

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента *Толстого Льва Олеговича*
(ПІБ)

академічної групи *122-18-3*
(шифр)

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*
(назва освітньої програми)

на тему: *Розробка веб-орієнтованого додатку з продажу
спортивного харчування на базі фреймворку Laravel*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>доц. Спірінцев В.В.</i>			
розділів:				
спеціальний	<i>доц. Спірінцев В.В.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2022

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

« » 2022 року

ЗАВДАННЯ

на кваліфікаційну роботу
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-18-3
(група)

Толстого Льва Олеговича
(прізвище та ініціали)

тема кваліфікаційної роботи

Розробка веб-орієнтованого додатку з
продажу спортивного харчування на базі фреймворку Laravel

затверджена наказом ректора НТУ «ДП» від 18.05.2022 р.

№ 268-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	На основі матеріалів проектно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми	13.05.2022 р.
Економічний	Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки	27.05.2022 р.

Завдання видав

(підпис)

доц. Спірінцев В.В.

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Толстой Л.О.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

РЕФЕРАТ

Пояснювальна записка: 92 с., 29 рисунків, 10 таблиць, 18 джерел.

Об'єкт розробки: веб-орієнтована інформаційна система для автоматизації діяльності підприємства в сфері електронної комерції.

Мета кваліфікаційної роботи: розробка програмного забезпечення веб-орієнтованої інформаційної системи для автоматизації діяльності компанії в сфері електронної комерції.

У вступі розглянуто аналіз та поточний стан проблеми, уточнено мету кваліфікаційної роботи та область її застосування, наведено обґрунтування актуальності теми та сформульовано постановку завдання.

У першому розділі було проаналізовано предметну галузь, визначено актуальність завдання та мету розробки, сформульована постановка завдання, визначені вимоги до впровадження програмного забезпечення, технологій і програмних засобів.

У другому розділі проаналізовані існуючі рішення, обрано платформу для розробки, виконано проектування та розробку веб-орієнтованої інформаційної системи, описана робота системи, алгоритм і структура його функціонування, а також виклик та завантаження додатку, визначено вхідні і вихідні дані, структурні характеристики параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, створено підрахунок собівартості роботи по розробленню додатку та розраховано час для його створення.

Практичне значення полягає у створенні веб-орієнтованої інформаційної системи, що забезпечує: об'єднання елементів прямого маркетингу та традиційної торгівлі; надання універсального інструменту управління візуальним контентом та адміністративної частини проекту; забезпечення відвідувачам сайту максимальної зручності роботи із системою, простоти і комфортності доступу до каталогу товарів за рахунок оптимально візуалізованих параметрів його вмісту; підвищення ефективності роботи компанії в цілому, за рахунок автоматизації управління бізнес-процесами.

Актуальність розробки інформаційної системи для автоматизації діяльності компанії в сфері електронної комерції не викликає сумніву та визначається великим попитом на подібні системи, оскільки електронна комерція робить торгівлю більш гнучкою, зрозумілою, відкритою й стандартизованою, а також надає значні переваги (що особливо відчутні в умовах COVID-19): мобільність, безконтактність, ціна, масштабованість, оперативність реагування на зміни ринку товарів і послуг, швидкість і ефективність здійснення комерційних операцій і т.ін.

Список ключових слів: ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРНЕТМАГАЗИН, БАЗА ДАНИХ, PHP, HTML, CSS, MYSQL.

ABSTRACT

Explanatory note: 92 pages, 29 figures, 10 tables, 18 sources.

Object of development: web-oriented information system for automation of enterprise activities in the field of e-commerce.

The purpose of the diploma project: to develop software for a web-based information system to automate the company's activities in the field of e-commerce.

The introduction considers the analysis and the current state of the problem, clarifies the purpose of the qualification work and its scope, provides a justification for the relevance of the topic and formulates the task.

In the first section the subject branch was analyzed, the urgency of the task and the purpose of development were determined, the task statement was formulated, the requirements for the implementation of software, technologies and software was specified.

The second section analyzes the existing solutions, selected a platform for development, designed and developed a web-based information system, describes the system, algorithm and structure of its operation, as well as calling and loading the application, defines input and output data, structural characteristics of technical parameters.

In the economic section, the complexity of the developed information system was determined, the calculation of the cost of work on the development of the application was created and the time for its creation was calculated.

Of practical importance is the creation of a web-based information system that provides: combining elements of direct marketing and traditional trade; providing a universal tool for visual content management and the administrative part of the project; providing site visitors with maximum convenience of working with the system, simplicity and comfort of access to the catalog of goods due to the optimal parameters of visualization of its content; increase the efficiency of the company as a whole, by automating business process management.

The relevance of developing an information system to automate the company's activities in the field of e-commerce is unquestionable and is determined by the high demand for such systems, as e-commerce makes trade more flexible, clear, open and standardized, and provides significant benefits (especially in COVID- 19): mobility, contactlessness, price, scalability, efficiency of response to changes in the market of goods and services, speed and efficiency of commercial transactions, etc.

List of keywords: INFORMATION SYSTEM, ONLINE STORE, DB, PHP, HTML, CSS, MYSQL.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	10
1.1. Загальні відомості з предметної галузі.....	10
1.1.1. Опис предметної галузі.....	10
1.1.2. Огляд та аналіз існуючих аналогів.....	12
1.1.3. Глосарій основних термінів.....	16
1.2. Призначення розробки та галузь застосування.....	17
1.3. Підстава для розробки.....	18
1.4. Постановка завдання.....	18
1.5. Вимоги до програми або програмного виробу.....	19
1.5.1. Вимоги до функціональних характеристик.....	19
1.5.2. Вимоги до інформаційної безпеки.....	20
1.5.3. Вимоги до складу та параметрів технічних засобів.....	22
1.5.4. Вимоги до інформаційної та програмної сумісності.....	23
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	24
2.1. Функціональне призначення системи.....	24
2.2. Опис застосованих математичних методів.....	24
2.3. Опис використаних технологій та мов програмування.....	24
2.4. Опис структури системи та алгоритмів її функціонування.....	38
2.4.1. Опис архітектури системи.....	42
2.5. Обґрунтування та організація вхідних та вихідних даних програми....	43
2.6. Опис розробленої системи.....	44
2.6.1. Використані технічні засоби.....	44

2.6.2. Використані програмні засоби.....	44
2.6.3. Виклик та завантаження програми.....	45
2.6.4. Опис інтерфейсу користувача.....	47
2.6.4.1. Опис клієнтської частини.....	49
2.6.4.2. Опис адміністрування системою.....	54
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	57
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту.....	57
3.2. Рахунок витрат на створення програми.....	61
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
Додаток А. Код програми.....	67
Додаток Б. Відгук керівника економічного розділу.....	91
Додаток В. Перелік файлів на диску.....	92

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПК – персональний комп'ютер.

HTML – hyper text markup language.

CSS – cascading style sheets.

PHP – hypertext preprocessor.

MySQL – structured query language.

ВСТУП

Автоматизація будь-якого виду надання послуг – завдання на сьогоднішній день відоме та поширене. У сучасному світі ні один бізнес не може обійтися без інтеграції в свою роботу інформаційної системи в якомусь її представленні, оскільки це істотно підвищує як зручність і якість взаємодії всередині бізнесу, так і значно підвищує охоплення клієнтської аудиторії та можливості взаємодії з нею, реклами.

Можна говорити про велику кількість успішно впроваджених проектів у різних галузях, у різних за масштабом компаніях, застосуванням різних засобів автоматизації. Коли ми говоримо про автоматизацію системи компанії з продажу спортивного харчування, ми маємо на увазі використання інтернет-додатку для взаємодії з клієнтською базою. Фактично поняття веб-сайту стало невід'ємною частиною будь-якого бізнесу у сфері надання послуг продажу. Результати аналізу існуючих даних свідчать, що нині більшість як великих мережевих магазинів, або невеликих локальних тією чи іншою мірою переглянули свій підхід до структури та процесів крування своїм бізнесом у бік автоматизації.

Це обумовило мету дослідження, яка полягає в удосконаленні процесу ведення бізнесу продажу спортивного харчування шляхом розробки та впровадження Web-додатку для взаємодії з існуючою та залучення нової клієнтської бази. Для досягнення поставленої мети було поставлено такі завдання:

- здійснити аналіз предметної області для вибору найкращих інструментів для вирішення поставленого завдання;
- вивчення необхідних матеріалів для роботи з мовами програмування;
- вивчення необхідних матеріалів для роботи з базою даних;
- розробити блок схеми принципу роботи інформаційної системи;
- побудова архітектури веб-додатку;
- розробка дизайну веб-додатку;
- створення інтерфейсу користувача веб-додатку;

- створення бекенд частини програми для взаємодії з базою даних;
- тестування системи;
- аналіз результатів роботи.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

Об'єкт управління є веб-сайтом для автоматизації роботи компанії з продажу спортивного харчування з передбачуваною назвою «SportCompanу».

Веб-сайт дає можливість клієнтам компанії ознайомитися з товарами, що продаються компанією, подати заявку на покупку товару, залишити відгук про роботу компанії, можливістю отримати інформацію про кожен товар та ціну на нього, а також залишити заявку на придбання франшизи компанії.

На сьогоднішній день існує велика конкуренція між компаніями з продажу спортивного харчування, тому потрібно мати власну стратегію, щоб забезпечити тривале та ефективне існування свого бізнесу на ринку послуг. Для того, щоб користувачеві отримати необхідну інформацію, її потрібно шукати в різних джерелах інтернету, або їхати в представництво компанії і отримувати її там. Для вирішення цієї проблеми найбільш ефективним способом буде створення веб-сайту, з докладною інформацією про компанію та всіма функціями, описаними вище.

Дана система дозволить користувачеві отримати всю необхідну інформацію про послуги, що надаються, не виходячи з дому, що підвищить кількість клієнтообігу для бізнесу і як наслідок підвищення виручки компанії.

1.1.1. Опис предметної галузі

Предметною областю даної кваліфікаційної роботи є створення інтернет магазину з продажу спортивного харчування.

Для того, щоб повністю представити функціонал сайту, виділено такі основні бізнес-процеси:

- перегляд товарів магазину компанії;
- отримання інформації про товари компанії;

Характеристика бізнес-процесу придбання товарів компанії представлена в Табл. 1.1.

Таблиця 1.1

Характеристика бізнес-процесу придбання послуг компанії

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Придбання товарів компанії
Основні учасники	Користувач
Вхідна подія	Запит на відправку форми покупки
Вихідна подія	Відправлення заповненої форми до БД
Вихідні документи	База даних
Клієнт бізнес процесу	Користувач

1.1.2. Огляд та аналіз існуючих аналогів

Перш ніж почати розробляти певну систему, потрібно виконати пошук її аналогів, визначити їх переваги та недоліки, щоб не допустити ті ж помилки та використовувати переваги проаналізованих систем.

Проаналізовано три веб-ресурси, що реалізують функції предметної області.

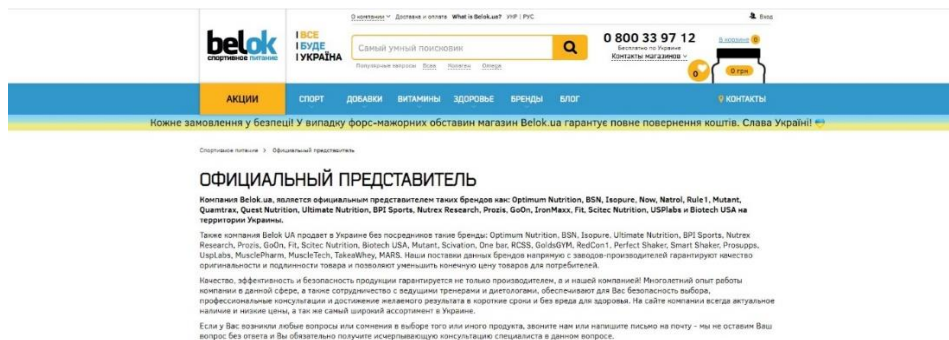
Спочатку вибрано для аналізу веб-сайт компанії «Belok» (режим доступу: <https://belok.ua/>)

Наступним – веб-сайт компанії " Всаа " (режим доступу: <https://bcaa.ua/>).

Третім аналогом обрано веб-сайт компанії «Bodymarket» (режим доступу: <https://bodymarket.ua/>).

Розглянемо докладніше подані вище аналоги. Першим розглянемо портал <https://belok.ua/>

На рис. 1.2 зображено вигляд головної сторінки веб-ресурсу.



ОФИЦИАЛЬНЫЙ ПРЕДСТАВИТЕЛЬ

Компания Belok.ua является официальным представителем таких брендов как: Optimum Nutrition, BSN, Isopure, Now, Metrol, Rule 1, Mutant, Quantum, Quest Nutrition, Ultimate Nutrition, BPI Sports, Nutrex Research, Prozis, Gator, IronMaxx, Fit, Scitec Nutrition, USPlabs и Biotech USA на территории Украины.

Также компания Belok UA продает в Украине без посредников такие бренды: Optimum Nutrition, BSN, Isopure, Ultimate Nutrition, BPI Sports, Nutrex Research, Prozis, Gator, Fit, Scitec Nutrition, Biotech USA, Mutant, Scivation, One Back, KCS, GoldGym, RedCon1, Perfect Shaker, Smart Shaker, Prosupps, Uppaboo, MusclePharm, MuscleTech, TakeMyMuscle, MARS. Наши поставки данных брендов напрямую с заводом-производителем гарантируют качество оригинальности и подлинности товара и позволяют уменьшить конечную цену товаров для потребителей.

Качество, эффективность и безопасность продукции гарантируется не только производителем, а и нашей компанией! Многолетний опыт работы компании в данной сфере, а также сотрудничество с ведущими тренерами и диетологами, обеспечивает для Вас безопасность выбора, профессиональные консультации и достижение желаемого результата в короткие сроки и без вреда для здоровья. На сайте компания всегда актуальными наливает и микширует, а так же самый широкий ассортимент в Украине.

Если у Вас возникли любые вопросы или сомнения в выборе того или иного продукта, звоните нам или напишите письмо на почту - мы не оставим Ваш вопрос без ответа и Вы обязательно получите исчерпывающую консультацию специалиста в данном вопросе.



Рис. 1.2. Вид головної сторінки ресурсу

Ця сторінка має зручний інтерфейс, за допомогою якого користувач може легко зорієнтуватися у подальших діях.

