

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
(повне найменування вищого навчального закладу)
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
(повне найменування інституту, назва факультету (відділення))
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

бакалавра

(освітньо-кваліфікаційний рівень)

на тему: «Розробка платформи для парсингу інтернет-магазинів
«Rozetka» та «Фокстрот» мовою JavaScript з використанням Node.js»

Виконав: здобувач 4 курсу, групи 4ПР1

напряму підготовки (спеціальності)

121 «Інженерія програмного забезпечення»

(шифр і назва напряму підготовки, спеціальності)

Ізмайлов Ян Данилович

(прізвище та ініціали)

Керівник к.т.н , ст. викладач Кибалко І.І

(прізвище та ініціали)

Рецензент к.т.н., доцент Вороненко М.О.

(прізвище та ініціали)

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут, факультет, відділення Інформаційних технологій та дизайну
Кафедра, циклова комісія Програмних засобів і технологій
Освітньо-кваліфікаційний рівень бакалавр
ОПП Програмна інженерія
(шифр і назва)
Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувача кафедри, голова циклової
комісії програмних засобів і
технологій

_____ к.т.н., доцент О.Е. Огнева
«___» _____ 2025 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ

Ізмайлову Яну Даниловичу

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) «Розробка платформи для парсингу інтернет-магазинів
«Rozetka» та «Фокстрот» мовою JavaScript з використанням Node.js»

керівник проєкту (роботи) к.т.н , ст. викладач Кибалко І.І.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «20 січня» 2025 року № 94-с

2. Строк подання здобувачем

проєкту(роботи)

11.06.2025

3. Вихідні дані до проєкту (роботи) літературні та періодичні джерела,
матеріали переддипломної практики, матеріали курсової роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження та аналіз предметної області, проєктування платформи для парсингу
інтернет-магазинів «Rozetka» та «Фокстрот» ,

програмна реалізація, тестування та впровадження

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

13 Рисунків

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів атестаційного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Отримання завдання	09.02.2024	Виконано
2	Підбір літератури	10.02.2024 – 20.02.2024	Виконано
3	Аналіз предметної області	20.02.2024 – 05.03.2024	Виконано
4	Розробка концептуальної моделі	06.03.2024 – 15.03.2024	Виконано
5	Розробка та проєктування системи	16.03.2024 – 28.03.2024	Виконано
6	Розробка інтерфейсу користувача	10.04.2024 – 15.04.2024	Виконано
7	Написання вихідного коду програми	16.04.2024 – 15.05.2024	Виконано
8	Оформлення пояснювальної записки	16.05.2024- 28.05.2024	Виконано
	Захист кваліфікаційної роботи	11.06.2024	Виконано

Здобувач _____
(підпис)

Я.Д. Ізмайлов _____
(прізвище та ініціали)

Керівник проєкту (роботи) _____
(підпис)

О.С.Комісаров _____
(прізвище та ініціали)

Реферат

Пояснювальна записка: 67 с., 13 рис., 8 додатки, 40 джерел.

Об'єкт дослідження: процес збору та аналізу даних з веб-сайтів інтернет-магазинів Rozetka та Фокстрот. Ці платформи надають велику кількість цінних даних для аналізу та їх обробки.

Предмет дослідження: алгоритми та методи як отримувати, обробляти та зберігати дані з веб-сайтів за допомогою серверної платформи, з акцентом на використанні TypeScript та Node.js для розробки аналітичних функцій та баз даних MySQL та MongoDB для зберігання отриманої інформації. Система. Він також досліджує оптимізацію процесу аналітики для забезпечення продуктивності, точності та масштабованості системи.

Мета роботи: створення ефективної серверної та клієнтської частини платформи, яка може автоматично отримувати дані з веб-сайтів Rozetka та Фокстрот. Основна увага приділяється розробці системи, яка може гнучко адаптуватися до змін у структурі веб-сайтів і точно зберігати дані в реляційних і документно-орієнтованих базах даних та буде зручна і зрозуміла кінцевому користувачу. Інша мета - створити архітектуру, яка дозволить розширити платформу для аналізу інших веб-сайтів.

