МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»	**«До захисту допущено»**
Директор інституту (декан факультету)	Завідувач кафедри
_____ Андрій ФОРСЮК	_____ Сергій ГРИБКОВ
(підпис) (прізвище та ініціали)	(підпис) (прізвище та ініціали)
«02» ____червня_____ 2025 р.	«02» ____червня_____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 комп'ютерні науки
(код і назва спеціальності)

освітньо-професійної програми Інформаційні системи та штучний інтелект

на тему: Розроблення вебзастосунку для вихователя ДНЗ

Виконав: здобувач 4 курсу, групи КН-4-4 .

Арзамасов Андрій Миколайович _____
(прізвище, ім'я та по батькові повністю) (підпис)

Керівник Ліманська Наталія Володимирівна _____
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти _____ _____
(ім'я та прізвище) (підпис)

_____ _____
(ім'я та прізвище) (підпис)

_____ _____
(ім'я та прізвище) (підпис)

Рецензент _____ _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач_____
(підпис)

Київ — 2025 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра інформаційних технологій, штучного інтелекту і кібербезпеки
Освітній ступінь бакалавр
Спеціальність 122 комп'ютерні науки
(код і назва)
Освітньо-професійна програма Інформаційні системи та штучний інтелект
(назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних технологій, штучного інтелекту і кібербезпеки _____ Сергій ГРИБКОВ

«28» квітня 2025 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Арзамасова Андрія Миколайовича
(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення вебзастосунку для вихователя ДНЗ»
, керівник роботи Ліманська Наталія Володимирівна,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «28» квітня 2025 р. № 254-кс
2. Строк подання здобувачем роботи: 30.05.2025 р.
3. Вихідні дані до роботи: *Дані про працівників, дані про облік предеметів в ДНЗ, дані про дітей*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
1. Розроблення архітектури веб-застосунку
2. Створення бази даних MySQL
3. Реалізація основних функцій для веб-застосунку

5. Перелік графічного матеріалу:
1. Організаційна структура установи
2. Функціональні моделі
3. Моделі та схеми бази даних
4. Скриншоти інтерфейсу користувача

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ліманська Н.В.	28.04.2025	12.05.2025
2	Ліманська Н.В.	28.04.2025	19.05.2025
3	Ліманська Н.В.	28.04.2025	26.05.2025

7. Дата видачі завдання: 28 квітня 2025

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз структури підприємства	29.04.2025	
2	Постановка задачі	30.04.2025	
3	Створення бізнес-моделі процесів	01.05.2025	
4	Огляд рішень-аналогів	02.05.2025	
5	Створення технічного завдання	03.05.2025	
6	Проектування БД	05.05.2025	
7	Створення інформаційної системи	07.05.2025	
8	Створення інструкції користувача	10.05.2025	
9	Оформлення пояснювальної записки	11.05.2025	
10	Створення презентації	17.05.2025	

Здобувач _____ Андрій АРЗАМАСОВ

Керівник роботи _____ Наталія ЛІМАНСЬКА

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці вебзастосунку для вихователя дошкільного навчального закладу №21. Кваліфікаційна робота складається з 80 сторінок, 13 рисунків, додатків та літературних джерел – 34.

У роботі досліджено інформаційні потоки та бізнес-процеси діяльності вихователя ДНЗ, виявлено проблеми, пов'язані з переважанням паперового документообігу та неефективним використанням робочого часу. Розроблено архітектуру веб-застосунку на основі тришарової моделі з використанням фреймворку Flask, бази даних MySQL та інтерфейсу користувача на основі Bootstrap. Реалізовано функціональні модулі для обліку дітей, ведення журналу відвідування, складання розкладу занять, обліку інвентарю, формування навчальних планів та звітів. Проведено комплексне тестування та впровадження веб-застосунку в діяльність ДНЗ №21. Техніко-економічний розрахунок підтвердив високу ефективність впровадження системи: термін окупності менше 3 місяців, рентабельність інвестицій понад 400%. Впровадження веб-застосунку дозволило скоротити час на ведення документації на 70%, підвищити якість обробки інформації, оптимізувати робочий час вихователів та покращити якість навчально-виховного процесу.

Ключові слова : MySQL, FLASK PYTHON, РОЗРОБКА ПЗ, ПЗ, ФОРМА, ЗАПИТ, ЗВІТ, ПРЕДСТАВЛЕННЯ, ІНТЕРФЕЙС КОРИСТУВАЧА, ІНФОРМАЦІЙНА СИСТЕМА

ABSTRACT

The thesis is devoted to the development of a web application for educators of preschool educational institution №21. The term paper consists of 80 pages, 13 figures, 34 appendices and references.

The paper examines information flows and business processes of preschool educators, identifies problems related to the predominance of paper document flow and inefficient use of working time. The architecture of the web application was developed based on a three-tier model using the Flask framework, MySQL database, and Bootstrap-based user interface. Functional modules were implemented for children's records, attendance tracking, scheduling, inventory management, lesson planning, and reporting. Comprehensive testing and implementation of the web application into the activities of preschool educational institution №21 were conducted. Technical and economic calculations confirmed the high efficiency of the system implementation: payback period of less than 3 months, return on investment over 400%. The implementation of the web application reduced documentation time by 70%, improved information processing quality, optimized educators' working time, and enhanced the quality of the educational process.

Keywords: MySQL, FLASK, PYTHON, SOFTWARE DEVELOPMENT, SOFTWARE, FORM, QUERY, REPORT, VIEW, USER INTERFACE, INFORMATION SYSTEM

ЗМІСТ

ВСТУП 6

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ОБ'ЄКТУ ДОСЛІДЖЕННЯ ... 9

1.1 Загальний опис ДНЗ №21 як об'єкту дослідження 9

1.2 Структурна та організаційна схема ДНЗ №21 14

1.3 Аналіз інформаційних потоків та бізнес-процесів у роботі вихователя ДНЗ №21 20

1.4 Стан комп'ютеризації та виявлені проблеми в інформаційних процесах ДНЗ №21 25

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ ВИХОВАТЕЛЯ ДНЗ №21 31

2.1 Проектування архітектури веб-застосунку 31

2.2 Розробка бази даних MySQL для зберігання інформації 36

2.3 Проектування інтерфейсу користувача 42

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ ВИХОВАТЕЛЯ ДНЗ №21 51

3.1 Програмна реалізація серверної частини веб-застосунку 51

3.2 Програмна реалізація клієнтської частини веб-застосунку 56

3.3 Тестування та впровадження веб-застосунку в діяльність ДНЗ №21 60

3.4 Техніко-економічний розрахунок ефективності впровадження розробленого веб-застосунку 66

ВИСНОВКИ 71

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 75

ДОДАТКИ 81

ВСТУП

Актуальність теми: зумовлена необхідністю оптимізації інформаційних процесів у дошкільних навчальних закладах та зменшення часових витрат вихователів на ведення документації. В умовах сучасного інформаційного суспільства та цифрової трансформації освіти, традиційна паперова система обліку та документообігу в дошкільних навчальних закладах стає неефективною, призводить до нераціонального використання робочого часу педагогічних працівників, дублювання інформації, складності її пошуку та аналізу. Вихователі ДНЗ №21 витрачають значну частину робочого часу на рутинну роботу з документацією, що зменшує можливості для безпосередньої взаємодії з дітьми, розробки цікавих та розвиваючих занять, індивідуального підходу до кожної дитини.

Розробка та впровадження веб-застосунку для вихователя ДНЗ дозволяє автоматизувати основні бізнес-процеси, оптимізувати інформаційні потоки, забезпечити електронний документообіг, створити єдиний інформаційний простір для всіх учасників освітнього процесу. Така система значно скорочує час на ведення документації, підвищує якість та оперативність обробки інформації, забезпечує прозорість діяльності закладу, покращує взаємодію між працівниками та комунікацію з батьками. Використання сучасних веб-технологій дозволяє створити зручний, інтуїтивно зрозумілий інтерфейс, доступний з різних пристроїв, що забезпечує гнучкість та мобільність роботи вихователів з інформаційною системою.

Мета і завдання дослідження: метою дослідження є підвищення ефективності роботи вихователя ДНЗ №21 шляхом розробки та впровадження веб-застосунку для автоматизації основних бізнес-процесів, оптимізації інформаційних потоків та зменшення часових витрат на ведення документації.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Провести аналіз інформаційних потоків та бізнес-процесів у діяльності вихователя ДНЗ №21.

2. Визначити стан комп'ютеризації та виявити проблеми в інформаційних процесах закладу.
3. Розробити архітектуру веб-застосунку для вихователя ДНЗ.
4. Спроектувати базу даних для зберігання та обробки інформації.
5. Розробити інтерфейс користувача з урахуванням принципів юзабіліті та адаптивного дизайну.
6. Здійснити програмну реалізацію серверної та клієнтської частин веб-застосунку.
7. Провести тестування та впровадження веб-застосунку в діяльність ДНЗ №21.
8. Оцінити техніко-економічну ефективність впровадження розробленого веб-застосунку.

Об'єкт дослідження: процес автоматизації діяльності вихователя дошкільного навчального закладу №21 міста Києва.

Предмет дослідження: методи, засоби та технології розробки веб-застосунку для автоматизації бізнес-процесів та оптимізації інформаційних потоків у діяльності вихователя ДНЗ №21.

Практичне значення роботи: полягає у розробці та впровадженні веб-застосунку, який вирішує реальні проблеми в інформаційних процесах ДНЗ №21 та підвищує ефективність роботи вихователів. Створений веб-застосунок дозволяє автоматизувати ведення журналу відвідування, облік дітей, складання розкладу занять, облік інвентарю, формування навчальних планів та звітів, що значно скорочує час на ведення документації, підвищує якість обробки інформації та забезпечує її доступність для всіх учасників освітнього процесу. Розроблені архітектурні та програмні рішення можуть бути використані для створення подібних систем в інших дошкільних навчальних закладах, що сприятиме цифровій трансформації дошкільної освіти в Україні.

Методи дослідження: у процесі виконання кваліфікаційної роботи використано комплекс наукових методів дослідження:

1. Теоретичні методи: аналіз науково-методичної літератури та нормативних документів з питань організації діяльності дошкільних навчальних закладів, систематизація та узагальнення теоретичних даних для формування концептуальних основ дослідження.
2. Емпіричні методи: спостереження за діяльністю вихователів ДНЗ №21, інтерв'ювання працівників закладу для виявлення проблем та потреб, аналіз документації та інформаційних потоків.
3. Моделювання: розробка концептуальної моделі веб-застосунку, проектування архітектури системи та моделі бази даних.
4. Програмно-технологічні методи: використання сучасних технологій веб-розробки (Python, Flask, SQLAlchemy, MySQL, HTML5, CSS3, JavaScript, Bootstrap) для реалізації веб-застосунку.
5. Експериментальні методи: тестування розробленого веб-застосунку, впровадження його в діяльність ДНЗ №21, збір та аналіз даних про ефективність використання системи.
6. Економіко-математичні методи: розрахунок економічної ефективності впровадження веб-застосунку, визначення терміну окупності інвестицій, рентабельності, чистої приведеної вартості проекту.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ОБ'ЄКТУ ДОСЛІДЖЕННЯ

1.1 Загальний опис ДНЗ №21 як об'єкту дослідження

Дошкільний навчальний заклад №21 міста Києва представляє собою освітню установу, що функціонує як цілісна педагогічна система, спрямована на забезпечення всебічного розвитку дітей дошкільного віку. Відповідно до Закону України "Про дошкільну освіту", дошкільний навчальний заклад визначається як навчальний заклад, який забезпечує реалізацію права дитини на здобуття дошкільної освіти, її фізичний, розумовий і духовний розвиток, соціальну адаптацію та готовність продовжувати освіту. ДНЗ №21 є живим втіленням цього визначення, оскільки створює оптимальні умови для гармонійного розвитку особистості дитини в період її становлення. Заклад функціонує як соціально-педагогічна система, інтегрована в освітній простір міста Києва, що надає освітні послуги відповідно до державних стандартів дошкільної освіти та реалізує базовий компонент дошкільної освіти України [1].

Структурно-функціональна організація ДНЗ №21 відображає комплексний характер діяльності закладу та включає взаємопов'язані підсистеми: адміністративну, освітньо-виховну, методичну, медичну та господарську. Адміністративна підсистема, очолювана директором закладу, забезпечує стратегічне керівництво, прийняття управлінських рішень, координацію діяльності всіх підрозділів, а також вирішення кадрових питань. Освітньо-виховна підсистема представлена вихователями, які безпосередньо взаємодіють з дітьми, реалізуючи освітні програми та організовуючи навчально-виховний процес. У цьому контексті особливого значення набуває методична підсистема, яка здійснює навчально-методичне забезпечення педагогічного процесу, розробляє навчальні програми, проводить заходи з підвищення кваліфікації педагогічних працівників, що сприяє постійному вдосконаленню освітнього процесу та впровадженню інноваційних методик навчання.

Медична підсистема ДНЗ №21 відіграє суттєву роль у забезпеченні фізичного благополуччя вихованців. Медичний працівник закладу здійснює моніторинг стану здоров'я дітей, проводить профілактичні заходи, організовує медичні огляди, контролює дотримання санітарно-гігієнічних норм у приміщеннях закладу. Данна підсистема функціонує у тісній взаємодії з освітньо-виховною, оскільки питання здоров'язбереження інтегруються у навчально-виховний процес. Господарська підсистема забезпечує матеріально-технічну базу закладу, проводить ремонтні роботи, контролює харчування та підтримання належних санітарних умов. Взаємодія між цими підсистемами створює цілісний механізм функціонування дошкільного навчального закладу, де кожна ланка виконує специфічні функції, сприяючи досягненню спільної мети – створенню сприятливих умов для розвитку дітей [2].

Аналіз інформаційних потоків у ДНЗ №21 дозволяє виявити складну мережу комунікацій, що пронизує всі рівні організаційної структури закладу. Зовнішні інформаційні потоки включають взаємодію з контролюючими органами (Міністерство освіти і науки України, місцеві управління освіти), батьками вихованців, постачальниками послуг та товарів, медичними установами. Внутрішні інформаційні потоки забезпечують циркуляцію інформації між структурними підрозділами закладу, передачу розпоряджень від керівництва до виконавців, зворотний зв'язок від виконавців до керівництва, а також горизонтальну комунікацію між працівниками одного рівня. Інформаційні потоки за змістом можна класифікувати на навчально-методичні (навчальні програми, методичні розробки, плани занять), адміністративно-управлінські (накази, розпорядження, звіти), фінансово-господарські (кошториси, інвентаризаційні відомості, договори), медичні (карти розвитку дітей, медичні довідки, журнали здоров'я) та комунікаційні (спілкування з батьками, консультації, батьківські збори).

Бізнес-процеси роботи вихователя ДНЗ №21 представляють собою послідовність взаємопов'язаних дій, спрямованих на реалізацію освітніх завдань та забезпечення повноцінного розвитку дітей. Ключовими бізнес-процесами є:

ранковий прийом дитини, що включає привітання, перевірку документації, оцінку самопочуття та взаємодію з батьками; оцінка стану здоров'я дитини з можливим залученням медичного працівника у разі виявлення проблем; проведення навчальних занять відповідно до програми з використанням дидактичних матеріалів; організація розвиваючих ігор та активностей, що сприяють формуванню різноманітних навичок та вмінь; оформлення документації, включаючи ведення щоденних записів про діяльність дитини та підготовку звітів; організація прогулянок на свіжому повітрі із забезпеченням безпеки дітей. Кожен з цих процесів має свою специфіку та потребує від вихователя відповідних компетенцій, знань та навичок [3].

Функціональна модель роботи ДНЗ №21 відображає системну взаємодію між різними підрозділами та посадовими особами закладу. Центральним елементом цієї моделі є директор, який здійснює загальне керівництво закладом, визначає стратегію розвитку, контролює фінансово-господарську діяльність, приймає кадрові рішення. Вихователі, перебуваючи у безпосередній взаємодії з дітьми, реалізують освітню програму, планують і проводять заняття, організовують дозвілля, здійснюють моніторинг розвитку дітей. Методист забезпечує навчально-методичну підтримку педагогічного процесу, розробляє методичні рекомендації, організовує семінари та тренінги для підвищення кваліфікації вихователів. Практичний психолог проводить діагностику психологічного стану дітей, надає консультації батькам та вихователям, здійснює корекційну роботу з дітьми, які мають особливі освітні потреби. Медичний працівник контролює стан здоров'я дітей, проводить профілактичні заходи, веде медичну документацію. Господарський персонал забезпечує функціонування матеріально-технічної бази закладу, створює належні санітарно-гігієнічні умови для перебування дітей.

