

Державний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Програмна реалізація Web-додатку визначення вартості
доставки логістичної компанії на основі Java»**

Студента 2 курсу, 3мз групи
спеціальності
122 «Комп'ютерні науки»

Тебенькова
Катерина
Олексіївна

підпис студента

Науковий керівник
доктор технічних наук, професор

Краскевич Валерій
Євгенович

підпис керівника

Гарант освітньої програми
доктор фізико-математичних наук,
професор

Пурський Олег
Іванович

підпис керівника

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Комп'ютерні науки»

Зав. кафедри _____ **Затверджую**
Пурський О.І.
«22» грудня 2022р.

Завдання на випускню кваліфікаційну роботу студенту

Тебенькової Катерини Олексіївни

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи
«Програмна реалізація Web-додатку визначення вартості доставки логістичної компанії на основі Java»
Затверджена наказом ректора від «20» грудня 2022 р. № 3550
2. Строк здачі студентом закінченої роботи 24 листопада 2023 року
3. Цільова установка та вихідні дані до роботи
Мета роботи: розробка сервісу, який допоможе компаніям з доставками продукції.
Об'єкт дослідження: створення сервісу з метою допомоги у доставці компаній.
Предмет дослідження: сервіси, методи та інформаційні технології для розробки.
4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Краскевич В.Є.	22.12.2022 р.	22.12.2022 р.
2	Краскевич В.Є.	22.12.2022 р.	22.12.2022 р.
3	Краскевич В.Є.	22.12.2021 р.	22.12.2022 р.

6. Зміст випускного кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. Огляд рішень та загальних понять

1.1. Структур

1.1.1 Загальні поняття

1.1.2 Види веб-додатків

1.1.3 Різниця між веб-додатком та сайтом

1.2 Огляд існуючих рішень

1.2.1 Сервіси

1.2.2 Bolt Food

РОЗДІЛ 2. Функціонал для роботи над сервісом

2.1. HTML

2.2. CSS

2.3. JavaScript

2.4 PHP

2.5 База даних MySQL

РОЗДІЛ 3. Розробка та огляд середовища розробки

3.1. Опис застосунку IntelliJ IDEA

3.2. Підключення до бази даних

3.3. Робота з базою даних

3.4 Розгляд форми реєстрації, авторизації

3.5 Модель

3.6 Використання CSS

3.7 Використання Java

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	01.11.2022	01.11.2022
2	Розробка та затвердження завдання на випускну кваліфікаційну роботу	09.12.2022	09.12.2022
3	Вступ	01.05.2023	01.05.2023
4	РОЗДІЛ 1. Огляд рішень та загальних понять	14.06.2023	14.06.2023
5	Підготовка статті у збірник наукових статей магістрів	20.06.2023	20.06.2023
6	РОЗДІЛ 2. Функціонал для роботи над сервісом	08.09.2023	08.09.2023
7	РОЗДІЛ 3. Розробка та огляд середовища розробки	20.10.2023	20.10.2023
8	Висновки	02.11.2023	02.11.2023
9	Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику	22.11.2023	22.11.2023
10	Попередній захист випускної кваліфікаційної роботи	29.11.2023	29.11.2023
11	Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи	04.12.2023	04.12.2023
12	Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі	06.12.2023	06.12.2023
13	Публічний захист випускної кваліфікаційної роботи	За розкладом роботи ЕК	

8. Дата видачі завдання «9» грудня 2022 р.

9. Керівник випускного кваліфікаційного проекту Краскевич В.Є.
(прізвище, ініціали, підпис)

10. Гарант освітньої програми Пурський О. І.
(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент Тебенькова К.О.
(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи

Випускна кваліфікаційна робота студентки Тебенькової Катерини Олексіївни виконана відповідно до затвердженої теми «Програмна реалізація Web-додатку визначення вартості доставки логістичної компанії на основі Java».

Актуальність дослідження обумовлена потребою в розробці програмного забезпечення, що допоможе в найкоротші терміни визначити вартість доставки вантажу у визначене місце з урахуванням потреби користувачів веб додатку.

За основу роботи було обрано мови веб програмування, що дають можливість розробити інтернет-додаток та не прив'язуватись до окремих пристроїв. Зміна семантики та функціоналу в додатку спрощується. За рахунок розміщення на сервері сайту, і не потребує окремого оновлення на пристроях користувачів.

За результатами дослідження було розроблено Web-додаток визначення вартості доставки логістичної компанії на основі Java.

Студентка Тебенькова Катерина Олексіївна вміє аналізувати теоретичний та практичний матеріал, працювати з літературними джерелами, приймати аналітичні рішення та застосовувати сучасні комп'ютерні технології.

Випускна кваліфікаційна робота відповідає вимогам та може бути представлена до захисту.

Керівник випускної кваліфікаційної роботи

(підпис, дата)

13. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента _____
(прізвище, ініціали)
може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми _____ Демідов П.Г.
(підпис, прізвище, ініціали)

Завідувач кафедри _____ Пурський О.І.
(підпис, прізвище, ініціали)

« _____ » _____ 2023 р.

Анотація

У цій роботі був розглянутий веб-сервіс для визначення вартості доставки товарів з актуальністю в сучасному світі. Можна побачити порівняння різних організацій які були покладені в основу розробки цього проекту. Додаток дозволяє проаналізувати процес доставку товарів. Була розглянута одна з найпопулярніших мов програмування - Java. За мету було покладена розробка сервісу, який допоможе компаніям з доставкою товарів.

Ключові поняття: HML, CSS, Java, SQL



Anotation

There is a web service for determining the cost of delivery of goods with relevance in the modern world. You can see a comparison of different organizations that were the basis for the development of this project. The application allows you to analyze the process of delivery of goods. One of the most popular programming languages, Java, was considered. The goal was to develop a service that would help companies deliver goods.

Key concepts: HML, CSS, Java, SQL

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 Огляд рішень та загальних понять	10
1.1 Структура	10
1.1.1 Загальні поняття	10
1.1.2 Види веб-додатків	11
1.1.3 Різниця між веб додатком та сайтом.....	12
1.2 Огляд існуючих рішень	14
1.2.1 Сервіси.....	14
1.2.2 Bolt Food.....	15
ВИСНОВОК ДО РОЗДІЛУ	16
РОЗДІЛ 2 Функціонал для роботи над сервісом	17
2.1 HTML.....	17
2.2. CSS.....	19
2.3. JavaScript	20
2.4 PHP.....	21
2.5 База даних MySQL	24
ВИСНОВОК ДО РОЗДІЛУ	27
РОЗДІЛ 3 Розробка та огляд середовища розробки	28
3.1 Опис застосунку IntelliJ IDEA	28
3.2 Підключення до бази даних	29
3.3 Робота з базою даних	30
3.4 Розгляд форми реєстрації, авторизації.....	30
3.5 Модель.....	33

3.6 Використання CSS	33
3.7 Використання Java	35
3.8 Формула для визначення вартості доставки.....	36
ВИСНОВОК ДО РОЗДІЛУ	38
ВИСНОВКИ	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	40
ДОДАТОК	43



ВСТУП

На теперішній час в сучасному суспільстві з'являється все більше нових, небажаних але неминучих глобальних проблем. Серед них була пандемія, яка сталась на початку 2020 року у всьому світі. Через це були започатковані вимушені обмеження щодо громадян. Виходячи з таких умов, своєї найбільшої популярності набули такі сервіси як онлайн-магазини, служби доставки, інтернет-зв'язок.

У сучасному світі логістика є невід'ємною складовою бізнесу, що потребує постійної оптимізації та удосконалення процесів. Однією з ключових галузей логістики є доставка товарів, яка забезпечує необхідний рівень послуг для клієнтів. У зв'язку з цим, розвиток програмних засобів для визначення вартості доставки стає все більш актуальним завданням.

У даному дипломному проекті буде розглянуто програмну реалізацію web додатку для визначення вартості доставки логістичної компанії на основі мови програмування Java. Додаток буде розроблено з урахуванням вимог сучасних стандартів безпеки та ефективності роботи. Так як поширеність такого виду - сервісів розпочалася відносно нещодавно, з'являється все більше й більше нових конкурентних рішень в цьому напрямку, які спроможні витіснити інших з ринку. Щоб стати поперед них нам необхідно проаналізувати всі переваги та недоліки схожих сервісів, робити свої висновки та реалізовувати нові нестандартні задумки.

Основними завданнями дипломного проекту є розробка web-інтерфейсу користувача, реалізація алгоритмів визначення доставки, інтеграція з базою даних та іншими додатками логістичної компанії.

Отримання нових знань та навичок в розробці програмного забезпечення та застосування їх у сфері логістики, буде важливою складовою моєї подальшої професійної діяльності.

