

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ
УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ПОБОДЖЕНО **ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**
НУБІП України

Декан факультету
Інформаційних технологій

Глазунова О.Г., д.пед.н, проф.

підпис

ПІБ, вчене звання і ступінь

Завідувач кафедри
Комп'ютерних систем і мереж

Лакно В.А., д.т.н., проф.

підпис

ПІБ, вчене звання і ступінь

« » 2021 р. « » 2021 р.
НУБІП України
МАГІСТЕРСЬКА РОБОТА

На тему: «Дослідження веб-додатку для прогнозування та замовлення товарів на склад»

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма _____

Орієнтація освітньої програми _____

НУБІП України

Керівник дипломного проекту: _____ / Шкарупило В. В. /

підпис

ПІБ

Виконав: _____ / Клементьєв Д.Р. /

підпис

ПІБ

НУБІП України
КИЇВ-2021

НУБІП України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

«ЗАТВЕРДЖУЮ»

завідувач кафедри

комп'ютерних систем і мереж

Лахно В.А., д.т.н., проф.

підпис

ПІБ, вчене звання і ступінь

20__ р.

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ

Клементьєв Дарій Романович

(прізвище, ім'я, по батькові)

Спеціальність (напрямок підготовки) комп'ютерна інженерія _____

Освітня програма _____

Орієнтація освітньої програми _____

Тема магістерської роботи _____

затверджена наказом ректора НУБіП України від “___” _____ 20__ р. № _____

Термін подання завершеної роботи на кафедру _____

Вихідні дані до магістерської роботи _____

Перелік питань, що підлягають дослідженню:

1. _____
2. _____
3. _____

Перелік графічного матеріалу (за потреби) _____

Дата видачі завдання “___” _____ 2021 р.

Керівник магістерської роботи _____ Шкарупило В. В., к.т.н., доц. _____

(підпис)

(прізвище та ініціали)

Завдання прийняв до виконання _____ Клементьєв Д. Р. _____

(підпис)

(прізвище та ініціали студента)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної області		Виконано
2	Проектування системи		Виконано
3	Реалізація системи		Виконано
4	Тестування системи		Виконано
5	Оформлення пояснювальної записки		Виконано
6	Оформлення графічного матеріалу		Виконано

Студент

(підпис) (ініціали та прізвище)

Керівник проекту (роботи)

(підпис) (ініціали та прізвище)

РЕФЕРАТ НУБІП України

Пояснювальна записка: __ сторінка, __ рисунка, __ таблиці, __ додатків, __

джерел. НУБІП України

КОМП'ЮТЕРНА СИСТЕМА, КЕРУВАННЯ СКЛАДОМ, ТОВАР,
АДМІНІСТРАТОР, ВІЗУАЛЬНА ЧАСТИНА, ІНТЕРФЕЙС, ВІКНО

КОРИСТУВАЧА, LARAVEL 6, MYSQL, ELASTICSEARCH, REDIS, VUE, ELEMENT UI НУБІП України

Мета – розробка максимально простого та інтуїтивного зрозумілого для користувача інтерфейсу для автоматизованої системи для прогнозування та замовлення товарів на склад, що повинно призвести до скорочення втрат часу роботи працівників.

НУБІП України

Об'єкт – веб-додаток для прогнозування та замовлення товарів на склад.

Предмет розробки – методи та програмні додатки для веб-додатку.

НУБІП України

Проект складається з трьох розділів.

Перший розділ присвячено аналізу предметної області. Проводиться детальний огляд об'єкта, опис вибраної мови програмування та її аналіз. Проведено огляд існуючих рішень,

НУБІП України

У другому розділі розкриті питання щодо вимог автоматизованої системи.

Третій розділ присвячено проектуванню та реалізації компонентів системи. Проведено моделювання поведінки та структури системи.

НУБІП України

В результаті виконання дипломної роботи проведено аналіз, моделювання та дослідження розробленої автоматизованої системи та розроблені рекомендації щодо її проектування.

НУБІП України ^{ЗМІСТ}

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ 7

НУБІП України

ВСТУП 8

1 АНАЛІТИЧНИЙ ОГЛЯД 10

1.1 Дослідження предметної області 10

1.2 Призначення системи 12

1.3 Огляд існуючих рішень 15

НУБІП України

2 ВИМОГИ ДО СИСТЕМИ 20

2.1 Загальні відомості про проект 20

НУБІП України

2.2 Цілі і задачі проекту 21

2.3 Структура проекту 22

2.4 Вимоги до навантаження 23

НУБІП України

2.5 Бізнес-вимоги до системи 24

2.6 Розподіл доступу до ресурсу 25

2.7 Загальні елементи сторінок 26

2.8 Сторінки веб-сайту 27

НУБІП України

2.9 Розробка SQL запитів 28

3 ПРОЕКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ 32

3.1 Моделювання поведінки системи 32

НУБІП України

3.2 Прототипування графічного інтерфейсу 38

3.3 Розгортання та налаштування системи 39

3.4 Розробка графічного інтерфейсу 43

	6
3.5 Перевірка системи на стресостійкість.....	50
3.6 Розміщення додатку на хостингу.....	53
ВИСНОВКИ.....	57

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	59
-------------------------------	----

Додаток А.....	60
НУБІП України	

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПК – персональний комп'ютер

HTML мова розмітки гіпертексту, HyperText Markup Language

CSS каскадні таблиці стилів, Cascading Style Sheets

front-end інтерфейс для взаємодії між користувачем і back-end

JS – JavaScript

DOM об'єктна модель документа, Document Object Model

BBC Британська Телерадіомовна Корпорація, British Broadcasting Corporation

cookies – дані отримані від веб-сайту на веб-сервері

pop-up – спливне вікно

OPD методологія об'єктних процесів, Object Process Methodology

ISO – міжнародна організація зі стандартизації, International Organization for Standardization

Mb мегабайт

UI інтерфейс користувача, User interface

Ajax – підхід до побудови призначених для користувача

інтерфейсів веб-додатків, що полягає в «фоновому» обміні

даними браузера з веб-сервером, Asynchronous JavaScript and XML

API – програмний інтерфейс застосунку, Application Programming Interface

ВСТУП

НУБІП України

Нині під час переходу до ринкової економіки багатьом підприємств і закупівельних організацій постає актуальна на сьогодні завдання – ведення обліку з допомогою технічних засобів. Значення автоматизації велике, оскільки за його впровадженні посилюються контрольні функції з виконання зобов'язань та своєчасних платежів податків у бюджет. При автоматизації обліку реалізації продукції скоротиться як час, витрачається обробку інформації, а й підвищиться якість цієї обробки. Процес реалізації продукції, виготовленої підприємством, є найважливішим у його діяльності, оскільки, завершуючи оборот коштів підприємства, він дозволяє відшкодувати витрати та виконати зобов'язання перед бюджетом за податковими платежами, позабюджетними фондами, установами банків за отриманими кредитами, постачальниками та кредиторами.

У процесі реалізації продукції формується кінцевий фінансовий результат виробничо-фінансової діяльності – прибуток чи збиток. Прибуток (збиток) від продукції (робіт, послуг) - це позитивна (негативна) різниця між виручкою від продукції в діючих цінах без ПДВ і акцизів і витратами її виробництва і

Підприємства можуть мати різний масштаб та обсяг документообігу. Тому автоматизована система дозволить вести обробку даних або документів як на одному робочому місці (малі компанії), так і на кількох (великі компанії), закріплюючи за кожним співробітником ділянку робіт: прийом товарів від постачальників, облік оптової реалізації, облік роздрібною реалізації, складання калькуляційних карт і т.д.

За такого обсягу оброблюваної інформації без використання технічних засобів нерідко виникають помилки, які обертаються в організацію відчутними фінансовими збитками. В даний час, при гострій конкуренції на ринку продовольчих товарів організація потребує оперативного отримання достовірних

даних про обсяги продажу, тенденції зростання або спаду попиту на окремі види товарів.

В результаті виконання роботи проведено аналіз процесів складської логістики приладобудівного підприємства, реінжиніринг бізнес-процесів

логістичних завдань, пов'язаних з урахуванням комплектацій збірок, що видаються у виробництво, а також товарно-матеріальних цінностей, що використовуються як засоби виробництва. Запропоновано методика, виконано проектування та розробку

автоматизованої системи управління складськими комплексами з урахуванням

їхньої інтеграції в єдиний інформаційний простір підприємства, розроблено автоматизовану систему складського обліку.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Дослідження предметної області

НУБІП УКРАЇНИ

Нині однією з найважливіших завдань виробничої діяльності є регулярне постачання населення продуктами харчування, одягу та непродовольчими товарами. Цією сферою діяльності займається така галузь, як логістика. Логістика – наукова сфера про ефективне планування та організацію матеріальних та інформаційних потоків, а також організація товароруку найкращим способом від виробника до споживача товару чи послуги.

Початок повсюдного використання логістики було покладено за часів Другої світової війни, яка застосовувалася при вирішенні складних стратегічних та політичних завдань, а також сприяла взаємодії всіх важливих підрозділів промисловості та постачальних баз. Потоки транспорту в логістиці вишикувалися з метою забезпечення зброєю, продовольчими товарами, паливом та продуктами харчування. З цього випливає, що логістика як наука стала затребуваною ще за часів другої світової, але особливу увагу логістичним процесам почали приділяти порівняно недавно.

Однією з основних завдань логістики по праву можна назвати розробку та проектування системи контролю за потоками матеріалів та інформації, створення ефективних транспортних потоків, раціональний розподіл фінансових потоків, а також створення результативного руху товарів на складах, виробництві, між постачальниками продукції та їх споживачами тощо... з метою економії часу та мінімізації витрат.

Об'єктами логістики є інформаційні, матеріальні, транспортні, фінансові потоки у виробничій, економічній та господарській діяльності.

НУБІП УКРАЇНИ

Логістика допомагає ефективно вирішувати такі питання як:

- раціональне пересування товарів та різних потоків;
- управління трудовими фондами;
- раціональне розміщення для зберігання запасів сировини, матеріалів,

необхідних безперервної роботи виробництва;

- обробка інформації, що надходить на склад;
- зниження витрат на товарорух і т.д.

При недостатній увазі до організації логістичних процесів відбувається зниження ефективності роботи персоналу, а внаслідок підприємства в цілому, а

також призводить до підвищення витрат, які будуть необхідні для реалізації обробки інформації та розміщення вхідних матеріальних потоків. Виходячи з сучасних технологій та економічних вимог, які пред'являються до складського приміщення та складських систем, вся діяльність має бути автоматизована.

Автоматизація – це напрямок наукового та технічного прогресу, який використовує технічні методи та математичний аналіз з метою звільнити людину від складних процесів на виробництві. Це складний комплекс заходів, який

призначений для реалізації контролю за всім виробничим процесом. Для отримання найоб'єктивнішої інформації можуть бути використані різні програмні модулі, такі як 1С, Vector, АСУС, WMS тощо. Дані модулі призначені для автоматизації практично всіх аспектів виробництва, включаючи облік продукції, управління

замовленнями, контроль за бракованими партіями та управління комплексною автоматизацією на складі.

Система управління складом – інформаційна система для управління, яка забезпечує автоматизацію на складі та оптимізує всі складські процеси. Система управління складом розробляється спеціально для забезпечення контролю за

матеріальними та інформаційними потоками. Для того, щоб вважати склад повністю автоматизованим, необхідно, щоб усі складські процеси, наприклад, такі як переробка тарно-пакувальних та штучних вантажів вироблялися автоматичними

пристроями без участі людини. Так само складське господарство повинно мати власну систему управління складом, розроблену з врахуванням усіх особливостей приміщення, розташування складу та виробництва, та товару, що зберігається.

Основною складовою технологічних процесів на харчовому виробництві є забезпечення особливих вимог зберігання продукції, їх транспортування та реалізація.

Склади є невід'ємною частиною технологічного процесу підприємств.

Автоматизована система складування на підприємстві може вирішувати всі завдання, пов'язані з урахуванням сировини на складі, її прийомом, відстеження дати постачання сировини на склад та термінів зберігання продукції.

Автоматизація складської діяльності надає конкурентне лідерство на ринку, полегшує роботу з інформаційними потоками, зменшує трудомісткість роботи людини, підвищує контроль на складі, а головне прискорює логістичний процес всього підприємства. Всі ці переваги дозволяють значно скоротити витрати на логістичні процеси.