Особливу роль у функціональній моделі ДНЗ №21 відіграє взаємодія вихователя з іншими працівниками закладу. Зокрема, вихователь взаємодіє з медичним працівником, повідомляючи про випадки нездужання дітей та отримуючи інформацію про оцінку їхнього стану здоров'я. Співпраця з

психологом та логопедом передбачає обмін інформацією про особливості розвитку дітей, реалізацію рекомендацій спеціалістів у навчально-виховному процесі, спільну оцінку динаміки розвитку дитини. Взаємодія з директором закладу включає звітування про результати роботи, обговорення проблемних ситуацій та пошук шляхів їх вирішення. Така багаторівнева взаємодія забезпечує комплексний підхід до організації освітнього процесу та створює умови для всебічного розвитку дітей відповідно до їхніх індивідуальних особливостей та потреб [1].

Інформаційне забезпечення діяльності ДНЗ №21 є невід'ємною складовою його функціонування та включає документацію різного призначення: нормативно-правову (закони, постанови, накази, інструкції), організаційно-розпорядчу (положення, статут закладу, посадові інструкції), навчально-методичну (програми, методичні рекомендації, конспекти занять), обліково-звітну (журнали відвідувань, щоденники спостережень, табелі), медичну (медичні карти дітей, журнали здоров'я), фінансово-господарську (кошториси, інвентаризаційні відомості, договори). Ефективне управління інформаційними потоками та документообігом є необхідною умовою для забезпечення якісного функціонування закладу та реалізації його освітніх завдань. Проте, наявна система інформаційного забезпечення ДНЗ №21 характеризується певними обмеженнями, пов'язаними з переважанням паперового документообігу, відсутністю інтегрованої інформаційної системи для управління бізнес-процесами, обмеженим застосуванням інформаційно-комунікаційних технологій у взаємодії з батьками та іншими зацікавленими сторонами.

Умови праці вихователя ДНЗ №21 характеризуються значним інформаційним навантаженням, необхідністю одночасного виконання різноманітних функцій, високим рівнем відповідальності за життя та здоров'я дітей. Вихователь повинен володіти глибокими знаннями у галузі дошкільної педагогіки та психології, застосовувати сучасні освітні методики, враховувати індивідуальні особливості кожної дитини, підтримувати постійний контакт з батьками. Окрім безпосередньої роботи з дітьми, значну частину робочого часу

вихователя займає ведення документації, підготовка до занять, участь у методичних заходах. Така багатофункціональність діяльності вихователя потребує ефективних інструментів для організації його роботи, які дозволять оптимізувати рутинні процеси, зменшити час на оформлення документації, забезпечити швидкий доступ до необхідної інформації та зосередити більше уваги на безпосередній роботі з дітьми [3].

Аналіз стану інформатизації та комп'ютеризації ДНЗ №21 виявляє певні прогалини в застосуванні сучасних інформаційно-комунікаційних технологій. Заклад оснащений комп'ютерною технікою здебільшого для адміністративних потреб, проте відсутня комплексна інформаційна система, яка б забезпечувала автоматизацію бізнес-процесів, електронний документообіг, оперативний обмін інформацією між працівниками, віддалений доступ до інформаційних ресурсів. Вихователі переважно використовують традиційні паперові форми для ведення документації, що призводить до значних затрат часу на оформлення звітів, планів роботи, характеристик дітей. Комунікація з батьками здійснюється переважно через особисті зустрічі, інформаційні стенди, батьківські збори, що обмежує можливості для оперативного інформування та зворотного зв'язку. Відсутність інтегрованої інформаційної системи ускладнює процеси моніторингу та аналізу різних аспектів діяльності закладу, знижує ефективність управлінських рішень, створює додаткове навантаження на персонал [1].

Зважаючи на виявлені проблеми, впровадження веб-застосунку для вихователя ДНЗ №21 є обґрунтованим та доцільним кроком для модернізації інформаційного забезпечення діяльності закладу. Веб-застосунок як спеціалізоване програмне забезпечення, що працює у веб-браузері, дозволить автоматизувати рутинні процеси, забезпечити електронний документообіг, створити єдиний інформаційний простір для комунікації між працівниками закладу та батьками. Функціональність веб-застосунку може включати електронні журнали відвідування, календарне планування занять, електронні портфоліо дітей, засоби комунікації з батьками, бази даних навчально-методичних матеріалів, аналітичні інструменти для оцінки розвитку дітей. Такий

підхід дозволить оптимізувати робочий час вихователя, підвищити якість освітнього процесу, забезпечити прозорість діяльності закладу для батьків та контролюючих органів, створити умови для впровадження інноваційних освітніх методик та технологій [4].

1.2 Структурна та організаційна схема ДНЗ №21

Структурна та організаційна схема ДНЗ №21 міста Києва відображає системну організацію закладу, що забезпечує реалізацію його основних функцій та завдань. Дошкільний навчальний заклад як соціально-педагогічна система характеризується наявністю чіткої ієрархічної структури, фіксованим розподілом повноважень та відповідальності, регламентованими відносинами між структурними підрозділами. Організаційна структура ДНЗ №21 побудована за функціональним принципом, що передбачає виділення підрозділів за основними напрямами діяльності закладу: адміністративним, методичним, медичним та господарським. Така структура забезпечує оптимальний розподіл завдань, компетенцій та відповідальності між працівниками, створює умови для ефективної взаємодії та координації діяльності різних підрозділів, сприяє досягненню стратегічних цілей закладу. Нижче, на рисунку 1.1, представлено організаційно-функціональну схему ДНЗ №21, яка візуалізує взаємозв'язки між структурними підрозділами та посадовими особами закладу.

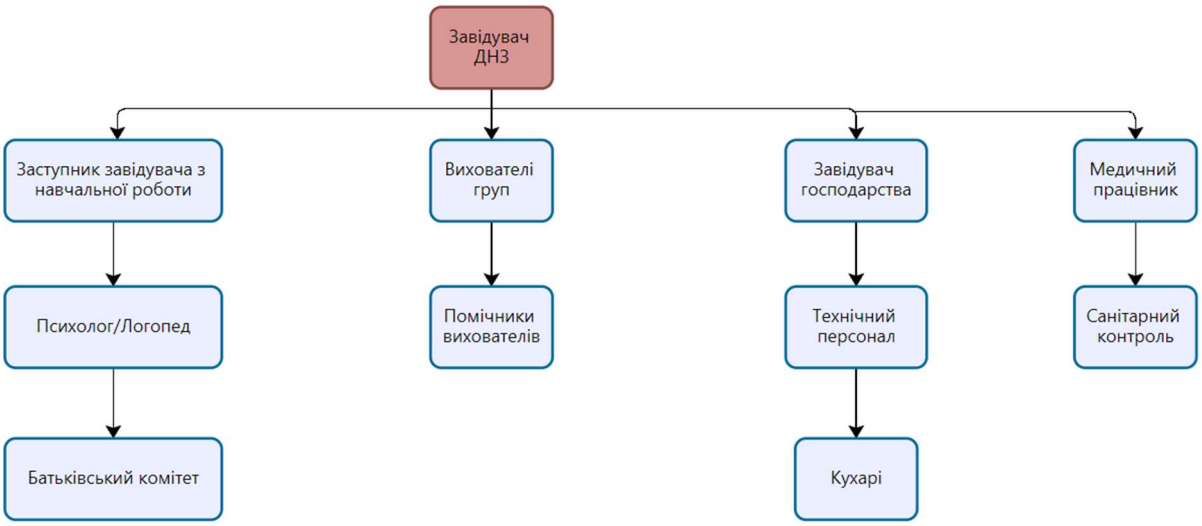

Рисунок 1.1 – Організаційно-функціональна схема ДНЗ

Адміністративний підрозділ, очолюваний директором (завідувачем) ДНЗ №21, є керівною ланкою в організаційній структурі закладу. Директор здійснює загальне керівництво діяльністю закладу, визначає стратегічні напрями його розвитку, приймає управлінські рішення, контролює фінансово-господарську діяльність, вирішує кадрові питання, представляє заклад у відносинах з органами державної влади, місцевого самоврядування, громадськими організаціями. Відповідно до Положення про дошкільний навчальний заклад, директор несе персональну відповідальність за виконання покладених на заклад завдань, результати фінансово-господарської діяльності, стан і збереження будівель та іншого майна. До адміністративного підрозділу також входять заступники директора, які здійснюють керівництво окремими напрямами діяльності закладу (навчально-виховною, адміністративно-господарською роботою). Адміністративний підрозділ виконує координаційну функцію, забезпечуючи узгоджену роботу всіх структурних елементів закладу, здійснює контроль за виконанням планів роботи, дотриманням нормативних вимог до організації освітнього процесу, створює умови для професійного розвитку педагогічних працівників [1].

Методичний підрозділ, який представлений методистом ДНЗ №21, відіграє суттєву роль у забезпеченні якості освітнього процесу. Методист відповідає за навчально-методичне забезпечення роботи закладу, здійснює планування та організацію методичної роботи, розробляє рекомендації щодо вдосконалення освітнього процесу, організовує заходи з підвищення кваліфікації педагогічних працівників, надає консультативну допомогу вихователям з питань планування та організації освітньої діяльності, впровадження інноваційних методик та технологій. Методичний підрозділ забезпечує формування єдиного освітнього простору закладу, сприяє впровадженню передового педагогічного досвіду, координує роботу з батьками щодо педагогічної просвіти. Ефективність функціонування методичного підрозділу визначає якість освітнього процесу, рівень професійної компетентності педагогічних працівників, результативність освітньої діяльності закладу загалом. Методист взаємодіє з усіма структурними підрозділами закладу, забезпечуючи інтеграцію освітньої, виховної, оздоровчої та розвивальної функцій дошкільної освіти [5].

Медичний підрозділ ДНЗ №21 представлений медичним працівником, який здійснює контроль за станом здоров'я дітей, організовує профілактичні заходи, проводить медичні огляди, контролює дотримання санітарно-гігієнічних норм у приміщеннях закладу, режиму харчування та якості продуктів. Відповідно до нормативних вимог, медичний працівник веде медичну документацію (медичні карти дітей, журнали здоров'я, журнали контролю), проводить санітарно-просвітницьку роботу серед дітей, батьків та працівників закладу, надає першу медичну допомогу у разі необхідності. Медичний підрозділ взаємодіє з адміністративним та освітньо-виховним підрозділами, забезпечуючи інтеграцію питань здоров'язбереження у навчально-виховний процес, консультує вихователів щодо організації оздоровчих заходів, індивідуального підходу до дітей з особливими потребами. Ефективна робота медичного підрозділу є невід'ємною складовою забезпечення повноцінного фізичного розвитку дітей, формування у них навичок здорового способу життя, попередження захворювань.

Господарський підрозділ ДНЗ №21 забезпечує матеріально-технічну базу закладу, проводить ремонтні роботи, контролює харчування та підтримання належних санітарних умов. До складу цього підрозділу входять кухарі та обслуговуючий персонал, які забезпечують харчування дітей відповідно до встановлених норм, підтримують чистоту та порядок у приміщеннях закладу, дбають про благоустрій території. Господарський підрозділ взаємодіє з адміністративним та медичним підрозділами, забезпечуючи створення безпечних та комфортних умов для перебування дітей у закладі, дотримання санітарно-гігієнічних норм, організацію раціонального харчування. Функціонування господарського підрозділу дозволяє реалізувати матеріально-технічну функцію ДНЗ, створити належні умови для організації освітнього процесу, забезпечити збереження та розвиток матеріальної бази закладу.

Основною ланкою освітньо-виховної діяльності ДНЗ №21 є вихователі, які безпосередньо взаємодіють з дітьми, реалізують освітні програми, організовують навчально-виховний процес. На рисунку 1.2 представлено схему бізнес-процесів роботи вихователя, яка відображає основні функціональні обов'язки та послідовність дій у процесі взаємодії з дітьми та іншими учасниками освітнього процесу [6].

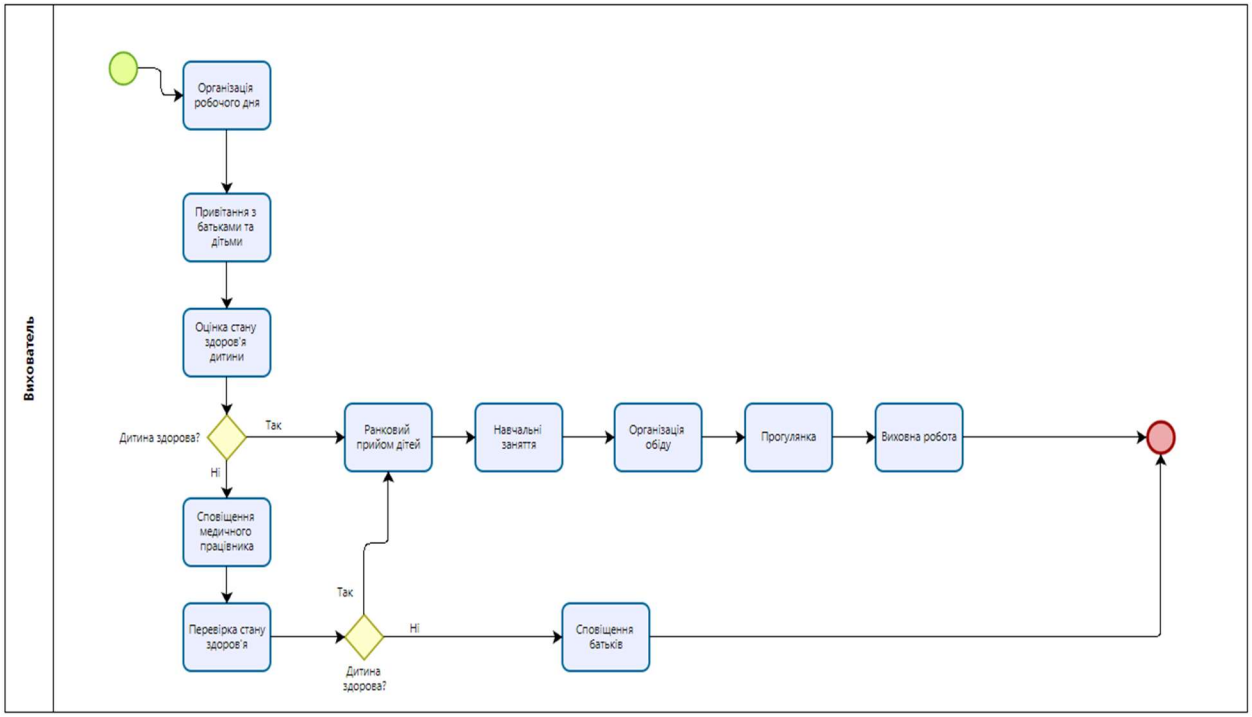

Рисунок 1.2 – Бізнес-процеси роботи вихователя

Аналізуючи схему бізнес-процесів, можна виділити основні функціональні обов'язки вихователя ДНЗ №21: здійснення ранкового прийому дітей, що включає привітання, перевірку документації, оцінку самопочуття; проведення навчальних занять відповідно до програми дошкільної освіти; організація ігрової та розвивальної діяльності; контроль за станом здоров'я дітей впродовж дня; ведення педагогічної документації; організація прогулянок на свіжому повітрі. Вихователь взаємодіє з різними структурними підрозділами закладу: адміністративним – щодо питань організації освітнього процесу, виконання планів роботи; методичним – щодо планування освітньої діяльності, застосування інноваційних методик; медичним – щодо стану здоров'я дітей, організації оздоровчих заходів; господарським – щодо створення належних умов для перебування дітей у групі. Ефективність роботи вихователя визначає якість освітніх послуг, що надаються дошкільним навчальним закладом, рівень розвитку та підготовки дітей до школи [1].

Для комплексного розуміння функціонування ДНЗ №21 необхідно розглянути функціональну схему роботи закладу, яка відображає взаємодію всіх

структурних підрозділів та основні процеси, що забезпечують реалізацію завдань дошкільної освіти. Схему зображено на рисунку 1.3.