На цьому ресурсі відображається стрічка основних послуг компаній, невелика навігація сайту представлена у вигляді вертикального меню. При натисканні вкладки відкривається докладна інформація, прикріплена до них. Щоб отримати інформацію про послугу, потрібно натиснути на пункт меню.

Наступним роздивимось веб-сайт <https://bcaa.ua/>:

На рис. 1.3 зображено вид головної сторінки веб-ресурсу.



Рис. 1.3. Вигляд головної сторінки веб-ресурсу

На даному ресурсі відображається банер компанії, навігація по сайту представлена у вигляді вертикального меню, є інформація про цінову політику компанії, а також меню основних послуг компанії. При натисканні на вкладки відкривається докладна інформація, прикріплена до них.

Останнім розглянемо портал <https://bodymarket.ua/>:

На рис. 1.4 зображено головну сторінку веб-ресурсу.

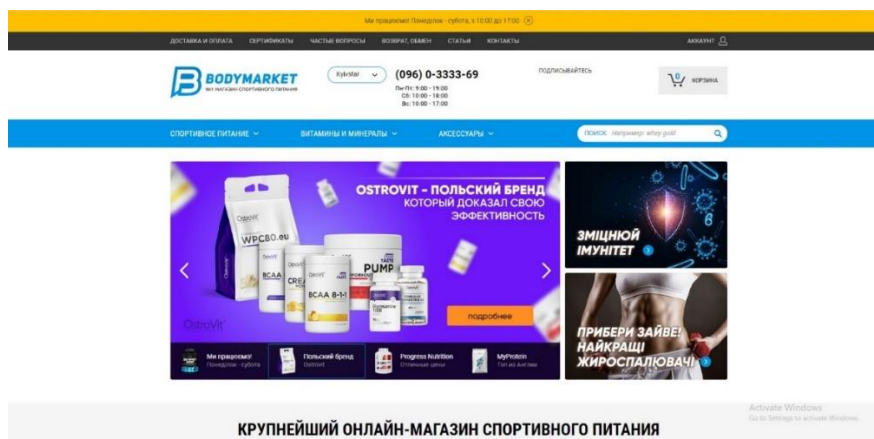


Рис. 1.4. Вигляд головної сторінки веб-ресурсу

На даному ресурсі відображається кнопка виклику форми зворотного дзвінка, навігація по сайту представлена у вигляді горизонтального меню, є посилання на соціальні мережі компанії. Щоб отримати консультацію з послуг, потрібно натиснути кнопку з номером компанії і заповнити викликану форму. Також присутня основна інформація про компанію, її графік роботи та адресу.

Порівняльна характеристика зазначених вище програмних продуктів представлена у табл. 1.2.

Таблица 1.2

Порівняльна характеристика програмних продуктів

Фірма розробник	Aiwe.pl	Reklama.lviv	Webera
-----------------	---------	--------------	--------

Закінч. табл. 1.2

Назва програмного продукту	Belok	Всаа	Bodymarket
Функціональність	На даному ресурсі користувач може ознайомитися з основними послугами компанії, інформацією про послуги, вибрати мову використання, а також переглянути інформацію про компанію	На даному ресурсі користувач може ознайомитися з основною інформацією про компанію, списком послуг компанії та її ціновими діапазонами	На даному ресурсі користувач може ознайомитися з основними послугами компанії, інформацією про послуги, вибрати мову використання, а також переглянути інформацію про компанію
Інтерфейс користувача	Є інтуїтивно зрозумілим, колірна гама підібрана вдало.	Є інтуїтивним, колірна гама підібрана вдало.	Не є інтуїтивно зрозумілим, колірна гама підібрана невдало.

Після огляду систем-аналогів було проаналізовано всі переваги та недоліки цих ресурсів. До переваг відносяться такі функції, як перегляд актуальних послуг компанії, контакти компанії, список товарів компанії, можливість віддаленої

покупки товарів, а також зручний інтерфейс користувача. Також значною перевагою є можливість написати відгук про компанію.

Основним недоліком є відсутність перерахованих вище функцій, а також не інтуїтивний інтерфейс користувача, що ускладнює роботу з системою.

Всі вище наведені переваги та недоліки були враховані під час розробки інтернет магазину з продажу спортивного харчування «SportCompanу».

1.1.3. Глосарій основних термінів

У табл. 1.3 представлений глосарій основних термінів, що використовуються при розробці інтернет магазину з продажу спортивного харчування «SportCompanу».

Таблиця 1.3

Глосарій основних використовуваних термінів

Веб-сторінка	Інформаційний ресурс, доступний в інтернеті, який можна переглянути за допомогою веб-браузера
Інтернет сервіс	Послуги, що надаються в інтернеті за допомогою спеціальних програм
Ключові слова	Слова, що мають суттєве смислове навантаження, вони є ключем при пошуку інформації в інтернеті або на сторінці сайту
Компанія з продажу спортивного харчування	Компанія, основною сферою діяльності якої є продаж спортивного харчування
Браузер	Програмне забезпечення, яке дозволяє користувачеві переглядати інформацію в інтернеті

СУБД	Програмне забезпечення, що надає користувачеві можливість створити, редагувати, зберегти та видалити інформацію з баз даних
------	---

Закінч. табл. 1.3

Web-server	Апаратне забезпечення, що надає користувачеві свої ресурси та доступ до віддалених сервісів
Адміністратор	Користувач системи, який підтримує роботу системи
Клієнт	Особа, яка використовує систему для отримання доступу до певних послуг
HTML 5	Мова розмітки
PHP	Мова серверної частини
JavaScript	Мова скриптів

1.2. Призначення розробки та галузь застосування

Розробка даного веб-сайту реалізує можливість інтеграції розробленого програмного продукту в бізнес з продажу спортивного харчування, з метою його автоматизації. Фактично поняття веб-сайту стало невід'ємною частиною будь-якого бізнесу у сфері надання послуг продажу спортивного харчування.

Результати аналізу існуючих даних свідчать, що нині більшість як великих мережеских магазинів спортивного харчування, або невеликих локальних тією чи іншою мірою переглянули свій підхід до структури та процесів керування своїм бізнесом у бік автоматизації. Проектоване програмне забезпечення надасть можливість клієнтам магазину переглядати наявні у продажу товари, отримувати інформацію про них, писати відгуки про роботу магазину та надсилати запит на придбання товару.

Також розроблене програмне забезпечення надасть можливість адміністратору магазину зручно наповнювати сайт новими товарами, або редагувати чи видаляти існуючі.

1.3. Підстава для розробки

Підставою для розробки є наявність попиту на розробку інформаційних систем автоматизації ведення бізнесу. Сайт розробляється для компанії «SportCompanу».

В результаті передпроектного аналізу були визначені наступні цілі, для досягнення яких розробляється сайт:

1. Презентація. Сайт повинен відображати діяльність компанії для наявних і потенційних клієнтів і партнерів.
2. Спілкування. Сайт повинен підтримувати комунікацію компанії з клієнтами та партнерами.
3. Наявність на сайті наступної інформації: інформація про організацію, інформація для потенційних партнерів організації, інформація для клієнтів, юридична адреса, поштова адреса, контактна інформація. Сайт має сприяти залученню клієнтів.
4. Можливість дистанційного придбання товарів компанії без залучення менеджерів компанії. Сайт повинен давати можливість купувати товари в автономному режимі.
5. Сайт повинен мати приємний та інтуїтивно зрозумілий інтерфейс користувача.

1.4. Постановка завдання

Метою даної кваліфікаційної роботи є створення інтернет магазину з продажу спортивного харчування "SportCompanу".

Завданням кваліфікаційної роботи є розробка спеціалізованої системи керування контентом для інформаційного наповнення сайту магазину. Обсяг

вимог до інформаційного наповнення даного сайту дозволяє створити швидко та комфортну для користувача систему керування на основі контентних блоків. Кожна сторінка формується із спеціалізованих блоків. Кожен блок має свої властивості і може містити тільки заздалегідь певну інформацію, представлену в детермінованому вигляді, з використанням стандартного стилю представлення і оптимізованого для кросбраузерного представлення.

Відсутність зайвих модулів і можливість формування сторінки тільки з блоків потрібного типу дозволяє створити невеликі за обсягом сторінки, що швидко генеруються, з різноманітною інформацією, текстом, графікою, таблицями, засобами зворотного зв'язку.

Система управління контентом веб-сайту, що розробляється, повинна мати такі можливості для адміністратора системи:

- додавати нові товари у базу даних магазину;
- редагувати існуючі у базі даних магазину товари;
- додавати до інформації про товар зображення;
- забезпечення зручного та наочного інтерфейсу адміністратора, який не має навичок роботи з базами даних та навичок верстки;
- забезпечення надійного зберігання інформації;
- забезпечувати захист від несанкціонованого доступу.
- для відвідувача сайту система управління має надавати такі можливості:
 - отримувати інформацію про актуальні товари магазину;
 - можливість передивлення сторінки товару з інформацією про нього;
 - можливість додання товару до кошика;
 - можливість написати відгук про роботу компанії;
 - можливість придбання доданих до кошика товарів;
 - можливість сортування товарів за категоріями.

Адміністратор системи після входу в систему управління (після введення логіну та паролю) отримує повний доступ до бази даних, має можливість додавати, редагувати та видаляти існуючі товари.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Список вимог до функціональних характеристик розроблюваного ПО представлений у табл. 1.4.

Таблиця 1.4

Список вимог до функціональних характеристик

Ідентифікатор вимоги	Назва вимоги	Приоритет	Складність	Контакт
1	Придбання товарів	Обов'язковий	Висока	Користувач Менеджер
2	Перегляд актуальних товарів	Обов'язковий	Висока	Користувач
3	Написання відгуку	Рекомендовано	Середня	Користувач Адміністратор
4	Наповнення бази новими товарами	Обов'язковий	Висока	Користувач
5	Сортування товарів за категоріями	Рекомендовано	Висока	Користувач Керівник
6	Перегляд інформаційних сторінок про товар	Обов'язковий	Висока	Користувач

1.5.2. Вимоги до інформаційної безпеки

Для забезпечення інформаційної безпеки особистих даних користувачів, таких як пароль, платіжні реквізити, ці дані буде зашифровано за допомогою алгоритмів симетричного шифрування. Симетричне шифрування передбачає використання одного й того самого ключа і для зашифрування, і для розшифрування. До симетричних алгоритмів застосовуються дві основні вимоги: повна втрата всіх статистичних закономірностей в об'єкті шифрування та відсутність лінійності. Прийнято розділяти симетричні системи на блокові та поточкові. У блокових системах відбувається розбиття вихідних даних на блоки з подальшим перетворенням ключа. У поточкових системах виробляється певна послідовність (вихідна гамма), яка у подальшому накладається саме повідомлення, і шифрування даних відбувається потоком у міру генерування гами.

Операція підстановки виконує першу вимогу до симетричного шифру, позбавляючись будь-яких статистичних даних шляхом перемішування бітів повідомлення за певним заданим законом. Перестановка необхідна до виконання другої вимоги – надання алгоритму нелінійності. Досягається за рахунок заміни певної частини повідомлення заданого обсягу на стандартне значення шляхом звернення до вихідного масиву.

Симетричні системи мають свої переваги, так і недоліки перед асиметричними. До переваг симетричних шифрів відносять високу швидкість шифрування, меншу необхідну довжину ключа за аналогічної стійкості, велику вивченість і простоту реалізації. Недоліками симетричних алгоритмів вважають насамперед складність обміну ключами через велику ймовірність порушення секретності ключа при обміні, який необхідний, і складність управління ключами у великій мережі.

+ Параметри










	id	login	name	lname	password	country
<input type="checkbox"/>   	7	VGvk	VGvkZHk=	QnJvd24=	MTIzNDU=	0KPQutGA0LDQuNC90LA=
<input type="checkbox"/>   	8	VG9t	VG9tbXk=	eGdmZGdz	ZHNnc2Rn	0KPQutGA0LDQuNC90LA=
<input type="checkbox"/>   	9	QWxleA==	QWxleGFuZHI=	QmFyc3Vr	MTIzNDU=	0KDQvtGB0YHQuNGP

Рис. 1.5. Зберігання зашифрованих даних у базі даних

1.5.3. Вимоги до складу та параметрів технічних засобів

Одним з найважливіших факторів, який необхідно врахувати під час розробки програми, є відповідність потреб у ресурсах наявному технічному забезпеченні. Програма має коректно працювати на сумісному з ним обладнанні. Реалізація проекту проводиться в системі розробки сценаріїв (PHP), включає інтерпретатор мови, набір функцій для доступу до баз даних та різних служб Інтернет. Використання цього продукту пред'являє такі вимоги до обладнання та програмного забезпечення:

- процесор Intel Pentium III 866 МГц та вище;
- оперативна пам'ять 1024 Мбайт та вище;
- дисковий простір щонайменше 100 Мб;
- відеокарта 256Мб;
- мінімальна роздільна здатність екрана 1024x768;
- 32-розрядна операційна система Windows (2000/XP/Vista/Seven/10);
- підтримка технології Flash;
- інтернет-браузер: Opera, Mozilla Firefox, Internet Explorer, Google Chrome;
- маніпулятор "миша";
- клавіатура.

Для коректного запуску програмного продукту на сервері, визначено наступні вимоги до серверного обладнання та програмного забезпечення:

- операційна система – Linux (CentOs, Debian тощо);
- PHP 7, Mariadb, Nginx, PHP-FPM;
- оперативна пам'ять від 2 Гб;

- наявність SSD;
- процесор щонайменше 2-х ядер;
- Vesta, ISPmanager

1.5.4. Вимоги до інформаційної та програмної сумісності

Програмний модуль «SportCompanu» може використовуватись на комп'ютерах із встановленою операційною системою Windows XP, Vista, 7, 10.

Програма має повну сумісність із офісними продуктами компанії Microsoft, та не викликає конфліктів з іншими програмами.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Дана програма є спеціалізованим програмним виробом та призначена для надання користувачеві можливості використання послуг компанії “SportCompany” за допомогою стаціонарного або переносного персонального комп’ютера (ноутбука).

Програма виконує наступні функціональні дії:

- відображення на інтерфейсі користувача HTML сторінок;
- виведення на інтерфейс користувача даних з бази даних магазину;
- відображення форм для комунікації з користувачем;
- взаємодія з базою даних магазину за допомогою даних, введених у форми;
- авторизація адміністратора для доступу до адмінпанелі;
- доповнення існуючого контенту сайту за допомогою адмінпанелі.

