0;
padding: 15px;
color: #FFffFF;
font-size: 14px;
-webkit-transition: all 0.3 ease;
transition: all 0.3 ease;
cursor: pointer;
}
```

```
.form input[type=submit]:hover,
.form input[type=submit]:active,
.form input[type=submit]:focus {
background: #194785;
}
```

```
.form .message {
margin: 15px 0 0;
color: #b3b3b3;
font-size: 12px;
}
```

```
.form .message a {
color: #4CAF50;
text-decoration: none;
}
```

```
body {
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
```

```
</style>
```

```
<script>
```

```
$('.message a').click(function() {
$('.form').animate({
height: "toggle",
opacity: "toggle"
}, "slow");
});
```

```
</script>
```

Б.12 Функція закінчення сесії працівника

```
<?php
```

```
ob_start();
```

```

session_start();
session_destroy();
$_SESSION['use'] = "";
echo "<h1 style=\"color:light-blue\">Ваш сеанс завершено, переадресую
на початкову сторінку </h1>";
//header('Location: ' . $_SERVER['HTTP_REFERER']);
header('Refresh: 1.5; URL=/');
?>

```

Б.13 Сторінка статусу складу

```

<?php
require("head.php");
require_once("connect.php");
$name = $_SESSION["lang"] . "_name";
$desc = $_SESSION["lang"] . "_desc";

?>

```

```

<div style="margin-top:5%;text-align:center;">
<?php
switch ($lang) {
case "en":
echo '<h1> Warehouse status page</h1>';
break;
case "ua":
echo '<h1> Сторінка статусу складу</h1>';
break;
case "ru":
echo '<h1> Страница состояния склада</h1>';

```

```

break;
default:
echo '<h1> Сторінка статусу складу</h1>';
break;
}
?>

```

```

<div class="form-storage">
    <form action="" method="POST">
        <a href="#" class="mybtn" onclick="show_form()"><?php echo
$translate[30][$lang]; ?></a>
        <input name="fillProd" type="submit" value="<?php echo
$translate[37][$lang]; ?>">
    </form>
</div>

<div id="div_form" style="display:none">
<div class="storage-add">
<form action="" method="post">
<table>
<thead>
<tr>
<th><?php echo $translate[31][$lang]; ?></th>
<th><?php echo $translate[32][$lang]; ?></th>
<th><?php echo $translate[33][$lang]; ?></th>
<th><?php echo $translate[34][$lang]; ?></th>
<th><?php echo $translate[35][$lang]; ?></th>
<th><?php echo $translate[36][$lang]; ?></th>
<th><?php echo $translate[25][$lang]; ?></th>
<th><?php echo $translate[28][$lang]; ?></th>
<th><?php echo $translate[26][$lang]; ?></th>

```

```

</tr>
</thead>
<tbody>
<tr>
<td><input required type="text" name="en_name"></td>
<td><input required type="text" name="ua_name"></td>
<td><input required type="text" name="ru_name"></td>
<td><input required type="text" name="en_desc"></td>
<td><input required type="text" name="ua_desc"></td>
<td><input required type="text" name="ru_desc"></td>
<td><input required type="text" name="fcode"></td>
<td><input required type="number" min="0" step="0.1" value="0"
name="price"></td>
<td><input required type="number" min="0" max="1000000" step="1"
value="0" name="quantity"></td>
<td><input type="submit" name="addProd" value="<?php echo
$translate[38][$lang]; ?>"></td>
</tr>
</tbody>
</table>
</form>
</div>
</div>
<?php
if (isset($_POST['addProd'])) {
$en_name = $_POST['en_name'];
$ua_name = $_POST['ua_name'];
$ru_name = $_POST['ru_name'];
$en_desc = $_POST['en_desc'];
$ua_desc = $_POST['ua_desc'];

```

```

$ru_desc = $_POST['ru_desc'];
$code=$_POST['fcode'];
$price=$_POST['price'];
$qty=$_POST['quantity'];

$sql      =      "INSERT      INTO      product_desc      VALUES
(nextval('product_desc_id_seq'::regclass),'$en_name','$en_desc','$ua_name','$ua_desc','$ru_name','$ru_desc',NULL,'$code');

insert into product_store values (nextval('product_desc_id_seq'::regclass)-
1,'$qty','$price');"

$sql_query_add = pg_query($con, $sql);
if ($sql_query_add) {
echo "Новий запис додано!";
} else {
echo "Помилка у запиті: " . $sql . " :-\n";
pg_last_error();

};
}

/*print_r($_POST);
print_r($_REQUEST);
print_r($_SESSION);*/

$product_array = runQuery($con, "select * from product_store inner join
product_desc using (id) ORDER BY id ASC;");
if (!empty($product_array)) { ?>
<div class="storage">
<table>
<thead>
<tr>

```

```

<th>ID</th>
<th><?php echo $translate[29][$lang]; ?></th>
<th><?php echo $translate[23][$lang]; ?></th>

<th><?php echo $translate[25][$lang]; ?></th>
<th><?php echo $translate[28][$lang]; ?></th>
<th><?php echo $translate[26][$lang]; ?></th>
</tr>
</thead>
<tbody>
<?php
foreach ($product_array as $key => $value) { ?>
<tr id="<?php echo $product_array[$key]["code"]; ?>">
<td><?php echo $product_array[$key]["id"]; ?></td>
<td>"></td>
<td><?php echo $product_array[$key][$name]; ?></td>

<td><?php echo $product_array[$key]["code"]; ?></td>
<td><?php echo $product_array[$key]["price"]*$curr_kof." ".$currency;
?></td>
<td><?php echo $product_array[$key]["qty"]; ?></td>
<?php if (isset($_POST['fillProd'])) { ?>
<td>
<form          action="storage.php?action=fill&code=<?php          echo
$product_array[$key]["code"]; ?>">
<input type="text" class="product-quantity" name="quantity" value="1"
size="2" />
</td>
<td>

```