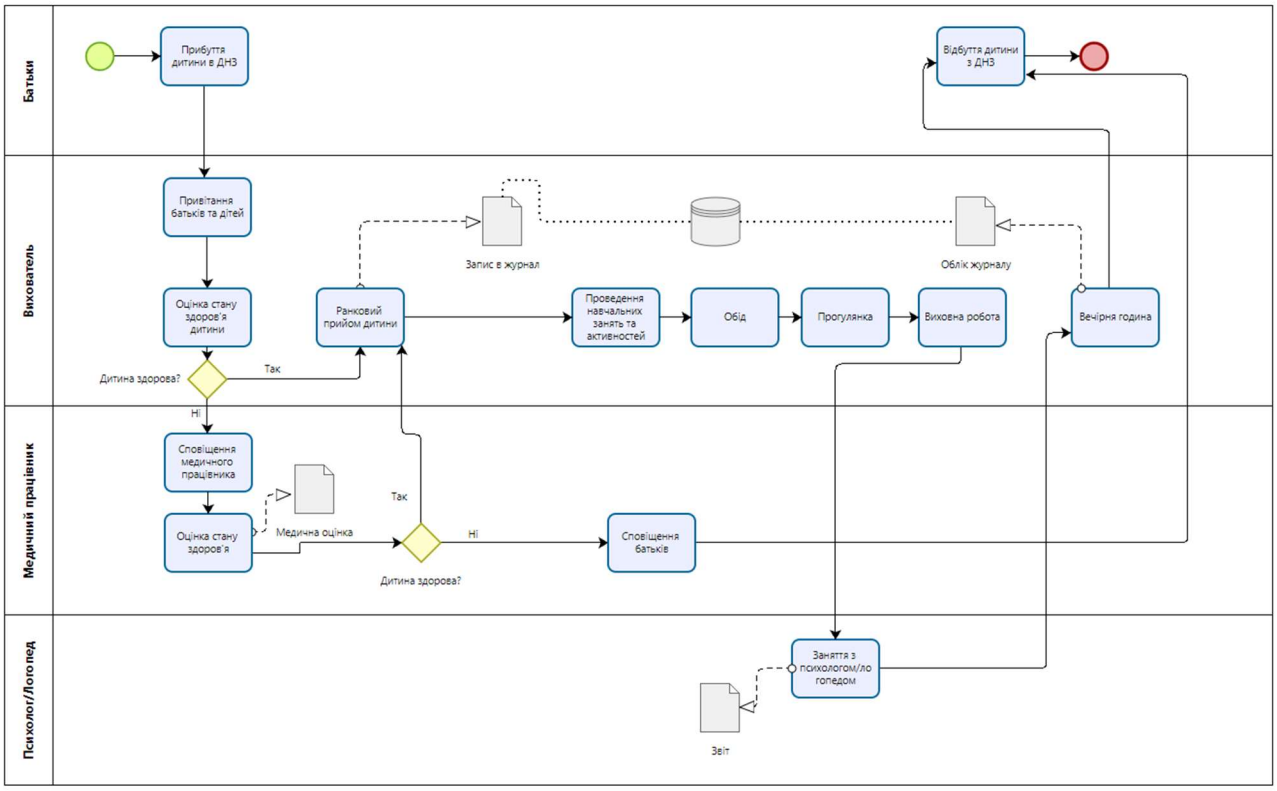

Рисунок 1.3 – Функціональна схема роботи ДНЗ

Представлена функціональна схема роботи ДНЗ №21 демонструє системний характер діяльності закладу, взаємозв'язок між різними структурними підрозділами та процесами. Центральну роль у цій схемі відіграє адміністративний підрозділ, який здійснює координацію та контроль за діяльністю всіх інших підрозділів. Освітньо-виховний процес, реалізований вихователями, спрямований на забезпечення різнобічного розвитку дітей відповідно до їхніх вікових та індивідуальних особливостей. Методичний супровід освітнього процесу забезпечує його якість та відповідність сучасним вимогам. Медичний і господарський підрозділи створюють необхідні умови для ефективної організації освітнього процесу, збереження та зміцнення здоров'я дітей. Взаємодія з батьками є невід'ємною складовою функціонування закладу, що забезпечує єдність вимог сім'ї та дошкільного закладу у вихованні дітей. Така

структурно-функціональна організація ДНЗ №21 забезпечує комплексний підхід до реалізації завдань дошкільної освіти, створює умови для гармонійного розвитку особистості дитини в період її становлення.

1.3 Аналіз інформаційних потоків та бізнес-процесів у роботі вихователя ДНЗ №21

Інформаційні потоки у роботі вихователя ДНЗ №21 являють собою складну систему циркуляції даних, що забезпечує реалізацію освітньо-виховного процесу та взаємодію з іншими учасниками освітнього середовища. Інформаційний потік, у контексті діяльності дошкільного навчального закладу, можна визначити як цілеспрямований рух даних від джерел інформації до її споживачів, що здійснюється в інформаційному просторі закладу. Аналізуючи діяльність вихователя ДНЗ №21, можна виділити декілька основних напрямів інформаційних потоків: вертикальні потоки (інформація, що циркулює між вихователем та адміністрацією закладу), горизонтальні потоки (обмін інформацією між вихователями та іншими фахівцями закладу), зовнішні потоки (взаємодія з батьками вихованців, контролюючими органами, іншими освітніми установами). За змістом інформаційні потоки у роботі вихователя включають: навчально-методичну інформацію (навчальні програми, плани, методичні рекомендації, конспекти занять), адміністративно-організаційну інформацію (накази, розпорядження, графіки роботи), інформацію про вихованців (особові справи, результати спостережень, характеристики), медичну інформацію (дані про стан здоров'я дітей, рекомендації щодо організації режиму), інформацію для батьків (повідомлення, консультації, рекомендації). Ефективність управління цими інформаційними потоками безпосередньо впливає на якість навчально-виховного процесу, своєчасність прийняття рішень, результативність взаємодії з батьками та колегами [2]. Завдання і функції вихователя представлені у таблиці 1.1.

Взаємодію вихователя з іншими працівниками можна побачити у таблиці 1.2.

Таблиця 1.1 “Завдання і функції вихователя ДНЗ”

№	Задачі	Функції
1	Ранковий прийом дитини	-Привітання дитини та перевірка документації. -Перевірка самопочуття дитини, взаємодія з батьками.
2	Оцінка стану здоров'я дитини	-Перший огляд стану здоров'я (може включати температуру, загальний стан). -У разі необхідності, сповіщення медичного працівника.
3	Проведення навчальних занять	-Планування та проведення заняття відповідно до навчальної програми з використанням навчальних матеріалів.
4	Організація активностей	-Розробка і організація розвиваючих ігор і активностей. -Співпраця з дітьми у групових іграх.
5	Оформлення документації	-Ведення щоденних записів про діяльність дитини. -Підготовка звітів і документації для внутрішнього користування.
6	Прогулянка	-Організація і контроль прогулянок на свіжому повітрі. -Забезпечення безпеки дітей під час прогулянок.

Таблиця 1.2 “Взаємодія вихователя з іншими працівниками”

№	Підрозділ	Одержання	Надання
1	Медичний працівник	Сповіщення про нездужання	- Оцінка стану здоров’я
2	Психолог/Логопед	Група дітей	- Реалізація рекомендацій - Оцінка розвитку дитини
3	Завідувач	Звіт	- Обговорення проблем - Пошук рішень

Аналіз взаємодії вихователя ДНЗ №21 з іншими працівниками закладу дозволяє виявити специфіку організації інформаційних потоків та визначити їх вплив на ефективність освітнього процесу. Відповідно до таблиці 1.2 "Взаємодія вихователя з іншими працівниками", можна простежити основні напрями обміну інформацією. У контексті взаємодії з медичним працівником вихователь надає інформацію про спостережувані ознаки нездужання дитини та отримує у

відповідь оцінку стану здоров'я. Така комунікація має оперативний характер і вимагає швидкого реагування, оскільки стосується життя та здоров'я дітей. У взаємодії з психологом та логопедом вихователь отримує групу дітей для спеціальних занять, реалізує надані фахівцями рекомендації щодо індивідуального підходу до дитини, надає інформацію про результати оцінки розвитку дитини. Цей інформаційний потік має переважно аналітично-консультативний характер і спрямований на забезпечення індивідуального підходу до дітей з особливими освітніми потребами. Взаємодія з завідувачем ДНЗ передбачає надання звітів про результати роботи, обговорення проблемних ситуацій та спільний пошук їх вирішення. Характерною особливістю цього інформаційного потоку є його формалізований характер, необхідність дотримання встановлених форм документації, періодичність звітування. Аналіз цих інформаційних потоків дозволяє виявити певні проблеми: переважання паперової форми документообігу, відсутність єдиної інформаційної системи, що ускладнює оперативний обмін даними, дублювання інформації у різних формах звітності, недостатнє використання сучасних інформаційно-комунікаційних технологій для організації взаємодії [1].

Бізнес-процеси у роботі вихователя ДНЗ №21, відповідно до таблиці 1.1 "Завдання і функції вихователя ДНЗ" та рисунка 1.2 "Бізнес-процеси роботи вихователя", можна систематизувати за основними напрямами діяльності. Ранковий прийом дитини як бізнес-процес включає такі функції: привітання дитини, перевірка наявності необхідної документації (медичної довідки після хвороби, довідки про відсутність тощо), первинна оцінка самопочуття та емоційного стану дитини, спілкування з батьками щодо особливостей стану дитини, сну, апетиту, поведінки вдома. Цей процес характеризується високою інтенсивністю інформаційного обміну, необхідністю одночасної взаємодії з різними суб'єктами (дітьми, батьками, медичним працівником), фіксацією отриманої інформації. Оцінка стану здоров'я дитини як бізнес-процес включає: проведення першого огляду стану здоров'я (спостереження за поведінкою, апетитом, активністю дитини), у разі виявлення відхилень – вимірювання

температури, огляд шкіри, слизових оболонок, сповіщення медичного працівника при підозрі на захворювання, інформування батьків про стан дитини, ведення відповідної документації. Цей процес вимагає від вихователя спеціальних знань з питань дитячої гігієни та первинної діагностики захворювань, вміння розпізнавати тривожні симптоми, що потребують медичного втручання. Проведення навчальних занять як бізнес-процес передбачає: планування занять відповідно до навчальної програми, підготовку дидактичних матеріалів, організацію освітнього середовища, безпосереднє проведення заняття з використанням відповідних методик, оцінку результатів засвоєння матеріалу, фіксацію результатів у відповідній документації. Цей процес є основним у діяльності вихователя і визначає якість освітніх послуг, що надаються закладом [5].

Продовжуючи аналіз бізнес-процесів у роботі вихователя ДНЗ №21, необхідно розглянути такі процеси, як організація активностей, оформлення документації та проведення прогулянок. Організація активностей як бізнес-процес включає: розробку і впровадження розвиваючих ігор та активностей відповідно до віку та інтересів дітей, створення ігрового середовища, що стимулює пізнавальну активність, участь у групових іграх для підтримки інтересу дітей, спостереження за взаємодією дітей у процесі гри для оцінки їхніх соціальних навичок. Цей процес вимагає від вихователя креативності, знання вікової психології, вміння організувати взаємодію між дітьми, здатності адаптувати активності відповідно до індивідуальних особливостей кожної дитини. Оформлення документації як бізнес-процес передбачає: ведення щоденних записів про діяльність кожної дитини (відвідуваність, активність на заняттях, особливості поведінки, досягнення, проблеми), підготовку планів роботи (щоденних, тижневих, місячних), складання характеристик на дітей, підготовку звітів для адміністрації закладу, оформлення інформаційних матеріалів для батьків. Цей процес є досить трудомістким і забирає значну частину робочого часу вихователя, що зменшує його можливості для безпосередньої взаємодії з дітьми. Прогулянка як бізнес-процес включає:

підготовку дітей до виходу на свіже повітря (одягання відповідно до погодних умов), організацію активності на прогулянці (рухливі ігри, спостереження за природою, трудова діяльність на ділянці), забезпечення безпеки дітей під час прогулянки, повернення до приміщення та переодягання дітей. Цей процес спрямований на забезпечення фізичного розвитку дітей, зміцнення їхнього здоров'я, ознайомлення з природним середовищем.

Аналіз інформаційних потоків та бізнес-процесів у роботі вихователя ДНЗ №21 дозволяє виявити певні проблеми та обмеження, що впливають на ефективність його діяльності. По-перше, спостерігається значне навантаження, пов'язане з оформленням документації, що відволікає вихователя від безпосередньої роботи з дітьми. Вихователь повинен вести різноманітні форми документації: плани роботи, журнали відвідування, щоденники спостережень, табелі, характеристики дітей, що вимагає значних затрат часу та ресурсів. По-друге, відсутність єдиної інформаційної системи призводить до дублювання інформації, ускладнює процес її пошуку та аналізу, знижує оперативність прийняття рішень. Наприклад, інформація про стан здоров'я дитини може одночасно фіксуватися у медичній карті, журналі відвідувань, щоденнику спостережень, що створює додаткове навантаження на персонал та підвищує ймовірність помилок. По-третє, обмежене використання сучасних інформаційно-комунікаційних технологій у роботі вихователя знижує ефективність комунікації з батьками, ускладнює процес обміну досвідом з колегами, обмежує доступ до сучасних освітніх ресурсів. По-четверте, існуючі методи збору та аналізу інформації про розвиток дітей не дозволяють повною мірою відстежувати динаміку їхнього прогресу, виявляти проблемні аспекти, розробляти індивідуальні стратегії розвитку. Ці проблеми зумовлюють необхідність оптимізації інформаційних потоків та автоматизації бізнес-процесів у роботі вихователя ДНЗ №21 [1].

Впровадження веб-застосунку для вихователя ДНЗ №21 дозволить суттєво оптимізувати інформаційні потоки та автоматизувати ключові бізнес-процеси. Веб-застосунок може забезпечити електронний документообіг, що дозволить

зменшити час на оформлення документації, уникнути дублювання інформації, забезпечити швидкий доступ до необхідних даних. Наприклад, електронний журнал відвідування автоматично формуватиме статистичні звіти, табелі, аналізуватиме динаміку відвідуваності, що зменшить рутинну роботу вихователя. Модуль планування дозволить створювати, зберігати та редагувати плани роботи, автоматично генерувати конспекти занять на основі шаблонів, відстежувати виконання програми, що оптимізує процес підготовки до навчальних занять. Електронне портфоліо дитини забезпечить систематизацію інформації про кожного вихованця, фіксацію результатів спостережень, аналіз динаміки розвитку, що дозволить реалізувати індивідуальний підхід до кожної дитини. Модуль комунікації з батьками надасть можливість оперативно інформувати їх про важливі події, досягнення дитини, рекомендації, що підвищить ефективність взаємодії з сім'єю. База методичних матеріалів забезпечить доступ до сучасних освітніх ресурсів, можливість обміну досвідом з колегами, що сприятиме професійному розвитку вихователя. Таким чином, впровадження веб-застосунку для вихователя ДНЗ №21 дозволить створити інтегроване інформаційне середовище, яке забезпечить оптимізацію інформаційних потоків, автоматизацію бізнес-процесів, підвищення якості освітніх послуг та ефективності управління дошкільним навчальним закладом [3].

1.4 Стан комп'ютеризації та виявлені проблеми в інформаційних процесах ДНЗ №21

Стан комп'ютеризації ДНЗ №21 міста Києва відображає загальні тенденції впровадження інформаційно-комунікаційних технологій у сферу дошкільної освіти та характеризується певними особливостями, зумовленими специфікою функціонування даного закладу. Комп'ютеризація у контексті дошкільного навчального закладу визначається як процес впровадження та використання комп'ютерної техніки та програмного забезпечення для оптимізації

адміністративних, освітніх та комунікаційних процесів. Аналіз наявного стану комп'ютеризації ДНЗ №21 свідчить про нерівномірний розподіл технічних засобів між структурними підрозділами закладу. Адміністративний підрозділ оснащено комп'ютерною технікою, яка використовується переважно для ведення документації, формування звітності, комунікації з контролюючими органами. Директор та методист закладу мають доступ до персональних комп'ютерів, підключених до мережі Інтернет, що дозволяє їм отримувати актуальну інформацію, відправляти звіти електронною поштою, здійснювати пошук методичних матеріалів. Однак, функціональні можливості наявної техніки та програмного забезпечення використовуються не повною мірою, оскільки відсутня інтегрована інформаційна система, яка б забезпечувала автоматизацію адміністративних процесів, електронний документообіг, аналітичну обробку даних. Для ведення бухгалтерського обліку та фінансової звітності використовується спеціалізоване програмне забезпечення, проте воно не інтегроване з іншими інформаційними системами закладу, що ускладнює оперативний аналіз фінансово-господарської діяльності, контроль за використанням ресурсів, планування бюджету [2].