РОЗДІЛ 1

Огляд рішень та загальних понять

1.1 Структура

1.1.1 Загальні поняття

Веб-додаток – це програма або програмне забезпечення, яка може бути запущена в будь-якому веб-браузері. Зовнішній інтерфейс веб-програми створюється за допомогою мов програмування, таких як HTML, CSS і JavaScript, які підтримуються всіма популярними браузерами, такими як Opera, Chrome, Mozilla, Yandex. Одночасно, для написання серверної сторони (back-end) можуть використовуватися різні мови програмування або фреймворки, такі як Python, PHP, Ruby, Java. [27]

Основні переваги веб-додатків:

- веб-додатки можуть запускатися на будь-якій операційній системі (Linux, Mac, Windows), оскільки всі вони підтримують сучасні браузери;
- завдяки використанню одного коду, веб-додатки набагато легше і дешевше підтримувати порівняно з десктоп-додатками;
- програмування веб-додатків спрощено, оскільки не вимагає багато роботи з елементами комп'ютера, такими як ядро, процесор, відеокарта;
- відмінною особливістю веб-додатків є те, що для їх розповсюдження не потрібно одержувати затвердження від будь-яких платформ;
- веб-додатки є більш економічним варіантом для підприємств, так як не потребують підписок або придбання ліцензій, а можуть бути використані як послуги SaaS, що значно дешевше.

Опис роботи веб-додатка: користувач створює запит, який потрібно відправити на веб-сервер, використовуючи веб-браузер або в самому веб-додатку. Потім цей

запит направляється на веб-сервер, пов'язаний з веб-додатком. Після отримання запиту сервер звертається до бази даних, щоб сформувати відповідь на запит користувача.

Завершальним етапом є відправлення необхідних даних на фронт-частину програми, що відображається користувачеві. Цей процес повторюється знову і знову, поки користувач створює запити в веб-додатку. Варто зауважити, що чим складніші і більші запити, тим триваліше буде передача відповіді з сервера.[28]

1.1.2 Види веб-додатків

Існують такі види веб-додатків:

- SPA

(Single Page Application) – односторінковий інтерактивний додаток. Він не тільки працює на одній сторінці, а й є інтерактивним, подібно до повноцінної програми. Не слід плутати односторінковий веб-сайт з SPA, оскільки в першому випадку може відсутній інтерактив. У SPA, користувач, перемикаючись між вкладками, залишається на одній сторінці. Лише необхідні частини контенту підвантажуються та оновлюються, що забезпечує високу швидкість. Gmail є прикладом односторінкової програми, де адреса сторінки не змінюється, коли користувач перемикається між списками повідомлень. [28]

SPA зазвичай створюють на JavaScript. Використання бібліотеки jQuery може бути доцільним для створення невеликих односторінкових додатків, проте для великих проектів краще використовувати фреймворки, наприклад Vue, React або Angular.

- MPA

MPA (Multi Page Application) описує традиційні багатосторінкові веб-програми, де нові HTTP-сторінки завантажуються при взаємодії користувача з веб-сайтом. Це сповільнює обмін даними порівняно з SPA, особливо, якщо існують проблеми

з інтернет-з'єднанням чи хостингом веб-сайту. Прикладами МРА є інтернет-магазини, такі як Rozetka та Amazon.

- PWA

PWA (Progressive Web Application) має можливості, функції та користувацький досвід нативних комп'ютерних та мобільних додатків. Хоча немає чіткого розрізнення між не-PWA та PWA додатками, але PWA повинен містити проксі-шар (Service Worker) та Web App маніфест.

Service Worker є проксі-шаром між серверною та клієнтською частинами, який знаходиться у браузері та обробляє всі запити. Таким чином, є два фронтенди шару, один для інтерфейсу та інший для логіки. Це дозволяє створювати повноцінні програми для Інтернету. Service Worker зазвичай програмують на чистому JavaScript. [28]

1.1.3 Різниця між веб додатком та сайтом

Хоча веб-додаток і веб-сайт можуть здаватися ідентичними, вони мають різні функції та компоненти розробки. Веб-додаток запускається в браузері та має більше елементів, які можна взаємодіяти, тоді як веб-сайт може містити лише інформацію.

Більш детальну характеристику чим відрізняється веб-додаток і веб-сайт ви можете побачити в таблиці 1.1.[28]

Різниця між веб-додатком та веб-сайтом

Параметр	Веб-додаток	Веб-сайт
Основне призначення	Створюється для того, щоб взаємодіяти з користувачем	Має на увазі лише наявність статей, без будь-якого можливого впливу на читача
Взаємодія з користувачем	Користувач може проводити маніпуляції з даними, але з обмеженим доступом	Користувач може читати інформаційний контент, проте не може його ніяк міняти
Аутентифікація	Більшість веб додатків вимагають автентифікації, щоб користувач міг скористатися програмою	Для даних сайтів не потрібна обов'язкова автентифікація. Сайт може лише мати пароль, який відсилає адміністратор ресурсу
Завдання та складність	Веб-програма має багато функцій і закриває безліч проблем	Веб-сайт відображає лише статичну сторінку, на якій зображена текстова інформація
Зміна проекту	Щоб внести будь-які зміни в проект, потрібно робити ревью всього коду і після вписувати те, що ви хочете змінити.	Досить просто вносити зміни, лише зробивши пару змін у html коді сторінки

1.2 Огляд існуючих рішень

На даний момент існує багато різних конкуруючих сервісів для доставки товарів на замовлення, кожен з яких посів свою нішу.

1.2.1 Сервіси

Glovo — іспанська компанія, яка надає сервіс доставки через мобільний застосунок. Станом на 2021 рік компанія оперує у більш, ніж 25 країнах. Із сервісом співпрацює більше 61 000 кур'єрів.[26]

Через мобільний додаток клієнти можуть замовити товар невеликого розміру (40x40x30см) вагою до 9 кг, який може включати їжу, ліки та документи. У Європі, головним складовим замовлень є їжа і доставка (близько 85%). Після оформлення замовлення, користувач може відстежити переміщення кур'єра на маршруті в режимі реального часу. Більшість кур'єрів рухається на велосипедах, мотоциклах і скутерах. Сервіс доступний через десктоп, IOS та Android.

Сервіс був заснований Оскаром П'єром та Сашою Мішо. У березні 2015 року було доставлено перше замовлення в Барселоні. У цьому ж місяці компанія отримала 140,000 євро у першому раунді інвестицій. У другому раунді інвестицій восени 2015 року компанія залучила 2 млн євро і почала свою діяльність у Мадриді та Валенсії. У березні 2016 компанія придбала італійську компанію Foodinho та почала свою діяльність у Мілані. У серпні 2016 року компанія залучила 5 млн євро в третьому раунді інвестицій. В червні 2016 року була укладена ексклюзивна угода з компанією McDonald's Іспанія та Італія для співпраці з послугою McDelivery. У вересні 2016 року було зібрано 30 млн євро інвестицій.

Станом на 2019 рік сервіс був успішно запущений у 10 містах: Київ, Харків, Одеса, Дніпро, Львів, Запоріжжя, Вінниця, Миколаїв, Чернігів, Черкаси, Полтава, Херсон, Івано-Франківськ, Житомир. Однак, станом на березень 2020 року, сервіс був запущений у 15 містах України.

Компанія стала жертвою кібератаки 29 квітня 2021 року. За даними компанії, зловмисники не отримали доступу до даних кредитних карток користувачів.

1.2.2 Bolt Food

Bolt Food — це онлайн-платформа для замовлення та доставки їжі, запущена компанією Bolt у 2019 році. Їжу та продукти доставляють кур'єри на автомобілях, скутерах, велосипедах або пішки. Станом на 2023 рік Bolt Food працює в понад 80 містах у 19 країнах.

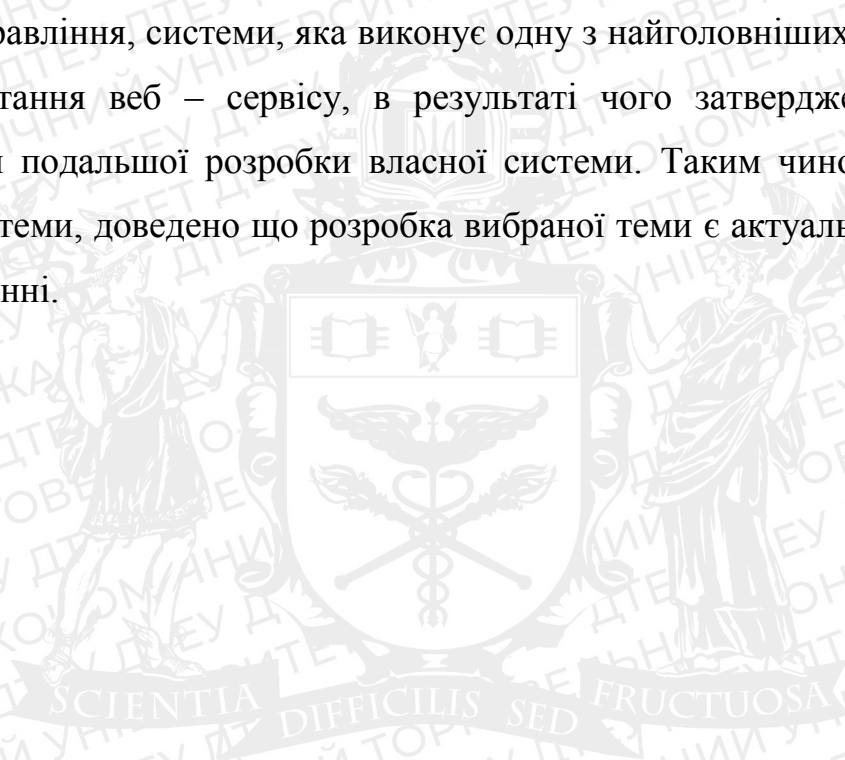
Сервіс налічує понад 6 мільйонів зареєстрованих користувачів у всьому світі та включає понад 30 000 партнерських ресторанів і мереж. У 2021 році Bolt Food запустила службу швидкої доставки продуктів Bolt Market. У серпні 2019 року транспортна компанія Bolt зайнялася доставкою їжі, запустивши Bolt Food у Таллінні, Естонія, і оголосивши про плани розширити послугу до Латвії, Литви та Південної Африки пізніше того ж року.

