Міністерство освіти і науки України
Кам’янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп’ютерних наук

Кваліфікаційна робота бакалавра

з теми: **«Проєктування та розробка вебзастосунку з функціональністю CRUD для ефективного керування базами даних на основі об’єктно-реляційного підходу»**

Виконав: здобувач вищої освіти групи KN1-B21
спеціальності 122 Комп’ютерні науки
Лискун Ростислав Володимирович

Керівник: Смалько Олена Аркадіївна,
кандидат педагогічних наук,
доцент кафедри комп’ютерних наук

Рецензент: Оптасюк Сергій Васильович,
кандидат фізико-математичних наук,
доцент, завідувач кафедри фізики

Кам’янець-Подільський – 2025 р.

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці вебзастосунку для обліку, управління та аналізу агровиробничих даних із реалізацією базової функціональності CRUD (створення, читання, оновлення, видалення). Такий інструмент спрямований на цифровізацію процесів збору та обробки інформації у малих та середніх фермерських господарствах, що сприяє ефективному управлінню аграрними ресурсами. Вебзастосунок розроблений із використанням мікрофреймворку Flask для побудови серверної логіки та MySQL як системи управління базами даних.

Окрім реалізації базових операцій взаємодії з агроданими, система доповнена підсистемою контролю витрат на добрива, що дозволяє проводити автоматизований розрахунок фінансових витрат на основі введених параметрів. Ще одним важливим компонентом є можливість генерації звітів у форматі PDF, що дає змогу зберігати результати у вигляді формалізованої документації, зручною для друку або подальшого аналізу.

Результати роботи свідчать про практичну доцільність застосування відкритого технологічного стеку для розробки агроінформаційних систем. Розроблений вебзастосунок може бути впроваджений у діяльність фермерських господарств для автоматизації агровиробничих процесів, а також використаний як дидактичний матеріал у межах освітніх програм, що охоплюють вебпрограмування, обробку даних і розробку CRUD-систем.

Ключові слова: вебзастосунок, агровиробничі дані, база даних, CRUD, Flask, автоматизація.

ABSTRACT

The qualification thesis is devoted to the development of a web application for the management, registration, and analysis of agri-production data with the implementation of CRUD functionality (Create, Read, Update, Delete). This tool aims to support the digital transformation of data management processes in small and medium-sized farms, contributing to more efficient agricultural resource control. The application is developed using the Flask microframework for server-side logic and MySQL as the database management system.

In addition to standard data operations, the system features a fertilizer cost control module that automates expense calculations based on input parameters. Another significant functionality is PDF report generation, allowing users to archive key results in a formalized format suitable for printing and further evaluation.

The results confirm the practical viability of using an open-source technology stack in building agricultural information systems. The developed application may be integrated into farm operations for process automation or used as educational material in courses related to web development, data processing, and CRUD system design.

Keywords: web application, agri-production data, database, CRUD, Flask, automation.

ЗМІСТ

АНОТАЦІЯ 2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ і СКОРОЧЕНЬ 5
ВСТУП 6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКІВ ТА УПРАВЛІННЯ БАЗАМИ ДАНИХ 9
1.1. Основи архітектури вебзастосунків 9
1.2. CRUD-операції як фундамент функціональності вебсистем 11
Висновки до розділу 1 13
РОЗДІЛ 2. ОБ'ЄКТНО-РЕЛЯЦІЙНИЙ ПІДХІД В УПРАВЛІННІ БАЗАМИ ДАНИХ 15
2.1. Переваги об'єктно-реляційного підходу 15
2.2. Обґрунтування вибору інструментів для реалізації ORM 17
Висновки до розділу 2 20
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ З ФУНКЦІОНАЛЬНІСТЮ CRUD 22
3.1. Проєктування та реалізація архітектури застосунку 22
3.2. Реалізація CRUD-функціоналу на основі ORM 25
3.3. Реалізація додаткового функціоналу вебзастосунку 26
Висновки до розділу 3 28
ВИСНОВКИ 30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 32
ДОДАТКИ 32

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ і СКОРОЧЕНЬ

CRUD – Create, Read, Update, Delete (Створення, Читання, Оновлення, Видалення) – базові операції з даними.