Стан комп'ютеризації освітньо-виховного процесу у ДНЗ №21 характеризується недостатнім рівнем забезпечення технічними засобами. Групові приміщення не обладнані комп'ютерною технікою, що обмежує можливості вихователів щодо використання інформаційно-комунікаційних технологій у навчально-виховному процесі. Вихователі не мають безпосереднього доступу до комп'ютера на робочому місці, що змушує їх виконувати значний обсяг роботи з документацією у паперовому форматі, витрачаючи на це значну частину робочого часу. Підготовка планів роботи, конспектів занять, характеристик дітей, оформлення інформаційних матеріалів для батьків здійснюється переважно вручну або з використанням особистої комп'ютерної техніки в позаробочий час. Відсутність автоматизованої системи обліку відвідуваності, моніторингу розвитку дітей, електронних журналів призводить до дублювання інформації у різних формах документації, підвищує

ймовірність помилок, ускладнює аналіз динаміки розвитку кожної дитини. Методичний кабінет оснащено комп'ютером з підключенням до мережі Інтернет, проте використання його функціональних можливостей обмежене через відсутність локальної мережі, що б забезпечувала доступ вихователів до електронних методичних ресурсів, баз даних, навчальних матеріалів. Відсутність мультимедійного обладнання (проекторів, інтерактивних дошок, планшетів) обмежує можливості для впровадження сучасних інтерактивних методик навчання, використання аудіовізуальних матеріалів, розвиваючих комп'ютерних програм [7].

Аналіз інформаційних процесів у ДНЗ №21 дозволяє виявити низку проблем, пов'язаних з недостатнім рівнем комп'ютеризації та автоматизації. Інформаційний процес, у контексті діяльності дошкільного навчального закладу, визначається як послідовність дій з отримання, обробки, зберігання, передачі та використання інформації, необхідної для забезпечення функціонування закладу. Основні інформаційні процеси у ДНЗ №21 включають: збір та реєстрацію даних (про вихованців, персонал, матеріальні ресурси), обробку та аналіз інформації (формування статистичної звітності, аналіз відвідуваності, оцінка розвитку дітей), зберігання інформації (архівування документації, створення баз даних), передачу інформації (комунікація між працівниками, взаємодія з батьками, звітування перед контролюючими органами), використання інформації (прийняття управлінських рішень, планування діяльності, організація освітнього процесу). Проблеми в інформаційних процесах ДНЗ №21 пов'язані з переважанням паперового документообігу, що призводить до значних затрат часу на оформлення документації, дублювання інформації, складності пошуку необхідних даних. Відсутність єдиної інформаційної системи ускладнює інтеграцію даних з різних джерел, знижує оперативність прийняття рішень, обмежує можливості для аналітичної обробки інформації. Недостатній рівень автоматизації адміністративних процесів (планування, контроль, звітність) призводить до неефективного використання людських ресурсів, підвищує вірогідність помилок через людський фактор.

Проблеми в інформаційних процесах найбільш гостро проявляються у роботі вихователя ДНЗ №21, оскільки саме вихователь є центральною ланкою освітньо-виховного процесу та залучений до найбільшої кількості бізнес-процесів. Аналіз бізнес-процесів, представлених у таблиці 1.1 "Завдання і функції вихователя ДНЗ", дозволяє виявити конкретні проблеми, пов'язані з недостатнім рівнем комп'ютеризації [1]. Вони відображені у таблиці 1.3

Таблиця 1.3 "Проблеми бізнес процесів у роботі вихователя ДНЗ №21"

№	**Бізнес-процес**	**Проблема інформаційного процесу**	**Наслідки**
1	Ранковий прийом дитини	Ведення журналу відвідування у паперовому форматі	Затрати часу на заповнення, складність аналізу динаміки відвідуваності, відсутність оперативного доступу до історії відвідувань
2	Оцінка стану здоров'я дитини	Відсутність електронної медичної картки, дублювання інформації у різних документах	Затримки у передачі інформації медичному працівнику, складність відстеження динаміки стану здоров'я, ризик втрати даних
3	Проведення навчальних занять	Рукописне оформлення планів та конспектів занять, відсутність доступу до електронних навчальних матеріалів	Значні затрати часу на підготовку, складність модифікації планів, обмежені можливості для використання мультимедійних ресурсів
4	Організація активностей	Відсутність електронної бази розвиваючих ігор та активностей	Обмеження у виборі активностей, складність адаптації до індивідуальних потреб дітей, затрати часу на розробку матеріалів
5	Оформлення документації	Паперовий формат документів, відсутність шаблонів та автоматизованих форм	Значні затрати часу, дублювання інформації, складність пошуку та аналізу даних, ризик помилок при ручному введенні
6	Прогулянка	Відсутність електронного планування прогулянок, обліку активностей	Складність координації дій з іншими вихователями, відсутність систематизації спостережень за дітьми під час прогулянок

Проблеми в інформаційних процесах поширюються також на взаємодію вихователя з іншими працівниками ДНЗ №21 та батьками вихованців. Відсутність ефективних електронних каналів комунікації призводить до затримок у передачі інформації, зниження оперативності реагування на проблемні ситуації, обмеження можливостей для регулярного зворотного зв'язку. Взаємодія з медичним працівником щодо стану здоров'я дітей здійснюється переважно в усній формі або через паперові записи, що підвищує ризик втрати інформації, ускладнює відстеження динаміки здоров'я кожної дитини. Співпраця з психологом та логопедом щодо особливостей розвитку дітей обмежена через відсутність інтегрованої електронної системи, яка б забезпечувала обмін даними про результати діагностики, рекомендації фахівців, динаміку корекційної роботи. Комунікація з батьками здійснюється переважно через особисті зустрічі, інформаційні стенди, паперові оголошення, що обмежує можливості для регулярного інформування про досягнення дитини, надання індивідуальних рекомендацій, залучення батьків до освітнього процесу. Відсутність електронних форм зворотного зв'язку ускладнює отримання інформації від батьків щодо особливостей поведінки дитини вдома, її інтересів, проблем, що знижує ефективність індивідуального підходу.

Виявлені проблеми в інформаційних процесах ДНЗ №21 обґрунтовують необхідність розробки та впровадження інтегрованої інформаційної системи, яка б забезпечувала автоматизацію основних бізнес-процесів, електронний документообіг, ефективну комунікацію між усіма учасниками освітнього процесу. Веб-застосунок для вихователя ДНЗ №21 може стати ефективним інструментом для вирішення виявлених проблем, оскільки забезпечить доступ до інформаційних ресурсів з будь-якого пристрою, підключеного до мережі Інтернет, не потребуватиме значних витрат на оновлення апаратного забезпечення, дозволить інтегрувати різні функціональні модулі в єдину систему. Основними функціональними вимогами до такого веб-застосунку є: електронне ведення журналів відвідування з можливістю формування

статистичних звітів; електронне портфоліо дитини, що містить інформацію про її розвиток, досягнення, особливості; електронне планування освітньої діяльності з бібліотекою конспектів занять та можливістю їх модифікації; електронна база методичних матеріалів з можливістю пошуку за різними критеріями; система комунікації з батьками, що забезпечує оперативне інформування, зворотний зв'язок, надання рекомендацій; аналітичний модуль для моніторингу та оцінки різних аспектів діяльності закладу. Впровадження такого веб-застосунку дозволить оптимізувати інформаційні процеси ДНЗ №21, підвищити ефективність роботи вихователя, забезпечити якість освітніх послуг, створити умови для інноваційного розвитку закладу.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ ВИХОВАТЕЛЯ ДНЗ №21

2.1 Проектування архітектури веб-застосунку

Проектування архітектури веб-застосунку для вихователя ДНЗ №21 базується на необхідності створення гнучкої, масштабованої та надійної системи, яка дозволить автоматизувати основні бізнес-процеси, оптимізувати інформаційні потоки та забезпечити ефективну взаємодію між усіма учасниками освітнього процесу. Архітектура веб-застосунку визначається як структурна організація програмної системи, що включає компоненти, їх зовнішні властивості та взаємозв'язки між ними. У контексті розроблюваного веб-застосунку оптимальною є тришарова архітектура, яка складається з презентаційного рівня (відповідає за взаємодію з користувачем), рівня бізнес-логіки (реалізує функціональні можливості системи) та рівня доступу до даних (забезпечує роботу з базою даних). Така архітектура забезпечує чіткий розподіл відповідальності між компонентами, полегшує розробку, тестування та підтримку системи, підвищує її гнучкість та масштабованість. Для реалізації цієї архітектури обрано фреймворк Flask, який є легким та гнучким мікрофреймворком для розробки веб-застосунків мовою Python. Flask дозволяє швидко створювати веб-додатки з мінімальними витратами ресурсів, забезпечує простоту розробки та підтримки коду, має широкі можливості для інтеграції з різними базами даних та розширеннями. Цей фреймворк добре підходить для проектів середнього масштабу, яким є розроблюваний веб-застосунок для вихователя ДНЗ [7].

Презентаційний рівень архітектури веб-застосунку реалізовано за допомогою шаблонізатора Jinja2, який є стандартним для фреймворку Flask. Jinja2 дозволяє створювати динамічні HTML-сторінки з використанням шаблонів, що значно спрощує розробку та підтримку інтерфейсу користувача. Для стилізації сторінок використовується фреймворк Bootstrap, який забезпечує адаптивний дизайн, сучасний вигляд інтерфейсу та кросбраузерну сумісність.

Такий підхід дозволяє створити інтуїтивно зрозумілий інтерфейс, який буде комфортним для використання як на стаціонарних комп'ютерах, так і на мобільних пристроях. Структура презентаційного рівня включає базовий шаблон (base.html), який містить спільні елементи для всіх сторінок (шапку, навігаційне меню, футер), та спеціалізовані шаблони для кожного функціонального модуля (діти, відвідування, розклад, інвентар, плани, звітність). Такий підхід забезпечує уніфікований вигляд усіх сторінок застосунку, спрощує навігацію та підвищує зручність використання. Аналіз файлу 'base.html' показує, що в ньому реалізовано загальну структуру сторінки, підключення стилів Bootstrap, створення навігаційного меню з посиланнями на всі функціональні розділи, а також систему відображення повідомлень для користувача (flash-повідомлення). Це забезпечує єдиний стиль оформлення для всього додатку та полегшує користувачам взаємодію з системою [5, с. 199].

Рівень бізнес-логіки веб-застосунку реалізовано через маршрутизатор (routes.py), який обробляє HTTP-запити, взаємодіє з моделями даних та повертає відповідні шаблони для відображення. Цей рівень відповідає за обробку введених користувачем даних, їх валідацію, виконання необхідних операцій з даними, формування відповіді для клієнта. Маршрутизатор Flask використовує декоратори для визначення URL-шляхів та HTTP-методів, які відповідають кожній функції представлення. Такий підхід забезпечує чіткий та зрозумілий код, простоту розширення функціональності, можливість розділення логіки на окремі модулі. Для аутентифікації та авторизації користувачів використовується розширення Flask-Login, яке надає зручний інтерфейс для роботи з сесіями користувачів, захисту маршрутів, перенаправлення неавторизованих користувачів [8]. Розглянемо приклад реалізації функції для додавання нової дитини до системи:

```
@main.route('/add_child', methods=['GET', 'POST'])
@login_required
def add_child():
    from .models import Child
```

```
    if request.method == 'POST':
        full_name = request.form['full_name']
        birth_date_str = request.form['birth_date']
        group_name = request.form['group_name']
        parent_name = request.form['parent_name']
        parent_phone = request.form['parent_phone']

        # Перетворення строки у об'єкт дати
        birth_date = datetime.strptime(birth_date_str, '%Y-%m-%d').date()
        new_child = Child(
            full_name=full_name,
            birth_date=birth_date,
            group_name=group_name
            parent_name=parent_name,
            parent_phone=parent_phone
        )
        db.session.add(new_child)
        db.session.commit()
        flash('Дитину додано успішно!', 'success')
        return redirect(url_for('main.children'))
    return render_template('add_child.html')
```

Цей фрагмент коду демонструє реалізацію маршруту '/add_child', який обробляє як GET-запити (відображення форми для додавання дитини), так і POST-запити (обробка даних форми та додавання нової дитини до бази даних). Декоратор '@login_required' забезпечує захист цього маршруту, дозволяючи доступ лише авторизованим користувачам. При отриманні POST-запиту функція отримує дані з форми, перетворює рядкове представлення дати народження у об'єкт дати, створює новий екземпляр моделі Child, додає його до сесії бази даних та фіксує зміни. Після успішного додавання дитини користувач отримує повідомлення про успіх та перенаправляється на сторінку зі списком дітей.

Такий підхід до організації бізнес-логіки забезпечує чіткий розподіл відповідальності, простоту розширення функціональності, можливість повторного використання коду [9].

Рівень доступу до даних веб-застосунку реалізовано за допомогою ORM (Object-Relational Mapping) SQLAlchemy, яка дозволяє працювати з базою даних на рівні Python-об'єктів, абстрагуючись від особливостей конкретної СУБД. SQLAlchemy надає зручний інтерфейс для визначення моделей даних, виконання CRUD-операцій (Create, Read, Update, Delete), побудови складних запитів. У файлі models.py визначено моделі даних, які відображають сутності предметної області: User (користувач системи), Child (дитина), Attendance (відвідуваність), Schedule (розклад занять), Inventory (інвентар), LessonPlan (навчальний план). Кожна модель містить атрибути, які відповідають полям відповідної таблиці в базі даних, а також методи для роботи з цими даними. Для зберігання даних використовується реляційна база даних SQLite, яка не потребує окремого сервера, проста в налаштуванні та використанні, підходить для невеликих та середніх проектів. Важливою особливістю архітектури є використання міграцій бази даних, які дозволяють відстежувати зміни в структурі бази даних, застосовувати ці зміни автоматично, повертатися до попередніх версій структури бази даних у разі необхідності. Це забезпечує гнучкість при розробці та модифікації системи, полегшує процес розгортання та оновлення застосунку.

Для забезпечення безпеки веб-застосунку впроваджено низку механізмів захисту на різних рівнях архітектури. На рівні аутентифікації використовується механізм хешування паролів за допомогою функції generate_password_hash з пакету werkzeug.security, що дозволяє зберігати в базі даних не самі паролі, а їх хеш-значення, які неможливо перетворити назад у паролі. При спробі входу в систему введений користувачем пароль хешується за тим же алгоритмом і порівнюється з хеш-значенням, збереженим у базі даних. Такий підхід забезпечує захист паролів користувачів навіть у випадку компрометації бази даних. На рівні авторизації використовується механізм декораторів @login_required, який захищає маршрути від несанкціонованого доступу,

дозволяючи їх використання лише авторизованим користувачам. Для захисту від CSRF-атак (Cross-Site Request Forgery) використовується механізм генерації та перевірки CSRF-токенів, вбудований у Flask-WTF. Для захисту від SQL-ін'єкцій використовується параметризований доступ до бази даних через ORM SQLAlchemy, який автоматично екранує параметри запитів. Безпека передачі даних забезпечується можливістю налаштування HTTPS-з'єднання для шифрування трафіку між клієнтом та сервером. Усі ці механізми в комплексі забезпечують надійний захист даних та функціональності системи від різних типів атак [10].

Масштабованість та розширюваність архітектури веб-застосунку забезпечується модульною структурою проекту, чітким розподілом відповідальності між компонентами, використанням принципів об'єктно-орієнтованого програмування та сучасних підходів до розробки веб-застосунків. Модульна структура дозволяє легко додавати нові функціональні модулі без значних змін в існуючому коді. Наприклад, для додавання нового модуля "Здоров'я дітей" достатньо створити відповідну модель даних у файлі models.py, додати маршрути для роботи з цим модулем у файлі routes.py, створити шаблони для відображення даних. Використання Blueprint у Flask дозволяє розділити функціональність на логічні блоки, що полегшує розробку та підтримку коду. Використання ORM SQLAlchemy забезпечує абстракцію від конкретної СУБД, що дозволяє за необхідності змінити базу даних без значних змін у коді застосунку. Розширюваність системи також забезпечується можливістю інтеграції з іншими сервісами та системами через API (Application Programming Interface). Наприклад, можна реалізувати API для обміну даними з системою електронного документообігу ДНЗ, системою обліку харчування, системою відеоспостереження тощо. Така архітектура дозволяє поступово розширювати функціональність системи відповідно до зростаючих потреб користувачів [11].