Під час презентації директор з продуктів Bolt Євгеній Кабанов сказав, що створення служби доставки їжі було популярним запитом серед існуючих користувачів Bolt протягом досить тривалого часу. У квітні 2021 року додаток Bolt Food було додано до Huawei AppGallery після того, як компанія Huawei заявила, що отримала понад 2000 запитів на підключення до мобільного додатка. У серпні 2021 року Bolt зібрав 600 мільйонів євро для запуску служби доставки продуктів Bolt Market. У січні 2022 року компанія оголосила, що залучила 628 мільйонів євро при оцінці 7,4 мільярда євро, щоб розширити свій супердодаток для транспорту та доставки їжі. Станом на січень 2023 року Bolt Food працює у понад 80 містах і 19 країнах Європи та Африки.

ВИСНОВОК ДО РОЗДІЛУ

В першому розділі було розглянуто поняття веб-додатку, проаналізовано веб-архітектуру, її суть, основні поняття та структуру.

Одним з досягнень даного розділу – аналіз та порівняння адміністративних панелей управління, системи, яка виконує одну з найголовніших ролей в побудові та використанні веб – сервісу, в результаті чого затверджено найголовніші функції для подальшої розробки власної системи. Таким чином, переглянувши існуючі системи, доведено що розробка вибраної теми є актуальною та корисною в використанні.



РОЗДІЛ 2

Функціонал для роботи над сервісом

2.1 HTML

HTML або "Мова розмітки гіпертексту" - це код, який використовується для створення веб-сторінок та веб-додатків, доступ до яких здійснюється в Інтернеті. Розробка HTML почалася наприкінці 1980-х - початку 1990-х років. Ця мова дозволяє розробникам веб-сайтів керувати відображенням елементів, таких як зображення, текст, форми та інтерактивні функції, в інтернет-браузерах.[16]

Для створення веб-сайтів та програм, доступних через Інтернет-браузерні програми, такі як Chrome, Firefox, Safari та Edge, веб-розробники часто використовують HTML разом із каскадними таблицями стилів (CSS) та мовою програмування JavaScript. HTML, CSS і JavaScript є основними мовами, які використовуються для розробки веб-сайтів і забезпечення їх функціональності.

Стандарти для HTML, CSS і JavaScript підтримуються всесвітньою веб-сторінкою (W3C). Це організація, яка розробляє та підтримує стандарти та протоколи Інтернету. У 1980 році Тім Бернерс-Лі, працюючи в Європейській організації ядерних досліджень, розпочав роботу над прототипом HTML. Протягом десятиліття Бернерс-Лі розробив мову розмітки гіпертексту, а також створив інтернет-браузер та серверне програмне забезпечення.

Протягом наступних двох десятиліть Всесвітній консорціум із вебсторінок (W3C) продовжить розробку за міжнародним стандартом коду. У 1995 році була опублікована друга версія HTML, і протягом наступних кількох років були внесені додаткові оновлення.

HTML побудовано на основі правил SGML (стандартна узагальнена мова розмітки), а XHTML побудовано на основі правил XML, суворішої підмножини SGML. XHTML документи повинні бути коректними XML документами, що дозволяє їх обробку стандартними інструментами обробки XML. HTML вимагає більш складних синтаксичних аналізаторів. XHTML можна розглядати як перетин HTML і XML, оскільки цей стандарт є переформулюванням HTML засобами XML. XHTML 1.0 став рекомендацією консорціуму W3C у січні 2000 року, а XHTML 1.1 - у травні 2001 року.

HTML - це мова розмітки, яка інтерпретується браузером і відображається у вигляді документа, зручного для людини. Вона є додатком SGML і відповідає міжнародному стандарту ISO 8879. HTML-документ є текстовим файлом, розміченим спеціальними командами. Текстовий формат вибирається з урахуванням простоти, можливості безпосередньої інтерпретації в будь-якій операційній системі, мінімального розміру файлу, зручності редагування і інтерпретації.

Мова розмітки гіпертекстових документів HTML дозволяє визначити різні типи елементів, такі як текстові фрагменти з параметрами форматування, списки, таблиці, зображення, гіперпосилання тощо. Елементи HTML оголошуються за допомогою тегів. Всі HTML-теги, що зустрічаються в тексті документа, інтерпретуються браузером при відображенні документа.

HTML - це мова розмітки, яка використовується для опису того, як програміст може використовувати код для позначення тексту. HTML та XML - це мови розмітки, що відрізняються від машинних мов, які записуються шістнадцятковим або двійковим кодом.

При використанні HTML для створення сторінки на веб-сайті автор використовує ключові компоненти або атрибути, які будуть прочитані будь-яким Інтернет-браузером. Багато з цих атрибутів використовуються разом з початковими та

кінцевими тегами. Наприклад, для позначення початку абзацу автор може використовувати тег <p>, а для позначення кінця абзацу - </p>.

Веб-сайт програміста може використовувати різні атрибути та елементи HTML для визначення зовнішнього вигляду кожної сторінки та її відображення користувачу. Додатково, програміст може використовувати CSS, JavaScript та інші мови, наприклад PHP, jQuery та XML, для створення веб-сайту.

Багато новачків у програмуванні вивчають HTML як свою першу мову та продовжують навчання з JavaScript та CSS. Навчання основ HTML може бути корисним для майбутніх програмістів і для тих, хто хоче зрозуміти, як створюються веб-сторінки, які вони щодня переглядають на своєму комп'ютері або смартфоні.

2.2. CSS

CSS - це мова каскадних таблиць стилів, яка використовується для опису стилів, які можуть бути застосовані до документів з кодом розмітки. Ця мова була створена Наном Вімом Ліе у 1994 році. CSS дозволяє веб-розробникам змінювати зовнішній вигляд та макет своїх веб-сторінок. Наприклад, за допомогою CSS можна змінювати шрифт, розмір та кольори тексту, фону, гіперпосилань та багато іншого.

CSS має декілька переваг. По-перше, він дозволяє економити час, оскільки стилі можна використовувати повторно на різних сторінках. Ви можете визначити стиль для кожного елемента HTML і застосувати його до багатьох веб-сторінок. Крім того, використання CSS допомагає знизити час завантаження сторінок, оскільки не потрібно повторювати атрибути HTML для кожного тегу. Всі стилі можна визначити в одному CSS-файлі і застосувати їх до відповідних тегів.

CSS також спрощує обслуговування веб-сторінок. Зміни в стилі можна вносити глобально, змінивши одне правило CSS, що автоматично оновить всі елементи на

всіх сторінках. Крім того, CSS надає більш широкий набір атрибутів порівняно з HTML, що дозволяє надати сторінці кращий вигляд.

Окрім цього, CSS дозволяє оптимізувати веб-сторінки для різних типів пристроїв. За допомогою аркушів стилів можна налаштувати вигляд веб-сторінки для портативних пристроїв, таких як КПК та мобільні телефони, або для друку. Використовуючи один і той же HTML-документ, можна створити різні версії веб-сторінок для різних пристроїв.[16]

2.3. JavaScript

JavaScript - скриптова мова, яка широко використовується для створення динамічної поведінки веб-сторінок. Ця мова була створена з метою надання життя веб-сторінкам. Програми на JavaScript називаються скриптами і можуть бути вбудовані безпосередньо в HTML-код сторінки та автоматично виконуватися під час завантаження. Скрипти JavaScript представлені у вигляді звичайного тексту і не потребують спеціальної підготовки або компіляції для запуску. У цьому відношенні JavaScript відрізняється від іншої мови програмування, відомої як Java.[9]

Початково JavaScript мав назву "LiveScript". Однак, через популярність мови Java в той час, вирішено було позиціонувати нову мову як "молодшого брата" Java. З часом JavaScript став повністю самостійною мовою зі своєю специфікацією під назвою ECMAScript і вже не має жодного відношення до Java.