Результати: була розроблена серверна та клієнтська частина платформи, яка автоматично аналізує дані з веб-сайтів Rozetka та Фокстрот і отримує інформацію про товари, характеристики, ціни та наявність. Система базується на Node.js і може ефективно обробляти великі обсяги даних завдяки асинхронній обробці запитів. Отримані дані зберігаються в MySQL для завдань аналізу, що вимагають реляційних структур, і в MongoDB для зберігання неструктурованих даних для більшої гнучкості. Архітектура платформи робить систему більш адаптивною за рахунок інтеграції з іншими джерелами даних та можливих змін у структурі веб-сайту.

Ключові слова: NODE.JS, TYPESCRIPT, ПАРСИНГ ВЕБ-САЙТІВ, ROZETKA, FOXROT, MYSQL, MONGODB, СЕРВЕРНА ЧАСТИНА, АВТОМАТИЗАЦІЯ, ОБРОБКА ДАНИХ.

АНОТАЦІЯ

У роботі розглядається розробка платформи для парсингу інтернет-магазинів, обробки та зберігання даних з провідних українських інтернет-магазинів Rozetka та Фокстрот. Платформа базується на сучасних технологіях Node.js та TypeScript, що дозволяють ефективно працювати з великими обсягами інформації. Для зберігання даних використовується комбінація двох типів баз даних: реляційної MySQL та документної MongoDB. Проєкт спрямований на автоматизацію процесу збору інформації про товари, їх ціни, наявність та акції, що підвищує швидкість і якість прийняття рішень користувачами та компаніями.

У першому розділі виконується аналіз цільової аудиторії платформи, визначаються основні потреби користувачів та бізнесу. Детально розглядаються існуючі рішення на ринку, зокрема платформи для агрегації каталогів, системи порівняння цін і моніторингу конкурентів, а також сервіси збору даних у реальному часі. Розділ виявляє основні функціональні та нефункціональні вимоги до розроблюваної системи. На основі проведеного аналізу визначається перелік необхідних функцій платформи, які будуть реалізовані у подальших етапах проєкту. Описані основні технічні та бізнес-цілі, що формують основу для проєктування та розробки. Зокрема, увага приділяється автоматизації збору даних та їх актуалізації. Визначено критерії ефективності системи, що забезпечують конкурентні переваги. Розділ містить обґрунтування вибору технологій і підходів.

Другий розділ присвячено проєктуванню системи, зокрема алгоритмів взаємодії клієнта з сервером. Описані архітектурні рішення, що забезпечують стабільну і масштабовану роботу платформи. В розділі викладено принципи побудови алгоритмів обробки даних для інтернет-магазинів, які дозволяють ефективно парсити та агрегувати інформацію. Значну увагу приділено проєктуванню баз даних, що є ключовим етапом системної архітектури. Окремо розглядаються особливості реляційної бази даних MySQL, яка використовується для збереження структурованих даних. Описується

структуру таблиць, зв'язки між ними та індексацію. Аналогічно, наводиться опис проєктування MongoDB для зберігання неструктурованих документів. Розглянуто питання масштабованості та гнучкості схеми документної бази. Розділ завершується обґрунтуванням вибору інструментів і технологій для роботи з базами даних.

Третій розділ присвячений практичній реалізації системи. У ньому детально описано створення баз даних, включаючи SQL-скрипти та налаштування моделей ORM для MySQL. Наведено приклади реалізації моделей даних у Sequelize, що спрощують роботу з реляційною базою. Аналогічно, описано створення схем і моделей для MongoDB з використанням Mongoose. Представлені алгоритми обробки даних для конкретних інтернет-магазинів, таких як Rozetka та Фокстрот. Вказано, як здійснюється парсинг сторінок, вилучення релевантної інформації та її подальша обробка. Описано механізми синхронізації даних між базами, а також роботу з асинхронними операціями. Розглянуто методи обробки помилок і логування процесів. Розділ містить також реалізацію підключення до баз даних через відповідні бібліотеки.

Четвертий розділ зосереджений на тестуванні розробленої системи. Описано побудову тест-плану, який включає як функціональне, так і нефункціональне тестування. Наведено приклади сценаріїв для перевірки коректності роботи серверної частини, обробки запитів і цілісності даних. Розглянуто тестування продуктивності, навантажень та стресових ситуацій. Описані методи автоматизації тестування та використання відповідних інструментів. Розділ також містить аналіз результатів тестування, виявлених помилок та способів їх усунення. Приділено увагу забезпеченню стабільності системи при масштабуванні. Визначено критерії успішності тестування для кожного виду перевірок. Заключна частина включає рекомендації щодо покращення системи.