```

        <div class="form-storage"><button onclick="" class="storage-button"
/></div>

    </form< /td>

    <?php } ?>

    <?php };

?>
</tbody>
</table>

</div>

<?php
}

?>
</div>

<script>
function show_form() {
var d = document;
var a = d.getElementById('div_form').style.display;
d.getElementById('div_form').style.display = (a == 'none') ? 'block' : 'none';
}
</script>

```

Б.14 Функція відстеження і додавання накладних

```

<?php
require("head.php");
require_once("connect.php");
$name = $_SESSION["lang"] . "_name";
$desc = $_SESSION["lang"] . "_desc";

?>

<script src="js/jquery.min.js"></script>

<?php
$userData = count($_POST["code"]);

if ($userData > 0) {
    $fprovider=$_POST["provider"];
    $fwaybill=$_POST["waybill"];
    $fpayed=$_POST["payed"];
    $fdate= date("Ymd");
    for ($i = 0; $i < $userData; $i++) {
        if (trim($_POST['code'] != "") && trim($_POST['quantity'] != "") &&
trim($_POST['price'] != "")) {
            $fcode=$_POST["code"][$i];
            $fquantity = $_POST["quantity"][$i];
            $fprice = $_POST["price"][$i];
            $query          =          "INSERT          INTO          waybills
(id_row,waybill,provider,product_code,qty,provider_price,payed,bill_date)

```

VALUES

```
(DEFAULT,'$fwaybill','$fprovider','$fcode','$fquantity','$fprice','$fpayed','$fdate');
";
```

```
$result = pg_query($con, $query);
}
}
// echo "Data inserted successfully";
//echo $query;
} else {
//print_r($result);
// echo "Please Enter data";
} ?>
```

```
<div style="margin-top:10%;margin-left:5%;" >
```

```
<?php /*print_r($_SESSION);
```

```
echo '\n post';
```

```
print_r($_POST);
```

```
echo '\n get';
```

```
print_r($_GET); */?>
```

```
<div class="form-group container-contact">
```

```
<form name="add_waybill" id="add_waybill" method="POST">
```

```
<div class="row">
```

```
<div class="col-25"><label for="waybill"><?php echo
$translate[39][$lang]; ?></label></div>
```

```
<div class="col-75"><input type="text" name="waybill"
placeholder="#00000"></div></div>
```

```
<div class="row">
```

```
<div class="col-25"><label for="provider"><?php echo
$translate[40][$lang]; ?></label></div>
```

```

        <div class="col-75"><input type="text" name="provider"
placeholder="ООО Арсеняя"></div></div>

```

```

<table style="width:100%" id="dynamic_field">
<thead>
<tr>
<th><?php echo $translate[23][$lang]; ?></th>
<th><?php echo $translate[26][$lang]; ?></th>
<th><?php echo $translate[41][$lang]; ?></th>
</tr>
</thead>
<tr>
<div class="row"><td>
<select style="padding:12;" name="code[]">
<?php
$query='select code,',$query .= $name.' as nn from (select * from
product_store inner join product_desc using (id) ORDER BY id) a;';
$allcodes = pg_query($con, $query);
while ($row = pg_fetch_assoc($allcodes)) {
echo '<option value="" . $row["code"] . ">' . $row["nn"] . '
$row["code"] . '</option>';
}
?>
</select>
</td>
<td><input type="text" name="quantity[]" placeholder="0" /></td>
<td><input type="text" name="price[]" placeholder="0.0" /></td><td>
<button type="button" name="add" id="add" class="mybtn"><?php echo
$translate[42][$lang]; ?></button></td></div>
</tr>

```

```

</table>

<div class="row">

    <div class="col-25"><label for="payed"><?php echo
$translate[43][$lang]; ?></label></div>

    <div class="col-75"><input type="text" name="payed"></div></div>

    <input type="submit" class="btn btn-success" name="submit" id="submit"
value="<?php echo $translate[44][$lang];?>">

</form>

</div>

<div>

<div style="text-align:center;" class="container">

<?php
switch ($lang) {
case "en":
echo '<h2>History of added waybills</h2>';
break;
case "ua":
echo '<h2>Попередні накладні</h2>';
break;
case "ru":
echo '<h2>Предыдущие накладные</h2>';
break;
default:
echo '<h2>Попередні накладні</h2>';
break;
}
?>

<?php

```

```

    $l_query=runQuery($con,"select  waybill,  count(waybill)  as  nn  from
waybills group by waybill;");
    $array[]="";$i=0;
    foreach($l_query as $k => $v){
    $array[$i]=$l_query[$k]["waybill"];
    $i++;
    }
    //print_r($array);
    $i--;
    for($i;$i>=0;$i--){
        $h_query=runQuery($con,"select
provider,product_code,qty,provider_price,payed,bill_date  from  waybills  where
waybill='$array[$i]' ");
        //print_r($h_query);
        echo '<table class="waybill-table"><thead><tr>
<th>'.$translate[39][$lang].'</th>
<th colspan="3">'.$array[$i].'</th></tr>
<tr>
<th>'.$translate[40][$lang].'</th>
<th>'.$h_query[0]["provider"].'</th>

<th>'.$translate[45][$lang].'</th>
<th>'.$h_query[0]["bill_date"].'</th>
</tr>
<tr>
<th>'.$translate[23][$lang].'</th>
<th>'.$translate[25][$lang].'</th>
<th>'.$translate[26][$lang].'</th>
<th>'.$translate[41][$lang].'</th>
</tr>

```

```

</thead>

<tbody>;
foreach($h_query as $kk => $vv){
echo '<tr>

<td>';

$tmp=$h_query[$kk]["product_code"];
$tmp_query='select  ';$tmp_query.=$name.'  from  product_desc  where
code=\'";