JavaScript є найпоширенішою мовою сценаріїв на стороні клієнта, що означає, що він використовується для написання коду, який виконується безпосередньо на сторінці HTML. Коли користувач запитує HTML-сторінку з JavaScript, скрипт надсилається до браузера, який виконує його.

Однак, варто зазначити, що JavaScript, який знаходиться на сторінці HTML, доступний для всіх, хто переглядає сторінку. Це може бути як перевагою, так і

недоліком. З одного боку, відкритість мови дозволяє переглядати, вивчати і використовувати будь-який JavaScript, зустрітий в Інтернеті. З іншого боку, це може призвести до проблем з безпекою, оскільки зловмисники можуть використовувати цей доступ для впровадження шкідливого коду.

JavaScript може використовуватися не тільки в браузері, але і на сервері або на будь-якому пристрої, що має механізм JavaScript. В різних браузерах використовуються різні двигуни для виконання JavaScript, такі як V8 у Chrome і Opera і SpiderMonkey у Firefox.

Сучасний JavaScript є безпечною мовою програмування, яка не надає прямого доступу до низькорівневих ресурсів, таких як пам'ять або процесор. Він був створений для використання в браузерах, тому не потребує такого доступу.

Можливості JavaScript значною мірою залежать від середовища, в якому він використовується. Наприклад, Node.js дозволяє JavaScript зчитувати та записувати файли, виконувати мережеві запити тощо. В браузері JavaScript може виконувати різні завдання, пов'язані з маніпуляцією веб-сторінками, взаємодією з користувачем та веб-сервером. Наприклад, він може змінювати вміст сторінки, реагувати на дії користувача, відправляти запити на сервер, зчитувати та завантажувати файли, встановлювати файли cookie, показувати повідомлення та зберігати дані на стороні клієнта.

2.4 PHP

PHP («препроцесор гіпертексту») - це скриптова мова програмування, що призначена для генерації сторінок на мові HTML на стороні веб-сервера та роботи з базами даних. Більшість провайдерів хостингу підтримують PHP.

Препроцесор - це програма, яка виконує попередню обробку даних, щоб вони могли використовуватись іншими програмами, наприклад, компіляторами. Дані, які виходять з препроцесора, знаходяться в препроцесованій формі, яка може бути оброблена іншими програмами (компіляторами). Препроцесори можуть

виконувати текстову підстановку або змагатися з іншими скриптовими мовами програмування. PHP є розширеною скриптовою мовою програмування, яка може бути використана для обробки первинного коду перед компіляцією. Це дозволяє значно розширити можливості мов програмування C/C++ та комп'ютерної верстки. Назва PHP є рекурсивним акронімом «PHP: Hypertext Preprocessor».

PHP є однією з найпопулярніших скриптових мов програмування в Інтернеті, разом з JSP, Perl, Ruby, Python та мовами, використовуваними в ASP.NET. PHP відрізняється наявністю ядра і підключаємих модулів, які розширюють його можливості, такі як робота з базами даних, сокетами, графікою, криптографією, документами формату PDF та інше. Багато розширень доступні, але стандартне постачання включає лише кілька десятків найбільш популярних розширень.

PHP може бути підключений до веб-сервера через модуль, створений спеціально для цього сервера (наприклад, для Apache або IIS), або використаний як CGI-додаток. Крім того, PHP може бути використаний для вирішення адміністративних завдань в різних операційних системах. Однак, у ролі серверної мови PHP не набув такої популярності, як Perl, Python або VBScript. Синтаксис PHP схожий на синтаксис мови C, і включає елементи, запозичені з Perl, такі як асоціативні масиви і цикл foreach.

PHP знайшов широке застосування серед тисяч програмістів. Мільйони веб-сайтів повідомляють про свою роботу з PHP, що складає більше ніж п'яту частину доменів Інтернету. Команда розробників PHP складається з багатьох людей, що добровільно працюють над ядром та розширеннями PHP, а також над суміжними проектами, такими якPEAR або документація мови.

PHP зазвичай використовується як серверна мова, яка виконується на веб-сервері. Це відрізняє його від мови JavaScript, яка виконується на стороні клієнта. Мови на стороні клієнта, такі як HTML, CSS та JavaScript, дозволяють веб-браузерам розбирати та змінювати контент на екрані комп'ютера. Проте JavaScript виконується на стороні клієнта, надаючи інструкції, які можуть бути розуміні та

виконані веб-браузером. Сторона клієнта - це та сторона, яку користувач бачить під час використання Інтернету. З іншого боку, серверна діяльність включає веб-браузер, який надсилає запити на веб-сервер, який відповідає на запит з HTML-кодом, який потім обробляється і надається веб-браузером, перетворюючись на контент на екрані користувача.[24]

Основна відмінність цього процесу від діяльності на стороні клієнта полягає в тому, що він включає зв'язок із сервером і не повністю завершується в браузері клієнта. Мова сценаріїв на стороні клієнта, така як JavaScript, може автоматизувати завдання, пов'язані з вмістом, який вже доступний користувачеві в його веб-браузері. З іншого боку, мова скриптів на стороні сервера, така як PHP, використовується для запиту вмісту з сервера веб-сайту або бази даних і створення вмісту, який видимий і доступний для користувача сайту. З метою підтримки ефективності веб-сайту, швидкості завантаження та ємності пам'яті, не всі активи можна постійно зберігати локально на веб-сторінках.

Наприклад, за допомогою PHP-скрипта можна автоматично відображати на головній сторінці вашого веб-сайту три останні публікації блогу. В цьому випадку самі публікації зберігаються на сервері сайту і викликаються, коли вони займають один із трьох останніх опублікованих слотів. Це дозволяє уникнути попереднього завантаження публікацій на ваш сайт і необхідності адміністратора завантажувати та оновлювати публікації при публікації нових історій. PHP-скрипти також можуть містити умовні заяви, які спрямовують ваш сайт на зміну його вигляду та додавання вмісту з вашого веб-сервера за потреби. Наприклад, скрипт може вказати, що якщо адміністратор завантажить відеопосилання на поле "x", то сайт завантажить відео з його сервера і відобразить його для користувача. Якщо адміністратор не завантажить посилання, то на цій сторінці буде відображено зображення "y" за замовчуванням. Серверні дії PHP вводять нові можливості на веб-сайт, які перевищують статичні функції, що пропонуються HTML і CSS, а також навіть динамічний вміст на стороні клієнта, реалізований за допомогою JavaScript.

2.5 База даних MySQL

Види баз даних:

1. Реляційна база даних - це тип сховища даних, яке організовує дані у вигляді таблиці, де дані у різних таблицях пов'язані між собою. Основою цих систем є структурована мова запитів, яка використовується для комунікації та управління даними. Системи управління базами даних (СУБД) мають фіксовану схему, що означає, що дані зберігаються у вигляді рядків (записів) та стовпців (атрибутів) зі строгою структурою. Кожен запис, як правило, має значення для кожного атрибута, що призводить до чітких залежностей між різними даними.

2. Нереляційна база даних (НБД) - це табличне сховище даних, яке використовує різні моделі даних для зберігання та управління. На відміну від більшості традиційних СУБД, тут не використовується таблична схема з рядками і стовпцями, а застосовується модель, оптимізована під конкретні вимоги типів даних, які зберігаються. Наприклад, дані можуть бути збережені у вигляді простих пар "ключ-значення", документів JSON або графів, які складаються з ребер та вершин.

MySQL - одна з найпопулярніших реляційних баз даних. Початково випущена як відкрите рішення, MySQL тепер належить корпорації Oracle. Сьогодні MySQL є основою для класичного серверного веб-стеку LAMP та LEMP (Linux, Apache/nginx, MySQL та Perl/PHP/Python). MySQL написаний на мовах програмування C та C++ і працює добре на таких платформах, як Linux, Windows, MacOS, IRIX та інші. СУБД - це програмне забезпечення або сервіс, який використовується для створення та управління базами даних заснованих на реляційній моделі.

Не будучи занадто технічним, основні принципи, що відбуваються у середовищі MySQL, залишаються однаковими:

1. MySQL створює базу даних для зберігання та обробки інформації, встановлюючи зв'язки між таблицями.

2. Клієнти можуть робити запити, використовуючи оператори SQL, щоб отримати необхідні дані з MySQL.
3. Серверна програма надає відповідну інформацію на запити клієнтів, яка потім відображається на їхніх пристроях.

Клієнти зазвичай вибирають графічний інтерфейс для роботи з MySQL. Легкий і зручний графічний інтерфейс дозволяє зручно управляти базами даних. Деякі популярні графічні інтерфейси для MySQL включають MySQL WorkBench, SequelPro, DBVisualizer та Navicat DB. Деякі з них є безкоштовними, деякі комерційними, і їх можна використовувати на різних операційних системах. Вибір графічного інтерфейсу залежить від потреб користувача. Для управління веб-базами даних, включаючи сайти на WordPress, часто використовується phpMyAdmin.