Для розгортання веб-застосунку спроектовано гнучку систему конфігурації, яка дозволяє адаптувати застосунок до різних середовищ (розробка, тестування, промислова експлуатація). Конфігурація здійснюється

через файл **init**.py, де визначаються основні параметри застосунку: секретний ключ для підписання сесій, URI для підключення до бази даних, налаштування логування, параметри для розширень тощо. У промисловому середовищі доцільно використовувати змінні середовища для зберігання конфіденційних параметрів конфігурації, що підвищує безпеку системи. Для розгортання застосунку можна використовувати різні методи: безпосереднє розгортання на виділеному сервері з використанням WSGI-серверів (Gunicorn, uWSGI), розгортання в контейнерах Docker для забезпечення ізоляції та портативності, розгортання в хмарних середовищах (AWS, Google Cloud, Microsoft Azure) для забезпечення високої доступності та масштабованості. Для моніторингу та діагностики застосунку планується використовувати системи логування та трасування, які дозволять відстежувати продуктивність системи, виявляти та усувати помилки, оптимізувати роботу застосунку. Такий підхід до проектування архітектури розгортання забезпечує гнучкість у виборі інфраструктури, можливість масштабування системи відповідно до зростання навантаження, надійність роботи застосунку в різних умовах експлуатації [12].

2.2 Розробка бази даних MySQL для зберігання інформації

Розробка бази даних для веб-застосунку вихователя ДНЗ №21 є фундаментальним етапом проектування інформаційної системи, оскільки саме база даних забезпечує структуроване зберігання та ефективний доступ до інформації, необхідної для функціонування застосунку. База даних у контексті інформаційних систем визначається як організована колекція даних, структурованих таким чином, щоб забезпечити зручний доступ, управління та модифікацію інформації. Для реалізації проєкту обрано систему управління базами даних MySQL, яка є однією з найпопулярніших реляційних СУБД з відкритим кодом. MySQL надає надійну платформу для зберігання даних, характеризується високою продуктивністю, масштабованістю, підтримкою транзакцій та забезпеченням цілісності даних. У початковій реалізації проєкту

використовувалась СУБД SQLite, проте для промислової експлуатації системи вибір зроблено на користь MySQL через її кращу продуктивність при одночасній роботі багатьох користувачів, підтримку складних запитів та транзакцій, розвинені механізми забезпечення цілісності даних, можливості резервного копіювання та відновлення. Перехід від SQLite до MySQL не потребує значних змін у коді застосунку завдяки використанню ORM SQLAlchemy, яка абстрагує роботу з базою даних на рівні об'єктів Python, що дозволяє змінювати СУБД з мінімальними змінами в коді, модифікуючи лише рядок підключення в конфігурації застосунку.

Проектування схеми бази даних для веб-застосунку вихователя ДНЗ №21 базується на аналізі інформаційних потоків та бізнес-процесів, визначених у першому розділі кваліфікаційної роботи. Спроектована база даних включає шість основних таблиць, кожна з яких відповідає певній сутності предметної області: User (користувачі системи), Child (діти), Attendance (відвідуваність), Schedule (розклад занять), Inventory (інвентар), LessonPlan (навчальні плани). Таблиця User зберігає інформацію про користувачів системи (вихователів, адміністраторів), включаючи їх ідентифікатори, імена, електронні адреси та хеш-значення паролів. Таблиця Child містить дані про дітей, які відвідують ДНЗ: повне ім'я, дату народження, назву групи. Таблиця Attendance призначена для обліку відвідуваності дітей і містить зв'язок з таблицею Child через зовнішній ключ, а також дату, час приходу та час виходу. Таблиця Schedule зберігає інформацію про розклад занять, включаючи день тижня, час та назву заняття. Таблиця Inventory призначена для обліку інвентарю в ДНЗ і містить назву предмета, його стан та кількість. Таблиця LessonPlan зберігає навчальні плани, включаючи назву, дату та опис. Такий набір таблиць повністю охоплює інформаційні потреби вихователя ДНЗ, визначені в результаті аналізу бізнес-процесів [13].

Реалізація моделей даних у веб-застосунку здійснюється через ORM SQLAlchemy, яка дозволяє визначати структуру таблиць та зв'язки між ними у вигляді Python-класів. Кожен клас моделі відповідає окремій таблиці в базі

даних, а атрибути класу – колонкам таблиці. Розглянемо приклад реалізації моделі даних для обліку відвідуваності дітей, взятий з файлу models.py розроблюваного веб-застосунку:

```
class Attendance(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    child_id = db.Column(db.Integer, db.ForeignKey('child.id'), nullable=False)
    date = db.Column(db.Date, nullable=False)
    time_in = db.Column(db.Time, nullable=True)
    time_out = db.Column(db.Time, nullable=True)
    child = db.relationship('Child', backref=db.backref('attendances', lazy=True))
```

У цьому фрагменті коду визначено клас Attendance, який відповідає таблиці для обліку відвідуваності. Атрибут id є первинним ключем таблиці, child_id – зовнішнім ключем, що посилається на таблицю Child, date зберігає дату відвідування, time_in та time_out – час приходу та виходу дитини відповідно. Атрибут child створює відношення між моделями Attendance та Child, дозволяючи звертатися до даних про дитину безпосередньо з об'єкту відвідуваності. Параметр backref створює зворотнє відношення, дозволяючи звертатися до всіх записів про відвідуваність конкретної дитини через атрибут attendances об'єкту Child. Такий підхід до проектування моделей даних забезпечує зручний доступ до пов'язаних даних, спрощує розробку та підтримку коду, дозволяє абстрагуватися від особливостей конкретної СУБД [14, с. 284].

Для забезпечення цілісності та несуперечності даних у базі даних впроваджено низку обмежень та правил. Перш за все, використано механізм первинних ключів (primary key) для однозначної ідентифікації записів у кожній таблиці. У всіх таблицях первинним ключем є поле id типу Integer з атрибутом автоінкременту, що забезпечує унікальність ідентифікаторів. Для забезпечення посилальної цілісності даних використано механізм зовнішніх ключів (foreign key), який гарантує, що значення у полі, яке посилається на іншу таблицю, існує в цій таблиці. Наприклад, поле child_id у таблиці Attendance є зовнішнім ключем, що посилається на поле id таблиці Child, що гарантує, що кожен запис про

відвідуваність пов'язаний з існуючою дитиною. Для забезпечення доменної цілісності даних використано обмеження типів даних для кожної колонки (Integer для цілих чисел, String для рядків, Date для дат, Time для часу, Text для довгих текстів). Також впроваджено обмеження NOT NULL для полів, які не можуть бути пустими (наприклад, ім'я дитини, дата відвідування), та обмеження UNIQUE для полів, які повинні мати унікальні значення (наприклад, електронна адреса користувача). Такі обмеження допомагають запобігти введенню некоректних даних, забезпечують несуперечність інформації в базі даних, підвищують надійність роботи системи [15, с. 22].

Для забезпечення ефективного доступу до даних та оптимізації продуктивності бази даних впроваджено систему індексів та оптимізовано структуру таблиць. Індекси є спеціальними структурами даних, які дозволяють швидко знаходити записи в таблиці без необхідності повного перегляду таблиці. У базі даних веб-застосунку створено індекси для полів, які часто використовуються у пошукових запитах: індекс для поля email у таблиці User для прискорення пошуку користувачів за електронною адресою, індекс для поля child_id у таблиці Attendance для прискорення пошуку записів про відвідуваність конкретної дитини, індекс для поля date у таблиці Attendance для прискорення пошуку записів за датою. Крім того, оптимізовано типи даних для кожної колонки з метою мінімізації обсягу пам'яті, необхідної для зберігання даних: для зберігання дат використано тип Date замість більш загального типу DateTime, для зберігання короткого тексту використано тип String з обмеженням довжини, для зберігання великих текстів – тип Text. Для таблиць, у яких очікується велика кількість записів (наприклад, таблиця Attendance), обрано двигун InnoDB, який забезпечує підтримку транзакцій, зовнішніх ключів та має кращу продуктивність при великих обсягах даних [16]. Структуру таблиць бази даних вебзастосунку для вихователя ДНЗ можна побачити у таблиці 2.1.

Таблиця 2.1 "Структура таблиць бази даних вебзастосунку для вихователя ДНЗ №21"

Назва таблиці	Призначення	Основні поля	Зв'язки
User	Зберігання інформації про користувачів системи	id (PK), name, email, password_hash	-
Child	Зберігання інформації про дітей	id (PK), full_name, birth_date, group_name, parent_name, parent_phone	One-to-Many з Attendance
Attendance	Облік відвідуваності дітей	id (PK), child_id (FK), date, time_in, time_out	Many-to-One з Child
Schedule	Зберігання розкладу занять	id (PK), day_of_week, time, activity	-
Inventory	Облік інвентарю	id (PK), name, condition, quantity	-
LessonPlan	Зберігання навчальних планів	id (PK), title, date, description, file_path	-

На рисунку 2.1 представлено діаграму класів, яка відображає структуру бази даних веб-застосунку та взаємозв'язки між таблицями. Діаграма наочно демонструє, що таблиця Attendance (Відвідування) має зв'язок типу "багато-до-одного" з таблицею Child (Дитина), оскільки одна дитина може мати багато записів про відвідування. Інші таблиці (User, Schedule, Inventory, LessonPlan) функціонують як незалежні сутності, що взаємодіють з іншими компонентами системи через бізнес-логіку застосунку [17].

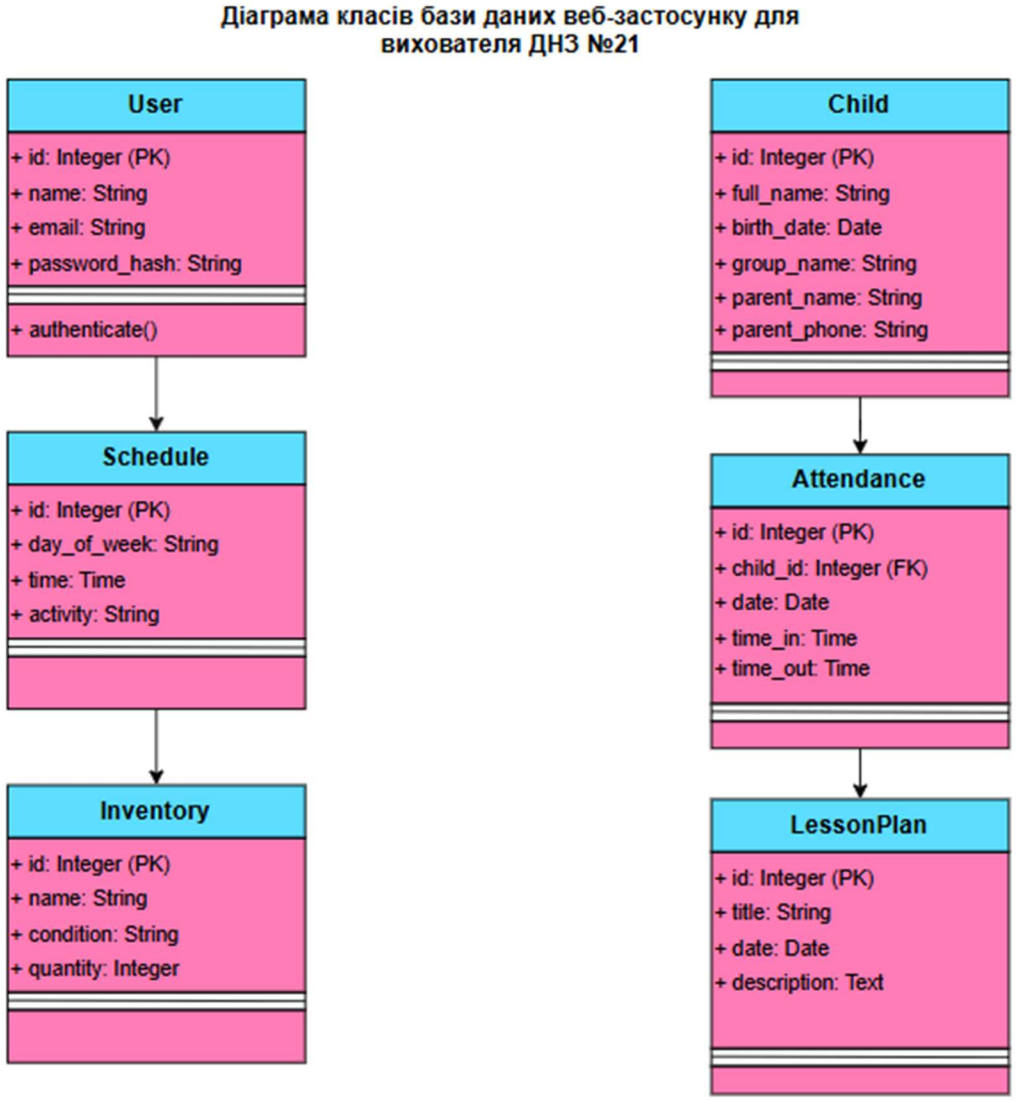

Рисунок 2.1 — Діаграма класів бази даних веб-застосунку для вихователя ДНЗ №21

Для забезпечення безпеки даних у базі даних впроваджено комплекс заходів, спрямованих на захист від несанкціонованого доступу, модифікації та втрати інформації. Перш за все, реалізовано механізм аутентифікації та авторизації користувачів, який обмежує доступ до даних неавторизованих осіб. Паролі користувачів зберігаються в базі даних не у відкритому вигляді, а у вигляді хеш-значень, обчислених за допомогою криптографічно стійких алгоритмів, що унеможливлює відновлення паролів навіть у випадку компрометації бази даних. Для захисту від SQL-ін'єкцій, які можуть призвести до несанкціонованого доступу до даних або їх модифікації, використовується

параметризований доступ до бази даних через ORM SQLAlchemy, який автоматично екранує параметри запитів. Для захисту від втрати даних реалізовано механізм регулярного резервного копіювання бази даних, який дозволяє відновити інформацію у випадку апаратних збоїв, програмних помилок або зловмисних дій. Крім того, всі критичні операції з даними (наприклад, додавання, редагування, видалення записів) протоколюються в журналі аудиту, що дозволяє відстежити всі зміни в базі даних та виявити потенційні проблеми. Такий комплексний підхід до забезпечення безпеки даних дозволяє мінімізувати ризики, пов'язані з використанням бази даних у веб-застосунку вихователя ДНЗ №21.

2.3 Проектування інтерфейсу користувача

Проектування інтерфейсу користувача для веб-застосунку вихователя ДНЗ №21 є невід'ємною складовою процесу розробки, яка визначає успішність взаємодії користувачів із системою та ефективність виконання ними професійних завдань. Користувацький інтерфейс (UI) визначається як сукупність засобів та методів, за допомогою яких користувач взаємодіє з різними пристроями та комплексами, у нашому випадку – з веб-застосунком. Якісно спроектований інтерфейс має відповідати принципам юзабіліті – сукупності показників зручності, ефективності, зрозумілості та суб'єктивної привабливості для користувача. Розробка інтерфейсу для веб-застосунку вихователя ДНЗ базується на результатах аналізу цільової аудиторії, основних бізнес-процесів та інформаційних потоків, виявлених у першому розділі кваліфікаційної роботи. Особливу увагу приділено врахуванню специфіки роботи вихователя, який має обмежений час на взаємодію з комп'ютером, потребує швидкого доступу до необхідної інформації, часто виконує типові операції з документацією. Виходячи з цих міркувань, інтерфейс проектувався з метою забезпечення максимальної інтуїтивної зрозумілості, швидкодії та зручності за рахунок логічного розташування елементів, уніфікованого стильового оформлення, продуманої

навігації. Для технічної реалізації інтерфейсу обрано фреймворк Bootstrap, який пропонує широкий набір готових компонентів, адаптивну сітку для різних розмірів екранів, типографіку та стилі форм, що значно спрощує розробку та забезпечує сумісність з більшістю сучасних браузерів [18].