$tmp_query.=$tmp.\';';
$t_query=runQuery($con,$tmp_query);
echo $t_query[0][$name];
echo '</td> <td>'.$h_query[$kk]["product_code"].'</td>

<td>'.$h_query[$kk]["qty"].'</td>

<td>'.$h_query[$kk]["provider_price"]*$curr_kof." ".$currency.'</td>

</tr>';
}
echo '
</tbody>

<tfooter>

<tr><td colspan="2">'.$translate[43][$lang].'</td>

<td
                                colspan="2">'.$h_query[$kk]["payed"]*$curr_kof."
"$currency.'</td></tr>

</tfoot></table>

<br>

<br>

<br>';

/*echo $tmp_query;
print_r($t_query);
echo $t_query[0][$name];
echo $h_query[0]["bill_date"];*/

```

```
}
?>
```

```
</div>
```

```
</div>
```

```
<script type="text/javascript">
```

```
$(document).ready(function() {
```

```
var i = 1;
```

```
$("#add").click(function() {
```

```
i++;
```

```
    $('#dynamic_field').append('<tr    id="row'    +    i    +    "'><div
class="row"><td><select name="code[]"><?php $query='select code,',$query .=
$name.' as nn from (select * from product_store inner join product_desc using (id)
ORDER BY id) a;'; $allcodes = pg_query($con, $query); while ($row =
pg_fetch_assoc($allcodes)) {echo '<option value="' . $row["code"] . "'>'
.$row["nn"].'    |    '. $row["code"] . '</option>'; }?></select></td><td><input
type="text" name="quantity[]" placeholder="0" /></td><td><input type="text"
name="price[]" placeholder="0.0" /></td><td><button type="button"
name="remove" id=""    +    i    +    ""    class="btn    btn-danger
btn_remove">X</button></td></tr></div>');
    });
```

```
$(document).on('click', '.btn_remove', function() {
```

```
var button_id = $(this).attr("id");
```

```
$('#row' + button_id + ").remove();
```

```
});

$("#submit").on('click', function() {
var formdata = $("#add_waybill").serialize();
$.ajax({
url: "action.php",
type: "POST",
data: formdata,
cache: false,
success: function(result) {
alert(result);
$("#add_waybill")[0].reset();
}
});
});
});
});
</script>
```

Б.15 Таблиці баз даних

```
--
-- PostgreSQL database dump
--
-- Dumped from database version 11.12 (Debian 11.12-0+deb10u1)
-- Dumped by pg_dump version 11.12 (Debian 11.12-0+deb10u1)
SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
```

```

SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;
SET default_tablespace = '';
SET default_with_oids = false;
--
-- Name: orders; Type: TABLE; Schema: public; Owner: www
--
CREATE TABLE public.orders (
    order_id integer NOT NULL,
    date date NOT NULL,
    customer_pii character varying NOT NULL,
    customer_phone character varying NOT NULL,
    customer_address character varying,
    status character varying NOT NULL
);
ALTER TABLE public.orders OWNER TO www;
--
-- Name: orders_detail; Type: TABLE; Schema: public; Owner: www
--
CREATE TABLE public.orders_detail (
    id_row integer NOT NULL,
    order_id integer NOT NULL,
    product_id integer NOT NULL,
    quantity integer NOT NULL,
    new_price real NOT NULL
);
ALTER TABLE public.orders_detail OWNER TO www;

```

```
--  
-- Name: orders_detail_id_row_seq; Type: SEQUENCE; Schema: public;  
Owner: www  
--  
CREATE SEQUENCE public.orders_detail_id_row_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
ALTER TABLE public.orders_detail_id_row_seq OWNER TO www;  
--  
-- Name: orders_detail_id_row_seq; Type: SEQUENCE OWNED BY;  
Schema: public; Owner: www  
--  
ALTER SEQUENCE public.orders_detail_id_row_seq OWNED BY  
public.orders_detail.id_row;  
--  
-- Name: orders_order_id_seq; Type: SEQUENCE; Schema: public;  
Owner: www  
--  
CREATE SEQUENCE public.orders_order_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
ALTER TABLE public.orders_order_id_seq OWNER TO www;
```

```
--
-- Name: orders_order_id_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: www
--
ALTER SEQUENCE public.orders_order_id_seq OWNED BY
public.orders.order_id;
--
-- Name: product_desc; Type: TABLE; Schema: public; Owner: www
--
CREATE TABLE public.product_desc (
    id integer NOT NULL,
    en_name character varying,
    en_desc character varying,
    ua_name character varying,
    ua_desc character varying,
    ru_name character varying,
    ru_desc character varying,
    picture character varying,
    code character varying
);
ALTER TABLE public.product_desc OWNER TO www;
--
-- Name: product_desc_id_seq; Type: SEQUENCE; Schema: public;
Owner: www
--
CREATE SEQUENCE public.product_desc_id_seq
AS integer
START WITH 1
INCREMENT BY 1
NO MINVALUE
```

```

NO MAXVALUE
CACHE 1;
ALTER TABLE public.product_desc_id_seq OWNER TO www;
--
-- Name: product_desc_id_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: www
--
ALTER SEQUENCE public.product_desc_id_seq OWNED BY
public.product_desc.id;
--
-- Name: product_store; Type: TABLE; Schema: public; Owner: www
--
CREATE TABLE public.product_store (
    id integer NOT NULL,
    qty integer DEFAULT 0,
    price real DEFAULT 0.00 NOT NULL
);
ALTER TABLE public.product_store OWNER TO www;
--
-- Name: product_store_id_seq; Type: SEQUENCE; Schema: public;
Owner: www
--
CREATE SEQUENCE public.product_store_id_seq
AS integer
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
ALTER TABLE public.product_store_id_seq OWNER TO www;

```