П'ятий розділ детально описує процес встановлення та розгортання проєкту. Наведено покрокові інструкції з інсталяції необхідних компонентів, налаштування середовищ розробки і виробництва. Розглянуто конфігурації для

підключення до баз даних та забезпечення їх безперервної роботи. Представлені методи розгортання серверної платформи на хостинг-платформах. Описано налаштування безпеки, зокрема роботу з обліковими записами та шифруванням даних. Розділ також містить рекомендації щодо підтримки і оновлення системи. Обговорено способи моніторингу працездатності та логування. Надано інструкції з резервного копіювання і відновлення баз даних.

У висновках роботи узагальнено основні результати виконаного дослідження та розробки. Детально проаналізовано ключові етапи реалізації проєкту, що дозволило досягти поставлених цілей. Наведено конкретні докази успішної реалізації, підтверджені практичними тестуваннями та результатами впровадження, які свідчать про ефективність запропонованих рішень.

Список використаних джерел містить фахову літературу та наукові публікації, присвячені розробці програмного забезпечення, роботі з базами даних а також методам тестування та забезпечення якості програмних продуктів. Крім того, до списку включено сучасні дослідження в галузі обробки великих обсягів даних та інтеграції різнорідних баз даних. Використані джерела забезпечують теоретичну та практичну базу для розробки системи, що дозволяє ефективно поєднувати реляційні та документні бази даних у межах проєкту.

ABSTRACT

This paper discusses the development of a platform for parsing online stores, processing and storing data from the leading Ukrainian online stores Rozetka and Foxtrot. The platform is based on modern technologies Node.js and TypeScript, which allow to work efficiently with large amounts of information. A combination of two types of databases is used to store data: relational MySQL and documentary MongoDB. The project is aimed at automating the process of collecting information about products, their prices, availability and promotions, which increases the speed and quality of decision-making by users and companies.

The first section analyzes the platform's target audience and identifies the main needs of users and businesses. The section also analyzes existing solutions on the market, including catalog aggregation platforms, price comparison and competitor monitoring systems, and real-time data collection services. The section identifies the main functional and non-functional requirements for the system to be developed. Based on the analysis, a list of necessary platform functions is determined, which will be implemented in the subsequent stages of the project. The main technical and business goals that form the basis for design and development are described. In particular, attention is paid to the automation of data collection and data updating. The system performance criteria that provide competitive advantages are identified. The section contains a justification for the choice of technologies and approaches.

The second section is dedicated to the system design, including algorithms for client-server interaction. It describes architectural solutions that ensure stable and scalable operation of the platform. The chapter outlines the principles of building data processing algorithms for online stores that allow efficient parsing and aggregation of information. Considerable attention is paid to database design, which is a key stage of system architecture. The features of the MySQL relational database, which is used to store structured data, are discussed separately. The structure of tables, relationships between them, and indexing are described. Similarly, a description of the design of MongoDB for storing unstructured documents is given. The issues of scalability and

flexibility of the document database schema are considered. The chapter concludes with a justification for the choice of tools and technologies for working with databases. The third section is devoted to the practical implementation of the system. It describes in detail the creation of databases, including SQL scripts and the configuration of ORM models for MySQL. Examples of implementing data models in Sequelize are provided to simplify the work with a relational database. Similarly, the creation of schemas and models for MongoDB using Mongoose is described. Data processing algorithms for specific online stores such as Rozetka and Foxtrot are presented. It is shown how pages are parsed, relevant information is extracted, and further processed. Mechanisms for synchronizing data between databases and working with asynchronous operations are described. Methods of error handling and logging of processes are considered. The section also contains the implementation of connecting to databases through the appropriate libraries.

The fourth chapter focuses on testing the developed system. It describes the construction of a test plan that includes both functional and non-functional testing. Examples of scenarios for checking the correctness of the server side, query processing, and data integrity are provided. Performance, load, and stress testing are considered. Methods of test automation and the use of appropriate tools are described. The chapter also contains an analysis of test results, errors found, and ways to fix them. Attention is paid to ensuring system stability during scaling. Test success criteria are defined for each type of test. The final part includes recommendations for improving the system.