Основою інтерфейсу веб-застосунку є базовий шаблон, який визначає загальну структуру та стильове оформлення всіх сторінок системи. Базовий шаблон реалізовано у файлі base.html, який містить DOCTYPE-декларацію, підключення CSS-стилів Bootstrap, створення навігаційного меню, області для відображення повідомлень та контейнера для основного контенту. Розглянемо цей файл детальніше:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>{% block title %}  Застосунок для вихователя ДНЗ№21
{% endblock %}</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
    <div class="container">
      <a class="navbar-brand" href="{{ url_for('main.dashboard') }}">
Застосунок для вихователя ДНЗ№21
</a>
      <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
```

```
            <div class="collapse navbar-collapse" id="navbarNav">
              <ul class="navbar-nav ms-auto">
                {% if current_user.is_authenticated %}
                <li class="nav-item">
                  <a class="nav-link" href="{{ url_for('main.children') }}">Діти</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{ url_for('main.attendance') }}">Відвідування</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{ url_for('main.schedule') }}">Розклад</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{ url_for('main.inventory') }}">Інвентар</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{ url_for('main.lesson_plans') }}">Плани</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{ url_for('main.reports') }}">Звітність</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{ url_for('main.logout') }}">Вихід</a>
```

```
                </li>
                {% else %}
                <li class="nav-item">
                    <a class="nav-link" href="{{ url_for('main.login') }}">Вхід</a>
                </li>
                {% endif %}
              </ul>
            </div>
          </div>
        </nav>

        <div class="container">
            {% with messages = get_flashed_messages() %}
              {% if messages %}
                {% for message in messages %}
                  <div class="alert alert-info">{{ message }}</div>
                {% endfor %}
              {% endif %}
            {% endwith %}

            {% block content %}{% endblock %}
        </div>
    </body>
    </html>
```

Цей код демонструє кілька суттєвих аспектів проектування інтерфейсу. По-перше, використання шаблонізатора Jinja2 дозволяє створювати блоки ({% block title %}, {% block content %}), які будуть замінені конкретним вмістом на кожній сторінці. По-друге, умовний оператор {% if current_user.is_authenticated %} забезпечує відображення різних елементів навігації для авторизованих та неавторизованих користувачів. По-третє, система flash-повідомлень ({% with

messages = get_flashed_messages() %}) дозволяє відображати користувачу інформацію про результати виконаних дій. По-четверте, навігаційне меню містить посилання на всі основні функціональні розділи застосунку, що забезпечує швидкий доступ до необхідних функцій. По-п'яте, використання адаптивної сітки Bootstrap (class="container") забезпечує правильне відображення контенту на різних пристроях. Такий підхід до проектування базового шаблону забезпечує єдиний стиль оформлення всіх сторінок застосунку, спрощує навігацію та підвищує зручність використання системи [19].

Проектування інтерфейсу для функціональних розділів веб-застосунку базується на аналізі бізнес-процесів, виявлених у першому розділі кваліфікаційної роботи. Кожен бізнес-процес трансформовано у відповідний функціональний розділ з власним інтерфейсом, який забезпечує виконання необхідних операцій. Розділ "Діти" (діти.html) надає можливість перегляду списку дітей, додавання нових дітей, редагування та видалення записів. Інтерфейс цього розділу спроектовано у вигляді таблиці з колонками для відображення основної інформації про дітей та кнопкою для додавання нових записів. Розділ "Відвідування" (attendance.html) дозволяє вести облік відвідуваності дітей, фіксувати час приходу та виходу, формувати звіти за різними періодами. Інтерфейс цього розділу включає таблицю з записами про відвідування та форму для відмітки відвідуваності. Розділ "Розклад" (schedule.html) надає можливість перегляду та редагування розкладу занять, додавання нових занять. Інтерфейс розділу "Інвентар" (inventory.html) забезпечує облік матеріальних цінностей, їх стану та кількості. Розділ "Плани" (lesson_plans.html) дозволяє створювати, редагувати та переглядати навчальні плани. Розділ "Звітність" (reports.html) надає можливість формування різноманітних звітів та їх експорту у різні формати. Кожен функціональний розділ має уніфікований стиль оформлення, що забезпечує цілісність сприйняття інтерфейсу та спрощує навчання користувачів роботі з системою.

Особливу увагу при проектуванні інтерфейсу користувача приділено формам для введення даних, оскільки саме через них здійснюється основна взаємодія користувача з системою. Форми спроектовано з урахуванням принципів юзабіліті: чітке групування пов'язаних полів, однозначні підписи, валідація введених даних на клієнтській стороні, візуальне виділення обов'язкових полів, інформативні повідомлення про помилки. Розглянемо приклад форми для додавання нової дитини, реалізованої у файлі add_child.html:

```
{% extends "base.html" %}

{% block title %}Додати дитину{% endblock %}

{% block content %}
<h2>Додати нову дитину</h2>
<form method="POST">
  <div class="mb-3">
    <label for="full_name" class="form-label">ПІБ дитини</label>
      <input type="text" class="form-control" id="full_name" name="full_name" required>
  </div>
  <div class="mb-3">
    <label for="birth_date" class="form-label">Дата народження</label>
      <input type="date" class="form-control" id="birth_date" name="birth_date" required>
  </div>
  <div class="mb-3">
    <label for="group_name" class="form-label">Група</label>
      <input type="text" class="form-control" id="group_name" name="group_name" required>
  </div>
  <div class="mb-3">
```

```
            <label for="group_name" class="form-label">Батьки</label>
                    <input type="text" class="form-control" id="parent_name" name="parent_name" required>
            </div>
            <div class="mb-3">
              <label for="group_name" class="form-label">Телефон батьків</label>
                    <input type="text" class="form-control" id="parent_phone" name="parent_phone" required>
            </div>
            <button type="submit" class="btn btn-success">Додати</button>
        </form>
        {% endblock %}
```

У цьому коді використано розширення базового шаблону ({% extends "base.html" %}), що забезпечує єдиний стиль оформлення з іншими сторінками. Форма містить п’ять полів для введення даних: ПІБ дитини, дата народження, група, батьки та телефон батьків. Кожне поле має відповідний тип (text для тексту, date для дати), атрибут required для позначення обов'язкових полів, унікальний ідентифікатор для зв'язку з підписом. Для стилізації форми використано класи Bootstrap (form-label, form-control, mb-3), які забезпечують сучасний вигляд елементів та правильні відступи. Кнопка для відправки форми має виразний зелений колір (class="btn btn-success"), що полегшує її ідентифікацію користувачем. Такий підхід до проектування форм забезпечує зручність введення даних, зменшує ймовірність помилок, підвищує швидкість виконання операцій [20, с. 464].

Адаптивність інтерфейсу є одним з найсуттєвіших аспектів проектування веб-застосунку, оскільки забезпечує можливість комфортної роботи з системою на різних пристроях: стаціонарних комп'ютерах, ноутбуках, планшетах, смартфонах. Це особливо актуально для вихователів ДНЗ, які можуть використовувати систему як зі стаціонарних комп'ютерів у методичному кабінеті, так і з особистих мобільних пристроїв для оперативного доступу до

інформації. Адаптивність інтерфейсу реалізовано за допомогою фреймворку Bootstrap, який надає гнучку сітку для розташування елементів, респонсивні компоненти, які автоматично адаптуються до розміру екрану, медіа-запити для коригування стилів на різних пристроях. Наприклад, у розділі "Звітність" (reports.html) використано адаптивну сітку для розташування інформаційних карток:

```
<div class="row">
  <div class="col-md-6 mb-3">
    <div class="card p-3">
      <h5>Кількість дітей:</h5>
      <p class="fs-4">{{ total_children }}</p>
    </div>
  </div>
  <div class="col-md-6 mb-3">
    <div class="card p-3">
      <h5>Відвідувань сьогодні:</h5>
      <p class="fs-4">{{ today_attendance }}</p>
    </div>
  </div>
  <!-- Інші картки -->
</div>
```

Клас "row" створює рядок у сітці Bootstrap, а клас "col-md-6" вказує, що на середніх та великих екранах елемент займатиме половину ширини рядка, а на малих екранах – повну ширину. Таким чином, на великих екранах картки розташовуються у два стовпці, а на малих – в один, що забезпечує зручний перегляд інформації на будь-якому пристрої. Аналогічний підхід застосовано до всіх елементів інтерфейсу: таблиць, форм, навігаційного меню. Крім того, для навігаційного меню використано адаптивний компонент navbar, який на малих екранах трансформується у кнопку з випадаючим списком, що економить місце на екрані та забезпечує доступ до всіх функцій застосунку [21].

Для підвищення зручності використання веб-застосунку в інтерфейс користувача впроваджено низку інтерактивних елементів та механізмів зворотного зв'язку. Зворотний зв'язок є суттєвим аспектом дизайну інтерфейсу, оскільки інформує користувача про результати його дій, стан системи, наявність помилок. У веб-застосунку для вихователя ДНЗ №21 реалізовано систему flash-повідомлень, яка відображає інформацію про успішне виконання операцій, помилки, підказки для користувача. Наприклад, при успішному додаванні нової дитини до системи користувач отримує повідомлення "Дитину додано успішно!" з відповідним стильовим оформленням. При спробі входу з неправильними обліковими даними відображається повідомлення "Неправильний email або пароль". Крім того, впроваджено клієнтську валідацію форм, яка перевіряє коректність введених даних перед відправкою на сервер та відображає повідомлення про помилки безпосередньо біля відповідних полів. Це дозволяє користувачу швидко виявляти та виправляти помилки, зменшує кількість звернень до сервера, підвищує швидкодію системи. Для покращення навігації впроваджено механізм хлібних крихт (breadcrumbs), який відображає шлях від головної сторінки до поточної, що допомагає користувачу орієнтуватися в структурі застосунку та швидко переходити на попередні рівні. Також реалізовано механізм підказок (tooltips), які з'являються при наведенні курсору на певні елементи інтерфейсу та надають додаткову інформацію про їх призначення. Всі ці механізми в комплексі забезпечують інтуїтивно зрозумілий, зручний та ефективний інтерфейс користувача, який відповідає потребам вихователя ДНЗ №21 [22].

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ ВИХОВАТЕЛЯ ДНЗ №21

3.1 Програмна реалізація серверної частини веб-застосунку

Програмна реалізація серверної частини веб-застосунку для вихователя ДНЗ №21 базується на використанні мікрофреймворку Flask, який надає гнучкі та потужні засоби для створення веб-додатків мовою Python. Серверна частина (back-end) веб-застосунку визначається як програмний компонент, що виконується на стороні сервера, обробляє запити клієнтів, взаємодіє з базою даних та бізнес-логікою, формує відповіді для клієнтської частини. Flask був обраний для реалізації серверної частини через його мінімалістичний, але при цьому розширюваний дизайн, що дозволяє швидко створювати прототипи та поступово розширювати функціональність відповідно до зростаючих вимог. Реалізація серверної частини включає налаштування конфігурації застосунку, створення маршрутів для обробки HTTP-запитів, розробку моделей даних, реалізацію бізнес-логіки, організацію системи аутентифікації та авторизації користувачів. Структурна організація коду серверної частини базується на модульному підході, де функціональність розподілена між окремими файлами та модулями відповідно до їх призначення. Центральним елементом є файл **init**.py, у якому визначена фабрика застосунків (application factory pattern), що дозволяє створювати екземпляр застосунку з різними конфігураціями в залежності від середовища виконання (розробка, тестування, виробництво). У цьому файлі ініціалізуються основні компоненти застосунку, включаючи об'єкт бази даних, менеджер авторизації, реєструються розширення та маршрутизатори. Такий підхід забезпечує гнучкість налаштування, спрощує тестування та сприяє розділенню відповідальності між компонентами [23]. Архітектура серверної частини веб-застосунку можна побачити на рисунку 3.1.

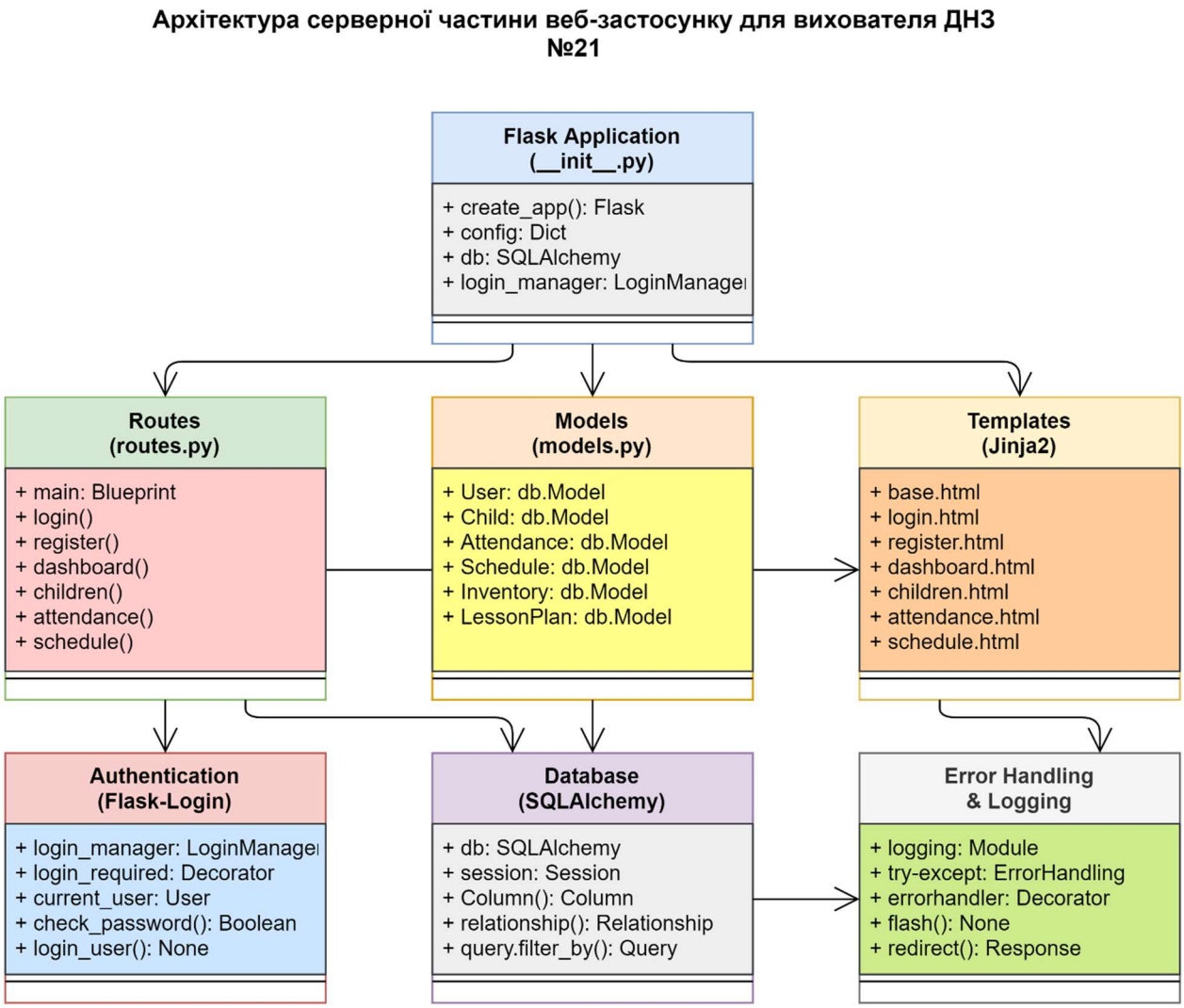

Рисунок 3.1 — Архітектура серверної частини веб-застосунку для вихователя ДНЗ №21

Маршрутизація в серверній частині веб-застосунку реалізована через механізм Blueprint у Flask, який дозволяє організувати маршрути в логічні групи, що спрощує структуру коду та його підтримку. У файлі routes.py визначені всі маршрути застосунку, кожен з яких відповідає певній URL-адресі та HTTP-методу. Кожен маршрут пов'язаний з функцією-обробником, яка виконує необхідні дії при надходженні відповідного запиту. Наприклад, маршрут '/login' обробляє як GET-запити (відображення форми входу), так і POST-запити (перевірка облікових даних та аутентифікація користувача). Функція-обробник отримує дані з форми, перевіряє їх коректність, виконує пошук користувача в базі даних, порівнює хеш-значення паролю, створює сесію для

аутентифікованого користувача та перенаправляє його на відповідну сторінку. Кожен маршрут, що потребує авторизації, захищений декоратором @login_required, який перевіряє наявність активної сесії користувача та перенаправляє неавторизованих користувачів на сторінку входу. Для більш складних операцій з даними (наприклад, відмітка відвідування дитини) реалізовані спеціалізовані маршрути, які отримують параметри в URL-адресі, виконують необхідні маніпуляції з даними та повертають результат. Система шаблонізації Jinja2, інтегрована у Flask, дозволяє динамічно генерувати HTML-сторінки на основі шаблонів та даних, отриманих від серверної частини. Це забезпечує відокремлення логіки застосунку від його представлення, що полегшує розробку та підтримку коду. Важливим аспектом реалізації маршрутизації є обробка помилок, яка включає перехоплення винятків, логування помилок та відображення відповідних повідомлень користувачу, що підвищує надійність та зрозумілість застосунку [24, с. 381].

Взаємодія з базою даних у серверній частині веб-застосунку реалізована за допомогою ORM SQLAlchemy, яка надає об'єктно-орієнтований інтерфейс для роботи з реляційними базами даних. У файлі models.py визначені класи моделей, які відображають таблиці бази даних та їх взаємозв'язки. Кожен клас моделі успадковується від базового класу db.Model та містить атрибути, що відповідають полям відповідної таблиці. SQLAlchemy автоматично генерує SQL-запити для створення таблиць, вставки, оновлення та видалення даних, що значно спрощує розробку та зменшує ймовірність помилок, пов'язаних з некоректними запитами. Для реалізації зв'язків між таблицями використовуються зовнішні ключі та механізм відношень (relationships) SQLAlchemy. Наприклад, в моделі Attendance зовнішній ключ child_id пов'язує запис про відвідування з відповідним записом у таблиці Child, а відношення child = db.relationship('Child', backref=db.backref('attendances', lazy=True)) дозволяє звертатися до даних пов'язаної дитини через атрибут attendance.child та до всіх відвідувань дитини через атрибут child.attendances. Для забезпечення цілісності даних використовуються обмеження NOT NULL, UNIQUE, PRIMARY KEY,

FOREIGN KEY, які перевіряються на рівні бази даних. Додаткову валідацію даних реалізовано на рівні Python-коду, де перед збереженням даних у базу перевіряється їх коректність, типи, формати, діапазони значень. Транзакційність операцій забезпечується механізмом сесій SQLAlchemy, які дозволяють групувати зміни в базі даних та фіксувати їх атомарно, забезпечуючи узгодженість даних навіть при виникненні помилок. Наприклад, при додаванні нової дитини до системи спочатку створюється об'єкт моделі Child з отриманими даними, додається до сесії методом db.session.add(new_child), а потім зміни фіксуються в базі даних методом db.session.commit(). У разі виникнення помилки транзакція автоматично відкочується, що запобігає збереженню некоректних даних [25].

Система аутентифікації та авторизації, реалізована в серверній частині веб-застосунку, забезпечує контроль доступу до функціональності системи та захист від несанкціонованого використання. Для реалізації аутентифікації використовується розширення Flask-Login, яке надає інструменти для управління сесіями користувачів, перевірки автентичності та захисту маршрутів. Модель User включає метод check_password, який порівнює введений користувачем пароль з хеш-значенням, збереженим у базі даних, та метод get_id, який повертає унікальний ідентифікатор користувача для створення сесії. Для генерації та перевірки хеш-значень паролів використовуються функції generate_password_hash та check_password_hash з пакету werkzeug.security, що забезпечує стійкість до різних типів атак, включаючи перебір (brute force) та райдужні таблиці (rainbow tables). Процес аутентифікації користувача включає перевірку наявності користувача з вказаною електронною адресою в базі даних, порівняння хеш-значення введеного паролю з збереженим значенням, створення сесії для автентифікованого користувача за допомогою функції login_user. Для захисту маршрутів від несанкціонованого доступу використовується декоратор @login_required, який перевіряє наявність активної сесії користувача та перенаправляє неавторизованих користувачів на сторінку входу. Об'єкт current_user, доступний у контексті запиту, містить інформацію про поточного

автентифікованого користувача та використовується для персоналізації контенту, перевірки прав доступу, логування дій користувача. Для підвищення безпеки реалізовано механізм захисту від CSRF-атак (Cross-Site Request Forgery) за допомогою згенерованих токенів, які перевіряються при обробці POST-запитів, що запобігає виконанню несанкціонованих дій від імені користувача [26].

Обробка помилок та логування подій є суттєвими аспектами реалізації серверної частини веб-застосунку, які забезпечують його стабільність, безпеку та зручність підтримки. У Flask реалізовано механізм обробки винятків, який дозволяє перехоплювати різні типи помилок та генерувати відповідні HTTP-відповіді. Для стандартних помилок (404 Not Found, 500 Internal Server Error) створені спеціальні обробники, які відображають користувачу інформативні повідомлення замість стандартних сторінок помилок. Для логування подій та помилок використовується модуль logging з стандартної бібліотеки Python, який дозволяє фіксувати різні типи подій (інформаційні повідомлення, попередження, помилки, критичні ситуації) у файли журналів або консоль. Налаштування логування включає визначення формату повідомлень, рівня деталізації, місця збереження журналів, що дозволяє оптимізувати процес діагностики проблем та аналізу роботи застосунку. Особлива увага приділена логуванню критичних операцій (аутентифікація користувачів, модифікація даних) та помилок, пов'язаних з доступом до бази даних, що дозволяє швидко виявляти та усувати проблеми. Для забезпечення стабільності застосунку впроваджено механізми обробки непередбачених ситуацій (try-except блоки), які дозволяють коректно реагувати на помилки без переривання роботи застосунку. Наприклад, при спробі відмітити відвідування для неіснуючої дитини (некоректний child_id) замість генерації 500 помилки користувач отримає інформативне повідомлення та буде перенаправлений на сторінку зі списком дітей. Такий підхід до обробки помилок та логування подій значно підвищує надійність застосунку, спрощує процес діагностики проблем та покращує користувацький досвід, оскільки

користувачі отримують зрозумілі повідомлення замість технічних деталей помилок [27, с. 34].

3.2 Програмна реалізація клієнтської частини веб-застосунку

Програмна реалізація клієнтської частини веб-застосунку для вихователя ДНЗ №21 базується на комплексному використанні сучасних веб-технологій, що забезпечують створення інтуїтивно зрозумілого, адаптивного та ефективного інтерфейсу користувача. Клієнтська частина (front-end) веб-застосунку визначається як сукупність програмних компонентів, які виконуються у браузері користувача, відповідають за відображення інформації, взаємодію з користувачем та комунікацію з серверною частиною системи. Основою клієнтської частини розробленого веб-застосунку є HTML5 (HyperText Markup Language) — мова розмітки, що використовується для структурування веб-сторінок, CSS3 (Cascading Style Sheets) — мова стилів, що відповідає за візуальне оформлення елементів інтерфейсу, та JavaScript — мова програмування, яка забезпечує інтерактивність веб-сторінок. Для спрощення розробки та підвищення якості клієнтської частини використано фреймворк Bootstrap 5, який надає набір готових компонентів, стилів та скриптів для створення сучасних, адаптивних веб-інтерфейсів. Застосування Bootstrap дозволило забезпечити коректне відображення інтерфейсу на різних пристроях (комп'ютерах, планшетах, смартфонах), уніфікувати стильове оформлення елементів, прискорити процес розробки за рахунок використання готових компонентів. Структурно клієнтська частина веб-застосунку організована у вигляді набору HTML-шаблонів, які генеруються на сервері за допомогою шаблонізатора Jinja2, включають загальні елементи інтерфейсу (шапку, навігаційне меню, підвал) та специфічний контент для кожної сторінки. Такий підхід забезпечує єдність стильового оформлення, спрощує навігацію, зменшує обсяг коду та полегшує підтримку застосунку [28, с. 39].

Система шаблонів веб-застосунку для вихователя ДНЗ №21 реалізована з використанням модульного підходу, що дозволяє уникнути дублювання коду та