```

--
-- Name: product_store_id_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: www
--
ALTER SEQUENCE public.product_store_id_seq OWNED BY
public.product_store.id;
--
-- Name: ui_web; Type: TABLE; Schema: public; Owner: www
--
CREATE TABLE public.ui_web (
    id integer NOT NULL,
    title character varying NOT NULL,
    en character varying NOT NULL,
    ua character varying NOT NULL,
    ru character varying NOT NULL
);
ALTER TABLE public.ui_web OWNER TO www;
--
-- Name: ui_web_id_seq; Type: SEQUENCE; Schema: public; Owner:
www
--
CREATE SEQUENCE public.ui_web_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE public.ui_web_id_seq OWNER TO www;
--

```

```

-- Name: ui_web_id_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: www

--
ALTER    SEQUENCE    public.ui_web_id_seq    OWNED    BY
public.ui_web.id;

--
-- Name: users; Type: TABLE; Schema: public; Owner: www
--
CREATE TABLE public.users (
    id integer NOT NULL,
    username character varying,
    pii character varying,
    pass character varying NOT NULL,
    adm boolean DEFAULT false NOT NULL
);
ALTER TABLE public.users OWNER TO www;

--
-- Name: users_id_seq; Type: SEQUENCE; Schema: public; Owner: www
--
CREATE SEQUENCE public.users_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE public.users_id_seq OWNER TO www;

--
-- Name: users_id_seq; Type: SEQUENCE OWNED BY; Schema: public;
Owner: www

```

```
--  
ALTER SEQUENCE public.users_id_seq OWNED BY public.users.id;  
--  
-- Name: waybills; Type: TABLE; Schema: public; Owner: www  
--  
CREATE TABLE public.waybills (  
    id_row integer NOT NULL,  
    waybill character varying,  
    provider character varying,  
    product_code character varying,  
    qty integer DEFAULT 0,  
    provider_price real DEFAULT 0.0,  
    payed real,  
    bill_date date  
);  
ALTER TABLE public.waybills OWNER TO www;  
--  
-- Name: waybills_id_row_seq; Type: SEQUENCE; Schema: public;  
Owner: www  
--  
CREATE SEQUENCE public.waybills_id_row_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
ALTER TABLE public.waybills_id_row_seq OWNER TO www;  
--
```

```

-- Name: waybills_id_row_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: www

--
ALTER SEQUENCE public.waybills_id_row_seq OWNED BY
public.waybills.id_row;

--
-- Name: orders order_id; Type: DEFAULT; Schema: public; Owner: www
--
ALTER TABLE ONLY public.orders ALTER COLUMN order_id SET
DEFAULT nextval('public.orders_order_id_seq'::regclass);

--
-- Name: orders_detail id_row; Type: DEFAULT; Schema: public; Owner:
www
--
ALTER TABLE ONLY public.orders_detail ALTER COLUMN id_row
SET DEFAULT nextval('public.orders_detail_id_row_seq'::regclass);

--
-- Name: product_desc id; Type: DEFAULT; Schema: public; Owner:
www
--
ALTER TABLE ONLY public.product_desc ALTER COLUMN id SET
DEFAULT nextval('public.product_desc_id_seq'::regclass);

--
-- Name: product_store id; Type: DEFAULT; Schema: public; Owner:
www
--
ALTER TABLE ONLY public.product_store ALTER COLUMN id SET
DEFAULT nextval('public.product_store_id_seq'::regclass);