2.2. Опис застосованих математичних методів

Розроблюваний програмний продукт не потребує використання математичних методів для проектування своєї структури. Математичні методи були використані при розрахунку економічної складової сайту, та наведені у розділі 3 кваліфікаційної роботи.

2.3. Опис використаних технологій та мов програмування

Розглянемо найпопулярніші мови веб-розробки та визначимо, яка з них найкраще підійде для вирішення нашого завдання.

JavaScript

Однією з найпопулярніших мов веб-програмування є JavaScript. Згідно з щорічними звітами різних популярних платформ, таких як Stack Overflow та Octoverse, JavaScript є однією з найбільш кращих та провідних мов програмування у світі технологій.

Одна з основних причин цього полягає в тому, що конкретна мова може використовуватися як для веб-розробки переднього плану, так і для веб-розробки внутрішнього інтерфейсу.

Дивлячись на деякі попередні тенденції та статистику, можна сказати, що популярність Node.js якимось чином збільшила використання JavaScript як внутрішньої мови для веб-розробки. Тим часом, мова надає вам чудові функції для серверної розробки, таких як полегшена мова сценаріїв, динамічна типізація, інтерпретація, підтримка об'єктно-орієнтованого програмування, перевірка на стороні клієнта, величезна підтримка спільноти та багато іншого.

Фреймворки JavaScript для серверної веб-розробки: Next.js, Express, MeteorJS і т.д.

Популярні веб-сайти, що використовують JavaScript: Faceitem, Google, eBay і т.д.

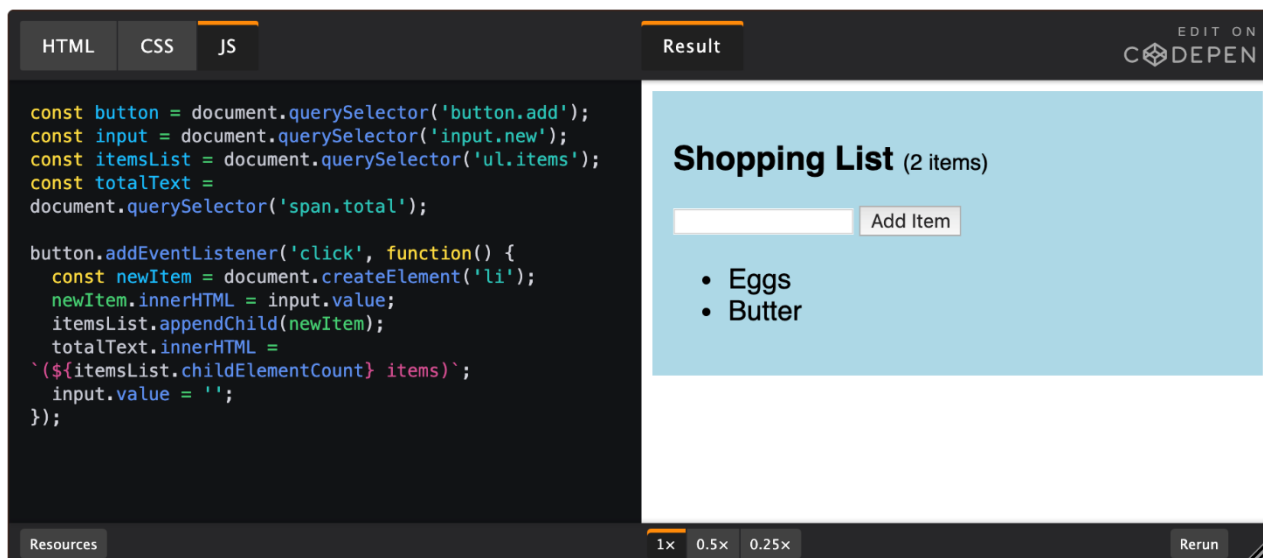
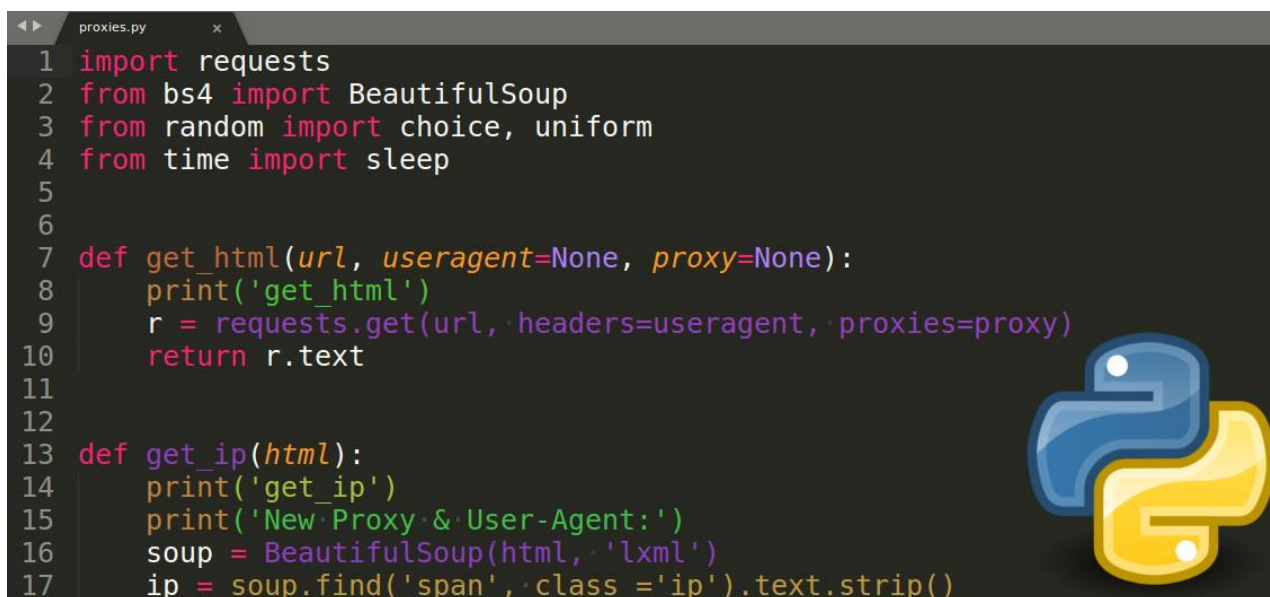


Рис. 2.2. Приклад розробки на JavaScript

Python

Хоча Python досить відомий серед людей своєю сумісністю з передовими технологіями, такими як машинне навчання, IoT, наука про дані. Ця збагачувальна мова програмування широко використовується і дуже підходить для серверної веб-розробки також. Навіть один із провідних ІТ-гігантів сучасності Google значною мірою покладається на Python, і це одна з трьох основних мов, які використовуються Google (дві інші — Java та C++). Однією з основних переваг використання Python для веб-розробки є величезний набір стандартних бібліотек, які роблять роботу розробників порівняно простішою та ефективнішою. Додаткові видатні та унікальні функції Python, такі як краща читаємість коду, більш проста інтеграція з іншими мовами, підтримка програмування з графічним інтерфейсом, переносимість та багато інших факторів роблять його більш переважною мовою серед веб-розробників. Фреймворки Python для серверної веб-розробки: Django, Flask, Pyramid тощо. Популярні веб-сайти, що використовують Python: Spotify, Pinterest, Instadray тощо.

A screenshot of a code editor window titled 'proxies.py'. The code is written in Python and includes imports for 'requests', 'BeautifulSoup', 'choice', 'uniform', and 'sleep'. It defines two functions: 'get_html' which uses 'requests.get' to fetch a URL with optional user agent and proxy, and 'get_ip' which uses 'BeautifulSoup' to parse the HTML and find an IP address. A Python logo is overlaid on the right side of the code block.

```
1 import requests
2 from bs4 import BeautifulSoup
3 from random import choice, uniform
4 from time import sleep
5
6
7 def get_html(url, useragent=None, proxy=None):
8     print('get_html')
9     r = requests.get(url, headers=useragent, proxies=proxy)
10    return r.text
11
12
13 def get_ip(html):
14    print('get_ip')
15    print('New Proxy & User-Agent:')
16    soup = BeautifulSoup(html, 'lxml')
17    ip = soup.find('span', class_='ip').text.strip()
```

Рис. 2.3. Приклад розробки на Python

PHP є досвідченим гравцем у світі веб-розробки. Ця серверна мова сценаріїв з відкритим вихідним кодом створена в 1994 році і спеціально використовується для веб-розробки. Оскільки це мова, що інтерпретується, вона також не потребує компілятора, а також може працювати практично у всіх основних операційних системах, таких як Windows, Linux, macOS, Unix і т. д., а також має багато переваг, таких як: простоту в освоєнні, крос-платформну сумісність, функції ООП, підтримка різних стандартних баз даних, таких як MySQL, SQLite і т. д., величезна підтримка спільноти та багато іншого. Крім цього, PHP дуже безпечний як мова сценаріїв на стороні сервера, оскільки в PHP доступно безліч хеш-функцій для шифрування даних користувача. Зокрема, якщо ви є новачком, ви можете вибрати PHP для серверної веб-розробки. PHP-фреймворки для серверної веб-розробки: Laravel, CodeIgniter, Symfony і т.д. Популярні веб-сайти, що використовують PHP: WordPress, MailChimp, Flickr і т.д.

```
<?php

namespace Email;

class Client
{
    public function send(string $emailAddress, Message $message): bool
    {
        if (!$this->validateParameters($emailAddress, $message)) {
            return false;
        }

        return $this->sendToMailServer([
            'to' => $emailAddress,
            'text' => $message->text(),
        ]);
    }
}
```

Рис. 2.4. Приклад розробки на PHP

Ruby

Ruby - ще одна мова програмування, яка відмінно підходить для веб-розробки. Подібно до PHP і Python, Ruby також простий в освоєнні і підходить для початківців.

Що робить Ruby особливим для веб-розробки, так це середовище Ruby on Rails, на якому працюють такі сайти, як Github, Shopify, Airbnb, Groupon, GoodReads та Kickstarter.

Rails – це середовище модель-представлення-контролер (MVC), що надає стандартні структури для бази даних, веб-служби та веб-сторінок. Воно заохочує та полегшує використання веб-стандартів, таких як JSON або XML, для передачі даних та HTML, CSS та JavaScript для взаємодії з користувачем.

Сама мова є високорівневою та чисто об'єктно-орієнтованою, що означає, що «кожне значення є об'єктом», і в Ruby немає примітивних типів даних. Він також має строгу динамічну типізацію та автоматичне складання сміття – форму для управління пам'яттю.

```
1  require "time"
2
3  class InvoiceItem
4    attr_reader :id, :item_id, :invoice_id, :created_at
5
6    attr_accessor :quantity, :unit_price, :updated_at
7
8    def initialize(params)
9      @id = params[:id].to_i
10     @item_id = params[:item_id].to_i
11     @invoice_id = params[:invoice_id].to_i
12     @quantity = params[:quantity].to_i
13     @unit_price = BigDecimal(params[:unit_price])
14     @created_at = Time.parse(params[:created_at].to_s)
15     @updated_at = Time.parse(params[:updated_at].to_s)
16   end
17
18   def unit_price_to_dollars
19     @unit_price.to_f
20   end
21 end
22
```

Рис. 2.5. Приклад розробки на Ruby

Java – ще одна зразкова мова програмування для серверної веб-розробки. Об'єктно-орієнтована мова програмування широко використовується для розробки веб-додатків масштабу підприємства, а також для розробки додатків для Android, настільних додатків, наукових додатків тощо. Основною перевагою використання Java є те, що вона працює за принципом «написати один раз і запустити будь-де», тобто скомпільований код Java може виконуватись на будь-якій платформі, що підтримує Java, без необхідності повторної компіляції. Говорячи конкретніше, код Java спочатку компілюється в байт-код, який залежить від машини, та був цей байт-код запускається на JVM незалежно від базової архітектури.

Крім того, Java підтримує багатопоточність, що дозволяє одночасно виконувати два або більше потоків для максимального використання ЦП. Іншими додатковими функціями Java є незалежність від платформи, дотримання концепцій ООП, багаті бібліотеки з відкритим вихідним кодом, автоматичне виділення пам'яті та складання сміття, переносимість тощо.

Java Frameworks для серверної веб-розробки: Spring, Struts, Grails.

Популярні веб-сайти, що використовують Java: LinkedIn, IRCTC, Yahoo і т.д.

```
def get_symbols(file_name):
    with open(file_name, "r") as in_file:
        records = []
        count = 0
        symbol_set = ""
        for line in in_file:
            symbol_set = symbol_set + line[:-1] + ','
            count = count + 1
            if count % 50 == 0:
                records.append(symbol_set)
                symbol_set = ""

        symbols.append(symbol_set)
    return records
```

Рис. 2.6. Приклад розробки на Java

C#

C# - одна з тих небагатьох мов, які останні кілька років постійно входять до п'ятірки найкращих мов програмування за різними стандартними індексами. Однак потрібно знати, що ця мова загального призначення спочатку була розроблена компанією Microsoft в першу чергу для платформи .Net. Поряд із серверною веб-розробкою в даний час C# широко використовується в багатьох областях, таких як розробка програм для Windows, розробка ігор і т. д. мова надає різні корисні функції, такі як швидка компіляція, функціональна сумісність, масштабованість та оновлюваність, компонентно-орієнтованість, структурована мова та багато інших. Крім того, C# пропонує широкий набір бібліотек, які допомагають розробникам прискорити та підвищити ефективність процесу розробки.

Фреймворком C# для серверної веб-розробки є .NET.

Популярні веб-сайти, що використовують C#: GoDaddy, Marketwatch, Stack Overflow і т.д.

```
public async Task<User> RegisterAsync(string login, string password)
{
    var hash = Hash(password);
    var exist = await _repository.Where<User>(s => s.Login == login && s.Password == hash).AnyAsync();

    if (exist)
        throw new AppError("User already exist");

    var id = Guid.NewGuid();
    var user = new User(id, login, hash);

    await _repository.SaveAsync(user);
    await _emailService.SendAsync(user.Login, "Добро пожаловать", "Бла бла бла");

    return user;
}
```

Рис. 2.7. Приклад розробки на C#

Для аналізу характеристик зазначених у попередньому пункті технологій, складемо порівняльні таблиці для проведення порівняння за такими

характеристиками як: Парадигми, Типізація, Управлінню пам'яттю, Об'єктно - орієнтованими можливостями та частотою використання.