1.2 Призначення системи

Склад є основним елементом руху потоку товару чи сировини від першоджерела сировини (постачальника) до споживача готової продукції. Склад - це складна технічна споруда, яка призначена для зберігання сировини, продуктів харчування та інших різних товарів народного споживання, з дотриманням необхідних режимів температури та з жорсткими вимогами до зберігання тієї чи іншої продукції [3]. Складське приміщення може включати безліч пристроїв, які виконують функції зберігання, транспортування, приймання товару на зберігання і видачі на виробництво в потрібній кількості. Переміщення потоків сировини в

будь-якому логістичному ланцюжку неможливе без зберігання запасів сировини або продовольчого товару у спеціально відведених для цього місцях. З цього випливає, що склад є ефективним засобом управління запасами та їх концентрації

на всіх ділянках логістичного ланцюга. Використання складів зумовлене такими

перевагами як:

- зменшення витрат за логістику підприємства;
- задоволення попиту споживачів з допомогою зберігання запасів великого асортименту продукції;

- виняток простоїв виробництва;

- зменшення собівартості продукції та витрат на транспортування сировини за рахунок власного сховища

Склад для того, щоб забезпечити безперебійну роботу виробництва повинен

мати такі структурні складові як:

- майданчики, будівлі для зберігання та інші споруди, призначені для розміщення товару та забезпечення його безпеки;

- пристрої для вантажно-розвантажувальних робіт, стелажі та інші пристрої, призначені для переміщення сировини та її зберігання;

- система управління складськими операціями, яка здійснює контроль за матеріальними та інформаційними потоками на складі, та для реалізації обліку сировини.

Існує три види потоків на складі: вхідні, вихідні та внутрішні. Вхідний потік

- це розвантаження товару, що прибув або сировини, перевірка його якості і кількості, а також перевірка всієї супроводжуючої документації. Вихідним потоком прийнято вважати необхідність завантаження транспортного засобу та підготовку

паketу супровідних документів. Внутрішній потік - це рух товарно-матеріальних

цінностей у рамках складського господарства та оформлення всіх документів, що регламентують дані переміщення [3].

Серед послуг складування особливу увагу займають матеріальні послуги (див. рис. 1.1)

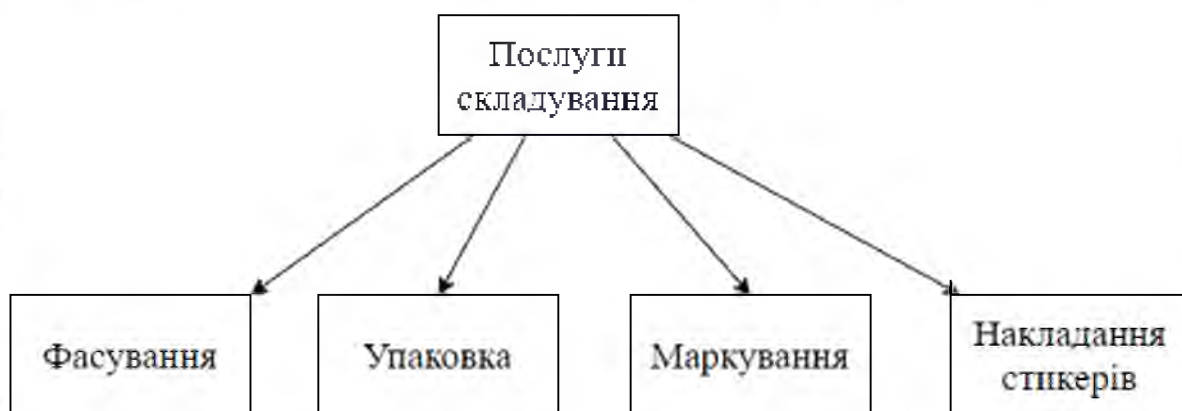


Рисунок 1.1 – Послуги складування

Фасування – це процес поміщення товару в упаковку при надходженні його на склад.

Упаковка – це розташування товарно-матеріальних цінностей у спеціалізовані засоби для забезпечення збереження товару під час зберігання чи транспортування. Існує два типи упаковки: транспортна тара та споживча упаковка.

При транспортуванні або зберіганні в транспортній тарі можливе нанесення спеціальних позначень, які дозволяють ідентифікувати товар (найменування товару, вага, одержувач вантажів тощо). Споживча упаковка служить для надання інформації про склад продукту, виробника і т. д.

Маркування – нанесення на тару умовних позначень, призначених для розпізнавання вантажу.

Стікерування – процес присвоєння товару чи сировині цифрового коду (штрих-коду), за допомогою наклеювання стікера. Інформація про товар заноситься до інформаційної системи, що дозволяє контролювати різні параметри товару, наприклад, термін придатності та температурні режими, а також спрощує процес обліку сировини.

Основним логістики складування є технологічний процес, тобто послідовність операцій з підготовки сховища до приймання товару або сировини, розміщення його на зберігання, суворий контроль за товаром, що зберігається, і підготовка сировини до відпуски зі складського приміщення. Рівень

конкурентоспроможності підприємства над ринком повністю залежить від цього, наскільки ефективним і результативним буде виконання всіх процедур, що з складської діяльністю.

Грунтуючись на дотриманні наступних принципів, можна раціонально організувати всі процеси всередині складу:

- автоматизовані та механізовані системи повинні стати головною особливістю технологічного процесу;
- ефективне використання площі та ємності сховища;
- планомірність та ритмічність складських операцій, повний контроль за збереженням товару;
- раціональний підхід до вибору обслуговуючого обладнання та інформаційного забезпечення складу.

1.3 Огляд існуючих рішень

Виробництво на сьогоднішній день неможливо уявити без систем автоматизованого управління. Склад сировини чи готової продукції є невід'ємною частиною логістики підприємства, що піддається автоматизації повною мірою.

Номенклатура найменувань товару може становити кілька десятків позицій, що потребує простоти пошуку необхідної назви та швидкого доступу до нього [4].

Наприклад, автоматизовані складські системи від компанії Kardex – одна з провідних компаній світового рівня, виробник автоматизованих систем

складування, забезпечує підприємство ефективним інструментом, який приносить помітну вигоду, зручність у роботі та підвищує продуктивність праці у складському господарстві. Kardex займається автоматизацією практично всіх аспектів складської діяльності від отримання замовлення до клієнта до повної комплектації відбраного товару та його відвантаження.

Для того щоб наочно продемонструвати впровадження автоматизації складу, наведемо наступний приклад. Проектне рішення групи компаній «Складовичок» фермерське господарство, що займаються виробництвом продуктів з індики, та ТОВ «ІТПрог» - українська компанія інформаційних технологій, що спеціалізується на роботі з м'ясопереробними підприємствами, метою спільної реалізації яких стала робота з проектування та реалізації розробка автоматизації складу м'ясної продукції. [5]. Для вирішення поставленого завдання було обрано систему управління під назвою «ISIT: WMS». Ця система повністю піддавалася інтеграції із роботизованим обладнанням. IT-рішення, яке було розроблено для складу м'ясної продукції, дозволяє автоматизувати практично всі логістичні процеси у сховищі. Автоматизована система складу вважає інтелектуальне керування параметрами складу за допомогою вироблених алгоритмів та рекомендацій, які приходять на термінали збору даних працівників (операторів) складу.

1.3.1 Класи систем керування складськими операціями.

Існує три основні класи систем керування складськими операціями.

Замовні системи переважно впроваджуються у великих складських будинках. Ця система призначена для сховища зі складною технологією виконання складських операцій.

Стандартні "коробкові" системи. Ця система передбачає продаж програмного забезпечення як готової програми. Система управління обмежена за функціональністю, і здебільшого є системами обліку чи контролю над складською діяльністю. Орієнтована дана система на середні та малі підприємства.

Адаптовані системи – це динамічніші системи управління, оскільки вони орієнтовані на середні підприємства зі складним технологічним процесом складу. Заснована ця система на стандартних складських операціях.

1.3.2 Система керування складом Vector

Автоматизована управлінська система Vector охоплює практично всі сторони складської діяльності. За допомогою даної системи сховище набуває вигляду єдиного механізму, в якому чітко налагоджені всі операції складування.

Ця система передбачає автоматизацію такої діяльності як:

- приймання сировини складу;
- розміщення сировини по задалегідь визначеним місцям;
- контроль якості продукції, що приходить;
- повний спектр операцій із комплектування замовлень;
- вихідний контроль продукції;
- інвентаризація;
- відвантаження замовлення;
- формування супровідних документів.

Застосування цієї системи дозволяє вирішувати такі завдання управління, як:

- контроль за складськими запасами;
- контроль за розміщенням замовлення;
- оцінка вхідних матеріальних потоків;
- оптимізація роботи операторів;
- координація черговості відвантаження замовлень.

"Vector" вимагає від оператора (персоналу) підтвердження кожної виконаної дії. Ця автоматизована система гарантує моніторинг кожної вхідної та вихідної одиниці товару (кількість, якість, статус комплектування тощо). Якщо система виявляє помилку, то на пульт оператора надходить інформація про несправність.

1.3.3 Система керування складом АСУС

Автоматизована система управління складом (АСУС) ґрунтується на методах економіки та математики при використанні обчислювального процесора [7]. Ця система повністю контролює такі процеси як: прийом сировини, зберігання та видача сировини (з) складського приміщення. АСУС має структурну одиницю, яка вирішує питання з бухгалтерського обліку, оперативного управління складськими процедурами та ін. Також система роздруковує супровідні документи, які потрібні на склад.

1.3.4 Система управління складом WMS

Система управління складом – це система із застосуванням комп'ютеризованих технологій, за допомогою якої вся складська діяльність оптимізується та автоматизується. Скороченою назва цієї системи - WMS («Warehouse Management System») [8]. Ця система призначена для ефективно роботи складських операцій, що покликана підвищити продуктивність робочого персоналу на складі.

WMS-система – це необхідний елемент для підприємства будь-якої галузі, чи то логістика, торгівля, виробництво, медицина тощо. Основне завдання цієї системи – це керування матеріальними потоками. WMS-система дозволяє вирішувати питання щодо обліку витрат, розрахунку собівартості товару, пропускної спроможності сховища, про якість та швидкість комплектування замовлення та його відвантаження.

1.3.5 Microsoft Business Solution – Ахарта

Microsoft Business Solution-Ахарта – це об'єднане рішення для виробничих, торгових та дистриб'юторських компаній. Основними характеристиками цієї системи є: необмежену кількість складських приміщень, складів, осередків та складської тари; облік товарів; контроль за завантаженням та розвантаженням

сировини в транспортній тарі; підтримка штрихкодуювання; зниження витрат за рахунок реалізації ефективного зберігання товарів і т.д.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

2 ВИМОГИ ДО СИСТЕМИ

2.1 Загальні відомості про проект

Сучасний склад – це складна технічна споруда, яка складається з численних взаємопов'язаних підсистем (комплекс будівель, сукупність вантажів, що переробляються, система інформаційного забезпечення), що мають певну структуру, і виконує функції з перетворення матеріальних потоків, а також накопичення, переробки та розподілу вантажів між споживачами.

У логістичні склад – це місце перетворення матеріальних потоків, вкладених у задоволення клієнтської бази. Склади зустрічаються по всьому шляху руху матеріальних потоків, починаючи зі складів постачальників сировини і матеріалів і закінчуючи складами системи розподілу. Саме тому складська логістика відіграє важливу роль у логістичній системі та у забезпеченні конкурентоспроможності організації в умовах ринкової економіки.

Основна мета існування складу – концентрація запасів, їх зберігання та забезпечення безперебійного та ритмічного постачання замовлень споживачів.

При цьому можливе різноманіття параметрів, технологічних та об'ємно-планувальних рішень, конструкцій обладнання та характеристик різноманітної номенклатури вантажів, що переробляються на складах, відносить склади до складних систем. У той же час склад сам є лише елементом системи вищого рівня – логістичного ланцюга, який і формує основні та технічні вимоги до складської системи, встановлює цілі та критерії її оптимального функціонування, диктує умови переробки вантажу.

Система для прогнозування та замовлення товарів на склад – це інтернет додаток який призначений для підприємницької діяльності, яка користується послугами тимчасових складських приміщень або має власний склад. Основний вік – усі. Основна конкурентна особливість – унікальність серед систем прогнозування та замовлення товарів на склад, а саме, створення специфічного додатку який

призначений для складських приміщень. Сайт повинен ознайомити відвідувача з можливостями сайту і їх властивостями, надати корисну інформацію відвідувачеві, підвищити лояльність користувачів. Організувати зручну і інтуїтивно зрозумілу взаємодію співробітників з товарами на складі з метою підвищення прибутку.

2.2 Цілі і задачі проекту

Цілі створення системи:

- автоматизація прогнозування та замовлення товарів на склад;
- скорочення часу на роботу з товаром за рахунок функціоналу сайту;
- донесення інформації про події, які пов'язані з товаром в зручній і зрозумілій формі;
- підвищення лояльності працівників.