ORM – Object-Relational Mapping (Об'єктно-реляційне відображення) – технологія програмування для роботи з базами даних.

DB – Database (База даних) – структурований набір даних.

SQL – Structured Query Language (Мова структурованих запитів) – мова для роботи з реляційними базами даних.

ВСТУП

У сучасних умовах цифрової трансформації суспільства все більшої актуальності набувають інформаційні технології, які забезпечують збирання, зберігання, обробку та візуалізацію даних. Особливу роль у цьому процесі відіграють вебзастосунки — динамічні програмні рішення, що функціонують у мережевому середовищі та забезпечують інтерактивну взаємодію користувача з даними. Підвищення попиту на цифрові платформи для автоматизації управлінських процесів, зокрема у галузях, які традиційно не пов’язуються з інтенсивною цифровізацією, як-от сільське господарство, обумовлює необхідність створення інструментів, здатних адаптуватися до специфічних потреб користувача та забезпечувати гнучке управління інформаційними ресурсами.

У центрі проєктування більшості веборієнтованих систем лежить концепція CRUD, яка передбачає чотири базові операції з даними — створення, читання, оновлення та видалення. Саме ця функціональність є основою будь-якої системи керування базами даних, незалежно від складності предметної області. Застосування CRUD-парадигми дозволяє забезпечити логічну цілісність даних, зручність взаємодії та можливість масштабування інформаційної системи відповідно до потреб користувача.

Реалізація вказаної парадигми у вебзастосунках потребує не лише належного проєктування логіки взаємодії з базою даних, але й вибору ефективного способу зв’язку між об’єктною моделлю застосунку та реляційною структурою бази. У цьому контексті об’єктно-реляційне відображення (ORM) постає як один із найефективніших підходів. ORM забезпечує автоматичне узгодження між об’єктами у програмному коді та таблицями в реляційній базі даних, що не лише спрощує розробку та зменшує обсяг низькорівневого коду, але й підвищує супровідність і безпеку програмного забезпечення.

Метою кваліфікаційної роботи є створення веборієнтованого застосунку з реалізацією функціональності CRUD, спрямованого на управління агровиробничими даними з використанням об'єктно-реляційного підходу. Реалізація такого підходу дозволяє ефективно структурувати облік аграрних ресурсів, раціоналізувати процеси застосування добрив та культур, а також забезпечити доступну і наочну візуалізацію інформації для подальшого аналізу. Наукова новизна дослідження полягає в інтеграції сучасних вебтехнологій у контексті цифровізації агропромислового виробництва.

Задля досягнення визначеної мети в роботі виконувалися наступні **завдання**:

- аналіз актуальних тенденцій цифрової трансформації в аграрній галузі з акцентом на обробку агровиробничих даних;[6]
- проєктування архітектури вебзастосунку відповідно до вимог модульності, масштабованості та адаптивності;
- реалізація CRUD-функціональності з безпосередньою взаємодією із реляційною базою даних MySQL та з використанням бібліотеки PyMySQL;
- розробка підсистеми автоматизованого обліку витрат на добрива на основі агрономічних параметрів;
- впровадження механізму генерації зведених звітів у форматі PDF з подальшим збереженням інформації у структурованому вигляді;
- тестування працездатності основних компонентів системи та валідацію даних на всіх етапах обробки.

Об'єктом дослідження є процеси організації, проєктування та реалізації веборієнтованих інформаційних систем, що забезпечують взаємодію з реляційними базами даних у режимі реального часу.

Предметом дослідження виступають технології реалізації CRUD-функціональності у вебзастосунках із використанням ORM як інструменту синхронізації між логічною (об’єктною) та фізичною (реляційною) моделями даних.