Переваги MySQL:

1. Гнучкість і простота використання.
2. Можливість змінювати вихідний код, щоб задовольнити свої потреби, без необхідності платити за нього. Процес встановлення простий і займає небагато часу.
3. Висока продуктивність.
4. Підтримка широкого спектру кластерних серверів, що дозволяє ефективно працювати з великими обсягами даних.
5. Визнана індустрією стандартом, що забезпечує наявність багатьох ресурсів для розробників.
6. Високий рівень безпеки завдяки системі управління доступом та шифруванню паролів.

Недоліки:

1. Недостатня надійність в деяких аспектах роботи з даними, таких як зв'язки, транзакції та аудит.

2. Низька швидкість розробки.[14]



ВИСНОВОК ДО РОЗДІЛУ

Цей розділ демонструє технології для використання при розробці веб-додатку. Було розглянуті технології HTML, CSS, JavaScript. Також було розглянута мова програмування PHP та база даних MySQL. Важливо одразу правильно обрати технології, щоб у подальшому не виникало складнощів з удосконаленням проекту.



РОЗДІЛ 3

Розробка та огляд середовища розробки

В попередньому розділі було проаналізовано та обрано саме ті мови програмування та компоненти, які комфортно використовувати для розробки.

3.1 Опис застосунку IntelliJ IDEA

Редактор коду в IntelliJ IDEA – один із найпотужніших в індустрії. Завдяки початковій індексації він розуміє всі нюанси вашого коду та виявляє помилки в міру введення, пропонує варіанти автодоповнення з урахуванням контексту, виконує безпечний рефакторинг та надає багато інших корисних функцій.

IntelliJ IDEA дозволяє працювати з елементами коду іншими мовами, вбудованими у ваш код. Ви навіть не помітите різницю між роботою з основним кодом Java і кодом, що підставляється. IDE забезпечує підсвічування, автодоповнення, виявлення помилок та інші функції допомоги для вбудованих фрагментів коду.

Функція автодоповнення коду IntelliJ IDEA працює з урахуванням контексту, пропонуючи тільки варіанти, дійсні для поточного положення курсору. Вам не потрібні жодні комбінації клавіш та додаткові налаштування: функція автодоповнення запускається, як тільки ви починаєте писати код у редакторі.

Функція автоматичного рефакторингу в IntelliJ IDEA дає можливість безпечного та ефективного оновлення коду. Він стає простішим, зрозумілішим, і його легше підтримувати. IDE пропонує широкий вибір рефакторингів для безпечного перейменування елементів коду, зміни сигнатури класу або методу, вилучення фрагмента коду метод і введення змінних.

IntelliJ IDEA вміє знаходити та усувати дублювання коду в проєкті. Просто вкажіть область пошуку, і IDE виявить ділянки коду, які схожі з вибраним методом чи константним полем. Потім вона запропонує замінити ці повторення на виклики оригінальних методів чи констант.

IntelliJ IDEA відмінно підходить для веб-розробки, забезпечуючи підтримку JavaScript та пов'язаних технологій, включаючи TypeScript, React, Vue, Angular, Node.js, HTML та файли стилів. Для всіх цих мов IntelliJ IDEA надає ключові функції: автодоповнення коду, контекстні дії, підсвічування помилок та багато іншого.

IntelliJ IDEA забезпечує інтеграцію з Git, дозволяючи клонувати проєкти з Git, записувати та відправляти зміни, працювати з кількома гілками, керувати списками змін та перевіряти зміни перед відправкою.

3.2 Підключення до бази даних

Приєднання до бази даних MySQL реалізується через PHP, який є мовою програмування.

В файлі connect.php ми встановлюємо з'єднання з сервером MySQL через функцію `mysql_connect()` і присвоюємо її змінній `"$conn"`. За замовчуванням, оскільки я використовую локальний хост, всі параметри мають наступні значення: адреса бази даних - `"127.0.0.1"`, назва - `"diplom"`, ім'я користувача - `"root"`, пароль - `""`.

```

<?php
// $servername = "127.0.0.1";
// $database = "diplom";
// $username = "root";
// $password = "";

$conn = mysqli_connect('127.0.0.1', 'root', '', 'diplom');

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// echo "Connected success";
// mysqli_close($conn);
?>

```

Рис. 3.1 Файл connect.php

Після успішного встановлення з'єднання до бази даних, ми маємо можливість взаємодіяти з нею, додавати та змінювати різноманітну інформацію.

3.3 Робота з базою даних

Розглянемо проблему про те, що після заповнення форми реєстрації та збереження даних в базу, пароль залишається незмінним, що створює значні ризики безпеки і може призвести до його швидкого зламу.

Для вирішення даної проблеми було використано функцію md5(), яка є однією з найпростіших функцій шифрування і підходить для початкової роботи з невеликим сервісом.

```

$password = md5($_POST['password']);

```

Рис. 3.2 Функція md5()

3.4 Розгляд форми реєстрації, авторизації

Насамперед розглянемо форму авторизації, що використовує PHP скрипти.

Рекомендується розпочати з виклику функції session_start(). Це необхідно для ініціалізації процесу обміну даними між користувачем та сервером. У разі, якщо користувач не здійснює жодних дій протягом певного періоду, сесія автоматично завершується.

Для роботи з сервером використовуються два HTTP-методи:

1. GET-запит. Використовується для отримання даних з веб-сайту. Наприклад, для доступу до сторінки, яку клієнт бажає переглянути.
2. POST-запит. Метод передачі даних на веб-сайт. Зазвичай його використовують для надсилання даних з форми на сервер.

```
<?php
session_start();
require_once 'connect.php';

$name_company = $_POST['name_company'];
$tel_number = $_POST['tel_number'];
$password = $_POST['password'];
$password_conf = $_POST['password_conf'];

if ($password !== $password_conf) {
    $_SESSION['error'] = 'Паролі не співпадають';
    header("Location: ../register.php");
}else{

    $hash_password = md5($password);

    mysqli_query($conn, "INSERT INTO `users` (`id`, `name_company`,
`tel_number`, `password`) VALUES (NULL, '$name_company', '$tel_number', '$hash_password')");
    $_SESSION['correct'] = 'Регістрація виконана!';
    header("Location: ../index.php");
}
?>
```

Рис. 3.3 Відправлення форми

За допомогою методу POST ми передаємо користувачеві дані, такі як пароль, підтвердження паролю, номер телефону та назва компанії, на веб-сервер для їх подальшого зберігання.

Для зберігання даних сеансу використовується глобальна змінна `$_SESSION`, яка дозволяє зберігати дані в файлі сесії на сервері. Цей файл створюється при першому запуску сесії і може використовуватись разом з кількома запитами.

Функція `mysqli_query()` використовується для виконання запиту `INSERT INTO`, який додає всі наші записи до бази даних. Виконуємо правильний SQL-запит на

вставку запису з усіма вказаними даними, а потім використовуємо функцію `mysqli_query()` для виконання цього запиту та додавання запису до таблиці.

```
<?php
session_start();
require_once 'connect.php';

$tel_number = $_POST['tel_number'];
$password = md5($_POST['password']);

$user_check = mysqli_query($conn, "SELECT * FROM `users` WHERE
`tel_number` = '$tel_number' AND `password` = '$password'");
if (mysqli_num_rows($user_check) > 0){

    $profile = mysqli_fetch_assoc($user_check);

    $_SESSION['profile'] = [
        'id' => $profile['id'],
        'tel_number' => $profile['tel_number'],
        'name_company' => $profile['name_company']
    ];

    header("Location: ../profile.php");
}
else{
    $_SESSION['error'] = 'Невірний логін або пароль';
    header("Location: ../index.php");
}
```

Рис. 3.4 Сесія авторизації

Іншою поширеною операцією при взаємодії з базами даних в PHP є отримання записів з таблиць (запити типу SELECT). Спочатку ми формуємо SQL-запит, використовуючи оператор SELECT. Потім ми викликаємо функцію `mysqli_query()`, щоб отримати дані з таблиці.

Для примусово виходу із сесії, скасовуємо поточну суперглобальну змінну, за допомогою:

```
<?php
session_start();
unset($_SESSION['profile']);
header("Location: ../index.php");
?>
```


3.5 Модель

В основі входу в адміністративну панель лежить мова розмітки HTML разом із скриптами PHP для забезпечення зв'язку з сервером. Під час розробки клієнтської частини основна увага приділяється розробці інтерфейсу, зручності використання для користувачів. Саме в цьому полягає суть даної частини.