Таблиця 2.1

Порівняльна характеристика за парадигмами

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Імперативна	+	+	+	+	+	+
Об'єктно-орієнтована	+	+	+	+	+	+
Функціональна	+/-	+/-	+/-	+/-	+	+
Рефлексивна	+/-	+/-	+	+	+	+
Узагальнене програмування	+	+	+	+	+	+
Логічна	-	-	-	-	-	-
Декларативна	+/-	-	+/-	+	+	+
Розподілена	+/-	+	-	+	+/-	+/-

Таблиця 2.2

Порівняльна характеристика за типізацією

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Статична типізація	+	+	+	+	+	+
Динамічна типізація	+	-	+	+	+	+
Явна типізація	+/-	+	-	+/-	+/-	-
Неявна типізація	+	-	+	+	+	+
Неявне приведення типів	-	+	+	+	+	+

Таблиця 2.3

Порівняльна характеристика за можливостями керування пам'яттю

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Створення об'єктів на стеку	+	-	-	-	-	-
Некеровані вказівники	+	-	-	-	-	-
Ручне керування пам'яттю	+	-	-	-	-	-
Збирання сміття	+	+	+	+	+	+

Таблиця 2.4

Порівняльна характеристика за функціональними можливостями

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Декларації чистоти функцій	-	-	-	-	-	-
First class функції	+	-	+	+	+	+
Анонімні функції	+	+	+	+	+	+
Лексичні замкнення	+	+	+	+	+	+

Закінч. табл. 2.4.

Часткове застосування	-	-	+	-	+	+
Карування	+	-	+	+	+	+

Таблиця 2.5

Порівняльна характеристика за об'єктно – орієнтованими можливостями

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Інтерфейси	+	+	+	+	+	-
Мультиметоди	+/-	-	-	-	-	-

Міксини	-	+	+	+	+	+
Перейменування при спадкуванні	-	-	-	-	-	-

Проведено аналіз усіх діючих сайтів за частотою використання та отримано наступний графік (рис. 2.8).

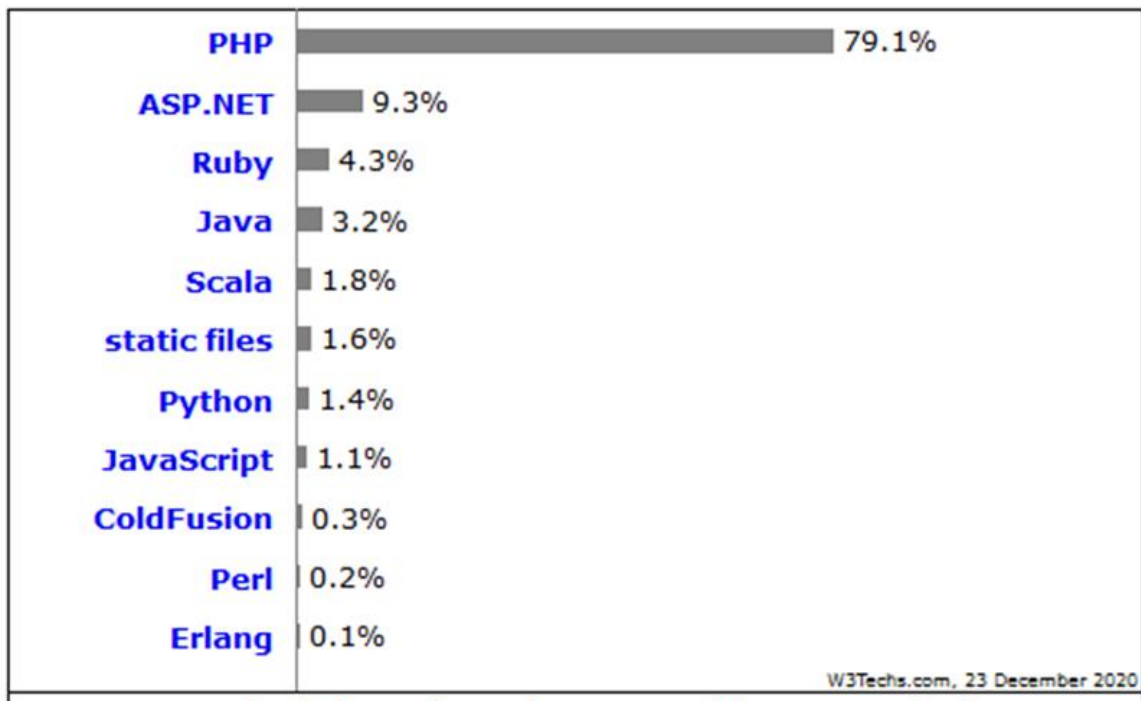


Рис. 2.8. Частота використання мов програмування у існуючих веб-сайтах

З проведеного аналізу видно, що біля 79% сайтів, про бекенд яких відомо, використовують PHP у своїй конструкції.

Аналіз існуючих СУБД для вирішення поставленої задачі:

Незалежно від того, яку інформаційну систему та з використанням яких мов програмування ми проектуватимемо, для повного її функціонування ми використовуватимемо бази даних.

База даних є систематизований набір даних. Вони підтримують електронне зберігання та маніпулювання даними. Бази даних значно спрощують управління даними.

Наводячи приклад бази даних: онлайн телефонний довідник використовує базу даних для зберігання даних про людей, телефонні номери та інші контактні дані. Постачальник електроенергії використовує базу даних для управління виставленням рахунків, проблемами, пов'язаними з клієнтами, обробкою даних про несправності тощо.

Давайте розглянемо також Faceitem. Він повинен зберігати, обробляти та подавати дані, пов'язані з учасниками, їх друзями, діями учасників, повідомленнями, рекламою та багатьом іншим. Можна надати безліч прикладів використання баз даних.



Рис. 2.9. Схема роботи бази даних

Розглянувши найпопулярніші системи управління базами даних можна буде визначити, яка з них найкраще підійде для вирішення завдання.

MySQL

Це одна з найпопулярніших систем реляційних баз даних. Спочатку рішення з відкритим вихідним кодом, MySQL тепер належить корпорації Oracle. Сьогодні MySQL є основою прикладного програмного забезпечення LAMP. Це означає, що він є частиною стека Linux, Apache, MySQL та Perl/PHP/Python.

Маючи під капотом C і C++, MySQL добре працює з системними платформами, як Windows, Linux, MacOS, IRIX та іншими.

MariaDB

MariaDB, форк MySQL з відкритим вихідним кодом, має комерційну підтримку. Він працює під Стандартною громадською ліцензією GNU і має ті ж команди, API та бібліотеки, що й MySQL.

Oracle

Oracle – це система управління реляційними базами даних, створена та керована корпорацією Oracle. В даний час вона підтримує кілька моделей даних, таких як документ, граф, реляційна модель та модель "ключ-значення" в рамках однієї бази даних. У останніх випусках він переорієнтувався на хмарні обчислення. Архітектура бази даних тепер спрощує упаковку багатьох баз даних та забезпечує плавне керування ними.

PostgreSQL

Ця система управління базами даних поділяє свою популярність із MySQL. Це об'єктно-реляційна СУБД, в якій об'єкти користувача та табличні підходи об'єднуються для створення більш складних структур даних. Крім того, PostgreSQL має багато спільного з MySQL. Він спрямований на зміцнення стандартів відповідності та розширюваності. Отже, він може обробляти будь-яке робоче навантаження як для одномашинних продуктів, так і для складних додатків.

Для аналізу зазначених СУБД складемо порівняльні таблиці для проведення порівняння за такими характеристиками як: переваги, недоліки, частота використання.

Таблиця 2.6

Порівняльна характеристика СУБД

Назва	Переваги	Недоліки
MySQL	<p>Чудово документована, пропонує багато функцій, навіть у безкоштовній версії. Пакет MySQL включений у стандартні репозиторії найпоширеніших дистрибутивів операційної системи Linux, що дозволяє дуже просто встановити її. Підтримує набір інтерфейсів користувача. Може працювати з іншими базами даних, включаючи DB2 та Oracle</p>	<p>Потрібно потратити багато часу і зусиль, щоб змусити MySQL виконувати нескладні завдання, хоча інші системи роблять це автоматично, наприклад: створювати інкрементні резервні копії. Відсутня підтримка XML або OLAP. Для безкоштовної версії доступна лише платна підтримка</p>

Закінч. табл. 2.6

<p>MariaDB</p>	<p>Продуктивність Індикатори дадуть вам знати, як обробляється запит. Розширювана архітектура та плагіни дозволяють налаштовувати інструмент відповідно до ваших потреб.</p>	<p>На сьогодні стабільність нижча, ніж у MySQL, тому навіть на нових проектах можна рекомендувати встановлювати mysql. Двигун досить новий, тому поки що немає жодних гарантій подальших оновлень.</p>
----------------	--	--

	Шифрування доступне в мережі, сервері та рівні програми.	Як і в інших безкоштовних базах даних, вам доведеться платити за підтримку.
PostgreSQL	Є масштабованим рішенням та дозволяє обробляти терабайти даних. Підтримує формат json. Існує безліч визначених функцій. Доступний ряд інтерфейсів.	Документація туманна, тому, можливо, відповіді на деякі запитання доведеться шукати в Інтернеті. Конфігурація може збентежити непідготовленого користувача. Швидкість роботи може зменшуватися під час пакетних операцій або виконання запитів читання.
Oracle	Найсвіжіші інновації та вражаючий функціонал уже впроваджено в цьому продукті, оскільки Oracle прагне тримати планку навіть на тлі інших розробників СУБД. Oracle є вкрай надійною.	Вартість Oracle може бути непомірно високою, особливо для невеликих організацій. Система може вимагати значних ресурсів відразу після установки, тому можливо потрібно модернізувати обладнання для впровадження Oracle

Провівши аналіз усіх випадків використання СУБД за частотою використання та отримуємо наступний графік (рис. 2.10).

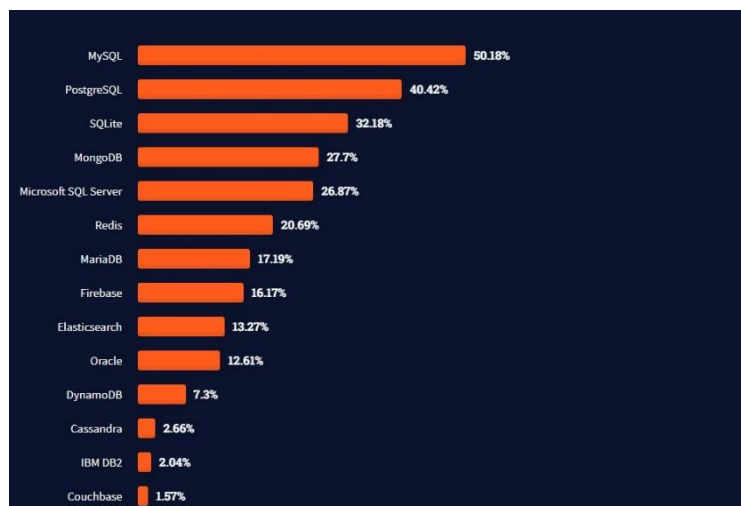


Рис. 2.10. Аналіз використання СУБД

Як показано на графіку, близько 50% сайтів, про бекенд яких відомо, використовують базу даних MySQL у своїй конструкції. Для збереження даних у проєктованому інтернет магазині буде використано саме цю базу даних, так як її функціонал ідеально підходить для виконання поставлених завдань.

2.4. Опис структури системи та алгоритмів її функціонування

Інформаційна модель відображує загальний склад інформаційних об'єктів автоматизованої системи, яка проєктується та зв'язки між ними. Структура інформаційної моделі системи наведена на (рис. 2.11).

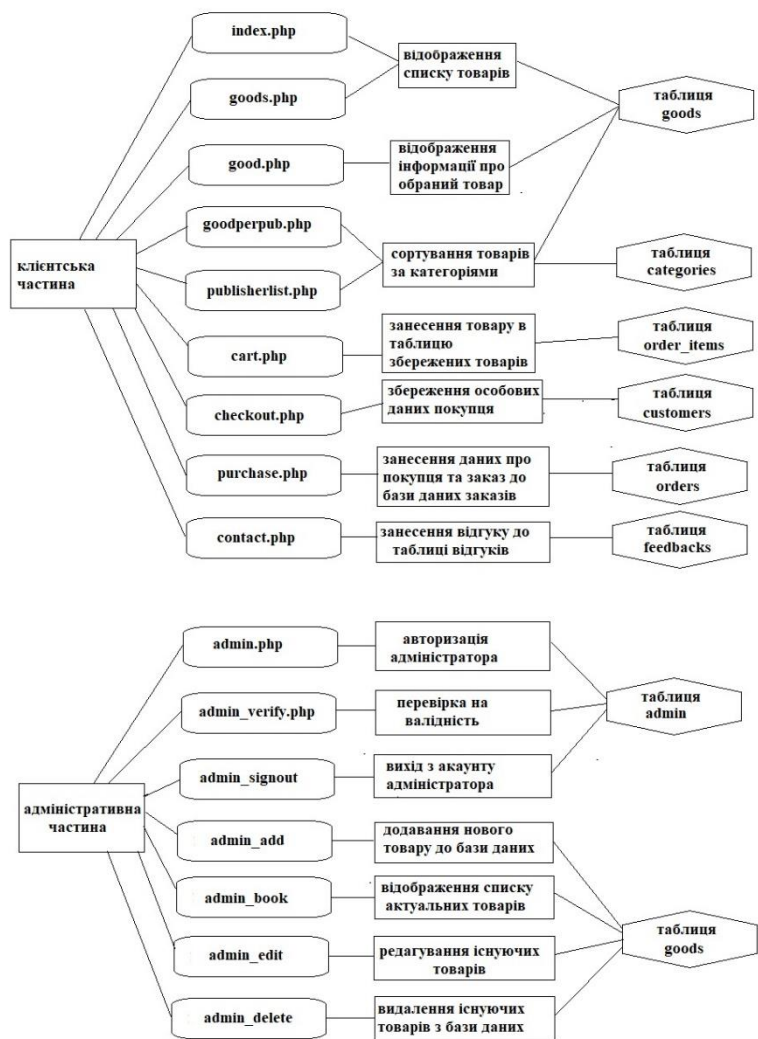


Рис. 2.11. Структура розроблюваного ПО

Програма складається з набору програмних модулів, кожний з яких виконує одну чи декілька функцій. У таблиці 2.7 наведено призначення програмних модулів.