Під час виконання дослідження було використано низку наукових методів. Теоретичну базу формували методи системного аналізу, які дозволили встановити структурні залежності між елементами вебсистеми та сформулювати технічні вимоги до неї. Методи моделювання застосовувалися для побудови логічної структури бази даних і взаємодії компонентів. Практичний етап реалізації спирався на методи об’єктно-орієнтованого програмування, зокрема при розробці серверної частини та взаємодії з ORM-бібліотекою. Прототипування стало основою для побудови інтерфейсу користувача, тоді як методи тестування були використані для перевірки коректності функціонування вебзастосунку в цілому.

Структура кваліфікаційної роботи побудована відповідно до логіки реалізації поставленої мети та охоплює три основні розділи. У першому розділі проаналізовано теоретичні основи побудови вебзастосунків, особливості реалізації CRUD-функціоналу та ключові підходи до побудови архітектури клієнт-серверних систем. У другому розділі визначено переваги використання об’єктно-реляційного підходу в процесі управління даними, а також обґрунтовано вибір технологій та інструментальних засобів серед популярних ORM-рішень. У третьому розділі представлено практичну реалізацію вебзастосунку, включно з описом архітектури системи, логіки взаємодії з базою даних, реалізації CRUD-функціональності та інтерфейсного компонента.

Список використаних джерел містить 35 позицій. Додатковий ілюстративний матеріал розміщено у додатках.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи здійснено повноцінний цикл дослідження, спрямованого на розробку веборієнтованого застосунку для управління агровиробничими даними. Запропоновано й реалізовано архітектурно-функціональну модель інформаційної системи, що базується на об'єктно-реляційному підході до структурування даних. Вебзастосунок підтримує повноцінний набір CRUD-операцій та забезпечує зручний інтерфейс користувача для ведення цифрового обліку аграрних ресурсів.

У процесі реалізації поставленої мети було проаналізовано поточні тенденції цифрової трансформації агросектору, визначено потреби щодо обліку агровиробничої інформації, а також спроєктовано та реалізовано низку ключових функціональних компонентів системи. Зокрема, створено інтерфейс для введення, редагування та видалення даних, розроблено підсистему автоматизованого обліку витрат на добрива та інтегровано механізм формування звітів у форматі PDF.

Виконане дослідження не лише підтвердило доцільність застосування сучасних технологій веброзробки у контексті аграрного управління, а й допомогло створити корисну платформу автоматизованого обліку витрат на добрива на основі агрономічних параметрів, здатну масштабуватись, розвиватись та відповідати потребам цифрового сільського господарства.

Практична цінність розробленого програмного забезпечення полягає у його здатності автоматизувати ключові процеси агровиробничого обліку, знизити навантаження на персонал, забезпечити оперативний доступ до актуальної інформації та мінімізувати ймовірність технічних або організаційних помилок при веденні записів. Розроблена система є корисною для малих і середніх фермерських господарств, яким часто бракує ресурсів для впровадження дорогих ERP-рішень. Завдяки відкритій архітектурі та використанню поширених технологій веброзробки, створене програмне забезпечення може бути легко адаптоване, масштабоване або інтегроване в