Мова розмітки документів складається з тегів, які утворюють систему разом із атрибутами.

```
<div class="wrapper">
  <form action="vendor/signin.php" method="post" enctype='multipart/form-data'>
    <h1>Bxid</h1>
    <label>Введіть номер телефону</label>
    <input type="tel_number" name='tel_number' required>
    <label>Введіть пароль</label>
    <input type="password" name='password' id="" required>
    <div class="registration">Ще немає профілю? <span style="font-weight:bold;color: black;">
      <a href="./register.php">Зареєструйтеся</a></span>
    </div>
    <button type='submit'>Зайти</button>
  <?php
    if ($_SESSION['error']) {
      echo '<p class="error_psw"> ' . $_SESSION['error'] . ' </p>';
    }
    unset($_SESSION['error']);
  <?>
  <?php
    if ($_SESSION['correct']) {
      echo '<p class="correct_psw"> ' . $_SESSION['correct'] . ' </p>';
    }
    unset($_SESSION['correct']);
  <?>
  </form>
  <div class="logo"></div>
  <div class="img_back"></div>
</div>
```

Рис. 3.5 Форма входу мовою HTML

3.6 Використання CSS

CSS - каскадна таблиця стилів, мова опису зовнішнього вигляду документа, яка дозволяє поліпшити інтерфейс та налаштувати веб-сайт для кращого сприйняття продукту користувачем. Завдяки CSS можна керувати розташуванням, кольорами,

шрифтами та іншими аспектами візуального представлення веб-сторінки, щоб зробити її більш привабливою та зручною для користувача.

SCSS - це розширена версія CSS, яка надає більші можливості. За допомогою препроцесора CSS ми можемо зручно писати код, який дозволяє використовувати цикли, функції, імпорт, змінні та математичні операції. Це значно підвищує потужність CSS і спрощує його написання.

```
.zakaz_block{
  padding: 15px 20px;
  background: #F4F4F4;
  border-radius: 3px;
  display: flex;
  flex-direction: row;
  .zakaz_block-info{
    margin-top: 10px;
    margin-left: 20px;
    padding-left: 20px;
    border-left: 1px solid #E6E6E6;
    width: 216px;
    .block-info_main-info{
      margin-top: 30px;
      margin-bottom: 140px;
      .block-info_main-info_item{
        font-weight: 700;

        border-bottom: 1px solid #B6B6B6;
        font-size: 14px;
        line-height: 17px;
        color: #B6B6B6;
        margin-top: 2.5px;
        margin-bottom: 10.5px;
      }
    }
    .item_top{
      margin-top: 10px;
    }
  }
}
```

Рис. 3.6 CSS


```

.zakaz_block {
  padding: 15px 20px;
  background: #F4F4F4;
  border-radius: 3px;
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-orient: horizontal;
  -webkit-box-direction: normal;
  -ms-flex-direction: row;
  flex-direction: row;
}
.zakaz_block .zakaz_block-info {
  margin-top: 10px;
  margin-left: 20px;
  padding-left: 20px;
  border-left: 1px solid #E6E6E6;
  width: 216px;
}
.zakaz_block .zakaz_block-info .block-info_main-info {
  margin-top: 30px;
  margin-bottom: 140px;
}
.zakaz_block .zakaz_block-info .block-info_main-info .block-info_main-info_item {
  font-weight: 700;
  border-bottom: 1px solid #B6B6B6;
}
.zakaz_block .zakaz_block-info .block-info_main-info .block-info_main-info_item p {
  font-size: 14px;
  line-height: 17px;
  color: #B6B6B6;
}

```

Рис. 3.7 CSS

Зробивши висновок, ми бачимо, що використання препроцесорів CSS-таблиць є набагато ефективнішим, компактнішим та швидшим, ніж звичайна верстка. Це пояснюється тим, що препроцесори можуть економити місце, розміщуючи дочірні елементи безпосередньо всередині батьківського блоку і використовуючи менше повторень завдяки наслідуванню, імпорту файлів та використанню міксінів.

3.7 Використання Java

Java використовується для розробки різних типів програм, включаючи веб-додатки, мобільні додатки, серверні додатки, вбудовані системи, фінансові

технології та багато інших. Вона має велику спільноту розробників і активно розвивається, що робить її популярною і потужною мовою програмування.

Для забезпечення адміністративної панелі вкладками на сторінці профілю, було розроблено алгоритм перемикання.

```
public void changeTabs(Event evt, String tabName) {
    Element[] mainSection = document.getElementsByClassName("main_section");
    for (int i = 0; i < mainSection.length; i++) {
        mainSection[i].style.display = "none";
    }

    Element[] tabButtons = document.getElementsByClassName("slidebar_buttons-link");
    for (int i = 0; i < tabButtons.length; i++) {
        tabButtons[i].className = tabButtons[i].className.replace(" active", "");
    }

    document.getElementById(tabName).style.display = "block";
    evt.currentTarget.className += " active";
}
```

Рис. 3.8 Java

```
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class Main {
    public static void main(String[] args) {
        Document doc = Jsoup.parse(html); // Парсинг HTML-документу

        Elements els = doc.getElementsByAttribute("name"); // Знаходження елементів з атрибутом "name"

        for (Element item : els) {
            item.addEventListener("click", new EventListener() { // Додавання події "click" для кожного знайденого елемента
                public void handleEvent(Event e) {
                    item.parentNode.parentNode.removeChild(item.parentNode); // Видалення батьківського елемента з DOM-дерева сторінки
                }
            });
        }
    }
}
```

Рис. 3.9 Java

3.8 Формула для визначення вартості доставки

Для систем доставки може бути використана формула, що враховує декілька факторів. Наприклад, з урахуванням типу товару (наприклад, крихкий,

небезпечний) і пов'язаних з цим особливостей доставки, вартості палива, зарплати працівників, амортизації обладнання тощо.

У цьому випадку формула може мати вигляд:

Вартість доставки = Фіксована плата + (Вага товару * Ставка за кілограм) + (Відстань * Ставка за кілометр) + (Додаткові витрати)

Фіксована плата - це основна плата за доставку, яка покриває загальні витрати компанії;

Ставка за кілограм - це плата за кожен кілограм товару;

Ставка за кілометр - це плата за кожен кілометр доставки;

Додаткові витрати - це будь-які інші витрати, пов'язані з обробкою, страхуванням, платежами і т.д.

Ці параметри можуть бути налаштовані відповідно до потреб логістичної компанії і можуть варіюватися залежно від специфіки товару, регіону, часу доставки та інших факторів.

ВИСНОВОК ДО РОЗДІЛУ

У цьому розділі було розглянуто розробку і продемонстровано основні моменти розробки. Перший етап розробки був виконаний з використанням мови PHP, підключенням до бази даних MySQL та написанням основних принципів серверного зв'язку. На другому етапі була розроблена клієнтська частина, а саме веб-інтерфейс та функціонал сервісу. Були наведені всі налаштування допоміжних програм, які використовувалися під час розробки. Детально розглянуті алгоритми рішень, методи та функції кодової частини програми.



ВИСНОВКИ

У данному дипломному проекті була проведена розробка та реалізація веб-додатку, який дозволяє визначити вартість доставки у логістичній компанії на основі мови програмування Java.

У ході розробки проекту було використано значну кількість програмних компонентів, різних мов програмування, середовищ та додатків, що відіграли важливу роль у процесі.

У процесі розробки були використані такі технології, як Java, HTML, CSS, JavaScript і база даних MySQL.

Розглянуті та використані технології показали, що розумний підбір технологій для розробки проекту може полегшити процес розробки та навіть зробити його цікавим.

Отриманий результат може бути використаний логістичними компаніями для автоматизації процесу доставки, що дозволить покращити ефективність роботи компанії та забезпечити точність розрахунків для клієнтів.

Також були розглянуті питання необхідності такої системи в наші часи, описано важливість покращення середовища для доставки, що в свою чергу дозволить покращити роботу підприємства.

В майбутньому цю розробку можна оновлювати та доповнювати новими функціональними можливостями, оскільки в цьому напрямку ще є багато корисних потреб. Проект розроблено таким чином, щоб стати одним з конкурентоздатних сервісів та успішно функціонувати на реальному ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Архитектура веба: основы для начинающих разработчиков. Tproger. URL: <https://tproger.ru/translations/web-architecture-101/>.
2. Клиент-серверна архітектура [Електронний ресурс] – Режим доступу до ресурсу: <https://doka.guide/js/web-app-works/>.
3. HostCMS. HostCMS. [Електронний ресурс] – Режим доступу до ресурсу: <http://demo.hostcms.ru/>.
4. DemoSimplaCms.[Електронний ресурс] – Режим доступу до ресурсу:<http://demo.simplacms.ru/>.
5. Драч В. Анализ популярных реляционных систем управления базами данных (2022 г.). Сайт Владимира Драча. URL: <https://drach.pro/blog/hitech/item/196-popular-relational-dbms-2022>.
6. Интерактивные онлайн-курсы HTML Academy. HTML Academy. [Електронний ресурс] – Режим доступу до ресурсу: <https://htmlacademy.ru/tutorial/php/mysql>.
7. Переваги та недоліки JavaScript. hackit-ukraine. [Електронний ресурс] – Режим доступу до ресурсу: <https://hackit-ukraine.com/627-the-advantagesand-disadvantages-of-javascript>.
8. Goodcn. Самые популярные языки программирования бэкенда: для чего они подходят лучше всего и какие компании их используют. Хабр. [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/skillbox/blog/534684/>.
9. Megida D. What is JavaScript? A Definition of the JS Programming Language. freeCodeCamp.org. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/>.