Призначення програмних модулів

№ з/п	Позначення	Призначення
1	Admin	Модуль входу у особистий кабінет адміністратора, містить форми для введення логіну та паролю
2	Admin_add	Модуль для додавання нових товарів у базу даних магазину. Містить форму для введення інформації про товар та завантаження його банера. Після введення даних якщо вони валідні, вони будуть занесені до відповідної таблиці
3	Admin_item	Модуль для перегляду існуючих у базі даних товарів та можливістю переходу на сторінки їх редагування. Містить блок з інформацією з бази та кнопками редагування та видалення
4	Admin_delete	Модуль для видалення обраного товару з бази даних магазину
5	Admin_edit	Модуль для редагування обраного товару у базі даних магазину
6	Admin_signout	Модуль для завершення сесії адміністратора
7	Admin_verify	Модуль для перевірки валідності введених даних при авторизації адміністратора
8	Good	Модуль для відображення інформації про обраний товар, яка зберігається у базі даних та витягується за запитом користувача
9	GoodPerPub	Модуль для відображення товарів відповідної категорії, обраної у модулі Publisher_list

10	Goods	Модуль для відображення повного списку товарів магазину. Інформація про товари знаходиться у базі даних та витягується за запитом користувача
11	Dray	Модуль для відображення корзини товарів. Користувач може додавати будь яку кількість товарів до корзини перш, ніж придбати їх
12	Checkout	Модуль для введення особистих даних покупця для оформлення замовлення. Містить форму для введення даних з полями особистих даних. Після підтвердження, якщо дані валідні, їх буде збережено у базі даних користувачів
13	Contact	Модуль для надання можливості користувачу залишити відгук про роботу порталу. Містить форму для заповнення особових даних та безпосередньо самого відгуку. Після підтвердження, введені дані буде занесено до бази даних відгуків
14	Index	Головний модуль програми. Містить привітальний банер та забезпечує перехід на інші сторінки сайту
15	Process	Модуль для опису роботи логіки процесу покупки. Не має інтерфейсної частини та відповідає тільки за процес роботи бронювання
16	Publisher_list	Модуль для відображення категорій товарів
17	Purchase	Модуль для введення платіжних даних покупця для оформлення замовлення. Містить форму для введення даних з полями платіжних даних. Після підтвердження, якщо дані валідні, замовлення буде збережено у відповідній таблиці

2.4.1. Опис архітектури системи

В основі системи, що розробляється, лежить архітектура «клієнт-сервер», в якій завдання або мережеве навантаження розподілені між постачальниками послуг (сервісів), званих серверами, та замовниками послуг, званих клієнтами. Як середовище взаємодії клієнта з сервером використовується інтернет (рис. 2.1).

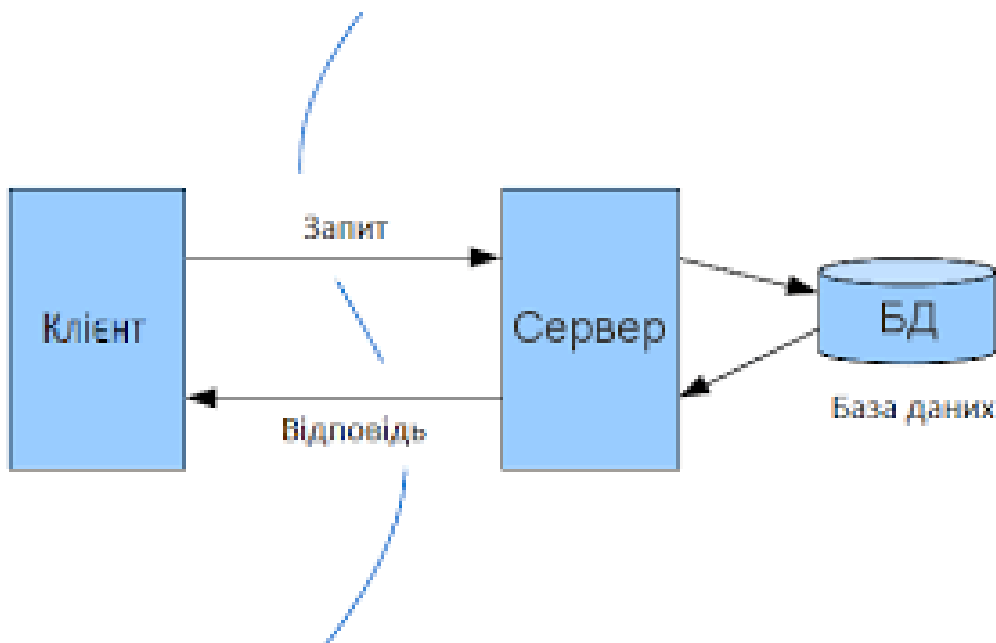


Рис. 2.1. Архітектура «клієнт-сервер»

Основними перевагами архітектури «клієнт-сервер» є:

- можливість, як правило, розподілити функції обчислювальної системи між кількома незалежними комп'ютерами у мережі. Це дозволяє спростити обслуговування обчислювальної системи. Зокрема, заміна, ремонт, модернізація або переміщення сервера, не зачіпають клієнтів;
- усі дані зберігаються на сервері, який, як правило, захищений набагато краще за більшість клієнтів. На сервері легше забезпечити контроль повноважень, щоб дозволяти доступ до даних лише клієнтам з відповідними правами доступу;
- дозволяє об'єднати різні клієнти. Використовувати ресурси одного сервера часто можуть клієнти з різними апаратними платформами, операційними системами тощо.

Основні недоліки:

- у разі використання централізованої системи, непрацездатність основного сервера може зробити непрацездатним всю програму;
- адміністрація даної системи потребує кваліфікованого професіонала;
- висока вартість обладнання.

Клієнтська частина програми повинна підтримувати такі технології:

- доступ до Інтернету;
- можливість роботи за протоколом HTTP;
- підтримка пристроїв взаємодії з людиною для введення даних.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

В якості вхідних даних в веб-сайтах виступають дані, за допомогою яких користувач може взаємодіяти з сервісом. Можна зробити висновок, що вхідними даними розроблюваного веб-сайту є:

Зі сторони покупця:

1. Прізвище покупця.
2. Ім'я покупця.
3. Місто покупця.
4. Поштовий індекс покупця.
5. Країна покупця.
6. Дані платіжного засобу покупця.

Зі сторони адміністратора:

1. Логін адміністратора.
2. Пароль адміністратора.
3. Назва товару.
4. Категорія товару.
5. Ціна товару.
6. Опис товару.

7. Банер товару.

8. Код товару.

Вихідними даними Web-сайту є вміст сторінок, а також сформоване замовлення у базі даних проекту.

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

Для виконання кваліфікаційної роботи використовувався персональний ЕОМ, який має такі характеристики:

- процесор: Intel Core i7 11375H 4x3.3GHz;
- пам'ять: 16GB DDR3, 1600MHz;
- відеокарта: Intel Iris Xe Graphics G7 96EUs;
- BIOS: E1563IMS.10C, 27/01/2021;
- жорст.диск: 1TB SSD;
- карта: Broadcom NetXtreme Gigabit Ethernet;
- монітор: MSI Optix MAG271C;
- опер.система: Microsoft Windows 10 Home (64-розрядна операційна система).

2.6.2. Використані програмні засоби

Провівши аналіз та порівняльну характеристику між актуальними у наш час та найбільш підходящими для вирішення нашого завдання інструментами, ми можемо зробити висновок, які з них використовуватимемо у процесі виконання роботи.

Для написання візуальної частини інтерфейсу веб-додатку інтернет магазину з продажу спортивного харчування буде використано HTML як основну розмітку для сторінок, які відображатимуться перед користувачем та CSS для стилізації цих сторінок та надання їм необхідного нам вигляду. В якості мови для

виконання бекнд частини буде використаний PHP, проведений аналіз показав, що ця мова найбільш підходить для виконання поставленого завдання, так як вона має досить зрозумілу структуру, є абсолютно вільно використовуваною, має хороші показники в області продуктивності і відмінно підходить для написання веб-додатків, що недвозначно підтверджується частотою її застосування. В якості СУБД була обрана MySQL, так як вона достатньо проста у використанні, має великий функціонал, безпечна, має хорошу швидкість роботи, а також має інтерфейс для взаємодії з PHP. Як IDE для розробки було вибрано PHPStorm, а для візуалізації роботи з базою даних PHPMyAdmin.

2.6.3. Виклик та завантаження програми

Розроблений веб-сайт може бути завантажений та запущений на віддаленому хостингу або локальному сервері. Для демонстрації роботи оптимальним буде запуск на локальному сервері засобами програми OpenServerPanel. OpenServer є портативною серверною платформою та програмним середовищем, створеним спеціально для веб-розробників з урахуванням їх рекомендацій та побажань.

Процес встановлення OpenServer та запуску на ньому розробленого програмного продукту описаний далі:

В процесі установки Open Server Panel позначити галочками пункт установки Microsoft Visual C++ 2005-2008-2010-2012-2013-2015-2019 Redistributable Package, пункт установки RuntimePack Lite, пункт налаштування Windows і пункт оптимізації системи для роботи з SSD, за його наявності. Без виконання цих дій робота Open Server Panel на комп'ютері не гарантується.

Інсталяцію/переустановку бібліотек від Microsoft та застосування налаштувань Windows необхідно виконувати з кожною новою версією Open Server.

Щоб запустити Open Server Panel, треба відкрити файл Open Server.exe. Після старту програми буде видно червоний прапорець в області повідомлень

Windows (область біля системного годинника). Щоб увімкнути безпосередньо веб-сервер треба натиснути [Меню → Запустити].

Перед початком роботи з Open Server Panel треба переконатися, що увімкнені потрібні модулі в налаштуваннях програми.

Нижче наведено встановлені в Open Server Panel початкові налаштування для підключення до різних модулів:

- підключення до MySQL;
- адреса: домен вашого сайту;
- порт: 3306;
- користувач: root;
- пароль: root.

Бажано використовувати звичну адресу localhost для підключення до MySQL, PostgreSQL, FTP або Memcache серверу.

Створення бази даних на локальному сервері Open Server:

1. Відкрити phpMyAdmin;
2. Натиснути "Створити БД";
3. У сторінці заповнити два поля під загальною назвою "Створити базу даних";
4. У лівому полі ввести ім'я бази даних: "www_project";
5. У правому полі вибрати: utf8_general_ci;
6. Натиснути на кнопку "Створити";
7. У сторінці, що відкрилась, натиснути "Привілеї", щоб створити користувача;
8. У блоці "Новий" натиснути "Додати користувача";
9. У сторінці, що відкрилася, в блоці "Інформація облікового запису" заповнити поля:
 - 1) ім'я користувача: вести таке саме, як і ім'я бази даних;
 - 2) хост: замість % ввести localhost;

- 3) пароль: ввести пароль бази даних сайту;
- 4) у блоці "База даних для користувача", поставити галочку в полі "Надати всі привілеї", на базу даних " www_project".

Блок "Глобальні привілеї" - не чіпати, "Відзначити все" - не відзначаємо. Натиснути кнопку "Вперед" - з'являється повідомлення: "Ви додали нового користувача". У списку таблиць БД (лівий стовпець) - виділити створену базу даних. Відкрити вкладку "Імпорт", у блоці "Імпортований файл" відзначити "Огляд вашого комп'ютера", натиснути на кнопку "Огляд", знайти файл бази даних (з розширенням .sql або його архів .sql .zip), натиснути на кнопку "Вперед". Зачекати кілька хвилин, якщо база велика. Після закінчення імпорту з'явиться повідомлення: "Імпорт успішно завершено, виконано xxxx запитів". Натиснути на кнопку "Вихід" (зверху ліворуч зелена кнопка "Exit") та закрити вікно phpMyAdmin. Перезапустити Open Server та відкрити сайт.

2.6.4. Опис інтерфейсу користувача

Інтерфейс програми побудований за стандартними схемами, що використовуються в розробці веб-додатків. Цей розділ допомоги докладно описує всі елементи управління та способи роботи з ними.

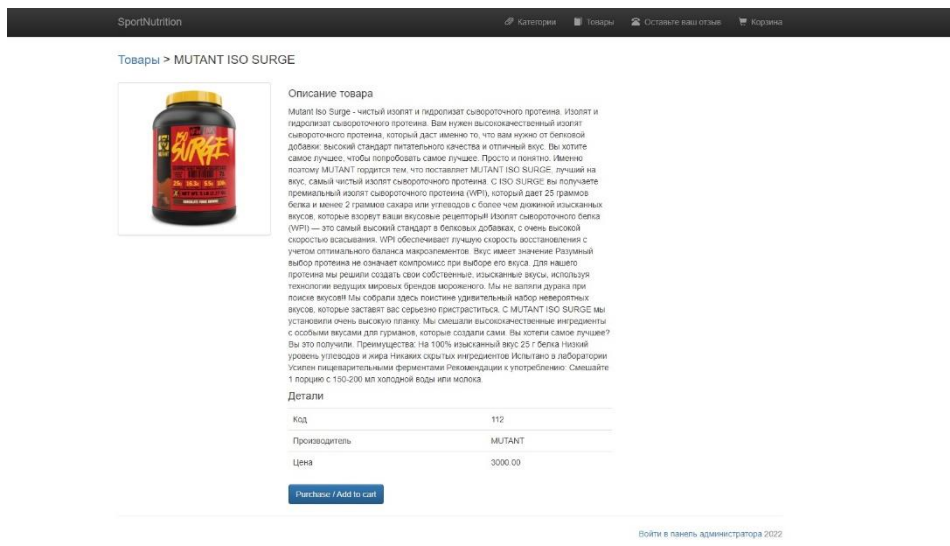


Рис. 2.11. рабочий экран програми

Экран програми (рис 2.11):

1. Header сторінки: у заголовку міститься назва сайту та меню для навігації між сторінками;
2. Основна зона: основна робоча зона вікна. У наведеному прикладі в ній розміщено банер товару та його опис, також присутня кнопка для збереження товару у кошику;
3. Footer сторінки: у футері розміщено посилання для переходу на сторінку адміністрування ресурсу.

Форма введення:

Форма введення (рис. 2.12) є елементом управління, який дає користувачеві можливість ввести інформацію. Це може бути будь-яка інформація – від імені користувача до завантаження банеру товару.

Код товара	<input type="text"/>
Название	<input type="text"/>
Фирма	<input type="text"/>
Банер	<input type="button" value="Выберите файл"/> Файл не выбран
Описание	<input type="text"/>
Цена	<input type="text"/>
Категория	<input type="text"/>
<input type="button" value="Добавить товар"/> <input type="button" value="Отмена"/>	

Рис. 2.12. форма введення

Редагування форми введення починається при виборі одного з полів цієї форми. Після заповнення форми користувач має можливість натиснути кнопку збереження, після чого програма виконає запрограмовану логіку.