Завдання інтернет-сайту:

- викликати у відвідувача відчуття довіри до підприємства;
- можливість реєструвати час прибуття товару на склад з максимальною автоматизацією процесу;
- автоматизація процесу перевірки наявного товару на складі викладачами і надання співробітникам звітної документації;
- надання інформації про підприємство та події у ньому в зручній формі;
- допомога у виборі звітності в автоматичному режимі. Наприклад, надаючи відвідувачеві інформацію по найменш популярні товари для кожної групи чи окремо по користувачам;
- скорочення взаємодії користувачів з співробітниками;
- зворотній зв'язок з відвідувачами.

Цільова аудиторія – чоловіки і жінки від 18 до 45 років з активний спосіб життя. Проживають у містах України. Рівень матеріального стану не важливий.

2.3 Структура проекту

Структура сайту має складатися з наступних елементів:

a) Головна сторінка:

- 1) Слайдер з демонстрацією товару;
- 2) Контактні дані (телефон, виклик живого чату);
- 3) Активація прогнозування у ручному режимі.

b) Загальні відомості сайту:

- 1) Опис сайту;
- 2) Дані для зв'язку з розробником.

c) Адміністративна панель:

- 1) Перегляд товару по назві;
- 2) Перегляд товару по характеристикам;
- 3) Перегляд товару який в дорозі;
- 4) Перегляд відсутнього товару;
- 5) Таблиці розташовуються сіткою;
- 6) Можливість переглянути детальну інформацію про товар;
- 7) Можливість створити звіт;
- 8) Можливість надіслати повідомлення на пошту.

d) сторінка товару:

- 1) Фотографія товару;
- 2) Назва товару;
- 3) Опис товару;

НУБІП України

- 4) Перегляд характеристик;
- 5) Можливість переглянути історію продажу.

e) Статті (блог?):

1) Корисні статті;

НУБІП України

2) Розширена демонстрація звітності.

f) Співробітництво:

1) Карусель наших розробників;

2) Форма для відправки запиту на багрепорт.

НУБІП України

g) Контакти:

1) Контактний телефон, email;

2) Адреса складу.

h) Згода на обробку персональних даних, cookies, отримання рекл. Розсилки.

НУБІП України

i) Сторінка «Реєстрації».

j) Сторінка «Персональний кабінет».

2.4 Вимоги до навантаження

НУБІП України

По закінченню робіт і при закінченні етапів розробки кожного сайту системи проводиться тестування навантаження системи. Робота Виконавцем вважається виконаним, якщо пройдено моніторинг продуктивності Бітрікс.

НУБІП України

Система повинна витримувати навантаження до 15000 унікальних користувачів на добу. Швидкість будь-якої сторінки сайту до 1 секунди під навантаженням.

НУБІП України

Наповнення сайту контентом замовника виконується замовником.

2.5 Бізнес-вимоги до системи

Оскільки основним завданням є створення системи моніторингу відвідування

студентів, то необхідно сформулювати основні бізнес-вимоги:

- система повинна сприяти і сприяти організації складського процесу в підприємстві;

- система буде використовуватися для ініціалізації процесів на великих

складських приміщеннях;

- система сприятиме прогнозуванню товарів на склад;

- система повинна допомагати в прийнятті рішень, спрямованих на запобігання негативних рішень працівників;

- система забезпечуватиме моніторинг успішності товарів на ринку;

- створення єдиного цифрового середовища для працівників.

Програма повинна виконувати такі завдання.

- організувати зберігання, обробку даних та виведення на друк;

- максимально зменшити витрати на обробку інформації;

- мати зручні форми вводу-виводу з хорошим інтерфейсом для користувача;

- виключити ймовірність допущення арифметичних та логічних помилок;

- контролювати дії користувача під час роботи з програмою і повідомляти

йому про припущення помилки та можливість її виправлення;

- виводити на друк певні форми документів (Накладна, книга складського обліку, інвентарний перелік основних засобів);

- організувати створення звітів та документів.

2.6 Розподіл доступу до ресурсу

Передбачається, що цю систему використовуватимуть працівники підприємства: відділ закупівлі, відділ збуту, зав.складом, що займаються обліком товарів на складах, і оприбуткуванням товарів, що надійшли на склад.

Працівникам підприємства доводиться виконувати велику кількість дій щодо пошуку зведеної інформації по підприємству в цілому. Виконання цієї роботи вручну потребує значного часу.

Інформаційна система, що розробляється, дозволити значно зменшити час роботи співробітників та спростить процес отримання різної зведеної інформації.

Гість – роль призначена для звичайних користувачів, які можуть лише переглядати список доступних товарів, мають можливість зробити замовлення через форму запити.

Працівник складу - роль призначена для адміністрування складом. Користувач може переглядати усю інформацію про товар та переглядати звітність по певним критеріям.

Працівник відділу закупівлі – роль призначена для адміністрування відділу закупівлі. Користувач може переглядати усю інформацію про товар та власноруч проводити замовлення товару на склад.

Працівник відділ збуту – роль призначена для адміністрування відділу збуту. Користувач може переглядати усю інформацію про товар та власноруч проводити продаж товару.

Режим обмеженого перегляду – роль призначена лише для перегляду товару на складі. Наприклад, це можуть співробітники складу, які хочуть перевірити наявність певного товару.

2.7 Загальні елементи сторінок

НУБІП України

Шапка:

НУБІП України

1) Верхнє горизонтальне меню. Містить посилання «Головна», «Загальні відомості», «Адміністративна панель», «Заняття», «Статті», «Контакти», «Реєстрація» або «Персональний кабінет», провідні на відповідні сторінки.

2) Логотип сайту. Містить посилання на головну сторінку сайту.

НУБІП України

3) Персональний кабінет

4) При першій сесії користувача система намагається визначити користувача за даними ІР адреси.

5) Вхід – при натисканні з'являється pop-up вікно авторизації.

НУБІП України

6) Реєстрація – при натисканні відкривається сторінка з можливістю зареєструватися.

Підвал:

НУБІП України

1) Логотип сайту з посиланням на головну сторінку

2) «Контакти», провідні на відповідні сторінки.

НУБІП України

3) Текстовий блок з адресою сайту та графіком роботи.

4) Блок, що містить телефон і Email. При зміні міста телефон в підвалі змінюється за аналогією з телефоном в шапці сайту.

5) Посилання на представництва компанії в соціальних мережах:

НУБІП України

«Facebook», «Twitter» і «Вконтакте».

6) Блок з копірайтом.

НУБІП України

2.8 Сторінки веб-сайту НУБІП України

«Головна сторінка»

Н

Головна сторінка містить загальну інформацію про проект (сайт) та кнопку для входу в обліковий запис користувача.

«Сторінка авторизації користувача»

Н

Сторінка повинна відчинятися як поп-ап вікно у якому пропонується ввести пошту та пароль користувача. Також, форма має містити кнопку “Забув пароль”, яка перенаправляє у вікно забув пароль.

«Сторінка відновлення паролю»

Н

Сторінка яка містить поле для вводу пошти користувача. При запиті на відновлення користувач отримує повідомлення на пошту у якому міститься посилання на зміну паролю. Для зміни паролю потрібно двічі ввести пароль який задовольняє потреби безпеки.

«Сторінка користувача»

Н

Сторінка містить основну інформацію про користувача: ім'я, прізвище, по батькові (якщо існує), пошту, телефон (якщо вказано) та фотографію. Сторінка містить кнопку для зміни даних, користувач може змінити пароль, пошту, телефон та фотографію профілю. Також на сторінці у вигляді списку відображено доступних товарів, при натисканні на предмет відкривається вікно предмету у якому можна переглянути статистику по товару.

«Сторінка реєстрації»

Н

Сторінка доступна лише користувачам з доступом до редагування даних. Користувач обирає кого саме потрібно зареєструвати користувача. Якщо обрано

Н студента, то на сторінці реєстрації користувач вказує ім'я, прізвище та по-батькові, вписує у поле назву посади яку попередньо додано до бази даних у вікні посад, поле має автоматично підказувати існуючі посади. Після чого автоматично створюється персональний акаунт з електронною поштою яке містить у собі ім'я та прізвище

Н користувача на трансліті та пароль - changepassword.

«Сторінка перегляду статистики»

Сторінка доступна усім користувачам. Користувач повинен обрати який

Н товар потрібно переглянути. Товар виводиться в одній таблиці в якій вказується кількість проданого товару за певним періодом, назва товару та кількість на складі.

«Сторінка редагування товару»

Сторінка доступна лише користувачам з доступом до редагування даних.

Н Користувач отримує можливість обрати який товар треба редагувати, після вибору товару користувач отримує можливість створення нового та редагування старих товарів. Користувач обирає назву, кількість на складі, номер складу та відповідального за товар.

НУБІП України

2.9 Розробка SQL запитів

НУБІП України

Laravel database query builder надає зручний і плавний інтерфейс для

створення і виконання запитів до бази даних. Він може використовуватися для

виконання більшості операцій з базами даних у вашому додатку працює на всіх підтримуваних системах баз даних.

НУБІП України

Будівельник запитів Laravel використовує прив'язку параметрів PDO для захисту вашої програми від атак SQL-ін'єкцій. Немає необхідності очищати рядки, що передаються в якості Прив'язок.

Можна використовувати табличний метод БД, щоб почати запит. Метод `table` повертає екземпляр `fluent query builder` для даної таблиці, що дозволяє пов'язати додаткові обмеження із запитом і потім, нарешті, отримати результати за допомогою методу `get`. Далі зображено запити на вибірку даних з БД.

Лістинг 2.5. Запити на вибірку даних

```
<?php
namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\URL;
use App\Schedule as Schedule;

class URLGenerationController extends Controller
{
    public function index()
    {
        date_default_timezone_set('Europe/Kiev');
        // date("H:i:s") // время по Киеву
        // date("N") // номер дня недели для поиска по дням
        // date("W")%2 // номер недели, для поиска верхней и
        // нижней недели

        $nowPair = DB::table('schedule_of_disciplines')
            ->whereTime('start_time', '<=', '09:00:00')
            ->whereTime('end_time', '>=', '09:00:00')
            ->select('number')
            ->get();

        /*$schedulesIdByCurrentTimeAndClassroom =
        DB::table('schedules')
            ->where('index_number', $nowPair[0]->number)
            ->where('day', 1)
            ->where('week', 1)
            ->join('classrooms', 'schedules.id_classroom', '=',
            'classrooms.id');
```

```

->where('classrooms.room_number', 1) // номер комнати
->where('classrooms.building_number', 1) // номер
корпуса
->select('schedules.id')
->get();*/

```

```

$schedules = DB::table('schedules')
->where('index_number', $nowPair[0]->number)
->where('day', 1)
->where('week', 1)
->join('classrooms', 'schedules.id_classroom', '=',
'classrooms.id')

```

```

->select('schedules.id', 'schedules.index_number', 'schedules.day', 'sche
dules.week', 'classrooms.building_number', 'classrooms.room_number')
->get();
foreach($schedules as $schedule){
    $oneSchedule = Schedule::where('id', $schedule->id)
->get()[0];

```

```

    $oneSchedule->link
    url("/getPairLink?building=$schedule->building_number&room=$schedule-
->room_number");
    $oneSchedule->save();
}
/* $pairs = DB::table('pairs')

```

```

->join('schedules', 'pairs.id_schedule', '=',
'schedules.id')
->join('classrooms', 'schedules.id_classroom', '=',
'classrooms.id')
->select('pairs.*', 'schedules.*', 'classrooms.*')
->get();
*/
return true;
}
}

```

Для заповнення БД використовується Laravel Seeding:

Лістинг 2.6 – Запити на заповнення БД

```

<?php
use Illuminate\Database\Seeder;
class GroupsSeeder extends Seeder
{

```

```

    /**
     * Run the database seeds.
     * @return void
     */
    public function run()
    {

```

```

        DB::table('faculties')->insert([
            'name' => 'Інформаційних технологій'
        ]);

```

```

        DB::table('departments')->insert([
            'name' => 'Комп'ютерних мереж і систем',
        ]);

```

```

        DB::table('groups')->insert(array(
            0 =>
                array(
                    'name' => "KI180080ck",
                    'id_faculty' => 1,
                    'id_department' => 1
                ),

```

```

            1 =>
                array(
                    'name' => "KI170080ck",
                    'id_faculty' => 1,
                    'id_department' => 1
                ),
        ));
    }
}

```

НУБІП України

НУБІП України

НУБІП України

НУБІП України

3 ПРОЕКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ

3.1 Моделювання поведінки системи

За допомогою діаграм можна візуалізувати систему з різних точок зору. Одна з діаграм, наприклад, може описувати взаємодію користувача з системою, інша - зміна станів системи у процесі її роботи, третя - взаємодія між собою елементів системи тощо.