--
-- Name: ui_web id; Type: DEFAULT; Schema: public; Owner: www

```

```

--
ALTER TABLE ONLY public.ui_web ALTER COLUMN id SET
DEFAULT nextval('public.ui_web_id_seq'::regclass);
--
-- Name: users id; Type: DEFAULT; Schema: public; Owner: www
--
ALTER TABLE ONLY public.users ALTER COLUMN id SET
DEFAULT nextval('public.users_id_seq'::regclass);
--
-- Name: waybills id_row; Type: DEFAULT; Schema: public; Owner:
www
--
ALTER TABLE ONLY public.waybills ALTER COLUMN id_row SET
DEFAULT nextval('public.waybills_id_row_seq'::regclass);
--
-- Data for Name: orders; Type: TABLE DATA; Schema: public; Owner:
www
--
COPY public.orders (order_id, date, customer_pii, customer_phone,
customer_address, status) FROM stdin;
\
--
-- Data for Name: orders_detail; Type: TABLE DATA; Schema: public;
Owner: www
--
COPY public.orders_detail (id_row, order_id, product_id, quantity,
new_price) FROM stdin;
\
--

```

-- Data for Name: product_desc; Type: TABLE DATA; Schema: public;
Owner: www

--

COPY public.product_desc (id, en_name, en_desc, ua_name, ua_desc,
ru_name, ru_desc, picture, code) FROM stdin;

2	Violet Variant	Violet Variant test flower	Фіалка Вариативна
	Фіалка Вариативна квітка для тесту	Фиалка Вариативная	Фиалка
	Вариативная тестовый цветок	pictures/3.jpg	v2
3	Rum Punch \N	Ромовий пунш \N	Ромовый пунш \N
	pictures/54.jpg	rm54	
4	Saintpaulia grotei \N	Гроте\N	Гроте\N
	pictures/25.jpg	rm12	
5	Optimara MyLove\N	Фіалка Оптимара \N	Фиалка Оптимара
	\N pictures/14.jpg	rmv14	
6	Melodie Kimi	Мелодії Кімі	Мелодии Кими
	pictures/61.jpg	rmvk61	
1	Taboo	Violet Regular Test flower	Фіалка Табуу
	звичайна тестова квітка	Фиалка Табуу	Фиалка
	Тестовый цветок	pictures/2.jpg	Обыкновенная
		v1	
7	AV-Tango \N	Фіалка АВ-Танго \N	Фиалка АВ-Танго
	\N pictures/46.jpg	jm46	
8	Rob's Dandy Lion \N	Фіалка Роб Денді \N	Фиалка Роб Денди
	\N pictures/3.jpg	robs3	
9	Gillian	Фіалка Джиллиан	Фиалка Джиллиан
	pictures/8.jpg	jul8	
10	Ramblin' Dots	Фіалка Рамблин Дотс	Фиалка
	Рамблин Дотс	pictures/28.jpg	sd12
11	Boo Man	Буу Мен	Буу Мен
	pictures/47.jpg	bo47	

12	Shining Bell	Сяючий Дзвіночок	Сияющий
	Колокольчик	pictures/49.jpg	sdf49
13	Marshmallow	Зефір	Зефир
		pictures/50.jpg	mall50
14	Blue Chiffon	Синій шифон	Синий шифон
		pictures/44.jpg	gtr44
15	White Queen	Біла Королева	Белая Королева
		pictures/43.jpg	queue43

\.
 --
 -- Data for Name: product_store; Type: TABLE DATA; Schema: public;
 Owner: www
 --
 COPY public.product_store (id, qty, price) FROM stdin;
 3 100 250
 4 100 325
 5 100 315
 6 500 250
 7 500 285
 8 500 170
 9 500 210
 11 500 195
 12 500 235
 13 500 265
 14 500 315.5
 15 100 185.5
 1 500 190
 10 500 305
 2 500 175.5
 \.

--

-- Data for Name: ui_web; Type: TABLE DATA; Schema: public; Owner:

www

--

COPY public.ui_web (id, title, en, ua, ru) FROM stdin;

1	#home	Home	Головна	Главная	
2	#gallery	Gallery	Галерея	Галерея	
4	#logout	Logout	Вихід	Выход	
5	#login	LogIn	Увійти	Войти	
7	#storage	Storage	Склад	Склад	
8	#orders	Orders page	Сторінка замовлень	Страница заказов	
9	#admin	Control	Керування	Управление	
10	#fname	First name	Ім'я	Имя	
11	#lname	Last name	Прізвище	Фамилия	
12	#subject	Subject	Тема	Тема	
14	#write_something	Write your questions...	Задайте ваші питання...		

Напишите ваш вопрос..

15	#submit_contacts	Send message	Надіслати	повідомлення
----	------------------	--------------	-----------	--------------

Отправить сообщение

13	#phone	Tel number: Телефон:	Телефон:
6	#order	Order	Замовити Заказ
3	#feedback	Feedback	Зворотній зв'язок Обратная связь
16	#total	Total: Усього:	Всего:
17	#remove	Remove item	Видалити Удалить
18	#addprod	Add to Cart	Додати у кошик Добавить в корзину
20	#waybill	Waybill	Накладні Накладные
19	#emptycart	Your cart is empty	Ваш кошик порожній Ваша

корзина пуста

21	#checkout	Checkout	Оформити замовлення Оформить заказ
22	#empty_cart	Empty Cart	Очистити кошик Очистить корзину

заказа	23	#shopping cart	Shopping Cart	Кошик замовлення	Корзина
товара	24	#product_name	Product Name	Назва товару	Название
	25	#product_description	Description	Опис	Описание
	26	\$code	Code	Код	Код
	27	#quantity	Quantity	Кількість	Количество
	28	#unit_price	Unit price	Ціна за шт.	Цена за шт.
	29	#price	Price	Вартість	Стоимость
	30	#pic	Picture	Зображення	Картинка
продукт	31	#new_product	New product	Новий продукт	Новый
название	32	#name_engl	Name English	Назва Англійською	Английское
название	33	#name_ukr	Name Ukrainian	Назва Українською	Украинское
название	34	#name_rus	Name Russian	Назва російською	Русское
описание	35	#desc_en	English Desc.	Опис англійською	Английское
описание	36	#desc_ukr	Ukrainian desc.	Опис українською	Украинское
	37	#desc_rus	Russian Desc.	Опис російською	Русское описание
Пополнить запасы товара	38	#fill_up_product	Fill up product	Поповнити запаси	товару
	39	#add_new_prod	Add	Додати	Добавить
№	40	#waybill_num	Waybill number	Накладна №	Накладная
поставщика	42	#provider_price	Provider price	Ціна постачальника	Цена

```

43 #add_row_form Add row Додати рядок Додати рядок
44 #payed_for_waybill Payed for waybill Сплачено за накладну
Оплачено по накладной
45 #add_waybill Add Waybill Додати накладну Додати
накладную
46 #bill_date Bill date Дата внесения Дата внесения
41 #organization Organization (provider) Організація-
постачальник Организация-поставщик
47 #address Adress Адреса Адрес
48 #full_name Full Name Прізвище Ім'я Фамилия Имя
49 #city City Місто Город
50 #state State Область Область
51 #zip Zip Поштовий індекс Почтовый индекс
52 #payment Payment Оплата Оплата
53 #accepted_card Accepted Card Карты до оплаты Карты для
расчета
54 #name_on_card Name on Card Прізвище, ім'я на карті
Фамилия, имя на карте
55 #card_num Card number Номер карти Номер карты
56 #exp month Exp Month Місяць Месяц
57 #exp_year Exp Year Рік Год
58 #pay&order Pay&Order Заовити та сплатити Заказать и
оплатить
59 #shipment Shipment Доставка Доставка
\
--
-- Data for Name: users; Type: TABLE DATA; Schema: public; Owner:
www
--
COPY public.users (id, username, pii, pass, adm) FROM stdin;

```

```

2      adminТестовий                               Адміністратор
pbkdf2_sha256$12000$+M5PZ2H0bMzPk/BMfEU1Pg==$oKGN/f2OxjKO
pLey7s/zhOVkPd+radNqwX9WuVbwyYQ=      t