The fifth section describes in detail the process of installing and deploying the project. It provides step-by-step instructions for installing the necessary components, setting up development and production environments. Configurations for connecting to databases and ensuring their continuous operation are considered. Methods for deploying the server platform on hosting platforms are presented. Security settings are described, including working with accounts and data encryption. The chapter also contains recommendations for maintaining and updating the system. Methods of

performance monitoring and logging are discussed. Instructions are provided for backing up and restoring databases.

The conclusions of the paper summarize the main results of the research and development. The key stages of the project implementation are analyzed in detail, which allowed us to achieve the set goals. Specific evidence of successful implementation is provided, confirmed by practical testing and implementation results, which demonstrate the effectiveness of the proposed solutions.

The list of references includes professional literature and scientific publications on software development, database management, and methods of testing and quality assurance of software products. In addition, the list includes current research in the field of big data processing and integration of heterogeneous databases. The sources used provide a theoretical and practical basis for the development of a system that allows you to effectively combine relational and document databases within a project.

ЗМІСТ

АНОТАЦІЯ	3
ABSTRACT	6
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ВИМОГ СИСТЕМИ	7
1.1. Аналіз цільової аудиторії	7
1.2. Розгляд існуючих рішень	8
1.2.1 Платформи для агрегації каталогів	9
1.2.2 Системи для порівняння цін і моніторингу конкурентів	10
1.2.3 Платформи для збору даних у реальному часі	11
1.3. Визначення функціоналу проєкту	12
РОЗДІЛ 2. ПРОЄКТУВАННЯ АЛГОРИТМІВ ТА БАЗ ДАНИХ	14
2.1. Проєктування взаємодії клієнта з сервером	14
2.2 Проєктування алгоритмів роботи програми	16
2.3 Проєктування бази даних	21
2.3.1 Проєктування реляційної бази даних MySQL	21
2.3.2 Проєктування не реляційної бази даних MongoDB	28
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМИ ПРОЄКТУ	31
3.1 Розробка баз даних	31
3.1.1 Розробка реляційної бази даних	31
3.1.2 Розробка не реляційної бази даних	35
3.2. Розробка алгоритмів проєкту	36
3.2.1 Алгоритми обробки даних інтернет магазину Фокстрот	36
3.2.2 Алгоритми обробки даних інтернет магазину Розетка	42
3.3 Розробка підключення до баз даних	47

	3
3.3.1 Підключення до MySQL за допомогою Sequelize	47
3.3.2 Підключення до MongoDB за допомогою Mongoose	49
РОЗДІЛ 4. ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	52
4.1. Побудова тест-плану програмного додатку	52
4.2. Функціональне тестування серверного додатку	52
4.3. Нефункціональне тестування серверного додатку	55
РОЗДІЛ 5 ВСТАНОВЛЕННЯ ТА РОЗГОРТАННЯ ПРОЄКТУ	58
5.1 Встановлення компонентів проєкту	58
5.2 Встановлення та розгортання баз даних	60
ВИСНОВКИ	63
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	65
ДОДАТОК А. КОД РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ	68
ДОДАТОК Б. КОД КЛАСУ ПІДКЛЮЧЕННЯ ДО MYSQL БАЗИ ДАНИХ	72
ДОДАТОК В. КОД КЛАСУ ПІДКЛЮЧЕННЯ ДО MONGODB БАЗИ ДАНИХ	73
ДОДАТОК Г. КОД АЛГОРИТМІВ ОБРОБКИ ТА ЗБОРУ ДАНИХ З МАГАЗИНУ ФОКСТРОТ	75
ДОДАТОК Д. КОД АЛГОРИТМІВ ОБРОБКИ ТА ЗБОРУ ДАНИХ З МАГАЗИНУ РОЗЕТКА	87
ДОДАТОК Е. ДІАГРАМА КЛАСІВ ВЗАЄМОДІЇ КОНТРОЛЕРІВ З БАЗОЮ ДАНИХ	96
ДОДАТОК Ж. ДІАГРАМА КЛАСІВ ВЗАЄМОДІЇ ДАНИХ З МАГАЗИНУ ФОКСТРОТ З БАЗОЮ ДАНИХ	97
ДОДАТОК З. ДІАГРАМА КЛАСІВ ВЗАЄМОДІЇ ДАНИХ З МАГАЗИНУ ROZETKA З БАЗОЮ ДАНИХ	98

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

БД — бази даних

Системи для зберігання та управління даними у структурованому вигляді.