При натисканні кнопки вибору користувач матиме змогу редагувати саме це поле. Навігація по елементах керування здійснюється кнопками "Вгору" та "Вниз", або за допомогою комп'ютерної миші.

Поля введення можуть трохи відрізнятися одна від одної. Крім цього можуть бути поля пароля та числові поля. Поля пароля відрізняються від простого текстового поля тим, що там під час редагування відображаються не введені символи, а знаки "*" для того, щоб пароль не можна було підглянути. Числові поля бувають як цілі, так і дробові. Вони дозволяють вам вводити лише цифри, а також (у дрібних полях) – десяткову точку.

Список (рис 2.13) є елементом управління, який дозволяє вибрати одне значення із запропонованих. Список може бути згорнутим та розгорнутим. Приклад використання списку наведено далі.

Список категорій

- 4 Протеїн
- 4 ВСАА
- 5 Креатин
- 3 Гейнер
- 4 Жиросжигатели
- 4 Трибулус
- Полный список товаров

Рис. 2.13. Список

2.6.4.1. Опис клієнтської частини

Для клієнтської частини було реалізовано сторінки (рис. 2.14, 2.15, 2.16, 2.17, 2.18, 2.19, 2.20, 2.21), необхідні для взаємодії клієнта з базою даних магазину. Функціонал сайту дає можливість клієнту ознайомитись з актуальними товарами на сайті магазину, додати сподобавшийся товар до кошика, придбати додані товари, сортувати товари за категоріями та залишати відгук про роботу порталу. Загальний вигляд сторінок наведено нижче. Код сторінок наведено у додатках.

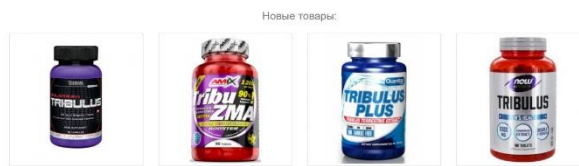
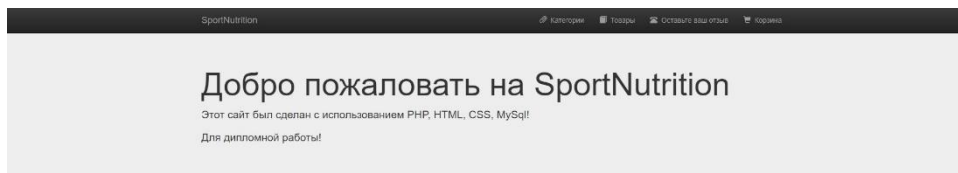


Рис. 2.14. Інтро сторінка

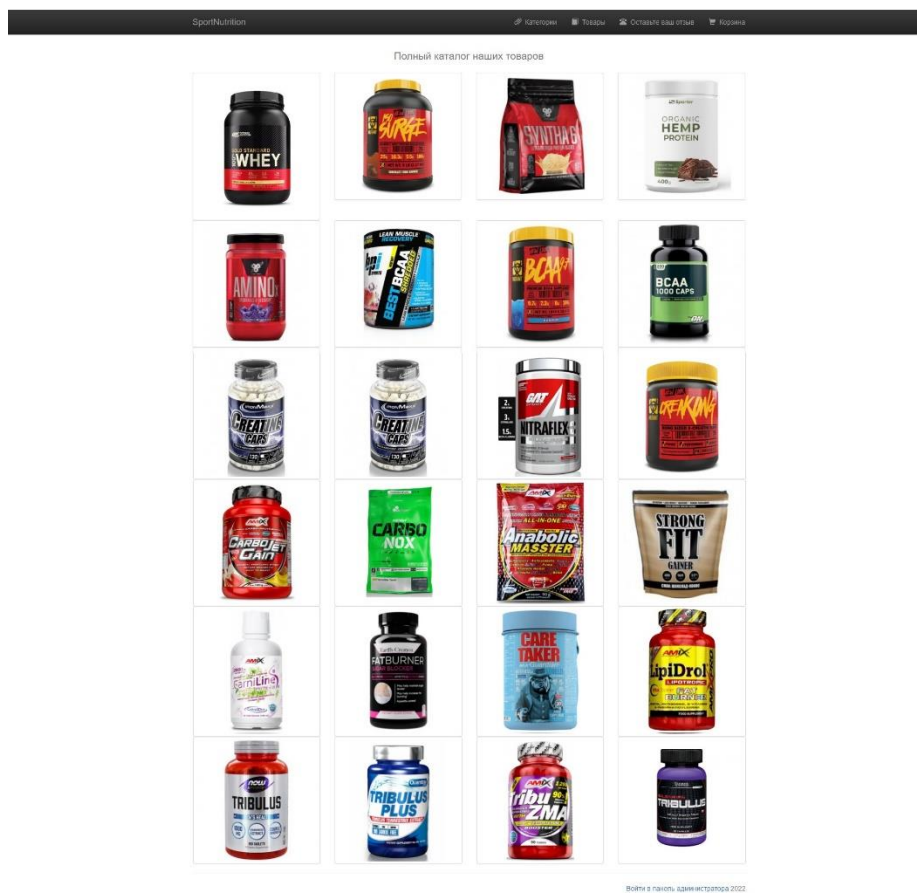


Рис. 2.15. Сторінка товарів

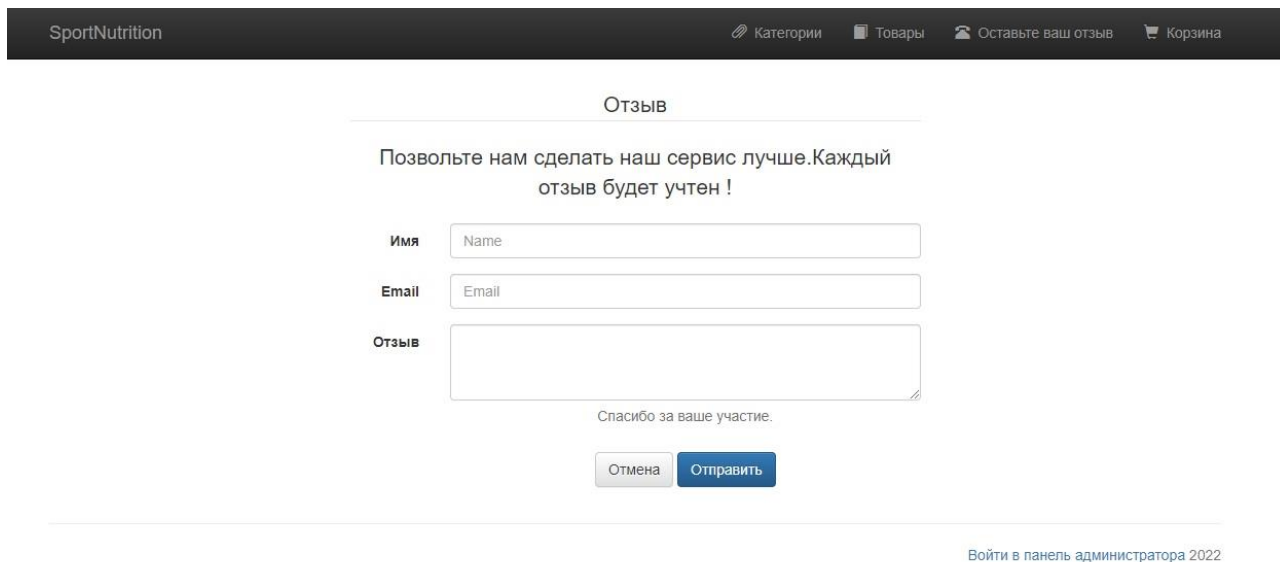


Рис. 2.16. Сторінка відгуків

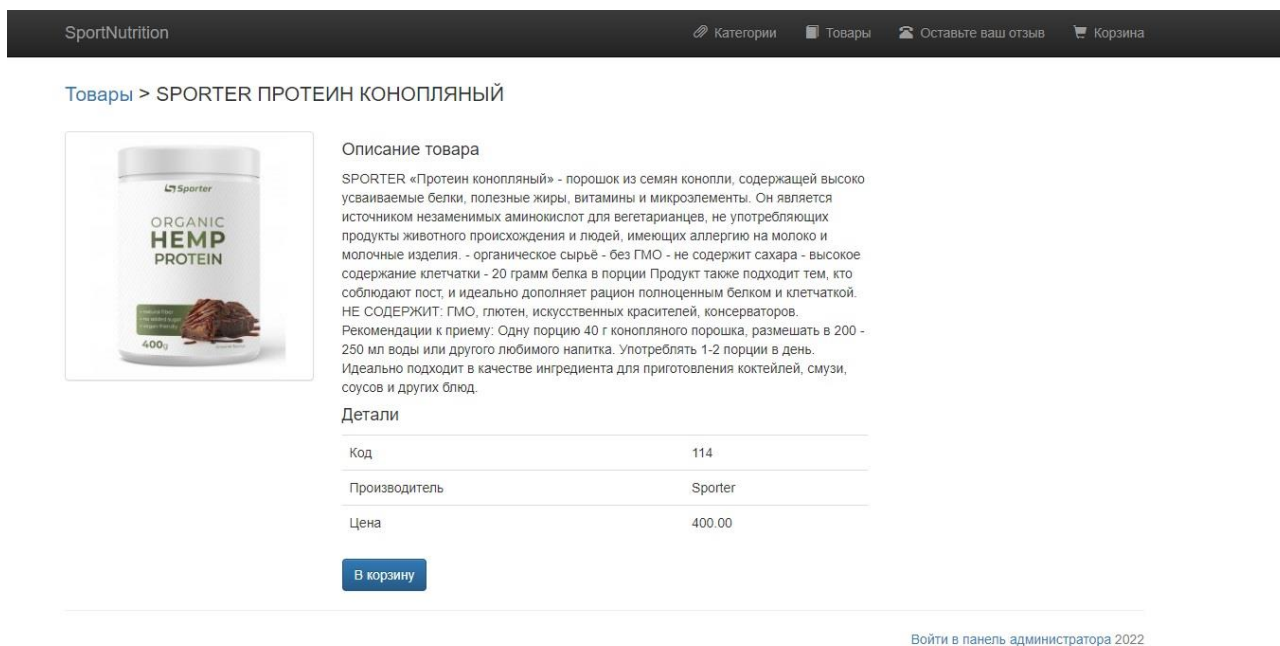


Рис. 2.17. Сторінка інформації про товар

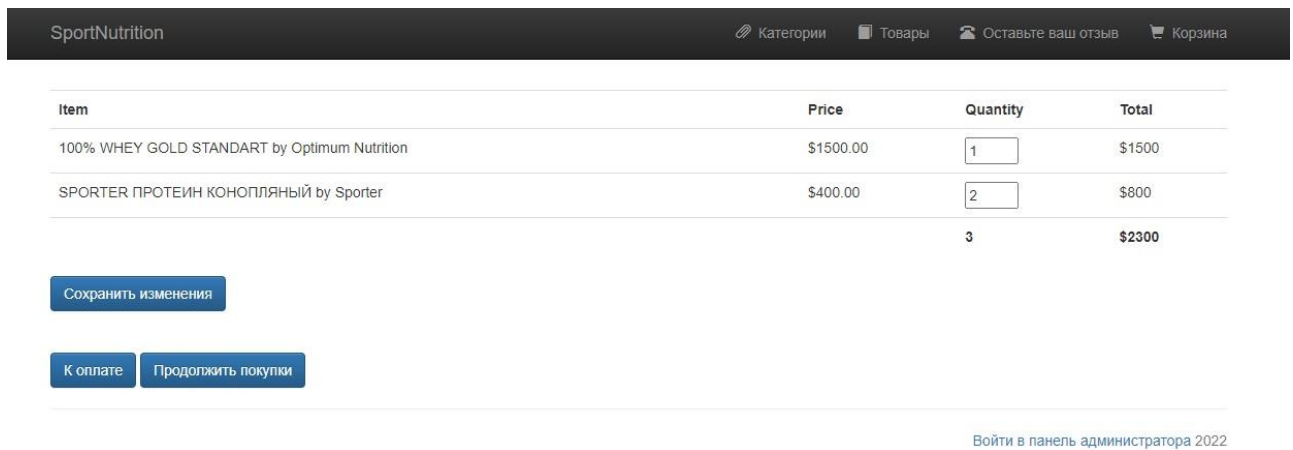


Рис. 2.18. Сторінка збережених користувачем товарів

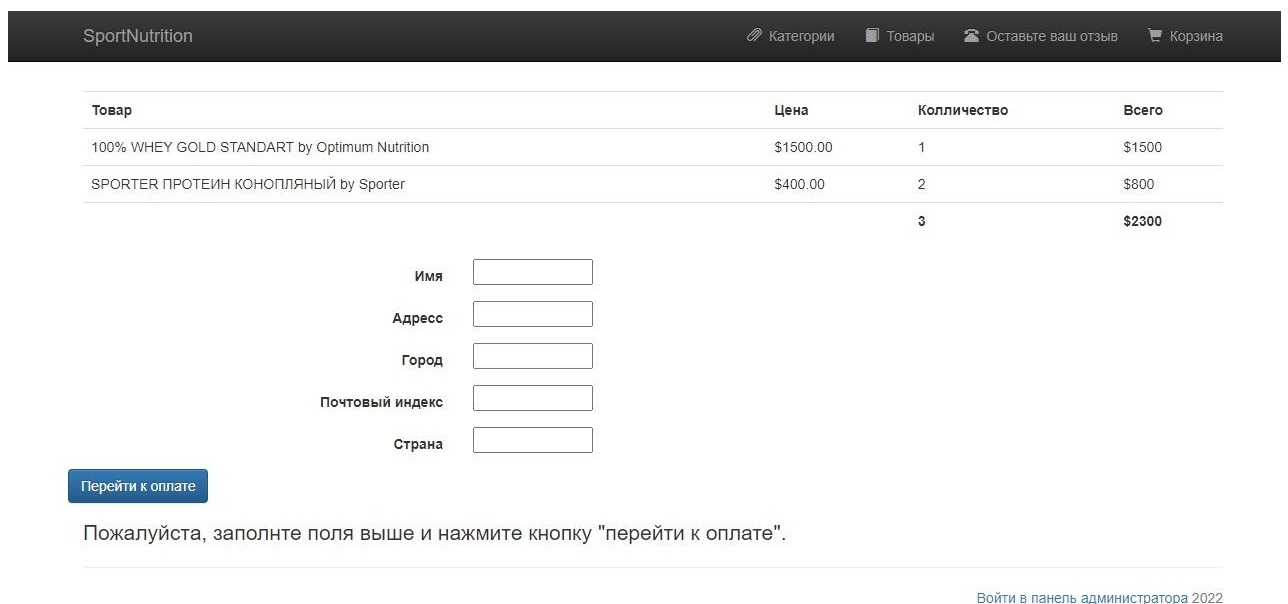


Рис. 2.19. Сторінка бронювання замовлення

SportNutrition Категории Товары Оставьте ваш отзыв Корзина

Товар	Цена	Количество	Всего
100% WHEY GOLD STANDART by Optimum Nutrition	\$1500.00	1	\$1500
SPORTER ПРОТЕИН КОНОПЛЯНЫЙ by Sporter	\$400.00	2	\$800
		3	\$2300
Доставка			20.00
Всего с доставкой			\$2320

Тип

Номер

CVV

Дата окончания

Name


Пожалуйста введите свои платежные данные и нажмите кнопку "Оплатить".