1      worker1      Тестовий                               Користувач
pbkdf2_sha256$12000$+M5PZ2H0bMzPk/BMfEU1Pg==$oKGN/f2OxjKO
pLey7s/zhOVkPd+radNqwX9WuVbwyYQ=      f

\
--
-- Data for Name: waybills; Type: TABLE DATA; Schema: public;
Owner: www
--

COPY public.waybills (id_row, waybill, provider, product_code, qty,
provider_price, payed, bill_date) FROM stdin;
10  12h55p      ООО "Цвет"      rmv14      100  235  100002021-
04-25
11  12h55p      ООО "Цвет"      sdf49 200  185  100002021-04-25
12  12h55p      ООО "Цвет"      queue43      50  250  100002021-
04-25
15  22Ag67      ООО "Садовод"  mall50      100  150  5000 2021-
04-25
16  22Ag67      ООО "Садовод"  sdf49 50  200  5000 2021-04-25
24  22Ag67      ООО "Садовод"  rm54 100  210  6000 2021-04-25
25  22Ag65      ООО "Садовод"  jul8 350  160  8000 2021-04-25
\
--
-- Name: orders_detail_id_row_seq; Type: SEQUENCE SET; Schema:
public; Owner: www
--

SELECT pg_catalog.setval('public.orders_detail_id_row_seq', 1, false);

```

```

--
-- Name: orders_order_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: www

--
SELECT pg_catalog.setval('public.orders_order_id_seq', 1, false);
--
-- Name: product_desc_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: www

--
SELECT pg_catalog.setval('public.product_desc_id_seq', 32, true);
--
-- Name: product_store_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: www

--
SELECT pg_catalog.setval('public.product_store_id_seq', 9, true);
--
-- Name: ui_web_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: www

--
SELECT pg_catalog.setval('public.ui_web_id_seq', 59, true);
--
-- Name: users_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
www

--
SELECT pg_catalog.setval('public.users_id_seq', 2, true);
--
-- Name: waybills_id_row_seq; Type: SEQUENCE SET; Schema: public;
Owner: www

--
SELECT pg_catalog.setval('public.waybills_id_row_seq', 25, true);

```

```
--  
-- Name: orders_detail orders_detail_pkey; Type: CONSTRAINT;  
Schema: public; Owner: www  
  
--  
ALTER TABLE ONLY public.orders_detail  
    ADD CONSTRAINT orders_detail_pkey PRIMARY KEY (id_row);  
  
--  
-- Name: orders orders_pkey; Type: CONSTRAINT; Schema: public;  
Owner: www  
  
--  
ALTER TABLE ONLY public.orders  
    ADD CONSTRAINT orders_pkey PRIMARY KEY (order_id);  
  
--  
-- Name: product_desc product_desc_pkey; Type: CONSTRAINT;  
Schema: public; Owner: www  
  
--  
  
ALTER TABLE ONLY public.product_desc  
    ADD CONSTRAINT product_desc_pkey PRIMARY KEY (id);  
  
--  
-- Name: product_store product_store_pkey; Type: CONSTRAINT;  
Schema: public; Owner: www  
  
--  
  
ALTER TABLE ONLY public.product_store  
    ADD CONSTRAINT product_store_pkey PRIMARY KEY (id);  
  
--
```

```

-- Name: ui_web ui_web_pkey; Type: CONSTRAINT; Schema: public;
Owner: www
--
ALTER TABLE ONLY public.ui_web
    ADD CONSTRAINT ui_web_pkey PRIMARY KEY (id);
--
-- Name: users users_pkey; Type: CONSTRAINT; Schema: public; Owner:
www
--
ALTER TABLE ONLY public.users
    ADD CONSTRAINT users_pkey PRIMARY KEY (id);
--
-- Name: users users_user_key; Type: CONSTRAINT; Schema: public;
Owner: www
--
ALTER TABLE ONLY public.users
    ADD CONSTRAINT users_user_key UNIQUE (username);
--
-- Name: waybills waybills_pkey; Type: CONSTRAINT; Schema: public;
Owner: www
--
ALTER TABLE ONLY public.waybills
    ADD CONSTRAINT waybills_pkey PRIMARY KEY (id_row);
--
-- Name: orders_detail orders_detail_order_id_fkey; Type: FK
CONSTRAINT; Schema: public; Owner: www
--
ALTER TABLE ONLY public.orders_detail

```

```
        ADD CONSTRAINT orders_detail_order_id_fkey FOREIGN KEY  
(order_id) REFERENCES public.orders(order_id);
```