існуючі інформаційні середовища. Також можливе його використання в якості базової платформи для створення комплексних агроінформаційних систем із розширеним функціоналом — наприклад, модулем супутникового моніторингу, агрохімічного аналізу або з мобільним доступом. Окрім практичного застосування в агропромисловому виробництві, розроблений вебзастосунок є цінним навчальним ресурсом для студентів спеціальності «Комп’ютерні науки», оскільки демонструє принципи побудови інтерактивних клієнт-серверних систем, реалізації CRUD-функціональності, інтеграції з реляційними базами даних та створення звітності на основі введених даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Grinberg M. The Flask mega-tutorial (2024 Edition). *Miguelgrinberg.com*. December 3. 2023. URL: https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world (дата звернення: 24.05.2025).
2. Krebs B., Martinez J. C. Developing RESTful APIs with Python and Flask. *Auth0 Blog*. January 7, 2025. URL: https://auth0.com/blog/developing-restful-apis-with-python-and-flask (дата звернення: 24.05.2025).
3. Flask Tutorial. *GeeksforGeeks*. March 16, 2025. URL: https://www.geeksforgeeks.org/flask (дата звернення: 24.05.2025).
4. PyMySQL Documentation. *PyMySQL*. Inada Naoki and GitHub, 2023. URL: https://pymysql.readthedocs.io (дата звернення: 24.05.2025).
5. Liu Q., Wu J. Research on agricultural data processing based on MySQL. *Agricultural & Forestry Economics and Management (2024).* DOI: 10.23977/agrfem.2024.070203.
6. Cravero A., Pardo S., Sepúlveda S., Muñoz L. Challenges to use Machine Learning in Agricultural Big Data. *Agronomy. 2022*. DOI:10.3390/agronomy12030748.
7. Clark W. E. REST APIs Step by Step: A Practical Guide with examples. Walzone Press, 2025. ISBN 9798227737656.
8. Bazhanau R. Mastering SQLAlchemy: A comprehensive guide for Python developers. *Medium*. November 6, 2024. URL: https://medium.com/@ramanbazhanau/mastering-sqlalchemy-a-comprehensive-guide-for-python-developers-ddb3d9f2e829 (дата звернення: 24.05.2025).
9. Brown T. SQL Injection Prevention Techniques Cheat Sheet. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html (дата звернення: 24.05.2025)
10. Habib A. Digital Transformation in Agriculture – Key Insights and Essential Guide for 2025. *Folio3 AgTech Blog*. November 15, 2024. URL:

https://agtech.folio3.com/blogs/digital-transformation-in-agriculture (дата звернення: 24.05.2025)

11. Digital Transformation in Agriculture: Innovation and the future. *Kaila.eu Blog*. December 5, 2024. URL: https://kaila.eu/blog/digital-transformation-in-agriculture-innovation-and-the-future (дата звернення: 24.05.2025)
12. Gard R. Quickly build secure microservices in Python. *DEV Community*. October 13, 2022. URL: https://dev.to/cowofevil/quickly-build-secure-microservices-in-python-1n44 (дата звернення: 24.05.2025)
13. Gray A. How will tech enable agrifood trends of 2024? *Purdue Agribusiness*, January 25, 2024. URL: https://agribusiness.purdue.edu/2024/01/25/how-will-tech-enable-agrifood-trends-of-2024 (дата звернення: 24.05.2025)
14. Savonin M. Agriculture technology trends for 2024 and beyond. *Keenethics Blog*. September 15, 2024. URL: https://keenethics.com/blog/agriculture-technology-trends-for-2024-and-beyond (дата звернення: 24.05.2025)
15. Revolutionizing agricultural extension: Digital technologies empower cooperative education and outreach. *Farmonaut Blog*. URL: https://farmonaut.com/blogs/digital-tech-transforming-agricultural-extension-in-2024 (дата звернення: 24.05.2025)
16. Oduor G., Macharia M., Rware H. I., Kaizzi K. C. Fertilizer use optimization approach: An innovation to increase agricultural profitability for African farmers. *African Journal of Agricultural Research*. 2016, 11(38):3587–3597. DOI: 10.5897/AJAR2016.11408.
17. Mărcuță C., MoldStud Team. Enhance the efficiency of your Python projects with Flask-security for effortless authentication solutions. *MoldStud*. February 15, 2025. URL: https://moldstud.com/articles/p-enhance-the-efficiency-of-your-python-projects-with-flask-security-for-effortless-authentication-solutions (дата звернення: 24.05.2025)
18. Tiwari S. Building Flask REST App with Flask-RESTful. *Medium*. May 30, 2025. URL: https://medium.com/analytics-vidhya/building-flask-rest-app-with-flask-restful-d22ba4c0e39f (дата звернення: 24.05.2025)