[Войти в панель администратора 2022](#)

Рис. 2.20. Сторінка оплати


SportNutrition Категории Товары Оставьте ваш отзыв Корзина

Категории > Креатин




QUAMTRAX KRE-ALKALYN 3000 - 120

[Детали](#)




IRONMAXX CREATIN - 130

[Детали](#)




GAT SPORT GAT NITRAFLEX+CREATINE 420 Г

[Детали](#)



MUTANT CREAKONG

[Детали](#)



OLIMP SPORT NUTRITION CARBO NOX

[Детали](#)

[Войти в панель администратора 2022](#)

Рис. 2.21. Сторінка категорії

2.6.4.2. Опис адміністрування системою

Для адміністрування магазину було реалізовано сторінки (рис. 2.22, 2.23, 2.24), необхідні для взаємодії адміністратора з базою даних магазину. Функціонал адмінпанелі дає можливість адміністратору бачити список усіх товарів, які присутні у базі даних магазину, редагувати чи видаляти ці дані, та додавати нові товари до бази даних. Загальний вигляд сторінок наведено нижче. Код сторінок наведено у додатках.



SportNutrition

Категории Товары Оставьте ваш отзыв Корзина

Логин

Пароль

Отправить

Войти в панель администратора 2022

Рис. 2.22. Сторінка входу в адмінпанель

Добавить новый товар

Выйти

Код	Название	Фирма	Банер	Описание	Цена	Категория		
134	ULTIMATE NUTRITION BULGARIAN TRIBULUS TERRESTRIS (90 КАП)	ULTIMATE NUTRITION	pGNC1-2332283dt-265x265.jpg	<p>Tribulus Terrestris - является абсолютно натуральной растительной альтернативой синтетическим гормонам-анаболическим. Повышенный уровень тестостерона способствует синтезу протеина и установлению положительного азотного баланса в организме. Для атлета это означает усиленный рост мышечных клеток и мышечной силы, а также более быстрое восстановление организма. Tribulus Terrestris от фирмы Ultimate Nutrition способен повысить уровень тестостерона более чем на 48% менее чем за 5 дней.</p> <p>месяца.</p>	446.00	Трибулус	Редактировать	Удалить
133	AMIX TRIBU-ZMA 1200MG - 90 ТАБ	AMIX	Amix Tribu-ZMA 1200.png	<p>Amix Tribu-ZMA сочетает в себе высокоэффективный растительный экстракт Tribulus Terrestris и запатентованный минеральный комплекс ZMA. Содержащийся натуральный экстракт Tribulus Terrestris имеет высокую концентрацию сапонинов (90%). Сапонины есть не что иное, как природные стимуляторы уровня тестостерона. Препарат одинаково эффективен и для мужчин и женщин для</p>	768.00	Трибулус	Редактировать	Удалить
130	AMIX AMIXPRO LIPIDROL FAT BURNER PLUS - 120	AMIX	AmixPro Lipidrol -265x265.png	<p>Amix LipiDrol Fat Burner это уникальный регулятор веса, содержащий усовершенствованную форму холина VitaCholine и комплекс других важных соединений, таких как инозитол, метионин и хорошо известный фосфолипид фосфатидилсерин. В сочетании с витаминами группы В играет ключевую роль в превращении жиров в энергию. VitaCholine это ведущий бренд холина, незаменимого питательного вещества на всех этапах</p>	463.00	Жиросжигатели	Редактировать	Удалить

Войти в панель администратора 2022

Рис. 2.23. Сторінка товарів з бази даних

SportNutrition [Категории](#) [Товары](#) [Оставьте ваш отзыв](#) [Корзина](#)

Код товара

Название

Фирма

Банер Файл не выбран

Описание

Цена

Категория

[Войти в панель администратора 2022](#)

Рис. 2.24. Сторінка додавання нового товару

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. Передбачуване число операторів програми - 500;
2. Коефіцієнт складності програми (1,25...2,0) - 1,4;
3. Коефіцієнт корекції програми в ході її розробки (0,05...0,1) - 0,05;
4. Годинна заробітна плата програміста - 60 грн/год;
5. Коефіцієнт збільшення витрат праці в наслідок недостатнього опису задачі (1,2...1,5) - 1,2;
6. Коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності (від 3-х до 5 років) – 1,1;
7. Вартість машино-години ЕОМ - 14 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_d, \text{ людино-годин, (3.1)}$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ - витрати праці на налагодження програми на ЕОМ;

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p),$$

де q - передбачуване число операторів програми ($q = 500$);

C - коефіцієнт складності програми ($C = 1,4$);

p - коефіцієнт корекції програми в ході її розробки ($p = 0,05$).

Звідси умовне число операторів в програмі:

$$Q = 500 \cdot 1,4 \cdot (1 + 0,05) = 735$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{75 \cdot 85 \cdot k}, \text{ люДИНО-ГОДИН,}$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 3 до 5 років він складає 1,1.

Приймемо збільшення витрат праці в наслідок недостатнього опису завдання не більше 50% ($B = 1,2$). З урахуванням коефіцієнта кваліфікації $k = 1,1$, отримуємо витрати праці на вивчення опису завдання:

$$t_u = (735 \cdot 1,2) / (80 \cdot 1,1) = 10,02 \text{ люДИНО-ГОДИН.}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20\dots25)*k}, \text{ люДИНО-ГОДИН,} \quad (3.2)$$

де Q – умовне число операторів програми;

k – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.2), отримаємо:

$$t_a = 735 / (21 \cdot 1,1) = 31,81 \text{ люДИНО-ГОДИН.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20\dots25)*k}, \text{ люДИНО-ГОДИН,}$$

$$t_n = 735 / (25 \cdot 1,1) = 26,72 \text{ люДИНО-ГОДИН.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4\dots5)*k}, \text{ люДИНО-ГОДИН.}$$

$$t_{отл} = 735 / (5 \cdot 1,1) = 133,63 \text{ люДИНО-ГОДИН.}$$

– за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 * t_{отл} , \text{ люДИНО-ГОДИН.}$$

$$t_{oml}^k = 1,5 \cdot 133,63 = 200,445 \text{ люДИНО-ГОДИН.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_d = t_{др} + t_{до} , \text{ люДИНО-ГОДИН,}$$

де $t_{др}$ – трудомісткість підготовки матеріалів і рукопису:

$$t_{др} = \frac{Q}{(15...20)*k} , \text{ люДИНО-ГОДИН.}$$

$t_{до}$ – трудомісткість редагування, печатки й оформлення документації:

$$t_{до} = 0,75 * t_d , \text{ люДИНО-ГОДИН.}$$

Підставляючи відповідні значення, отримаємо:

$$t_{др} = 735 / (18 \cdot 1,1) = 37,12 \text{ люДИНО-ГОДИН.}$$

$$t_{до} = 0,75 \cdot 37,12 = 27,84 \text{ люДИНО-ГОДИН.}$$

$$t_d = 37,12 + 27,84 = 64,96 \text{ люДИНО-ГОДИН.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 10,02 + 31,84 + 26,72 + 133,63 + 64,96 = 317,17 \text{ люДИНО-ГОДИН.}$$

3.2. Розхунок витрат на створення програми

Витрати на створення ПЗ $K_{ПО}$ включають витрати на заробітну плату виконавця програми $Z_{ЗП}$ і витрати машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ} , \text{ грн.}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t * C_{ПР} , \text{ грн.}$$

де: t - загальна трудомісткість, людино-годин;

$C_{ПР}$ - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 60 грн / год, отримуємо:

$$Z_{ЗП} = 317,17 \cdot 60 = 19030,2 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} * C_{мч} , \text{ грн.} \quad (3.3)$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (14 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$Z_{me} = 215,85 \cdot 14 = 3021,9 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{ПО} = 19030,2 + 3021,9 = 22052,1 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.}$$

де B_k - число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Звідси витрати на створення програмного продукту:

$$T = 317,17 / 1 \cdot 176 = 1,8 \text{ міс.}$$

Висновок

Програмне забезпечення призначене для реалізації веб-орієнтованої інформаційної системи з метою автоматизації діяльності компанії в сфері електронної комерції. Вартість даного програмного забезпечення становить 22052,1 грн. і не вимагає додаткових витрат при впровадженні та експлуатації програми. Очікуваний час розробки становить 1.8 місяці. Цей термін пов'язаний зі значним числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження програми та підготовку документації.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи була проведена робота з вивчення технологій розробки веб-сайтів з використанням технологій PHP, MySQL, HTML, CSS. За підсумками виконання роботи було розроблено інформаційну систему Web-застосунку інтернет магазину з продажу спортивного харчування.

Реалізовано функції користувача та адміністратора, що забезпечить можливостями перегляду основної інформації про товари компанії, сотрування товарів за категоріями, придбання товарів, написання відгуків з боку користувача та повного адміністрування порталу за допомогою панелі адміністратора.

Створена система дозволяє адміністратору керувати вмістом сайту, змінювати, видаляти та додавати інформацію до нього. При створенні системи були враховані всі плюси та мінуси систем-аналогів, вибраних для порівняння.

Розроблено всі необхідні діаграми для проектування та програмування системи та бази даних. Після реалізації системи обраними мовами програмування та середовищах, було проведено випробування на працездатність, яке засвідчило, що система виконуємо усі поставлені нами функції.

Оскільки порушень не виявлено, всі функції відповідають вимогам, тому веб-ресурс готовий до повної функціональної роботи, тобто система для впровадження, яка може бути використана бізнесом підвищення клієнтообігу.

У ході роботи були виконано такі завдання:

- аналіз предметної області для вибору найкращих інструментів і вирішення поставленого завдання;
- вивчення необхідних матеріалів для роботи з мовами програмування;
- вивчення необхідних матеріалів для роботи з базою даних;
- розробка блок схеми принципу роботи інформаційної системи;
- побудова архітектури інтернет-додатки;

- розробка дизайну веб-програми;
- створення інтерфейсу користувача веб-додатку;
- створення бекенд частини програми для взаємодії з базою даних;
- тестування системи;
- аналіз результатів роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дронов А. В. «PHP, MySQL и Dreamweaver MX 2004 - Разработка интерактивных Web-сайтов». СПб: БХВ-Петербург, 2005, 448 с.
2. L. Welling, L. Thomson. « PHP and MySQL Web Development (4th Edition)». Addison-Wesley Professional, 2008, 848 с.
3. Колісниченко Д.М. «Joomla 1.5. Руководство пользователя». СПб.: БХВ-Петербург, 2009, 224 с.
4. Колісниченко Д.М. «Двигун для вашого сайту. CMS Joomla! Slaed, PHP-Nuke». СПб.: БХВ-Петербург, 2008, 368 с.
5. Норт Берри. «Joomla! Практическое руководство». СПб.: СимволПлюс, 2008, 448 с.
6. Ден Рамел. «Самоучитель Joomla!». БХВ-Петербург, 2008, 448 с.
7. Хаген Граф. «Создание веб-сайтов с помощью Joomla! 1.5». Издавничий будинок "Вільямс". 2009, 304 с.
8. David Lane, Hugh E. Williams. «Web DB Applications with PHP & MySQL, 2nd Edition». O'Reilly Media, 2004, 680с.
9. Котеров Д. В. «PHP7 в подлиннике». СПб.: БХВ-Петербург, 2016, 1069 с.
10. MySQL 4 – Строкові функції. URL: <http://www.codenet.ru/db/mysql/mystring4>. дата звернення: 08.05.2022.
11. Web-розробка. URL: <http://fcit.tneu.org/web-rozrobka/>. дата звернення: 08.05.2022.
12. Астахова І.Ф., Толстобров А.П., Мельников В.М. «СУБД. Язык SQL в примерах и задачах». Воронеж, 2001, 160 с.
13. Бурсов М.В., Миколаїв Д.Г., Гаврилін Д.А. «Основы работы с HTML редактором Dreamweaver». Навчально-методичний посібник. Санкт-Петербург: Издавництво ГТМТО, 2002 – 101 с. / Режим доступа: <https://vdoc.pub/download/html-dreamweaver-65m9gh5mqcm0>. дата звернення: 08.05.2022.

14. Хансен Г., Хансен Д. «Базы данных. Разработка и управление». М: Бином, 2010, 704 с.
15. Маклафлин Б. «PHP и MySQL, исчерпывающее руководство». СПб.: Питер, 2014, 544с.
16. Види тестування ПО. Функціональне тестування. URL: <http://qlearning.com.ua/theory/lectures/material/testing-types-functional/>. дата звернення: 08.05.2022.
17. Перенесення файлів із localhost на сервер. URL: <http://joomlaportal.ru/faq/installation-and-update/51-perenos-sajta-slocalhost-na-server>. дата звернення: 08.05.2022.
18. Архітектура клієнт-сервер. URL: <http://inter.ptngu.com/kompyuterni-merezhi/arhitektura-kliyant-server>. дата звернення: 08.05.2022.