10. phpMyAdmin. phpMyAdmin. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.phpmyadmin.net>.

11. PHP | unset() Function - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/php-unset-function/>.

12. RushenRoshen. Рейтинг языков программирования 2021: доля Python падает, а TypeScript обошел C++, в лидерах JavaScript, Java, C#. Хабр. URL: <https://habr.com/ru/post/543346/>.

13. WebForMySelf - Как создать сайт с нуля, самому и бесплатно. WebForMySelf - публикация тематических материалов по вебразработке и сайтостроению. [Электронный ресурс] – Режим доступа до ресурсу: <https://webformyself.com/>.

14. What is MySQL: MySQL Explained For Beginners. Hostinger Tutorials. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.hostinger.com/tutorials/what-is-mysql>.

15. What is UI design? What is UX design? UI vs UX: What's the difference [Электронный ресурс] – Режим доступа до ресурсу: <https://uxplanet.org/what-is-ui-vs-ux-design-and-the-differenced9113f6612de>.

16. HTML и HTML5. Описание тегов по основным разделам. HTML5BOOK.RU - HTML, CSS, JavaScript и jQuery. [Электронный ресурс] – Режим доступа до ресурсу: <https://html5book.ru/html-html5/>

17. Гущин Д. Стандарты, стили та правила оформления коду PHP. КовельPost: Актуальні новини міста Ковель. URL: <https://kovelpost.com/blogs/9229>.

18. UX/UI design principles to help create an engaging experience. Fresh Consulting. URL: <https://www.freshconsulting.com/insights/blog/7-uiuxprinciples-fresh-web-experiences/>.

19. Реляционные СУБД – сравнение MySQL и SQL сервер. Руководства Hostinger. [Электронный ресурс] – Режим доступа до ресурсу:

<https://www.hostinger.com.ua/rukovodstva/reljacionnye-subdsravnenie-mysql-i-sql-server/>.

20. Выбор CMS для интернет-магазина: сравниваем 4 движка. Продвижение сайтов от 1PS.RU. [Электронный ресурс] – Режим доступа до ресурсу: <https://1ps.ru/blog/sites/vyibiraem-cms-dlya-internet-magazina1/#simpla>.

21. Основы асинхронного програмування на JavaScript. Codeguida. URL: <https://codeguida.com/post/445>.

22. Simplilearn. Header in PHP: The Ultimate Guide to Header Function. Simplilearn.com. URL: <https://www.simplilearn.com/tutorials/phptutorial/header-in-php#>.

23. mysql_fetch_assoc. Mathematical and Computer Sciences- Heriot-Watt University. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.macs.hw.ac.uk/~hwloidl/docs/PHP/function.mysqlifetch-assoc.html#>

24. Why Use PHP in 2022? Main Advantages and Disadvantages. Web and Mobile App Development Company | Light IT. URL: <https://light-it.net/blog/why-usephp-main-advantages-and-disadvantages/>

25. Какая разница между SCSS и Sass? - программирование. Ответы на вопросы пользователей по программированию на русском языке stackoverflow. URL: <https://ask-dev.ru/info/10892/whats-the-differencebetween-scss-and-sass>

26. [Электронный ресурс] - <https://uk.wikipedia.org/wiki/Glovo>

27. [Электронный ресурс] - <https://webcase.com.ua/uk/blog/cho-takoe-web-prilozhenie-vse-vidy/>

28. [Электронный ресурс] - <https://webcase.com.ua/uk/blog/cho-takoe-web-prilozhenie-vse-vidy/>

ДОДАТОК

```
*
body{
  height: 100vh;
  font-family: 'Open Sans';
  overflow: hidden;
}
.wrapper{
  height: 100%;
  margin: 0 auto;
  width: 335px;
  position: relative;
}
h1{
  font-style: normal;
  font-weight: 800;
  font-size: 24px;
  line-height: 33px;
  margin-bottom: 19px;
}
input{
  width: 335px;
  height: 49px;
  border: 1px solid #F8B505;
  border-radius: 3px;
  margin-bottom: 15px;
  outline: none;
  font-style: normal;
  font-weight: 700;
  font-size: 16px;
  line-height: 19px;
}
button{
  height: 55px;
  font-family: inherit;
  display: flex;
  align-items: flex-start;
  font-weight: 800;
  color: #FFF;
  font-size: 18px;
  padding: 20px 0px 20px 20px;
  background-color: #FFD700;
  border-radius: 3px;
  transition: background-color 0.5s;
```

```
cursor: pointer;
&:hover{
    background-color: #fbc550;
}
.checkbox{
    display: flex;
    flex-direction: row;
    margin-top: 20px;
    font-style: normal;
    font-weight: 800;
    font-size: 16px;

    input{
        height: 20px;
        width: 20px;
        margin-right: 5px;
    }
}
.error_psw{
    text-align: center;
}
.img_back{
    position: absolute;
    top: 63.61%;
    left: -793px;
    width: 100%;
    z-index: -1;
}
.logo{
    position: absolute;
    bottom: 87.9%;
    left: 60px;
}
.correct_psw{
    font-weight: bold;
    margin-top: 15px;
    text-align: center;
    border: 2px solid rgb(238, 241, 20);
    border-radius: 10px;
    box-shadow: 0px 7px 24px 2px rgb(255, 230, 5);
    padding: 5px 0 5px;
}
```



```
*::before,
body {
  height: 100vh;
  font-family: "Open Sans";
  overflow: hidden;
}

.wrapper {
  height: 100%;
  margin: 0 auto;
  width: 335px;
  position: relative;
}

form .registration {
  color: #919191;
  margin-bottom: 30px;
}
form label {
  color: #919191;
}

.error_psw {
  text-align: center;
}

.img_back {
  position: absolute;
  top: 63.61%;
  left: -793px;
  width: 100%;
  z-index: -1;
}

.logo {
  position: absolute;
  bottom: 87.9%;
  left: 60px;
}

.correct_psw {
  font-weight: bold;
  margin-top: 15px;
  text-align: center;
}
```

```

border: 2px solid rgb(238, 241, 20);
border-radius: 10px;
-webkit-box-shadow: 0px 7px 24px 2px rgb(255, 230, 5);
        box-shadow: 0px 7px 24px 2px rgb(255, 230, 5);
padding: 5px 0 5px;
}
.logo_profile {
margin-top: 4.1666666667%;
text-align: center;
}
.exit {
margin-top: 500px;
margin-left: 29.2834890966%;
}
button.active {
background-color: #FFF;
color: #F8B505;
}
section {
-webkit-box-flex: 0;
-ms-flex: 0 0 83.28125%;
flex: 0 0 83.28125%;
}
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class Main {
    public static void main(String[] args) {
        Document doc = Jsoup.parse(html); // Парсинг HTML-
        документу

        Elements els = doc.getElementsByAttribute("name");
        // Знаходження елементів з атрибутом "name"

        for (Element item : els) {
            item.addEventListener("click", new
            EventListener() { // Додавання події "click" для кожного
            знайденого елемента
                public void handleEvent(Event e) {
                    item.parentNode.parentNode.removeChild(item.parentNode); //
                    Видалення батьківського елемента з DOM-дерева сторінки
                }
            });
        }
    }
}

```



```

    }
    }
    }
}

public void changeTabs(Event evt, String tabName) {
    Element[] mainSection =
document.getElementsByClassName("main_section");
    for (int i = 0; i < mainSection.length; i++) {
        mainSection[i].style.display = "none";
    }

    Element[] tabButtons =
document.getElementsByClassName("sidebar_buttons-link");
    for (int i = 0; i < tabButtons.length; i++) {
        tabButtons[i].className =
tabButtons[i].className.replace(" active", "");
    }

    document.getElementById(tabName).style.display = "block";
    evt.currentTarget.className += " active";
}
<?php
// $servername = "127.0.0.1";
// $database = "diplom";
// $username = "root";
// $password = "";

$conn = mysqli_connect('127.0.0.1', 'root', '', 'diplom');
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// echo "Connected success";
// mysqli_close($conn);
?>
<?php
    session_start();
    unset($_SESSION['profile']);
    header("Location: ../index.php");
?>
<?php
    session_start();
    require_once 'connect.php';