Складну систему можна і потрібно представити у вигляді набору невеликих і майже незалежних моделей-діаграм, причому жодна з них не є достатньою для опису системи та отримання повного уявлення про неї, оскільки кожна з них фокусується на певному аспекті функціонування системи та висловлює різний рівень абстракції. Іншими словами, кожна модель відповідає певній, приватній точці зору на проєктовану систему.

Будь-які (у тому числі і програмні) системи проєктуються з урахуванням того, що в процесі своєї роботи вони будуть використовуватися людьми та взаємодіяти з іншими системами. Сутності, з якими взаємодіє система у процесі своєї роботи, називаються екторами, причому кожен ектор описує, що система неводитиметься строго певним, передбачуваним чином.

Прецеденти позначаються дуже простим чином - у вигляді еліпса, всередині якого вказано його назву. Прецеденти та ектори з'єднуються за допомогою ліній.

Часто на одному з кінців лінії зображують стрілку, причому спрямована вона до того, у кого запитують сервіс, тобто, чиїми послугами користуються. Це просте пояснення ілюструє розуміння прецедентів як сервісів, яке пропагує компанія IBM.

Розглянемо поведінку автоматизованої системи контролю доступу до приміщення як набір функцій та акторів, що з ними взаємодіють. Модель поведінки системи подано в нотації UML, діаграмою прецедентів, ще описує доступні дії і випадки (рис. 3.1).

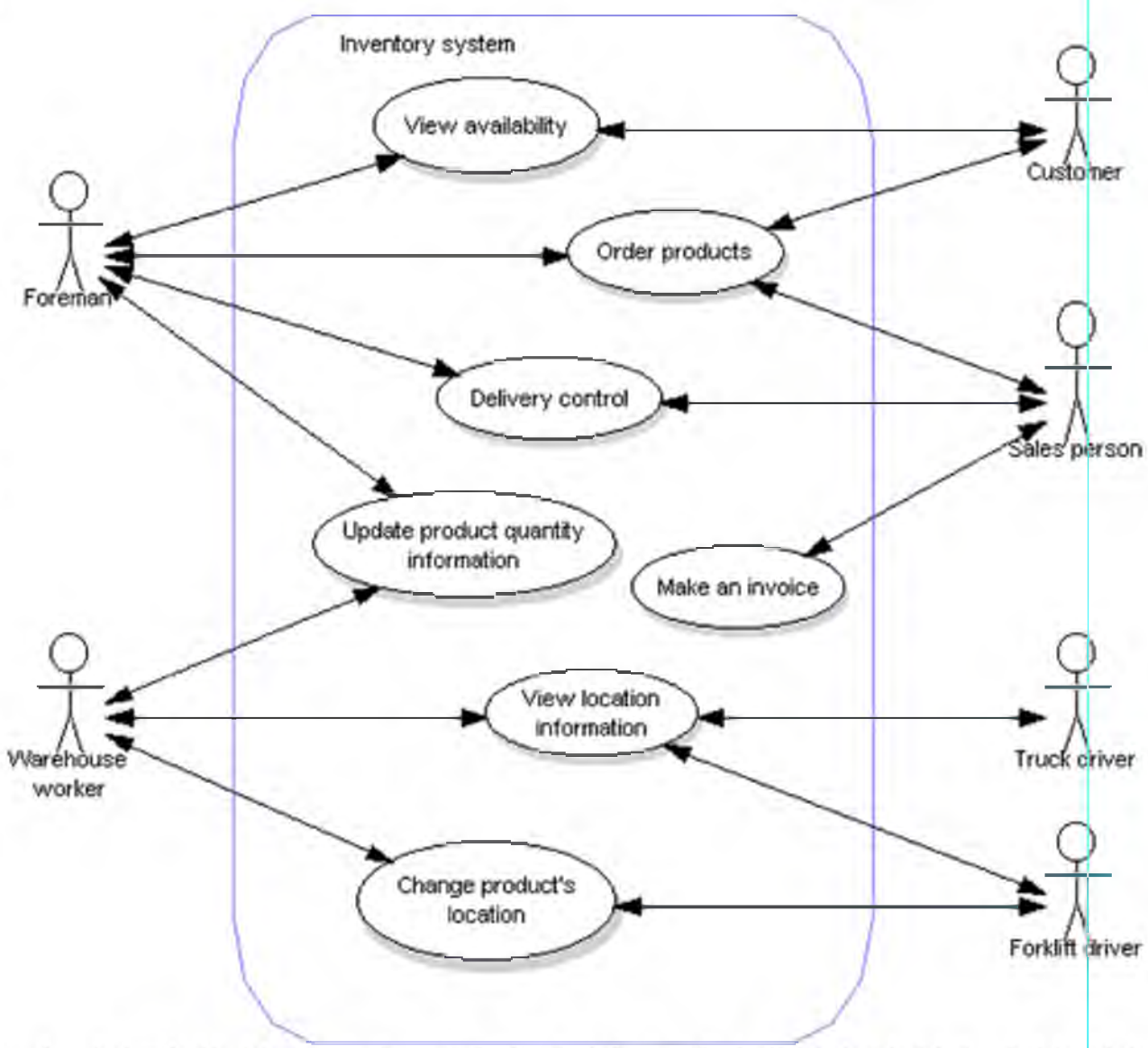


Рисунок 3.1 – Діаграма USE Case

Класи використовуються в процесі аналізу предметної області для складання словника предметної області системи, що розробляється. Це може бути як абстрактні поняття предметної області, і класи, куди спирається розробка і які описують програмні чи апаратні сутності.

Діаграма класів — це набір статичних декларативних елементів моделі. Діаграми класів можуть застосовуватися і при прямому проектуванні, тобто в процесі розробки нової системи, і при зворотньому проектуванні - описі існуючих

систем. Інформація з діаграми класів безпосередньо відображається у вихідний код програми - у більшості існуючих інструментів UML-моделювання можлива кодогенерація певної мови програмування. Таким чином, діаграма класів - кінцевий результат проектування та відправна точка процесу розробки.

За замовчуванням існує одна діаграма класів, звана Головною (Main), де показують пакети класів моделі, представлені відповідно до рис 3.2.

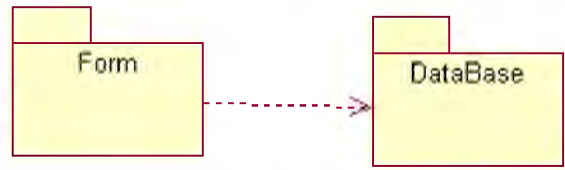


Рисунок 3.2 – Головна діаграма Класів системи обліку товарів на складі магазину

Всередині кожного пакета також є «головна діаграма», що включає всі класи цього пакета, представлені відповідно до рис. 3.3 та 3.4.

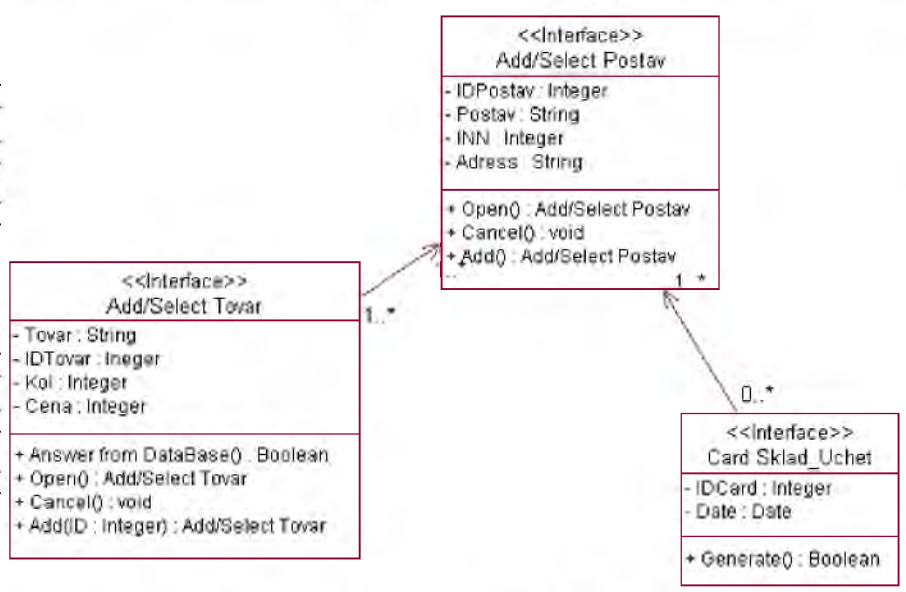


Рисунок 3.3 – Діаграма класів "form"

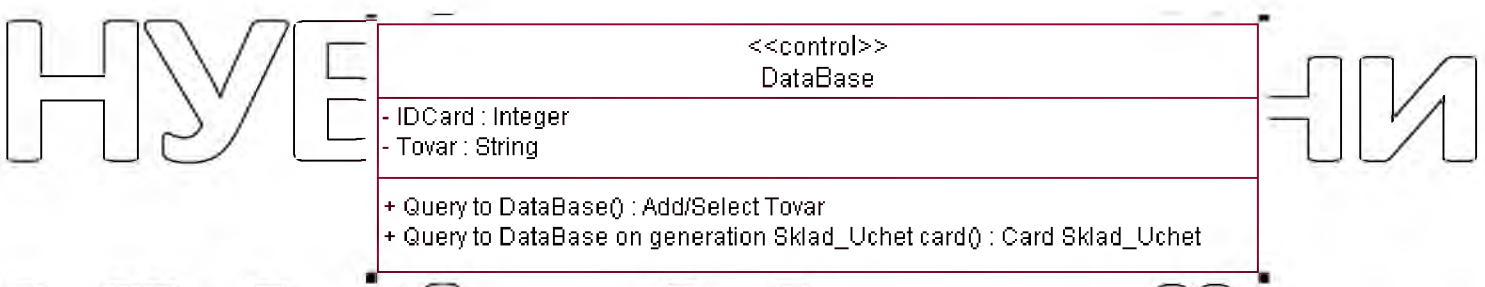


Рисунок 3.4 – Діаграма класів DataBase

Після створення розділом діаграми класів створюється діаграма класів для сценарію «get_tovar» (прийняти товар), де відображаються всі класи, атрибути і зв'язок між ними, представлена відповідно до рис. 3.5

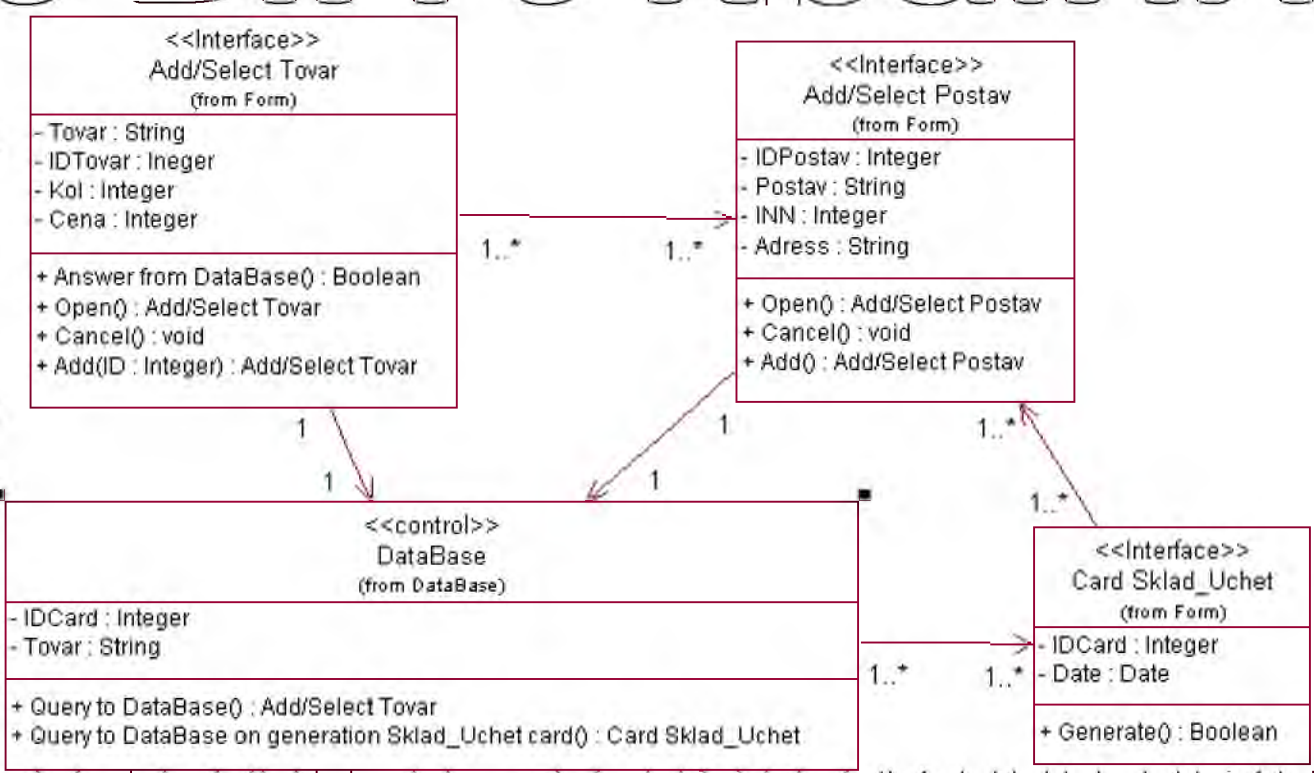


Рисунок 3.5 – Діаграма класів "get_tovar" (прийняти товар)

Діаграма послідовності відображає потік подій, що відбуваються у рамках варіанта використання. У цьому розділі буде розроблено діаграму послідовності, яка представлена відповідно до рис. 3.6.