ЛІСТИНГ ПРОГРАМИ

Лістинг файлу Index.php:

```

<?php
    session_start();
    $count = 0;
    // conno DB

    $title = "Index";
    require_once "./tmp/header.php";
    require_once "./functions/DB_functions.php";
    $conn = db_conn();
    $row = select4LatestItem($conn);
?>
    <!-- Example row of columns -->
    <p class="lead text-center text-muted">Новые товары:</p>
    <div class="row">
        <?php foreach($row as $item) { ?>
        <div class="col-md-3">
            <a href="item.php?itemibn=<?php echo $item['item_ibn']; ?>">
                
            </a>
        </div>
        <?php } ?>
    </div>
<?php
    if(isset($conn)) { mysqli_close($conn); }
    require_once "./tmp/footer.php";
?>

```

Лістинг файлу Admin.php:

```

<?php
    $title = "Administration section";
    require_once "./tmp/header.php";
?>

<form class="form-horizontal" method="post" action="admin_verify.php">
    <div class="form-group">
        <label for="name" class="control-label col-md-4">Логин</label>
        <div class="col-md-4">
            <input type="text" name="name" class="form-control">

```

```

        </div>
    </div>
    <div class="form-group">
        <label for="pass" class="control-label col-md-4">Пароль</label>
        <div class="col-md-4">
            <input type="password" name="pass" class="form-control">
        </div>
    </div>
    <input type="submit" name="submit" class="btn btn-primary">
</form>

<?php
    require_once "./tmp/footer.php";
?>

```

Лістинг файлу Admin_verify.php:

```

<?php
    session_start();
    if(!isset($_POST['submit'])){
        echo "Warning! Check again!";
        exit;
    }
    require_once "./functions/DB_functions.php";
    $conn = db_conn();

    $name = trim($_POST['name']);
    $pass = trim($_POST['pass']);

    if($name == "" || $pass == ""){
        echo "Name or Pass is empty!";
        exit;
    }

    $name = mysqli_real_escape_string($conn, $name);
    $pass = mysqli_real_escape_string($conn, $pass);
    $pass = sha1($pass);

    // get from db
    $query = "SELECT name, pass from admin";
    $result = mysqli_query($conn, $query);
    if(!$result){
        echo "Empty data " . mysqli_error($conn);
        exit;
    }
    $row = mysqli_fetch_assoc($result);

    if($name != $row['name'] && $pass != $row['pass']){
        echo "Name or pass is warning. Check again!";
        $_SESSION['admin'] = false;
        exit;
    }

```

```

}

if(isset($conn)) {mysqli_close($conn);}
$_SESSION['admin'] = true;
header("Location: admin_item.php");
?>

```

Лістинг файлу Good.php:

```

<?php
session_start();
$item_ibn = $_GET['itemibn'];
// DB conn
require_once "./function/DB_functtion.php";
$conn = db_conn();

$query = "SELECT * FROM items WHERE item_ibn = '$item_ibn'";
$result = mysqli_query($conn, $query);
if(!$result){
    echo "Can't get data " . mysqli_error($conn);
    exit;
}

$row = mysqli_fetch_assoc($result);
if(!$row){
    echo "Empty item";
    exit;
}

$title = $row['item_title'];
require "./tmp/header.php";
?>

<!-- Example clmns -->
<p class="lead" style="margin: 25px 0"><a href="boooks.php">Товары </a> <?php echo $row['item_title'];
?></p>
<div class="row">
    <div class="col-md-3 text-center">
        
    </div>
    <div class="col-md-6">
        <h4>Описание товара</h4>
        <p><?php echo $row['item_descr']; ?></p>
        <h4>Детали</h4>
        <table class="table">
            <?php foreach($row as $key => $value){
                if($key == "item_descr" || $key == "item_image" || $key == "itemid" || $key == "item_title"){
                    continue;
                }
                switch($key){
                    case "item_ibn":
                        $key = "Код";
                        break;
                    case "item_title":

```

```

        $key = "Название";
        break;
    case "item_author":
        $key = "Производитель";
        break;
    case "item_price":
        $key = "Цена";
        break;
    }
?>
<tr>
    <td><?php echo $key; ?></td>
    <td><?php echo $value; ?></td>
</tr>
<?php
    }
    if(isset($conn)) { mysqli_close($conn); }
?>
</table>
<form method="post" action="dray.php">
    <input type="hidden" name="itemibn" value="<?php echo $item_ibn;?>">
    <input type="submit" value="В корзину" name="dray" class="btn btn-primary">
</form>
</div>
</div>
<?php
    require "./tmp/footer.php";
?>

```

Лістинг файлу Goods.php:

```

<?php
    session_start();
    $count = 0;
    // conn to Db
    require_once "./functions/DB_funcntions.php";
    $conn = db_conn();

    $query = "SELECT item_ibn, item_image FROM items";
    $result = mysqli_query($conn, $query);
    if(!$result){
        echo "Can't retrieve data " . mysqli_error($conn);
        exit;
    }

    $title = "Full Catalogs of Goods";
    require_once "./tmp/header.php";
?>
<p class="lead text-center text-muted">Полный каталог наших товаров</p>
<?php for($i = 0; $i < mysqli_num_rows($result); $i++){ ?>
    <div class="row">
        <?php while($query_row = mysqli_fetch_assoc($result)){ ?>

```

```

        <div class="col-md-3">
            <a href="item.php?itemibn=<?php echo $query_row['item_ibn']; ?>">
                
            </a>
        </div>
    <?php
        $count++;
        if($count >= 4){
            $count = 0;
            break;
        }
    } ?>
</div>
<?php
}
if(isset($conn)) { mysqli_close($conn); }
require_once "../tmp/footer.php";
?>

```

Лістинг файлу Dray.php:

```

<?php
// sh dray needs start
/*
    Array of session(
        dray => array (
            item_ibn (get from $_POST['item_ibn']) => number of items
        ),
        items => 0,
        full_price => '0.00'
    )
*/
session_start();
require_once "../functions/DB_functions.php";
require_once "../functions/dray_functions.php";

// item_ibn got from form post method, change this place later.
if(isset($_POST['itemibn'])){
    $item_ibn = $_POST['itemibn'];
}

if(isset($item_ibn)){
    // new item selected
    if(!isset($_SESSION['dray'])){
        // $_SESSION['dray'] is associative array that itemibn => qty
        $_SESSION['dray'] = array();

        $_SESSION['full_items'] = 0;
        $_SESSION['full_price'] = '0.00';
    }
}

```



```

if(!isset($_SESSION['dray'][$item_ibn])){
    $_SESSION['dray'][$item_ibn] = 1;
} elseif(isset($_POST['dray'])){
    $_SESSION['dray'][$item_ibn]++;
    unset($_POST);
}
}

// if save change button is clicked , change the qty of each itemibn
if(isset($_POST['save_change'])){
    foreach($_SESSION['dray'] as $ibn => $qty){
        if($_POST[$ibn] == '0'){
            unset($_SESSION['dray'][$ibn]);
        } else {
            $_SESSION['dray'][$ibn] = $_POST[$ibn];
        }
    }
}

// print out header here
$title = "Your shopping dray";
require "../tmp/header.php";

if(isset($_SESSION['dray']) && (array_count_values($_SESSION['dray']))) {
    $_SESSION['full_price'] = full_price($_SESSION['dray']);
    $_SESSION['full_items'] = full_items($_SESSION['dray']);
?>
<form action="dray.php" method="post">
    <table class="table">
        <tr>
            <th>Item</th>
            <th>Price</th>
            <th>Quantity</th>
            <th>Full</th>
        </tr>
        <?php
        foreach($_SESSION['dray'] as $ibn => $qty){
            $conn = db_conn();
            $item = mysqli_fetch_assoc(getItemByIbn($conn, $ibn));
?>
        <tr>
            <td><?php echo $item['item_title'] . " by " . $item['item_author']; ?></td>
            <td><?php echo "$" . $item['item_price']; ?></td>
            <td><input type="text" value="<?php echo $qty; ?>" size="2" name="<?php echo $ibn; ?>"></td>
            <td><?php echo "$" . $qty * $item['item_price']; ?></td>
        </tr>
        <?php } ?>
        <tr>
            <th>&nbsp;</th>
            <th>&nbsp;</th>
            <th><?php echo $_SESSION['full_items']; ?></th>
            <th><?php echo "$" . $_SESSION['full_price']; ?></th>
        </tr>
    </table>

```

```

        <input type="submit" class="btn btn-primary" name="save_change" value="Сохранить изменения">
    </form>
    <br/><br/>
    <a href="checkout.php" class="btn btn-primary">К оплате</a>
    <a href="items.php" class="btn btn-primary">Продолжить покупки</a>
<?php
    } else {
        echo "<p class='text-error'>Your dray is empty! Please make sure you add some goods in it!</p>";
    }
    if(isset($conn)){ mysqli_close($conn); }
    require_once "./tmp/footer.php";
?>

```

Лістинг файлу Process.php:

```

<?php
    session_start();

    $_SESSION['err'] = 1;
    foreach($_POST as $key => $value){
        if(trim($value) == ""){
            $_SESSION['err'] = 0;
        }
        break;
    }

    if($_SESSION['err'] == 0){
        header("Location: purchase.php");
    } else {
        unset($_SESSION['err']);
    }

    require_once "./functtions/DB_functtions.php";
    // print out header here
    $title = "Purchase Process";
    require "./tmp/header.php";
    // conn DB
    $conn = db_conn();
    extract($_SESSION['ship']);

    // validate post section
    $credit_card_number = $_POST['credit_card_number'];
    $credit_card_PID = $_POST['credit_card_PID'];
    $credit_card_expire_date = strtotime($_POST['credit_card_expire_date']);
    $credit_card_owner = $_POST['credit_card_owner'];

    // find user
    $userid = getUserId($name, $address, $city, $zip_code, $country);
    if($userid == null) {
        // insert user into DB and return
        $userid = setUserId($name, $address, $city, $zip_code, $country);
    }

```

```

$date = date("Y-m-d H:i:s");
insertIntoOrder($conn, $userid, $_SESSION['full_price'], $date, $name, $address, $city, $zip_code, $country);

// take orderid from order to insert order items
$orderid = getOrderid($conn, $userid);

foreach($_SESSION['dray'] as $ibn => $qty){
    $itemprice = getitemprice($ibn);
    $query = "INSERT INTO order_items VALUES
    ($orderid, '$ibn', '$itemprice', '$qty')";
    $result = mysqli_query($conn, $query);
    if(!$result){
        echo "Insert value false!" . mysqli_error($conn2);
        exit;
    }
}

session_unset();
?>
<p class="lead text-success">Your order has been processed successfully. Please check your email to get your order
confirmation and shipping detail!.
Your dray has been empty.</p>

<?php
if(isset($conn)){
    mysqli_close($conn);
}
require_once "./tmp/footer.php";
?>

```

Лістинг файлу Purchase.php:

```

<?php
session_start();
$_SESSION['err'] = 1;
foreach($_POST as $key => $value){
    if(trim($value) == ""){
        $_SESSION['err'] = 0;
    }
    break;
}

if($_SESSION['err'] == 0){
    header("Location: checkout.php");
} else {
    unset($_SESSION['err']);
}

$_SESSION['ship'] = array();
foreach($_POST as $key => $value){
    if($key != "submit"){

```

```

        $_SESSION['ship'][$key] = $value;
    }
}
require_once "../functions/DB_functions.php";
// print out header here
$title = "Purchase";
require "../tmp/header.php";
// conn DB
if(isset($_SESSION['dray']) && (array_count_values($_SESSION['dray']))) {
?>
<table class="table">
    <tr>
        <th>Товар</th>
        <th>Цена</th>
        <th>Количество</th>
        <th>Всего</th>
    </tr>
    <?php
        foreach($_SESSION['dray'] as $ibn => $qty){
            $conn = db_conn();
            $item = mysqli_fetch_assoc(getItemByIbn($conn, $ibn));
        ?>
    <tr>
        <td><?php echo $item['item_title'] . " by " . $item['item_author']; ?></td>
        <td><?php echo "$" . $item['item_price']; ?></td>
        <td><?php echo $qty; ?></td>
        <td><?php echo "$" . $qty * $item['item_price']; ?></td>
    </tr>
    <?php } ?>
    <tr>
        <th>&nbsp;</th>
        <th>&nbsp;</th>
        <th><?php echo $_SESSION['full_items']; ?></th>
        <th><?php echo "$" . $_SESSION['full_price']; ?></th>
    </tr>
    <tr>
        <td>Доставка</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>20.00</td>
    </tr>
    <tr>
        <th>Всего с доставкой</th>
        <th>&nbsp;</th>
        <th>&nbsp;</th>
        <th><?php echo "$" . ($_SESSION['full_price'] + 20); ?></th>
    </tr>
</table>
<form method="post" action="process.php" class="form-horizontal">
    <?php if(isset($_SESSION['err']) && $_SESSION['err'] == 1){ ?>
    <p class="text-danger">All fields have to be filled</p>
    <?php } ?>
    <div class="form-group">
        <label for="credit_card_type" class="col-lg-2 control-label">Тип</label>

```

```

<div class="col-lg-10">
  <select class="form-control" name="credit_card_type">
    <option value="VISA">VISA</option>
    <option value="MasterCredit_card">MasterCredit_card</option>
    <option value="American Express">American Express</option>
  </select>
</div>
</div>
<div class="form-group">
  <label for="credit_card_number" class="col-lg-2 control-label">Номер</label>
  <div class="col-lg-10">
    <input type="text" class="form-control" name="credit_card_number">
  </div>
</div>
<div class="form-group">
  <label for="credit_card_PID" class="col-lg-2 control-label">CVV</label>
  <div class="col-lg-10">
    <input type="text" class="form-control" name="credit_card_PID">
  </div>
</div>
<div class="form-group">
  <label for="credit_card_expire_date" class="col-lg-2 control-label">Дата окончания</label>
  <div class="col-lg-10">
    <input type="date" name="credit_card_expire_date" class="form-control">
  </div>
</div>
<div class="form-group">
  <label for="credit_card_owner" class="col-lg-2 control-label">Name</label>
  <div class="col-lg-10">
    <input type="text" class="form-control" name="credit_card_owner">
  </div>
</div>
<div class="form-group">
  <div class="col-lg-10 col-lg-offset-2">
    <button type="reset" class="btn btn-default">Отмена</button>
    <button type="submit" class="btn btn-primary">Оплатить</button>
  </div>
</div>
</form>
<p class="lead">Пожалуйста введите свои платежные данные и нажмите кнопку "Оплатить".</p>
<?php
} else {
  echo "<p class='text-error'>Your dray is empty! Please make sure you add some items in it!</p>";
}
if(isset($conn)){ mysqli_close($conn); }
require_once "./tmp/footer.php"; ?>

```


ПЕРЕЛІК ДОКУМЕНТІВ НА ДИСКУ

Ім'я файла	Опис
Пояснювальні документи	
Диплом_Толстой.docx	Пояснювальна записка до дипломного проекту. Документ Word.
Диплом_Толстой.pdf	Пояснювальна записка до дипломного проекту в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація_Толстой.pptx	Презентація дипломного проекту