```
--
```

```
-- PostgreSQL database dump complete
```

```
--
```

ДОДАТОК В
Слайди презентації

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЗАПОРІЗЬКА ПОЛІТЕХНІКА»
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ

Дипломна кваліфікаційна робота

ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН СИСТЕМИ СКЛАДСЬКОГО ОБЛІКУ

Виконала
студентка групи КНТ-227

Колташева А.О.

Керівник
доцент, к.т.н.

Колпакова Т.О.

1

Рисунок В.1 – Слайд №1

Об'єкт дослідження – програмне онлайн забезпечення для обліку складу.

Предмет дослідження – онлайн програми або сайти для автоматизації складського обліку.

Мета роботи – розробка програмної онлайн системи для зниження навантаження на персонал через автоматизацію трудомісткої роботи та для підвищення прибутку підприємства.

2

Рисунок В.2 – Слайд №2

ОСНОВНІ ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

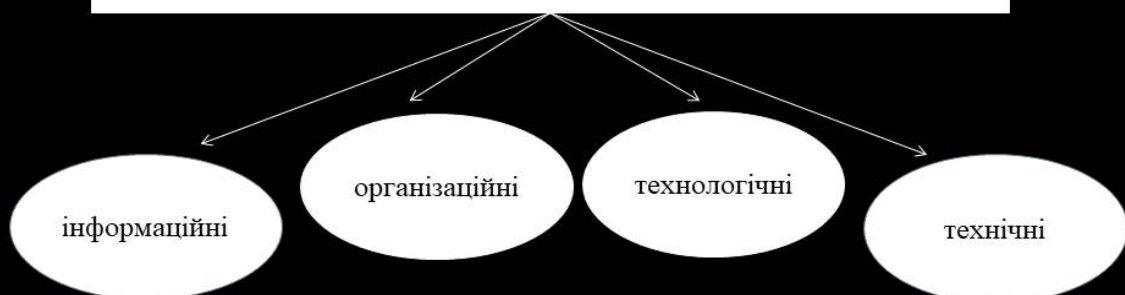
- Визначити і розібрати проблеми складського обліку
- Важливі вимоги до програмного забезпечення
- Провести аналіз та порівняння існуючих аналогічних розробок
- Розробити структурну схему та таблицю функцій онлайн забезпечення
- Провести SWOT-аналіз ручного засобу ведення обліку та за допомогою веб-пристроїв
- Обрати середовище проєктування веб-застосунку
- Обрати технологію з наявних варіантів мов програмування
- Охарактеризувати результативну інформацію програми

3

Рисунок В.3 – Слайд№3

АНАЛІЗ ПРОБЛЕМИ СКЛАДСЬКОГО ОБЛІКУ

Групи проблем, які вирішуються з **WMS-системою**



4

Рисунок В.4 – Слайд№4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Сумісність з іншими програмами та обладнанням.	Можливість завантаження і вивантаження бази в інші сервіси.
Оновлення в режимі реального часу.	Використання хмарних технологій.

5

Рисунок В.5 – Слайд №5

ПОРІВНЯННЯ ІСНУЮЧИХ АНАЛОГІВ ДЛЯ ОБЛІКУ СКЛАДУ

Критерії порівняння	Incrates	Контур.Маркет	ІBS Торгівля і склад
Загальні			
Робота в ОС Windows	+	+	+
Робота в ОС Linux	-	-	-
Вартість (USD)	14\$ в місяць	60-130\$ в рік	28\$ в місяць; 106\$ - безлімітне використання
Можливість працювати у програмі без її установки	-	+	+
Функціональні			
Облік техніки	+	+	+
Облік товарів	+	+	+
Можливість працювати на сервері онлайн	-	+	+
Підтримка баз даних SQL	+	+	+
Права та користувачі	+	+	+
Пошук по БД	-	+	+
Історія змін	-	+	+
Облік інвентарних номерів	+	+	+
Можливість налаштування інтерфейса	-	-	+
Створення власних шаблонів	+	-	+
Вбудовані друковані форми і звіти	+	+	-
Облік ремонтів	-	-	+
Облік витратних матеріалів	-	-	-
Всього:	8	10	13

6

Рисунок В.6 – Слайд №6

СТРУКТУРНА СХЕМА ОНЛАЙН ЗАБЕЗПЕЧЕННЯ



7

Рисунок В.7 – Слайд №7

ФУНКЦІЇ ОНЛАЙН ЗАБЕЗПЕЧЕННЯ

Функції системи	
Користувач	Адміністратор
Використання інтерфейсу	Додавання, редагування та видалення записів, інформації щодо товарів
Перегляд цікавих товарів	Перегляд сформованих таблиць та звітів організаційного характеру
Формування замовлення	Ведення зручного каталогу товарів
Заповнення особистих даних для оформлення замовлення	Формування звітів

8

Рисунок В.8 – Слайд №8

SWOT-АНАЛІЗ ВЕДЕННЯ ОБЛІКІВ СКЛАДСЬКОГО ОБЛАДНАННЯ РУЧНИМ ЗАСОБОМ

	Позитивні фактори	Негативні фактори
	Сильні сторони (внутрішній потенціал) (S)	Слабкі сторони (внутрішні недоліки) (W)
Внутрішнє середовище	Звичний, перевірений і надійний метод заповнення необхідної інформації вручну. Необхідність користування будь-якою програмою з таблицями окремо від сайту компанії. Немає потреби в потужному обладнанні.	Застарілий формат ведення документації (якщо це таблиці паперового формату). Необхідні базові вміння користування комп'ютером і ПЗ. Розрізнена структура файлів. Немає інтеграції з іншими системами. Немає вбудованих шаблонів звітності. Відсутній «розрахований на багато користувачів» режим. Відсутнє розмежування прав доступу.
Зовнішнє середовище	Можливість передачі інформації звітів на носіях або за коштами локальної та глобальної мережі.	Відсутність конфіденційності інформації. Безпека інформації.

9