НУБІП України

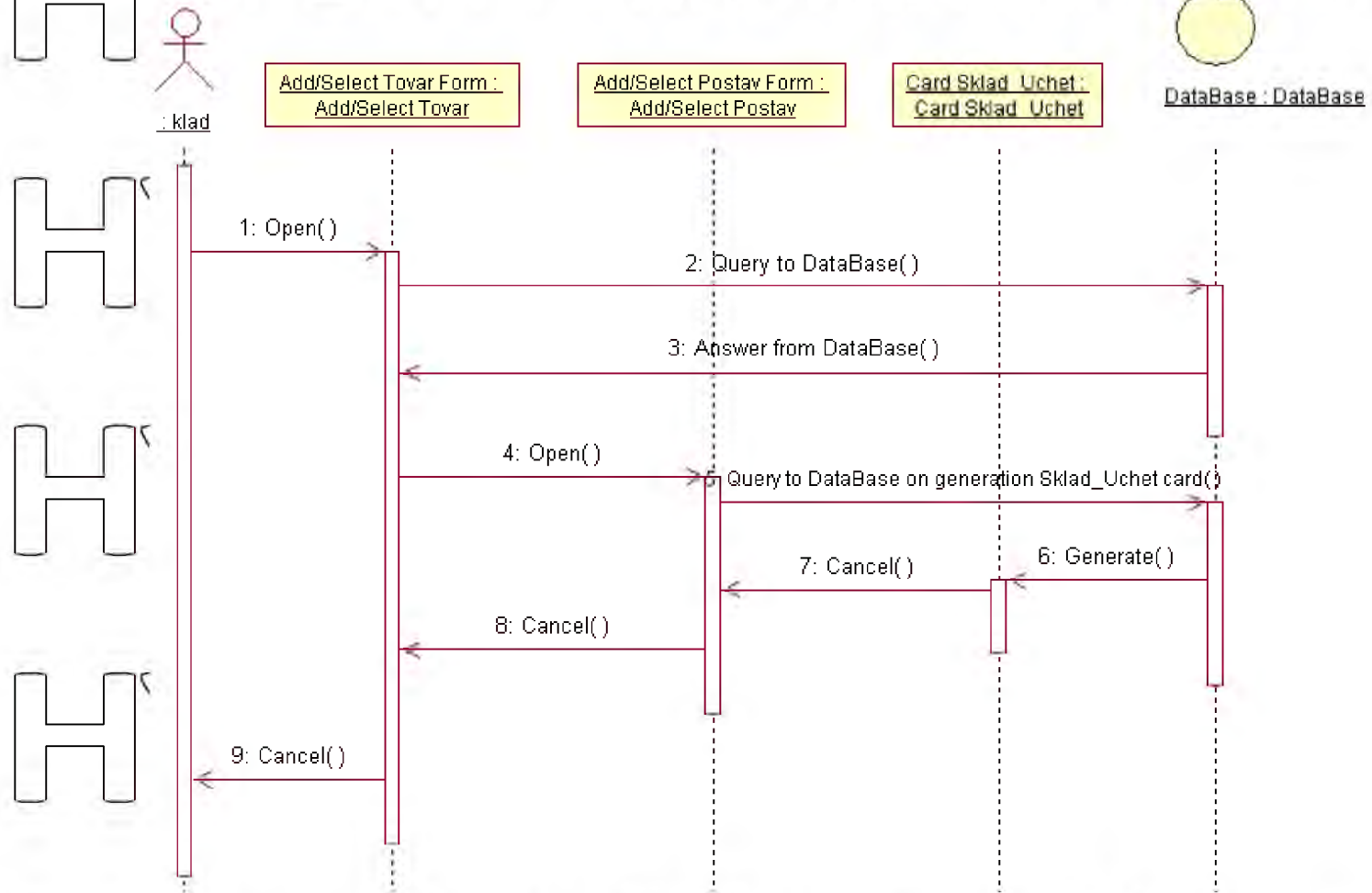


Рисунок 3.6 – Діаграма послідовності

НУБІП України

На цю діаграму розміщені такі об'єкти:

- «klad» (комірник) – дійова особа;
- "Add/Select Tovar Form" – містить форму введення або вибору товару;
- "Add/Select Postav Form" – містить форму введення або вибору постачальника товару;
- «Card Sklad_Uchet» – форма картки складського обліку, яка створюється після введення всіх даних та є підсумковим документом;
- «DataBase» – містить інформацію про постачальників та товари, на підставі інформації цього об'єкта формується карта складського обліку

НУБІП України

Повідомлення між об'єктами на діаграмі:

- Open – відкрити форму;
- "Cancel" – скасування дії;
- Query to DataBase – запит до бази даних на вибір товару;
- "Answer from DataBase" – найменування товару;
- "Query to DataBase on generation Sklad_Uchet card" – запит до бази даних на вибір постачальника та генерацію карти складського обліку;
- "Generate" – карта складського обліку.

Після створення об'єктів та повідомлень між ними було виконано співвідношення об'єктів з класами, а повідомлень з операціями. Усі назви об'єктів та повідомлень збігаються з назвами класів та операцій відповідно.

Діаграму діяльності наведено у рис. 3.7.

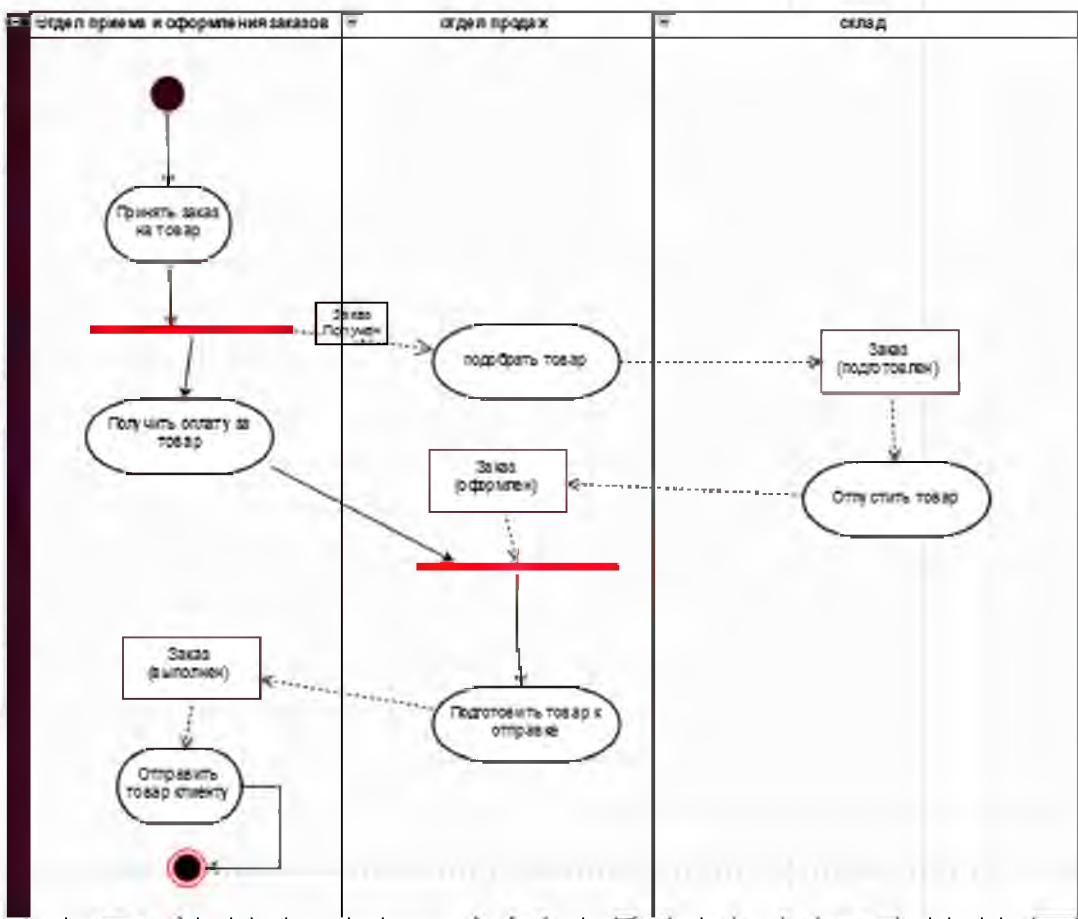


Рисунок 3.7 – Діаграма діяльності

3.2 Прототипування графічного інтерфейсу

Структура інтерфейсу системи прогнозування та замовлення товарів на склад була зазначена в технічному завданні на розробку. При розробці системи ніяких змін не передбачається.

Власне, коли попередня робота проведена і в голові є чітка картина того, що потрібно зробити, можна опрацьовувати прототип.

Перед тим як писати на React код потрібно проробити деталізовані прототипи, які можна обговорювати з замовником і тестувати.

Коли всі екрани опрацьовані, обговорені і затверджені, їх об'єднують в карту навігації, яка детально відображає логіку роботи продукту.

Шаблон веб-дизайну, також відомий як шаблон дизайну користувальницького інтерфейсу, - це набір вказівок для проектування аспекту або компонента інтерфейсу користувача. Шаплони веб-дизайну розробляються для конкретних завдань користувача і можуть бути прийняті та реалізовані на будь-якому веб-сайті.

Шаблон веб-дизайну - це не стандартизований шаблон або точно визначений дизайн. Скоріше, це набір найкращих практик, які вирішують проблему, орієнтовану на користувача.

При розробці дизайну для зручності користувача простір завжди обмежений. Можна розмістити лише стільки елементів сторінки в області перегляду (тобто у видимій області веб-сторінки), перш ніж утвориться візуальний безлад. Це є проблемою для великих веб-сайтів з великою кількістю сторінок.

Розробник має контроль над конкретними параметрами, функціоналом, естетикою і навіть мовою, на якій воно запрограмовано. Але шаблон дизайну спадних меню надає рекомендації щодо зручності їх реалізації.

Зрештою, є дві ключові причини чому веб-дизайнери дотримуються цих шаблонів:

По-перше, шаблони дизайну забезпечують хороший UX. Шаблон веб-дизайну інструктує дизайнера, як розташувати та впорядкувати елементи сторінки,

щоб найкращим чином задовольнити потреби відвідувачів і створити критичну довіру.

По-друге, шаблони дизайну спрощують процес проектування. Шаблони вирішують поширені проблеми UX, які зустрічаються на веб-сайтах, тому дизайнерам не потрібно знову і знову винаходити рішення однієї і тієї ж проблеми.

Одним із рішень є вертикальне спадне меню. Цей шаблон проектування визначає список параметрів, які з'являються після деякого активатора, як правило, події наведення курсора або клацання миші. Користувачі можуть вибрати будь-яку опцію з цього нового меню. Меню закривається, коли користувач наводить курсор на нього.

3.3 Розгортання та налаштування системи

React виходить з принципу, що логіка рендеринга нерозривно пов'язана з іншою логікою UI: з тим, як обробляються події, як стан змінюється в часі і як дані готуються до відображення.

Замість того, щоб штучно розділити технології, розміщуючи розмітку і логіку в різні файли, React розділяє відповідальність за допомогою слабо пов'язаних одиниць, званих «компоненти», які містять і розмітку, і логіку. Ми ще повернемося

до теми компонентів в наступному розділі, але якщо ідея тримати розмітку в JavaScript коді все ще викликає у вас дискомфорт, ця доповідь може переконати вас.

React можна використовувати і без JS, але більшість людей цінує його за наочність при роботі з UI, що живе в JavaScript-коді. Крім цього, JSX допомагає React робити повідомлення про помилки і попередження зрозуміліше.