```

```
$tel_number = $_POST['tel_number'];
$password = md5($_POST['password']);

$user_check = mysqli_query($conn, "SELECT * FROM
`users` WHERE
`tel_number` = '$tel_number' AND `password` =
'$password'");
if (mysqli_num_rows($user_check) > 0) {

    $profile = mysqli_fetch_assoc($user_check);

    $_SESSION['profile'] = [
        'id' => $profile['id'],
        'tel_number' => $profile['tel_number'],
        'name_company' => $profile['name_company']
    ];

    header("Location: ../profile.php");

} else {
    $_SESSION['error'] = 'Невірний логін або
пароль';
    header("Location: ../index.php");
}

?>
<?php
    session_start();
    require_once 'connect.php';

    $name_company = $_POST['name_company'];
    $tel_number = $_POST['tel_number'];
    $password = $_POST['password'];
    $password_conf = $_POST['password_conf'];

    if ($password !== $password_conf) {
        $_SESSION['error'] = 'Паролі не співпадають';
        header("Location: ../register.php");
    } else {

        $hash_password = md5($password);

        mysqli_query($conn, "INSERT INTO `users` (`id`,
```



```

`name_company`,
`tel_number`, `password`) VALUES (NULL,
'$name_company', '$tel_number', '$hash_password');
$_SESSION['correct'] = 'Регістрація виконана!';
header("Location: ../index.php");
}
?>
<?php
    session_start();
    if(!$_SESSION['profile'] ) {
        header("Location: index.php");
    }
?>
<!DOCTYPE html>
<html lang="en">
<body>
    <div class="wrapper_profile">
        <aside class="slidebar">
            <div class="slidebar_function">
                
            <p><?=$_SESSION['profile']['name_company'] ?></p>
            <p><?=$_SESSION['profile']['tel_number'] ?></p>
            </div>
            <div class="slidebar_buttons">
                <button class="slidebar_buttons-link"
onclick="changeTabs(event, 'firstSection')">Додати
товар</button>
                <button class="slidebar_buttons-link"
onclick="changeTabs(event, 'secondSection')">Список
товарів</button>
                <button class="slidebar_buttons-link"
onclick="changeTabs(event, 'thirdSection')">Закази</button>
                <button class="slidebar_buttons-link"
onclick="changeTabs(event, 'forthSection')">Додати
магазин</button>
            </div>
            <div class="exit"><a

```

```

href="vendor/logout.php">Вихід</a></div>
</aside>
<section class="main_section" id="firstSection">
  <div class="main_section_header">
    <div class="main_section_header-
name">Додати товар</div>
    <div class="main_section_header-
logo"></div>
  </div>
  <div class="main_section_main">
    <div class="main_section_main-blocks
block1">
      <div class="main-blocks-name">Головна
інформація</div>
      <form action="">
        <input type="product" name
='product'>
          <div
class="checkfloat">Кілограм<div
class="checkbox"><div><input type="checkbox"
></div></div></div>
          <div class="checkfloat">Грам<div
class="checkbox"><div><input type="checkbox" >
</div></div></div>
          <div
class="checkfloat">Упаковка<div
class="checkbox"><div><input type="checkbox" >
</div></div></div>
          <div class="checkfloat">Штука<div
class="checkbox"><div><input type="checkbox" >
</div></div></div>
        </form>
      </div>
    <div class="main_section_main-blocks
block2">
      <div class="main-blocks-name">Харчова
цінність</div>
      <form id = "flex_start" action="">
        <label>Калорійність</label>
        <input type="product" name
='product'>
        <label>Калорійність</label>
        <input type="product" name
='product'>

```



```

        <label>Калорійність</label>
        <input type="product" name
    ='product'>
        <label>Калорійність</label>
        <input type="product" name
    ='product'>
    </form>
</div>
<div class="main_section_main-blocks
block3">
    <div class="main-blocks-name">Загальна
інформація</div>
    <form id = "flex_start" action="">
        <label>Країна виробник</label>
        <input type="product" name
    ='product'>
        <label>Торгова марка</label>
        <input type="product" name
    ='product'>
        <label>Час придатності</label>
        <input type="product" name
    ='product'>
        <label>Час придатності</label>
        <input type="product" name
    ='product'>
        <button type='button' style = 'max-
width:335px; margin-top:150px'
onclick="addProduct()">Додати товар</button>
    </form>
</div>
</section>
<section class="main_section" id="secondSection">
    <div class="main_section_header">
        <div class="main_section_header-
name">Список товарів</div>
        <div class="main_section_header-
logo"></div>
    </div>
    <div class="main_section_main">
        <div class="main_section_main-blocks
block1">

```

```

<div class="main-blocks-
name">Продукти</div>
  <div class="main-blocks_catalog">
    <div class="main-blocks_catalog-
product" id = 'grid_catalog'>
      <div class="product_main-info">
        <div class="main-
info_text">
          <input name="del"
type="button" value="Видалити"/>
          <a href="" class="red"
>Редагувати</a>
        </div>
        <div class="product_main-info">
          <div class="main-
info_text">
            <input name="del"
type="button" value="Видалити"/>
            <a href="" class="red"
>Редагувати</a>
          </div>
          <div class="product_main-info">
            <div class="main-
info_text">
              ul.insertAdjacentHTML('afterEnd', htmlCode);
            </script>
            <!-- <script>
            var els =
document.getElementsByName("del");
            els.forEach(function(item) {
item.addEventListener("click", function() {
item.parentNode.parentNode.removeChild(item.parentNode);
});
});
            </script> -->
          </div>

```




```
</div>
</div>
</section>
<section class="main_section" id="thirdSection">
  <div class="main_section_header">
    <div class="main_section_header-
name">Закази</div>
    <div class="main_section_header-
logo"></div>
  </div>
  <div class="main_section_main">
    <div class="main_section_main-blocks
block1">
      <div class="zakaz_block">
        <div class="zakaz_block-products">
          <h3 class="zakaz_block-
products_name">Заказ №3465</h3>
          <div class="zakaz_block-
products-grid">
            <div class="products-grid-
item">
              
              <div
class="item_name">Пицца</div>
              <div
class="item_count">1 шт</div>
            </div>
            <div class="products-grid-
item">
              
              <div
class="item_name">Пицца</div>
              <div
class="item_count">2 шт</div>
            </div>
          <div class="products-grid-
item">
            
<div class="item_name"
style='margin-top:12px'>Яблуко</div>
<div
class="item_count">3 кг</div>
</div>
<div class="products-grid-
item">

<div
class="item_name">Піца</div>
<div
class="item_count">1 кг</div>
</div>
<div class="products-grid-
item">

<div
class="item_name">Помідори</div>
<div
class="item_count">2 кг</div>
</div>
<div class="products-grid-
item">

<div
class="item_name">Кавун</div>
<div
class="item_count">1 шт</div>
</div>
<div class="products-grid-
item">

<div class="item_name"
style='margin-top:12px'>Яблуко</div>
<div
class="item_count">2 кг</div>
</div>
<div class="products-grid-

```



```

item">
    
    <div
    class="item_name">Пица</div>
    <div
    class="item_count">1 шт</div>
    </div>
    <div class="products-grid-
item">
    
    <div
    class="item_name">Помідори</div>
    <div
    class="item_count">3 кг</div>
    </div>
    </div>
    <div class="zakaz_block-info">
    <div class="block-info_title"
    style = "font-weight: 700; font-size: 22px; line-height:
    30px;">Умови закау</div>
    <div class="block-info_main-
info">
    <div class="block-
info_main-info_item">Адреса
    <p>Вул. Незалежності,
    б</p>
    </div>
    <div class="block-
info_main-info_item item_top">Спосіб оплати
    <p>Готівка</p>
    </div>
    <div class="block-
info_main-info_item item_top">Спосіб доставки
    <p>Доставка
    кур'єром</p>
    </div>
    </div>
    <div class="info_weight" style
    = 'font-size: 14px; line-height: 17px; color: #B6B6B6;'>Вага
    кошику: 4кг</div>
    <div class="main_price" style =
    "font-weight: 700; font-size: 19px; line-height:

```

25px;"]>Загальна вартість:</div>

грн</div> <div class="item_name">750

</div>

</div>

</div>

</div>

</section>

<section class="main_section" id="forthSection">

<div class="main_section_header">

<div class="main_section_header-
name">Додати магазин</div>

<div class="main_section_header-
logo"></div>

</div>

<div class="main_section_main add_form">

<div class="main_section_main-blocks
block1">

<div class="main-blocks-name"></div>

<div class="wrapper" style = 'padding:
160px 0px 160px 0px; '>

<form action="" method="post">

<h1>Додавання магазину</h1>

<label>Назва

компанії</label>

<input type="name_company"

name='name_company'>

<label>Номер

телефону</label>

<input type="tel_number"

name='tel_number' id= "phone" required>

<button type='button'

class='button_add-document' style='background: #FBDA82;
color: black'>Додати документи</button>

<button type='submit'

style= 'margin-top:15px'>Додати</button>

<div class="checkbox" style

= 'margin-top:15px'><div><input type="checkbox" required>Я
приймаю умови користувальницької <span
style="color: #F8B505 ;">оферти</div></div>

</form>

</div>

</div>

</section>

</div>

</body>
</html>