19. Walia A. S. How to optimize the performance of a flask application. *DigitalOcean Community*. September 13, 2024. URL: https://www.digitalocean.com/community/tutorials/how-to-optimize-flask-performance (дата звернення: 24.05.2025)
20. How to Optimize Your Flask Web App for performance. *MuneebDev*. September 25, 2024. URL: https://muneebdev.com/how-to-optimize-your-flask-web-app-for-performance (дата звернення: 24.05.2025)
21. Ghimire A. Machine Learning Model with Flask REST API. *HackerNoon*. January 9, 2020. URL: https://hackernoon.com/machine-learning-w22g322x (дата звернення: 24.05.2025)
22. How to Build a Machine Learning API with Python and Flask. *Statworx Blog*. July 29, 2020. URL: https://www.statworx.com/en/content-hub/blog/how-to-build-a-machine-learning-api-with-python-and-flask (дата звернення: 24.05.2025)
23. Martinho P., Franco T. How to build a Flask API with Python: The complete guide. *Imaginary Cloud Blog*. April 15, 2025. URL: https://www.imaginarycloud.com/blog/flask-python (дата звернення: 24.05.2025)
24. Galindo A. T. Flask API tutorial: Build, document, and secure a REST API. *Zuplo.com*. March 29, 2025. URL: https://zuplo.com/blog/2025/03/29/flask-api-tutorial (дата звернення: 24.05.2025)
25. Thai C. Performance optimization in flask: Tips and tricks for making flask applications faster and more scalable. *Medium.com*. April 2024. URL: https://medium.com/@christopherthai/performance-optimization-in-flask-tips-and-tricks-for-making-flask-applications-faster-and-more-07b9327277b3 (дата звернення: 24.05.2025)
26. Bhatt M. Building scalable web applications with python flask: Best practices. July 2024. URL: https://codymohit.com/building-scalable-web-applications-with-python-flask-best-practices (дата звернення: 24.05.2025)
27. Patel H. Scaling strategies for microservices: Best practices and implementation. *Medium*. March 24, 2025. URL: https://medium.com/

@hitendra.patel2986/scaling-strategies-for-microservices-best-practices-and-implementation-42093d835b25 (дата звернення: 24.05.2025)

28. Mahalias I. Microservices Python Development: 10 Best *Practices*. *Planeks* Blog. March 27, July 2024. URL: https://www.planeks.net/microservices-development-best-practices (дата звернення: 24.05.2025)
29. Yadav R. Complete guide to Python frameworks for scalable microservices. *Peerbits Blog*. October 09, 2024. URL: https://www.peerbits.com/blog/guide-to-python-frameworks-for-scalable-microservices.html (дата звернення: 24.05.2025)
30. Andersen G., MoldStud Research Team. Best practices for designing scalable microservices — A Guide for architects. *MoldStud Blog*. June, 2025. URL: https://moldstud.com/articles/p-best-practices-for-designing-scalable-microservices-a-guide-for-architects (дата звернення: 24.05.2025)
31. Dyouri A., MacDonald B. How to use Flask-SQLAlchemy to interact with databases in a Flask application. *DigitalOcean Community*. March 10, 2022. URL: https://www.digitalocean.com/community/tutorials/how-to-use-flask-sqlalchemy-to-interact-with-databases-in-a-flask-application (дата звернення: 24.05.2025)
32. Dyouri A. How To Create Your First Web Application Using Flask and Python 3. Tutorial. *DigitalOcean Community*. August 19, 2021. URL: https://www.digitalocean.com/community/tutorials/how-to-create-your-first-web-application-using-flask-and-python-3 (дата звернення: 24.05.2025).
33. Theofilou A., Nastis S. A., Tsagris M., Rodriguez-Perez S., Mattas K. Design and Implementation of a Scalable Data Warehouse for Agricultural Big Data. *Sustainability*. 2025, 17(8), article 3727. doi:10.3390/su17083727.
34. Wu S., Mueller T. A User-Friendly NoSQL Framework for Managing Agricultural Field Trial Data. *Scientific Reports*. 2024, 14, article 29819. doi:10.1038/s41598-024-81609-2.
35. Ngo V. M., Le-Khac N.-A., Kechadi M. An Efficient Data Warehouse for Crop Yield Prediction. *arXiv*. 2018. Proceedings of the 14th International Conference on Precision Agriculture, Montreal, 24–27 June 2018.