Як ви організуєте компоненти в React?

Компоненти в React складаються з двох основних компонентів: state та props. Ці два компоненти означають, що є можливість організувати компоненти в ієрархічній структурі, яка забезпечує односторонній потік даних (через props).

Ця структура називається деревом, і вона дозволяє покласти відповідальність за стан на один компонент. Нижче компонент буде відповідати за масив і передавати інформацію через props своїм дочірнім компонентам тільки тоді, коли вони цього потребують. Нашадки компонента не можуть оновити дані, які він отримує від свого батька – якщо дані повинні бути оновлені, діти отримують інший реквізит від свого батька з функцією для його оновлення.

Дерево компонентів дозволяє нам створювати складні UI, які не плутають стану. Наприклад, додаток не буде плутати один стан зі станом іншого, який може надати суперечливу інформацію. Замість цього наш додаток слідує логічному шляху, щоб відобразити правильну інформацію, і буде оновлювати її відповідним чином (див рис. 3.8).

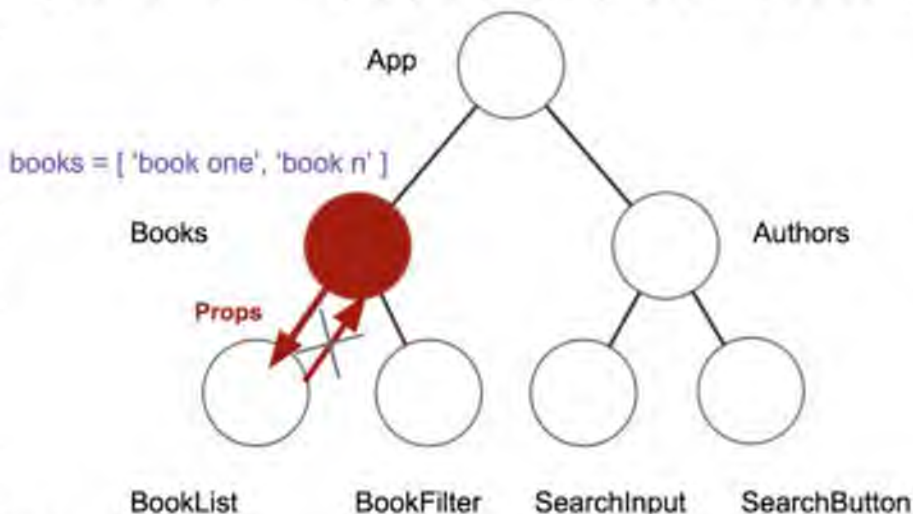


Рисунок 3.8 – Ієрархія класів у React

У Laravel є функція, яка дозволяє вам підлаштувати код для React компонентів за допомогою команди `preset react`. У попередніх версіях Laravel Налаштування React з Laravel було не таким простим. Якщо ви використовуєте останню версію Laravel, виконайте наступну команду, щоб додати в проект пресет

React [12].

Лістинг 3.1 Інсталяція пресету React в Laravel

```
php artisan preset react
```

Laravel за замовчуванням поставляється з попередньо встановленою Vue, і вищезгадана команда замінює всі екземпляри Vue на React. Примітно, що якщо вам не потрібен пресет, ви можете видалити його повністю, використовуючи команду `php artisan preset none`.

Laravel використовує Laravel Mix, який є прошарком для webpack. Webpack, є збирачем модулів. Він дозволяє всі залежності модуля і генерує необхідні статичні ресурси для JavaScript і CSS. React вимагає для роботи збирач, і webpack відмінно вписується в цю роль. Таким чином, Laravel Mix – це шар, який знаходиться поверх webpack і спрощує використання webpack в Laravel [13].

Гарне розуміння того, як працює Laravel Mix, важливо, якщо потрібно налаштувати конфігурацію webpack пізніше. Команда `React preset` не дає жодної інформації про те, як все працює під капотом. Отже, потрібно видалити передумовку React.

React додаток побудовано навколо компонентів. Компоненти є найважливішою структурою в React, і є каталог для компонентів.

Для програми, яку потрібно створити, треба починати з базового компонента, який відображатиме всі продукти, що повертаються сервером. Компонент повинен спочатку подбати про наступне:

- виийняти всі продукти з API (`GET /api/products`);

зберігання даних про продукт в його стейті;
 відображення даних продукту.

React не є повноцінним фреймворком, отже він не має ніяких Ajax функцій сам по собі. Буде використовуватися fetch(), який є стандартним JavaScript API для вилучення даних з сервера. Але є багато альтернатив для виконання Ajax запитів на сервер.

Лістинг 3.2 – Головна сторінка проекту

```
import React, { Component, useEffect, setState, useState } from
'react';
import ReactDOM from 'react-dom';
import List from '@material-ui/core/List';
import ListItem from '@material-ui/core/ListItem';
import ListItemIcon from '@material-ui/core/ListItemIcon';
import ListItemText from '@material-ui/core/ListItemText';
import Divider from '@material-ui/core/Divider';
import Grid from '@material-ui/core/Grid';

function Main() {
  const [pairs, setPairs] = useState([]);

  useEffect(() => {
    fetch('/api/pairs')
      .then(response => {
        return response.json();
      })
      .then(result => {
        setPairs(result);
      });
  }, []);

  function renderPairs() {
    return pairs.map(pair => {
      return (
        <ListItem button key={ pair.id } >
          <ListItemIcon >
            <ListItemText primary={ pair.number } />
          </ListItemIcon >
        </ListItem >
      );
    });
  }

  return (
    <div >
```

```

<Grid container>
  <Grid item sm={3}></Grid>
  <Grid item sm={6}>
    <List>
      { renderPairs() }
    </List>
  </Grid>
</Grid>
<Grid item sm={3}></Grid>
</Grid>
</div>
);
}

```

```

export default Main;
if (document.getElementById('pairs')) {
  ReactDOM.render(<Main />, document.getElementById('pairs'));
}

```

Тут було додано `createProduct` в `state` і ініціалізували його значенням `null`. Рядок `onClick={() => this.handleClick(product)}` викликає метод `handleClick()` при кліці на елемент списку. Метод `handleClick()` оновлює стейт `current Product`.

Тепер, щоб відобразити дані продукту, можна або відобразити його всередині компонента `Main`, або створити новий компонент. Як згадувалося раніше, поділ UI на більш дрібні компоненти – це `React way`.

Компонент `renderPairs` вкладено в компонент `Main`. Компонент `Main` передає свій стейт як `props`. Компонент `renderPairs` приймає цей `props` в якості вхідних даних і відображає відповідну інформацію.

3.4 Розробка графічного інтерфейсу

Для розробки візуальних компонентів було використано фреймворк Material-

НУБІП України

Material-UI – це набір компонентів React, який реалізує Google Material

Design. Ці компоненти працюють ізольовано, це означає, що вони є

самопідтримуючими і вводять тільки ті стилі, які вони повинні відображати [14].

Щоб почати роботу з матеріалом-ui, все, що потрібно, це запустити команду терміналу, наведену нижче.

Лістинг 3.3 – Встановлення фреймворка Material-UI

```
npm install @material-ui/core
```

Для початку роботи потрібно авторизуватися в системі, за допомогою вікна авторизації, що зображено на рис. 3.9.

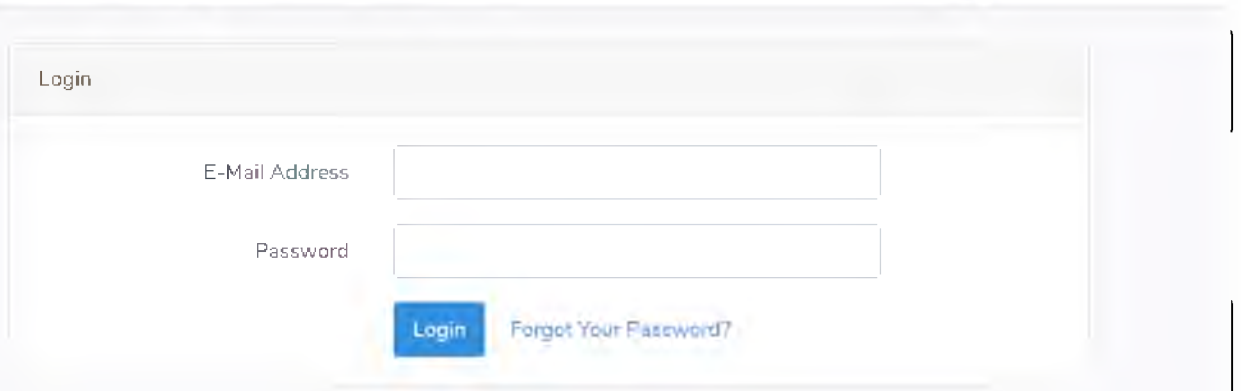


Рисунок 3.9 – Вікно входу в систему

При авторизації використовується валідація даних. Валідація форми – це коли користувач вводить дані, веб-додаток перевіряє, що дані коректні. Якщо Дані вірні,

додаток дозволяє даними бути відправленими на сервер і (як правило) бути збереженими в базі даних; якщо ні – видає повідомлення про помилку, що пояснює

НУБІП України

які виправлення необхідно внести. Перевірка форми може бути реалізована кількома різними способами [15].

Існує два різних типи перевірки форми:

1) Перевірка на стороні клієнта – це перевірка, яка відбувається в браузері, перш ніж дані будуть відправлені на сервер. Це зручніше, ніж перевірка на стороні сервера, так як дає миттєвий відповідь. Її можна далі поділити на:

– JavaScript перевірка виконується з використанням JavaScript. Повністю настраюється;

– вбудована перевірка форми, використовуючи функції перевірки форми HTML5. Для цього зазвичай не потрібно JavaScript. Вбудована перевірка форми має кращу продуктивність, але вона не така настраюється, як з використанням JavaScript.

2) Перевірка на стороні сервера – це перевірка, яка виникає на сервері після відправки даних. Серверний код використовується для перевірки даних перед їх збереженням в базі даних. Якщо дані не проходять перевірку валідності, відповідь відправляється назад клієнту, щоб повідомити користувачеві, які виправлення повинні бути зроблені. Перевірка на стороні сервера не така зручна, як перевірка на стороні клієнта, оскільки вона не видає помилок до тих пір, поки не буде відправлена вся форма. Тим не менш, перевірка на стороні сервера – це остання лінія захисту програми від неправильних або навіть шкідливих даних. Всі популярні серверні фреймворки мають функції для перевірки та очищення даних (що робить їх безпечними).

В проекті було використано перевірку на стороні клієнта.

Для перегляду приходу товару на склад використовувати вікно перегляду приходу, що зображено на рис 3.10.

Num5 from 01-09-21_7_8_7p_8p_X_XS_XR_Xsmax_TL_11pro_11proMax		Date: Aug 31, 2021	Status: Completed			
			Qty Ordered	Qty Received	Price	Amount
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_White_] SKU: cw-7			5	5	\$1.89	\$9.45
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_Green_] SKU: cgn-7			5	5	\$1.89	\$9.45
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_Red_] SKU: cr-7			5	5	\$1.89	\$9.45
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_Pink_] SKU: cpk-7			5	5	\$1.89	\$9.45
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_Rose Gold_] SKU: crg-7			10	10	\$1.89	\$18.90
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_Silver_] SKU: cs-7			5	5	\$1.89	\$9.45
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_Grey_] SKU: cgr-7			5	5	\$1.89	\$9.45
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_Dark Blue_] SKU: cdb-7			10	10	\$1.89	\$18.90
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 7_Black_] SKU: cbl-7			15	15	\$1.89	\$28.35
iPhone XS Max XR Case Slide Armor Wallet Card Slots Holder Cover for Apple[For iPhone 8_White_] SKU: cw-8			5	5	\$1.89	\$9.45

Рисунок 3.10 – Вікно приходу товару на склад

Поля для введення даних використовують функцію Autosuggest. Цей віджет використовується для установки значення однорядкового текстового поля. Він корисний в одному з двох випадків:

Значення для текстового поля має бути вибрано з визначеного набору допустимих значень, наприклад, поле розташування повинно містити дійсне ім'я місця розташування поле зі списком.

Текстове поле може містити будь-яке довільне значення, але доцільно пропонувати користувачеві можливі значення. Наприклад, поле пошуку може запропонувати аналогічні або попередні пошуки, щоб заощадити час користувача: free solo [16].

Для роботи з віджетом потрібно мати покращену версію пакетів "react-select" і "downshift".