ПЗ — програмне забезпечення

Комплекти програм, що виконують певні функції на комп'ютерах чи інших пристроях.

СУБД — система управління базами даних

Програмне забезпечення для створення, організації, керування, обробки та захисту баз даних.

API — прикладний програмний інтерфейс

Інтерфейс для взаємодії між програмами або компонентами програмного забезпечення. API дозволяє одній програмі взаємодіяти з іншою через чітко визначені функції або методи.

HTTP — протокол передачі гіпертексту

Протокол передачі даних, що використовується для обміну інформацією в інтернеті, зокрема для роботи вебсайтів.

MySQL — система керування реляційними базами даних

Популярна СУБД, яка використовує мову SQL для керування та обробки реляційних баз даних. Підтримує різноманітні платформи та має відкритий вихідний код.

MongoDB — документо-орієнтована система керування базами даних

Система керування базами даних, що зберігає інформацію у вигляді документів (JSON-подібні структури). Використовується для великих обсягів неструктурованих даних.

JSON — текстовий формат обміну даними між комп'ютерами

Формат для зберігання та передачі даних у вигляді пар "ключ-значення". Зручний для обміну інформацією між різними мовами програмування.

Visual Studio Code (VS Code) – це безкоштовне інтегроване середовище розробки (IDE) від Microsoft. Розробка сайтів і додатків включає Visual Studio

Code. VS Code має розширення для роботи з різними мовами програмування, включаючи PHP, і надає інструменти для редагування коду та роботи з системами контролю версій

Node.js — це однопоточне кросплатформове середовище виконання з відкритим вихідним кодом і бібліотека, яка використовується для запуску веб-додатків, написаних на JavaScript, поза браузером клієнта.

Cheerio — це бібліотека для роботи з HTML та XML документами в середовищі Node.js. Вона дозволяє зручно парсити HTML-код, маніпулювати ним та витягувати необхідні дані, подібно до того, як працює бібліотека jQuery в браузерах.

Puppeteer — це бібліотека для Node.js, яка надає високорівневий API для управління браузером Chrome або Chromium за допомогою протоколу DevTools. Puppeteer дозволяє автоматизувати різноманітні завдання в браузері, такі як веб-скрапінг, тестування інтерфейсу користувача, генерація скріншотів, створення PDF-документів і багато іншого.

ВСТУП

Сьогодні онлайн-покупки стали невід'ємною частиною повсякденного життя, надаючи доступ до широкого спектру товарів та послуг. Такі платформи, як Rozetka та Фокстрот, є лідерами українського ринку онлайн-покупок, надаючи мільйонам користувачів можливість швидко знаходити необхідні товари, порівнювати ціни та характеристики, а також визначати найкращі пропозиції. Однак, щоб отримати актуальну інформацію про товари, ціни, акції та наявність, користувачам доводиться відвідувати безліч сторінок і вручну перевіряти наявність оновлень на різних сайтах. Це забирає багато часу та зусиль і не завжди призводить до вибору найкращих варіантів. Тому важливо автоматизувати цей процес як для компаній, так і для споживачів. В умовах постійних змін важливо не лише збирати актуальну інформацію, а й отримувати стислу та змістовну інформацію для прийняття обґрунтованого рішення. Для того щоб це було можливо необхідно мати інструменти, які дозволяють швидко аналізувати ці дані. Метою цього проєкту є розробка серверної платформи, яка автоматично збирає дані з веб-сайтів Rozetka та Фокстрот. Завдяки використанню сучасних технологій, таких як Node.js та TypeScript, платформа буде ефективно обробляти великі обсяги даних та зберігати їх у відповідних базах даних MySQL та MongoDB для подальшого аналізу. Рішення має значні переваги як для кінцевих користувачів, так і для компаній. Користувачі отримують зручний інструмент для швидкого порівняння товарів, заощаджують час та роблять шопінг приємнішим. Для компаній автоматизований збір даних дозволяє швидко реагувати на зміни на ринку, коригувати ціни та асортимент, розробляти стратегії продажів на основі достовірної та актуальної інформації. Збір даних з різних джерел також дозволяє їм краще розуміти конкурентне середовище та адаптуватися до нових ситуацій. Тому розробка такої платформи є важливим кроком не лише в автоматизації збору даних, а й у створенні корисного інструменту аналізу та планування для всіх учасників ринку.