Рисунок В.9 – Слайд №9

SWOT-АНАЛІЗ ВЕДЕННЯ СКЛАДСЬКОГО ОБЛІКУ ЗАСОБАМИ ВЕБ-ДОДАТКІВ

	Позитивні фактори	Негативні фактори
	Сильні сторони (внутрішній потенціал) (S)	Слабкі сторони (внутрішні недоліки) (W)
Внутрішнє середовище	Безкоштовно, доступ до системи онлайн. Простота поводження з програмою. Здатність швидко навчатися співробітникам. Немає потреби в потужному обладнанні. Можливість розмежування прав доступу. Вбудовані шаблони звітів, динамічні звіти. Інтеграція з іншими системами.	Базові вміння поводитися з комп'ютером і ПЗ. Досить незвичний вигляд програми.
	Потенційні можливості (O)	Наявні загрози (T)
Зовнішнє середовище	Скорочення часу на обробку даних. Можливість передачі інформації на носіях і за коштами локальної та глобальної мережі. Управління правами доступу до інформації. Робота в системі з будь-якої точки доступу до інтернету. Незалежність від більшості зовнішніх факторів.	Збіг системи. Поломка устаткування. Відсутність інтернет з'єднання

10

Рисунок В.10 – Слайд №10

ПОРІВНЯННЯ ВЕБ-ТЕХНОЛОГІЙ

Веб-технологія	Розробник	Операційна система	Ліцензія	Стабільність	Популярність	Підтримка
JavaScript	Sun Microsystems и Oracle	Windows, Mac OS, Solaris	GPL, BCL	+	+	За гроші
PHP	PHP Group, Zend Technologies	Unix або Linux, OS X та Microsoft Windows	Вільне і відкрите програмне забезпечення, дозволена ліцензія	+	+	Безкоштовно
CGI	Weta Digital	Unix або Linux, Windows, MAC	Комерційна	+	-	Безкоштовно
DHTML	WHATWG	Windows, MAC	AGPL	+	-	За гроші
ASP	Microsoft	Microsoft Windows	Пропрієтарна (Власницьке програмне забезпечення)	+	-	За гроші
CSS	CSS Working Group	Microsoft Windows, Linux, Mac OS	GPL	+	+	Безкоштовно
SSI	Storage Solutions	Unix	Комерційна	+	-	За гроші

11

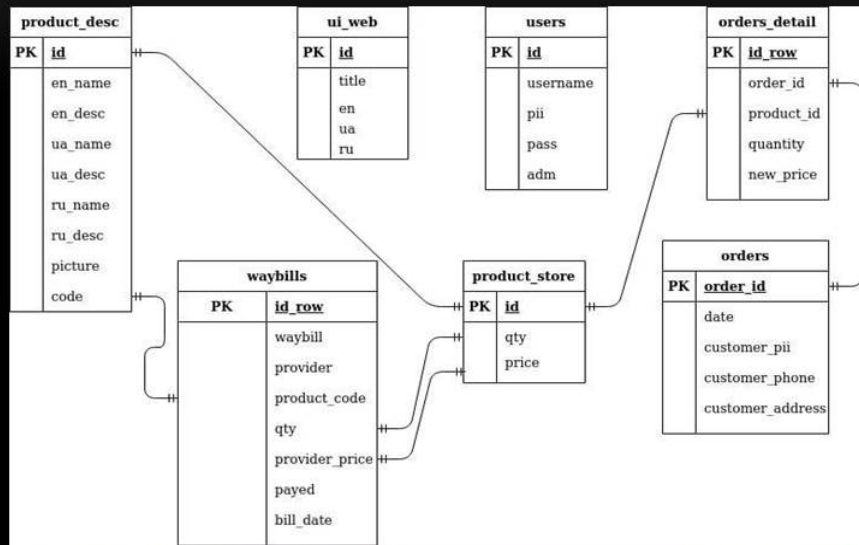
Рисунок В.11 – Слайд №11



12

Рисунок В.12 – Слайд №12

СХЕМА ТАБЛИЦЬ БАЗ ДАНИХ



13

Рисунок В.13 – Слайд №13

ФУНКЦІЇ ЗАСТОСУНКУ

Ласкаво просимо до Violet Flower Shop

Тут ви можете переглянути галерею красивих квітів та придбати деякі з них
Не соромтеся запитувати все, що вам потрібно, у розділі зворотного зв'язку

Галерея



Сторінка статусу складу

Новий продукт		Використання товару	
Зображення	Назва товару	Код	Вартість
1	Фіалка Табуру	т11	190 UAH
2	Фіалка Варієтальна	т12	175 UAH
3	Ромашка цукрова	т154	250 UAH
4	Горіч	т112	325 UAH
5	Фіалка Остінська	т1114	315 UAH
6	Мелодія Кіна	т11161	250 UAH
7	Фіалка АБ-Танго	т1146	285 UAH
8	Фіалка Рубі Літл	т1143	170 UAH
9	Фіалка Джамілія	т118	210 UAH
10	Фіалка Рамона Дот	т112	305 UAH
11	Блу Мен	т117	195 UAH
12	Скромний Діамант	т1149	225 UAH
13	Зефір	т1150	265 UAH
14	Соней шифон	т1141	315 UAH
15	Біла Королева	т1143	195 UAH

14

Рисунок В.14 – Слайд №14

ФУНКЦІЇ ЗАСТОСУНКУ


Попередні накладні

Накладна №	22Ag67		
Організація-постачальник	ООО "Садовод"	Дата внесення	2021-04-25
Назва товару	Код	Кількість	Ціна постачальника
Зефір	mal50	100	150 UAH
Сяючий Дзвіночок	sdf49	50	200 UAH
Ромовий пунш	rm54	100	210 UAH
Сплачено за накладну		6000 UAH	

Накладна №	12h55p		
Організація-постачальник	ООО "Цвет"	Дата внесення	2021-04-25
Назва товару	Код	Кількість	Ціна постачальника
Фіалка Оптимара	tmv14	100	235 UAH
Сяючий Дзвіночок	sdf49	200	185 UAH
Біла Королева	quue43	50	250 UAH
Сплачено за накладну		10000 UAH	



Накладна №

Організація-постачальник

Назва товару	Кількість	Ціна постачальника	
Фіалка Табуу v1	0	0.0	Додати рядок

Сплачено за накладну

[Додати накладну](#)

15

Рисунок В.15 – Слайд №15

ВИСНОВКИ

- проведено аналіз існуючих аналогічних розробок, розглянуто альтернативний, більш сучасний варіант
- розроблено структурну та функціональну схеми онлайн застосунку
- з'ясовано перелік важливих для додатка функцій
- проведено SWOT-аналіз позитивних та негативних факторів застарілого, ручного методу ведення обліку та обліку за допомогою веб-ресурсів
- обрано мову PHP для проектування самого застосунку, з допомогою ресурсів HTML та CSS для оформлення стилів
- описано характеристики та функції розробленого застосунку

16

Рисунок В.16 – Слайд №16