Лістинг 3.4 – Код приходу товару на склад

```

/* eslint-disable no-use-before-define */
import React from 'react';

```

```

import TextField from '@material-ui/core/TextField';
import Autocomplete from '@material-ui/lab/Autocomplete';
export default function ComboBox()
return (
  <Autocomplete
    id="combo-box-demo"
    options={data}
    getOptionLabel={({option}) => option.title}
    style={{ width: 300 }}
    renderInput={({params} => <TextField {...params}
label="Combo box" variant="outlined" />)}
  />
);
}

```

Для прогнозування товару на складі спроектовану спеціальне вікно, що зображено на рис 3.11.



Рисунок 3.11 – Вікно прогнозування товару

Для перегляду інформації щодо кожного товару, скільки продано, скільки є в наявності, скільки всього було замовлено, було написано скрипт, який завантажує дані з БД та відображає у вікні, що зображено на рис. 3.12.

Easy Lazy No Tie Elastic Silicone Shoe Laces Cool Guy Shoelaces Unisex(Adults,Colourful)

SKU: ssla-colourful Stock: 8

Total Qty Ordered At All	125	To Order Price	1.53
Total Qty Sold At All	137	To Order Qty	3
Units Sold Oct 14, 2021 - Nov 12, 2021	11	Total Price	\$ 4.59
Last Purchase Order Qty Ordered	50		
Last Purchase Order Qty Received	50		

Рисунок 3.12 – Вікно інформації щодо кожного товару

Є можливість виведення інформації щодо продажу товару за вказаний період та виведення лінії зростання чи падіння продажів (див. рис. 3.13.)

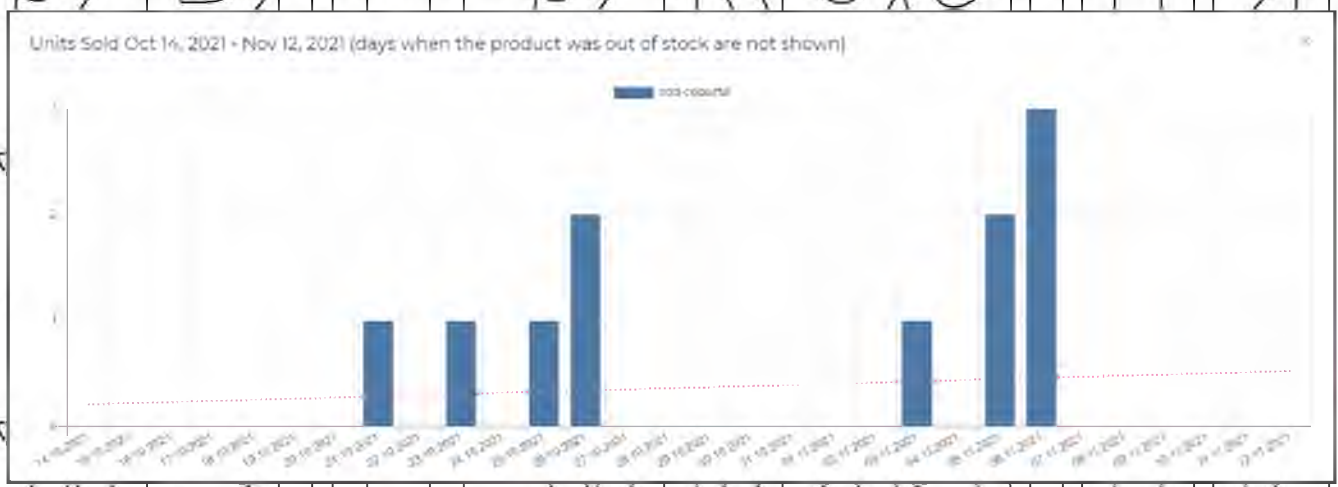


Рисунок 3.13 – Вікно інформації щодо продажу товару за вказаний період

Вікно з розрахунками для товару підтримує два основних метода розрахунку: середнє значення та метод регресії. Є можливість виведення діаграм (див. рис. 3.14 та 3.15).

Calculations

Forecasted value is 27, but you have 33 items on stock for now. So, you don't need to order more ones.

$27 / 30 = 0.75$

$0.75 * 30 = 22.5$

Рисунок 3.14 – Вікно виведення формули

Н
Н

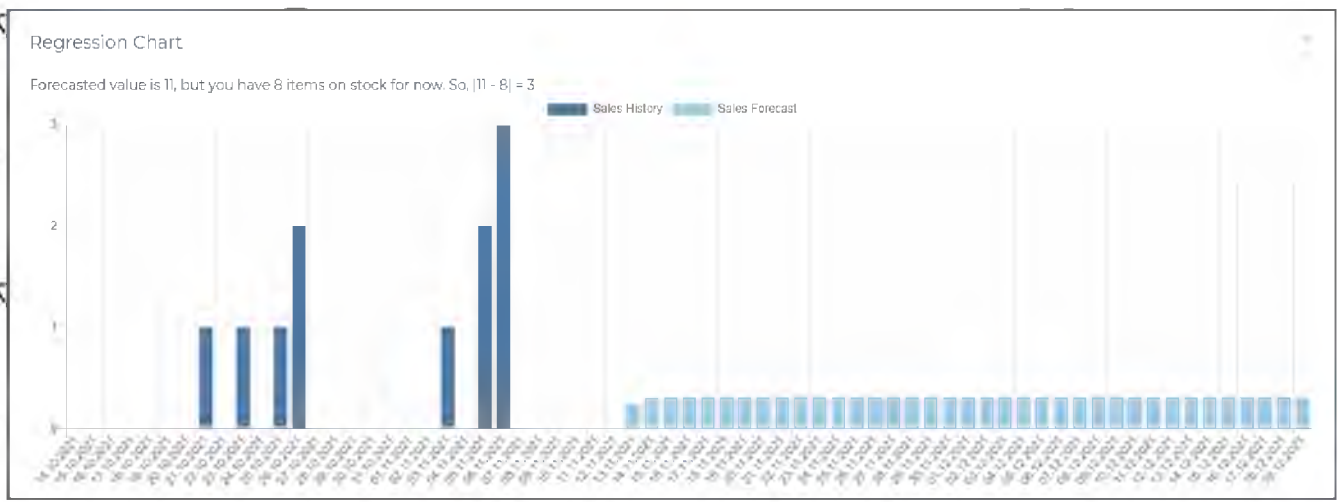


Рисунок 3.15 – Вікно виведення діаграм

Є можливість виведення кількості проданих одиниць між різними поставками товару, історію постачання товару, скільки було замовлено, скільки в результаті отримано, де на кожен рядок можна відкрити графік, подивитися продажі, постачання і т.д., кнопка калькулятора показує як керувався скрипт, але так само кількість можна ввести вручну, що зображено на рис. 3.16. Всі дані можна відразу вивести в таблицю ексель.

Н
Н

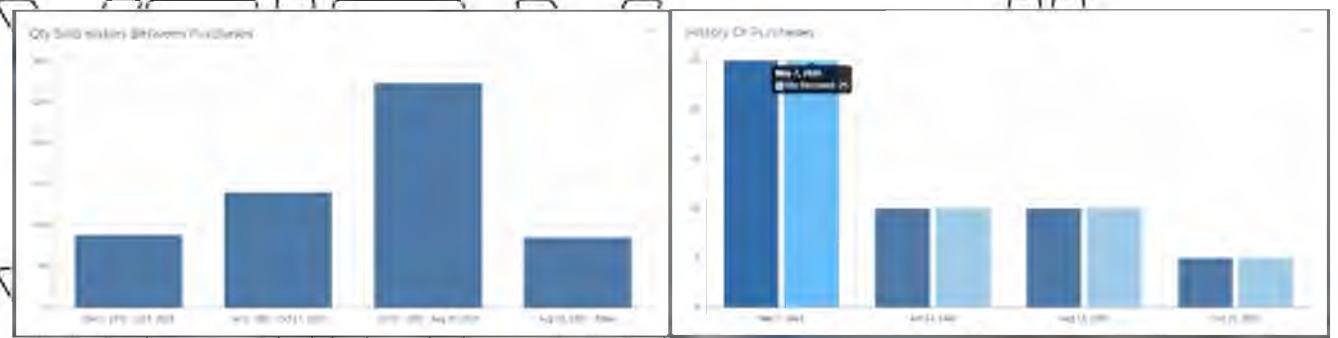


Рисунок 3.16 – Вікно виведення кількості проданих одиниць між різними поставками товару

Н
Н

3.5 ~~Перевірка системи на стресостійкість~~

DDoS-атака – це атака на сайт, основною метою якої є виведення його з ладу шляхом подачі великої кількості помилкових запитів. В результаті такої атаки сервера, які обслуговують сайт, змушені обробляти надмірний обсяг помилкових запитів, і сайт стає недоступним для простого користувача.

Основною метою для зловмисників, ddos застосовую для усунення конкуренції в тій чи іншій галузі, шляхом повного припинення роботи атакованого сервера за рахунок подачі на нього великої кількості помилкових запитів, з якими не буде справлятися ваш сайт.

Популярними жертвами таких атак стають комерційні та інформаційні сайти.

Хакери останнім часом використовують такий вид атак з метою вимагання, вимагаючи грошей за припинення атаки, або ведуть інформаційну війну.

Принаймні існує три різні методи організації ddos атак.

- по смузі пропускання – даний вид атаки передбачає що на веб сайт направляється велика кількість запитів по протоколах TCP, UDP і ICMP і таким чином повністю заповнюють його пропускну здатність. Викликаючи при цьому відмову в обслуговуванні.

- на основі протоколу сервера – даний вид атаки спрямований на конкретні сервіси сервера. І може виконуватися за допомогою TCP, UDP і ICMP. Часто такі атаки називають SYN-флуд, сенс яких у відсиланні на веб сервер великої кількості SYN запитів на які сервер повинен відповісти запитом ACK. Через велику повноту таких запитів, сервер часто не справляється з навантаженням і падає.

- на основі помилок конкретного веб сайту – цей вид атаки є найскладнішим в плані виконання і застосовується як правило високо-професійними хакерами. Суть його полягає в тому що на сайті-жертві знаходяться уразливості,

використовуючи як створюється високе навантаження на сервер і він отримує відмову в обслуговуванні.

Для тестування використовується сервіс HOIC.

HOIC був розроблений в ході операції Payback by Praetox тією ж командою що створила LOIC. Ключова відмінність в тому, що HOIC використовує HTTP протокол і з його допомогою посилає потік рандомізованих HTTP GET і POST запитів. Він здатний одночасно вести атаку на 256 доменів (див. рис. 3.17).

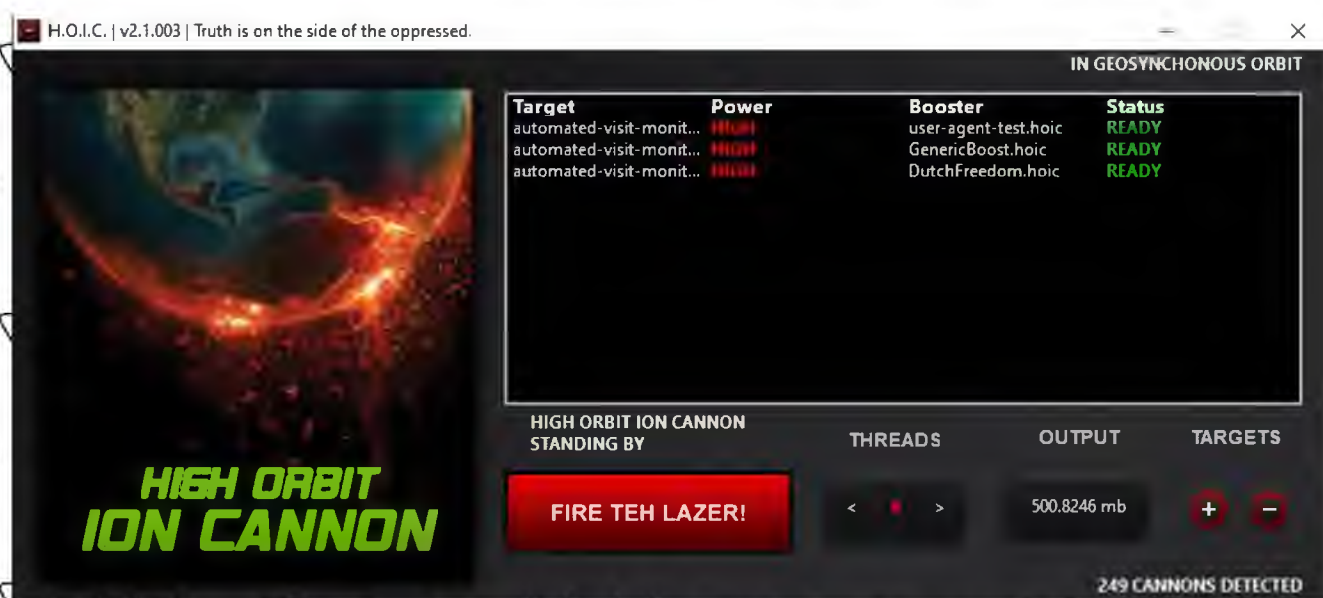


Рисунок 3.17 – Вікно HOIC

The Low Orbit Ion Cannon (LOIC) Низькоорбітальна іонна гармата. Можливо найпопулярніша DDoS програма. Вона може розсилати масові запити по протоколах ICMP, UDP тим самим забиваючи канал до сервера жертви. Найвідоміша атака за допомогою LOIC була здійснена групою Anonymous в 2009 році і спрямована проти PayPal, Visa, MasterCard в помсту за відключення WikiLeaks від системи збору пожертвувань (див. рис. 3.18).



Рисунок 3.18 – Вікно LOIC

На рисунках 3.19 та 3.20 зображено результати тестування системи.

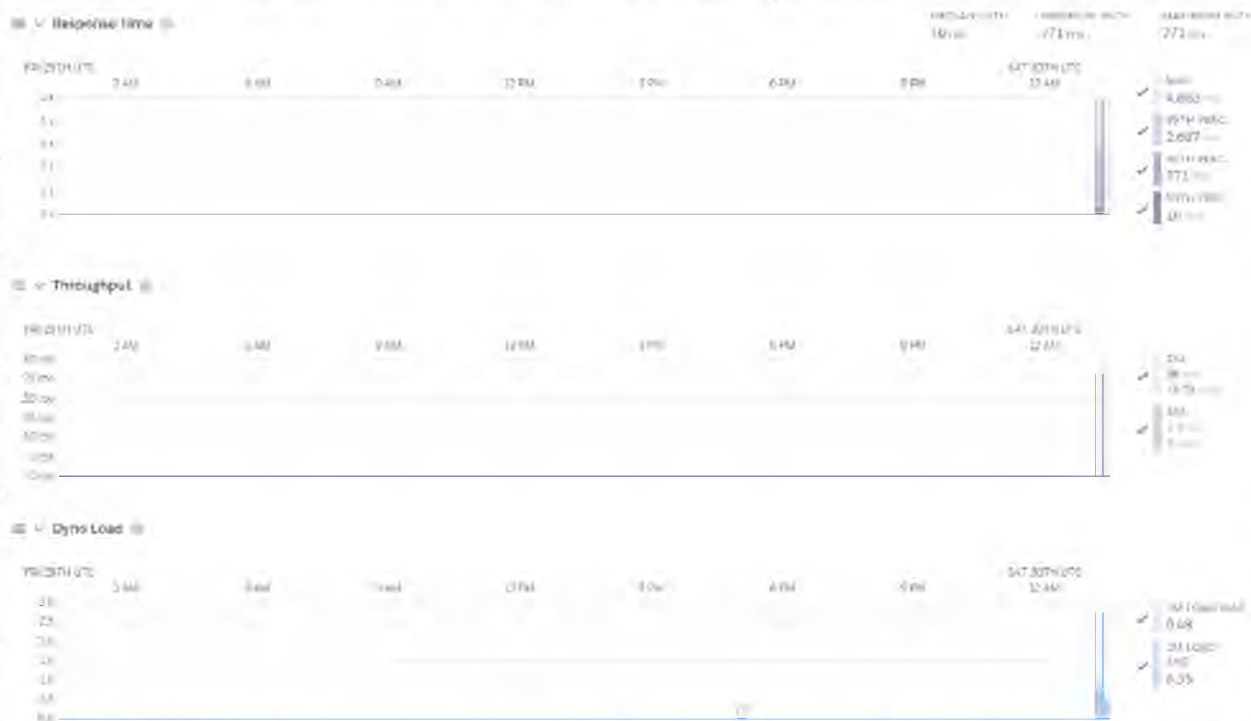


Рисунок 3.19 – Результати навантаження на систему



Рисунок 3.20 — Результати навантаження на ОЗУ та кількість користувачів

В результаті навантаження було виявлено, що система при 1000 підключених користувачів витримує з 15 відсотковим навантаженням на процесор та 10-ти відсотковими втратами ОЗУ. Тестування показало, що система спокійно витримує більше 15 тисяч одночасно підключених користувачів.

3.6 Розміщення додатку на хостингу

В розділі Суджс визначено, що додаток буде розміщено на SaaS сервісі Heroku. Для початку роботи з сервером потрібно інсталиювати інтерфейс командного рядка Heroku (CLI). Heroku CLI використовується для керування та масштабування додатків, надання надбудов, перегляду журналів додатків та локального запуску програми.

Після встановлення інтерфейс можливо використовувати команди Heroku через командну оболонку.

У Windows потрібно запустити командний рядок (cmd.exe) або Powershell для доступу до командної оболонки. Для входу в систему Heroku CLI потрібно використати команду Heroku login:

Лістинг 3.5 – Авторизація через Heroku CLI

```

$ heroku login
heroku: Press any key to open up the browser to login or q to exit
> Warning: If browser does not open, visit
> https://cli-auth.heroku.com/auth/browser/**
heroku: Waiting for login...
Logging in... done
Logged in as me@example.com

```

Ця команда відкриває веб-браузер на сторінці входу Heroku. Якщо браузер вже увійшов до акаунту Heroku, потрібно просто натиснути кнопку "Увійти", що відображається на сторінці.

Ця аутентифікація необхідна для правильної роботи команд heroku і git.

Перш ніж продовжити, потрібно переконатися, що необхідні компоненти встановлені правильно. Потрібно ввести кожну команду і переконатися, що вона відображає встановлену версію. Якщо версія не відображається, треба встановити необхідні компоненти.

Лістинг 3.6 – Перевірка встановлення компонентів

```

$ php -v
PHP 7.0.5 (cli) (built: Apr 26 2016 04:39:48) ( NTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies

```

```

$ composer -V
Composer version 1.4.1 2017-03-10 09:29:45

```

```

$ git --version
git version 2.12.2

```

Наступним кроком потрібно розгорнути додаток в Heroku. Для цього потрібно створити додаток на Heroku, який генерується в Heroku, щоб отримати вихідний код:

Лістинг 3.7 Створення додатку Heroku

```
$ heroku create
Creating sharp-rain-871... done, stack is heroku-18
http://sharp-rain-871.herokuapp.com/
https://git.heroku.com/sharp-rain-871.git
Git remote heroku added
```

Потрібно створити Procfile, текстовий файл у кореневому каталозі програми, щоб чітко оголосити, яка команда повинна бути виконана для запуску вашої програми. Профіль у прикладі розгорнутого додатка виглядає наступним чином:

Лістинг 3.8 Procfile

```
web: vendor/bin/heroku-php-apache2 web/
```

Це оголошує один тип процесу, web, і команду, необхідну для його запуску.

Тут дуже важлива назва web. Він оголошує, що цей тип процесу буде приєднаний до стеку маршрутизації http Heroku і отримувати веб-трафік при розгортанні.

Профілі можуть містити додаткові типи процесів. Наприклад, можна оголосити його для фонового робочого процесу, який обробляє елементи поза чергою.

Коли Додаток створюється, Git remote (називається heroku) також створюється та зв'язується з вашим локальним сховищем git.

Heroku генерує випадкове ім'я для програми, або можете передати параметр, щоб вказати своє власне ім'я програми. Після цього, потрібно розгорнути код на сервері:

Лістинг 3.9 – Розгортка на сервері

```
git push heroku master
Enumerating objects: 241, done.
Counting objects: 100% (209/209), done.
Delta compression using up to 16 threads
Compressing objects: 100% (159/159), done.
Writing objects: 100% (167/167), 116.12 KiB | 2.04 MiB/s, done.
```

```

Total 167 (delta 119), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> PHP app detected
remote: -----> Bootstrapping...
remote: -----> Installing platform packages...
remote: https://automated-visit-sys.herokuapp.com/ deployed
to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/automated-sys.git
   ef3eb20..dc7a7fa master -> master
$ heroku ps:scale web=1

```

Тепер з'явилася можливість відкрити додаток за URL-адресою, створеною за допомогою Heroku. В якості зручності можливо відкрити веб-сайт допомогою команди heroku open. На рис. 3.22 зображено домене ім'я та вікно входу.

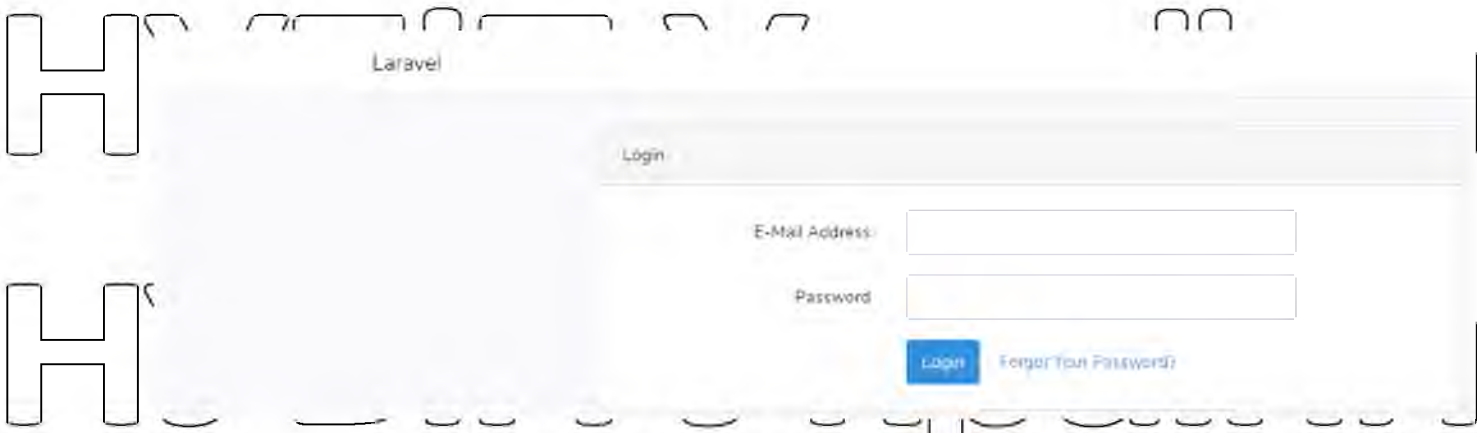


Рисунок 3.22 – URL адреса додатку

ВИСНОВКИ

НУБІП України

У дипломній роботі на тему дослідження веб-додатку для прогнозування та замовлення товарів на склад було розглянуто основні завдання, а також заходи щодо їх вирішення для своєчасного та швидкого виконання складського логістичного процесу.

НУБІП України

Правильне та ефективне впровадження автоматизованих систем дуже корисне для підприємства. У разі використання автоматизованої системи о значне підвищення продуктивності. Вплив «людського фактора» зведено до мінімуму, оскільки система вирішує всі питання з організацією звітності та документообігу.

НУБІП України

Система повністю контролює параметри сировини, що передається у виробництво:

НУБІП України

номенклатуру, обсяг, партію і якість. Система також забезпечує кадрову безпеку підприємства, тому що інформація про закупівлю та розташування сировини на складах централізовано зберігається і доступна для аналізу авторизованому користувачеві. За рахунок прозорості та оперативності обліку значно зростає

НУБІП України

керованість підприємством, аналітичні звіти показують відхилення вартості сировини, втрати маси, терміни придатності партій, обсяги споживання сировини виробництвом у режимі online, запаси сировини днями. Система реалізує всі необхідні параметри автоматизованого обліку товару. За допомогою програми

НУБІП України

формується звітність, реалізується ведення бази даних постачальників та конкурентів.

Одним з найбільш важливих параметрів, контрольованих системою є контроль якості товару, що передається, та аналіз термінів придатності, що у свою чергу дозволить продавати тільки нову продукцію .

НУБІП України

У першій частині магістерської роботи представлено техніко-економічне обґрунтування, яке дозволяє оцінити поточний стан логістичного процесу на підприємстві та виявлено проблеми роботи складу.

В основній частині випускної кваліфікаційної роботи було проведено ABC - аналіз складу, на основі якого було обрано номенклатурну позицію для розрахунку та вибору системи управління запасами. Було спроектовано додаток для прогнозування та замовлення товарів на склад, який забезпечить кращий контроль

за запасами та дозволить контролювати рівень обслуговування клієнтів.

Було розроблено структуру бази даних, необхідну зберігання даних системи. Також була обрана операційна система, в якій повинна буде функціонувати програма, що розробляється, і обрана мова програмування і система програмування, за допомогою якої буде проводитися розробка.

Результатом проектування став розроблений додаток, що автоматизує процес складського обліку.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України