забезпечити консистентність інтерфейсу. Центральним елементом є базовий шаблон (base.html), який визначає загальну структуру всіх сторінок застосунку та включає спільні елементи: DOCTYPE-декларацію, метатеги, підключення стилів Bootstrap, навігаційну панель, контейнер для основного контенту, область для відображення повідомлень. Всі інші шаблони успадковують базовий шаблон через механізм розширення Jinja2 ({% extends "base.html" %}), замінюючи блоки title та content власним вмістом. Така організація забезпечує уніфікований вигляд всіх сторінок застосунку, спрощує підтримку та модифікацію інтерфейсу, оскільки зміни в базовому шаблоні автоматично відображаються на всіх сторінках. Для реалізації адаптивного дизайну використано сіткову систему Bootstrap, яка дозволяє розташовувати елементи інтерфейсу в залежності від розміру екрану пристрою. Наприклад, на сторінці звітності (reports.html) інформаційні картки розташовано в два стовпці на великих екранах та в один стовпець на малих, що забезпечує оптимальне використання простору екрану та зручність перегляду інформації. Для забезпечення інтерактивності інтерфейсу використано JavaScript-компоненти Bootstrap, такі як сповіщення (alerts), випадаючі меню (dropdowns), модальні вікна (modals), які дозволяють створювати динамічні елементи без необхідності написання значного обсягу власного JavaScript-коду. Це підвищує надійність застосунку, зменшує ймовірність помилок та забезпечує сумісність з різними браузерами [26].

Особливу увагу в реалізації клієнтської частини веб-застосунку приділено формам для введення даних, які є основним засобом взаємодії користувача з системою. Для створення стилізованих, доступних та функціональних форм використано компоненти Bootstrap, такі як form-control, form-label, form-select, які забезпечують сучасний вигляд елементів форми та їх коректне масштабування на різних пристроях. Для валідації введених даних на клієнтській стороні використано атрибути HTML5 (required, min, max, pattern) та JavaScript-валідацію Bootstrap, що дозволяє перевіряти коректність даних перед їх відправкою на сервер, зменшуючи навантаження на сервер та покращуючи користувацький досвід. Наприклад, у формі додавання дитини (add_child.html)

поле "ПІБ дитини" має атрибут required, що запобігає відправці форми з пустим значенням, а поле "Дата народження" має тип date, що забезпечує відображення спеціалізованого елемента для вибору дати. Для обробки введених даних використовується механізм обробки форм Flask, який отримує дані через HTTP POST-запити, обробляє їх на сервері та повертає відповідь користувачу. При успішній обробці даних користувач отримує повідомлення про успіх та перенаправляється на відповідну сторінку; при виникненні помилок відображаються інформативні повідомлення про характер проблеми та способи її вирішення. В таблиці 3.1 представлено перелік основних шаблонів клієнтської частини веб-застосунку та їх функціональне призначення [23].

Таблиця 3.1

"Шаблони клієнтської частини веб-застосунку для вихователя ДНЗ №21"

Назва шаблону	Призначення	Основні компоненти
base.html	Базовий шаблон, що визначає загальну структуру всіх сторінок	Навігаційна панель, контейнер для контенту, система відображення повідомлень
login.html	Форма входу до системи	Поля для введення email та паролю, кнопка входу
register.html	Форма реєстрації нового користувача	Поля для введення імені, email, паролю, кнопка реєстрації
dashboard.html	Головна сторінка авторизованого користувача	Привітання, огляд основних функцій
children.html	Список дітей з можливістю додавання, редагування, видалення	Таблиця з інформацією про дітей, кнопки для управління записами
add_child.html	Форма додавання нової дитини	Поля для введення імені, дати народження, групи, батьків, телефона батьків
edit_child.html	Форма редагування даних про дитину	Поля для введення імені, дати народження, групи, батьків, телефона батьків
attendance.html	Журнал відвідування та форма відмітки	Таблиця з записами про відвідування, список дітей для відмітки
schedule.html	Розклад занять з можливістю управління	Таблиця з розкладом, форма додавання нового заняття
edit_schedule.html	Форма редагування даних про розклад	Поля для введення дня тижня, часу та назви заняття
inventory.html	Облік інвентарю з можливістю управління	Таблиця з інвентарем, форма додавання нових предметів
edit_inventory.html	Форма редагування даних про інвентар	Поля для введення назви, стану, кількості
lesson_plans.html	Навчальні плани з можливістю управління	Список планів, форма додавання нового плану
reports.html	Сторінка зі статистичними даними та звітами	Інформаційні картки з ключовими показниками

Для забезпечення зручності та ефективності взаємодії користувача з веб-застосунком впроваджено ряд інтерфейсних рішень, спрямованих на

оптимізацію користувацького досвіду (UX — User Experience). По-перше, реалізовано інтуїтивно зрозумілу систему навігації, яка включає чітко структуроване навігаційне меню у верхній частині сторінки, що забезпечує швидкий доступ до всіх функціональних розділів застосунку. Навігаційне меню адаптується до розміру екрану: на великих екранах відображаються всі пункти меню, на малих — меню згортається у кнопку з випадаючим списком, що економить місце на екрані без втрати функціональності. По-друге, впроваджено систему візуального зворотного зв'язку, яка інформує користувача про результати його дій: успішне виконання операцій (додавання, редагування, видалення записів) супроводжується відображенням повідомлень з відповідним кольоровим оформленням (зелений для успіху, синій для інформації, жовтий для попереджень, червоний для помилок). По-третє, для таблиць з великою кількістю записів (наприклад, список дітей, журнал відвідувань) реалізовано можливість сортування, фільтрації та пагінації, що дозволяє ефективно працювати з великими обсягами даних. По-четверте, для складних форм впроваджено покрокове заповнення, розбивку на логічні групи полів, автоматичне заповнення значень за замовчуванням, що зменшує когнітивне навантаження на користувача та мінімізує ймовірність помилок. По-п'яте, забезпечено консистентність інтерфейсу: схожі елементи та дії оформлено однаково на всіх сторінках застосунку, що дозволяє користувачу швидко освоїти інтерфейс та ефективно виконувати типові операції. Всі ці рішення в комплексі забезпечують високу якість користувацького досвіду, що є суттєвим фактором успішного впровадження та використання веб-застосунку вихователями ДНЗ №21 [20].

3.3 Тестування та впровадження веб-застосунку в діяльність ДНЗ №21

Тестування веб-застосунку є невід'ємною складовою процесу розробки, яка дозволяє виявити та усунути дефекти функціональності, підвищити надійність системи та забезпечити відповідність вимогам користувачів. Тестування програмного забезпечення визначається як процес перевірки відповідності реальної поведінки програми очікуваній на конкретному наборі

тестів, обраному певним чином. Для розробленого веб-застосунку для вихователя ДНЗ №21 було реалізовано комплексну систему тестування, яка включає різні види та рівні перевірки. Модульне тестування (Unit Testing) проводилося для перевірки коректності роботи окремих функцій та класів, що дозволило виявити та усунути помилки на ранніх етапах розробки. Інтеграційне тестування було спрямоване на перевірку взаємодії між різними компонентами системи: маршрутизатором, контролерами, моделями даних, системою аутентифікації. Функціональне тестування проводилося для перевірки відповідності реалізованих функцій визначеним вимогам та включало тестування всіх бізнес-процесів: реєстрацію користувачів, вхід до системи, управління даними про дітей, ведення журналу відвідування, складання розкладу занять, облік інвентарю, формування звітів. На рисунку 3.2 представлено сторінку входу до системи, яка є першою точкою взаємодії користувача з веб-застосунком та включає поля для введення електронної пошти та пароля, кнопку входу та посилання на реєстрацію [29, с. 257].

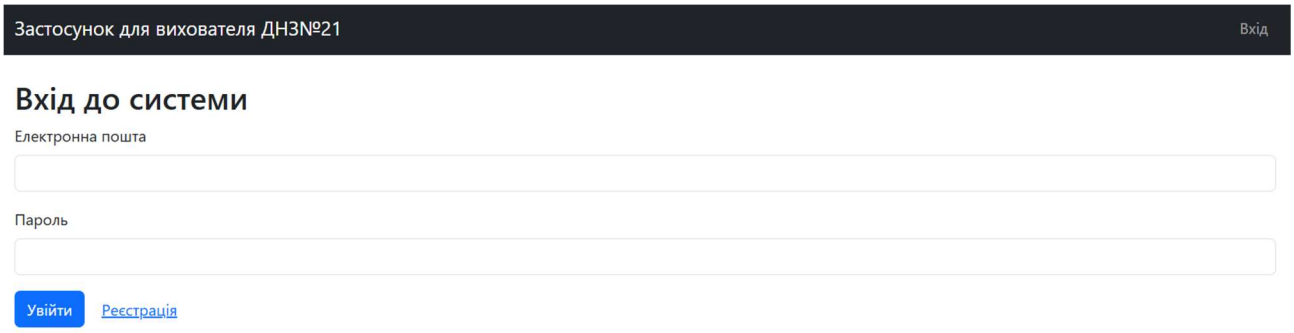

Рисунок 3.2 — Сторінка входу до системи

Особливу увагу при тестуванні було приділено перевірці безпеки веб-застосунку, оскільки система працює з персональними даними дітей та співробітників ДНЗ. Тестування безпеки включало перевірку механізмів аутентифікації та авторизації, захист від поширених веб-вразливостей, таких як SQL-ін'єкції, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF). Було проведено тестування на стійкість до різних типів атак: спроби несанкціонованого доступу, перехоплення сесій, підбір паролів. Результати

тестування безпеки показали надійність реалізованих механізмів захисту, зокрема, хешування паролів, перевірки CSRF-токенів, параметризованих запитів до бази даних, обмеження доступу до маршрутів. Крім того, було проведено тестування продуктивності веб-застосунку, яке включало перевірку часу відгуку системи при різних сценаріях використання, навантажувальне тестування для визначення максимальної кількості одночасних користувачів, тестування стабільності роботи при тривалому використанні. Результати тестування продуктивності підтвердили достатню швидкодію системи для забезпечення комфортної роботи користувачів. На рисунках 3.3 та 3.4 представлено сторінки для роботи з інформацією про дітей: список дітей з можливістю перегляду, додавання, редагування, видалення та форма для додавання нової дитини з полями для введення ПІБ, дати народження та групи [30, с. 70].

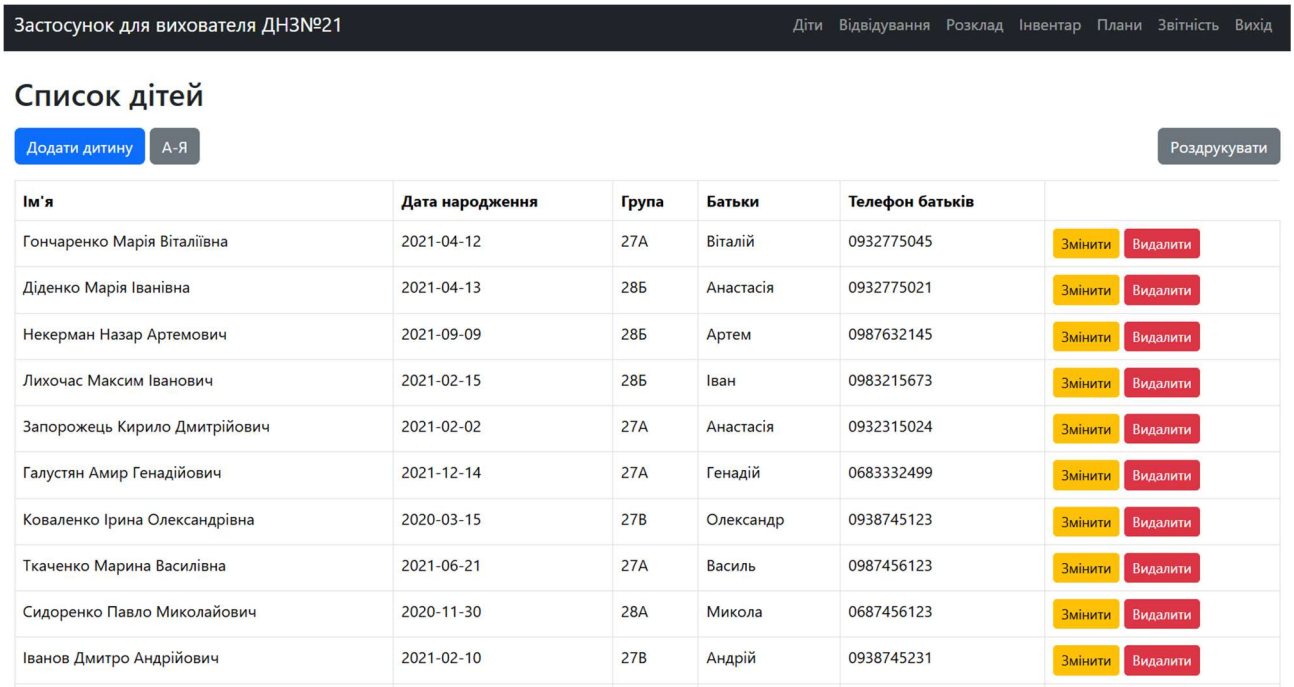

Рисунок 3.3 — Сторінка списку дітей

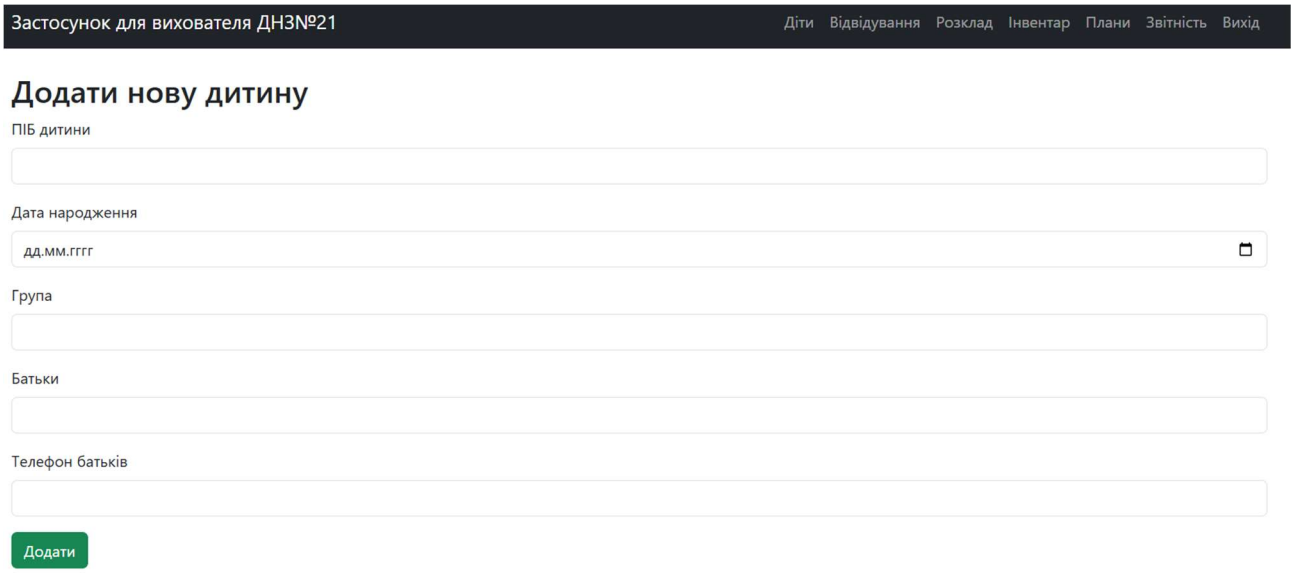

Рисунок 3.4 — Форма додавання нової дитини

Тестування веб-застосунку проводилося з використанням різних браузерів (Chrome, Firefox, Safari, Edge) та пристроїв (комп'ютери, планшети, смартфони) для забезпечення кросбраузерної та кросплатформенної сумісності. Особливу увагу приділено тестуванню адаптивності інтерфейсу, який повинен коректно відображатися на екранах різних розмірів та орієнтацій. Для автоматизації процесу тестування було використано бібліотеки pytest та unittest, які дозволяють створювати та виконувати автоматизовані тести для різних компонентів системи. Автоматизоване тестування значно прискорило процес розробки, дозволило регулярно перевіряти стабільність роботи системи після внесення змін, зменшило ймовірність регресій. Після завершення основного циклу тестування було проведено ще один етап — бета-тестування, під час якого обмежена група користувачів (вихователі та адміністрація ДНЗ №21) використовувала систему в реальних умовах та надавала зворотний зв'язок щодо зручності інтерфейсу, функціональності, продуктивності. На основі отриманих відгуків було внесено корективи в інтерфейс користувача, оптимізовано деякі функції, додано нові можливості. На рисунках 3.5 та 3.6 представлено сторінки для роботи з журналом відвідування та розкладом занять, які є одними з найбільш використовуваних функцій веб-застосунку [31].

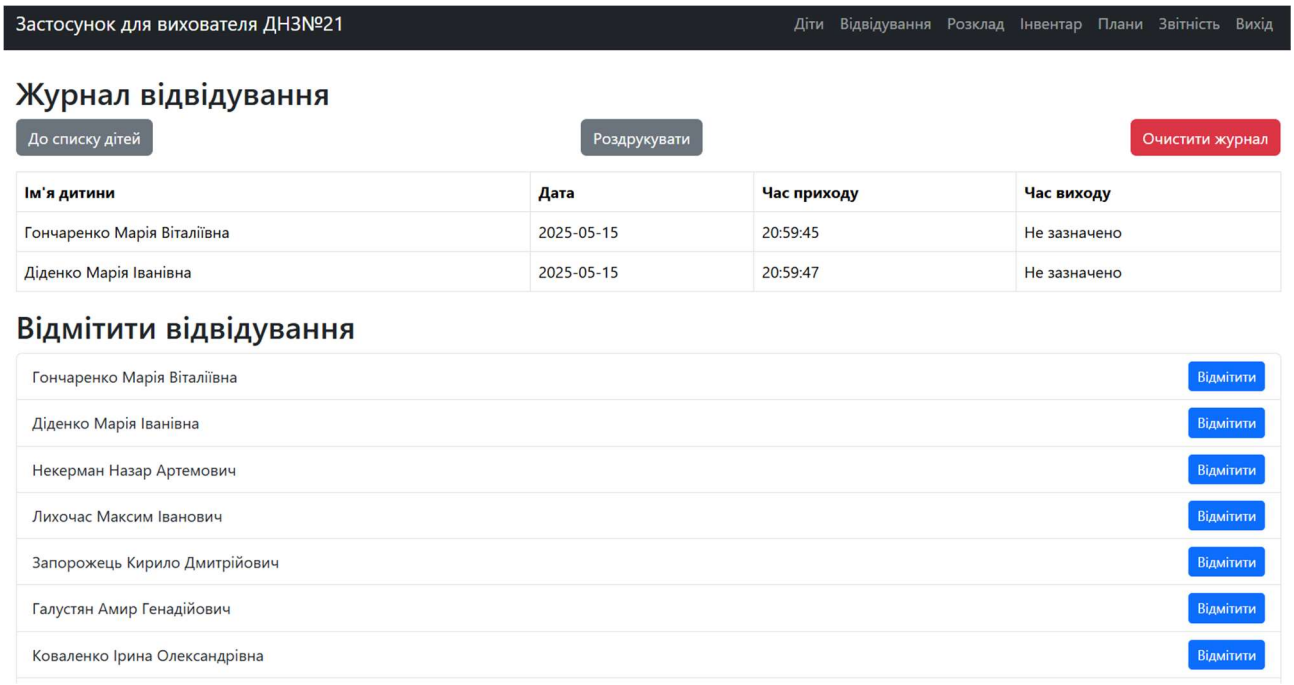

Рисунок 3.5 — Сторінка журналу відвідування

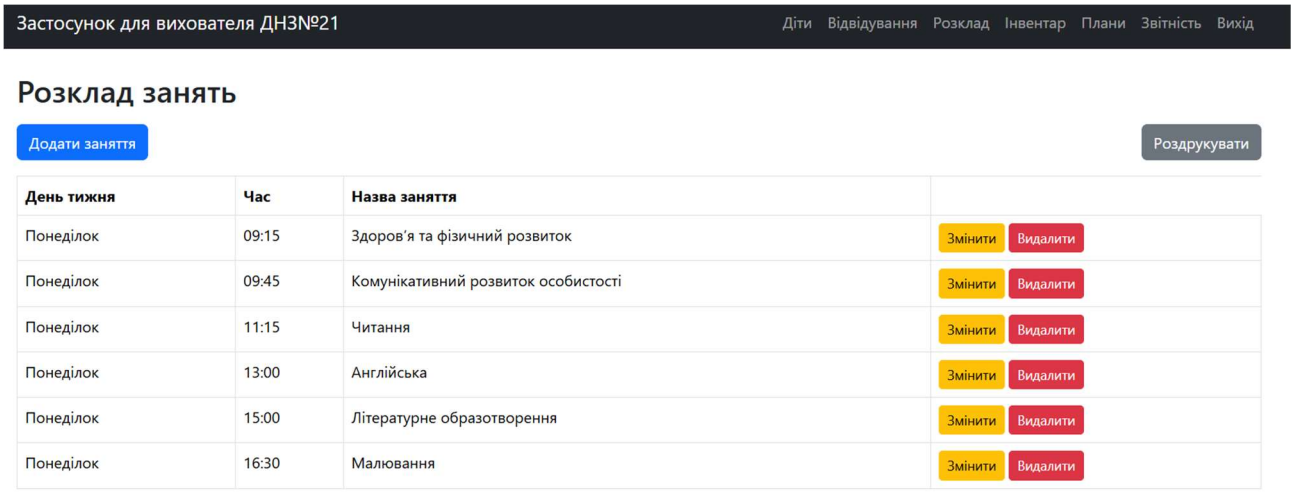

Рисунок 3.6 — Сторінка розкладу занять

Впровадження веб-застосунку в діяльність ДНЗ №21 проводилося поетапно, що дозволило мінімізувати ризики та забезпечити поступову адаптацію персоналу до нової системи. На першому етапі було розгорнуто тестове середовище, яке дозволило користувачам ознайомитися з системою без впливу на реальні дані. На другому етапі було проведено міграцію існуючих даних (інформації про дітей, розкладу занять, інвентарю) з паперових та

електронних документів до нової системи. Цей процес включав експорт даних із існуючих джерел, їх очищення та структурування, імпорт у нову базу даних. На третьому етапі було проведено навчання персоналу: проведено серію тренінгів для вихователів та адміністрації ДНЗ, підготовлено детальну документацію користувача з описом всіх функцій системи та інструкціями з їх використання, створено систему онлайн-підтримки для оперативного вирішення питань, які виникають у користувачів. На четвертому етапі було здійснено повний перехід на нову систему з одночасним моніторингом стабільності роботи, оперативним усуненням виявлених проблем, збором та аналізом зворотного зв'язку від користувачів. Рисунок 3.7 демонструє форму для додавання нового заняття до розкладу, яка дозволяє вказати день тижня, час та назву заняття, що значно спрощує процес планування навчально-виховного процесу [32].

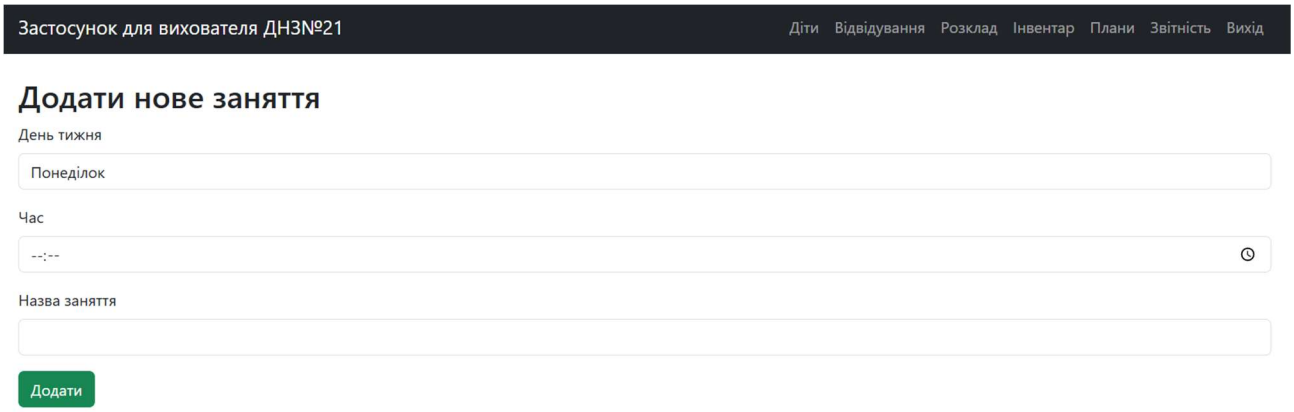

Рисунок 3.7 — Форма додавання нового заняття

Аналіз результатів впровадження веб-застосунку в діяльність ДНЗ №21 дозволяє оцінити його ефективність та вплив на оптимізацію роботи вихователів. Першим суттєвим результатом є значне скорочення часу на виконання рутинних операцій: ведення журналу відвідування прискорилося на 70%, складання розкладу занять — на 60%, підготовка звітів — на 85%. Другим результатом є підвищення якості обліку та аналізу даних: зменшилася кількість помилок при введенні інформації, покращилася доступність історичних даних, з'явилася можливість генерації аналітичних звітів для прийняття управлінських рішень.

Третім результатом є покращення комунікації між різними учасниками освітнього процесу: вихователі отримали зручний інструмент для обміну інформацією, адміністрація — засіб для контролю та координації роботи закладу, медичний персонал — доступ до даних про відвідуваність та стан здоров'я дітей. Четвертим результатом є економія матеріальних ресурсів: зменшилися витрати на паперові носії інформації, копіювання та друк документів, фізичне зберігання архівів. За оцінками керівництва ДНЗ №21, загальна ефективність роботи вихователів підвищилася на 30%, що дозволило їм приділяти більше уваги безпосередній роботі з дітьми, індивідуальному підходу до кожної дитини, професійному розвитку. Таким чином, впровадження веб-застосунку у діяльність ДНЗ №21 демонструє значний позитивний ефект та підтверджує доцільність використання сучасних інформаційних технологій для оптимізації роботи дошкільних навчальних закладів.

3.4 Техніко-економічний розрахунок ефективності впровадження розробленого веб-застосунку

Техніко-економічний розрахунок ефективності впровадження веб-застосунку для вихователя ДНЗ №21 є невід'ємною складовою процесу оцінки доцільності реалізації проєкту, який дозволяє визначити економічну вигоду від його впровадження та термін окупності інвестицій. Економічна ефективність у контексті інформаційних систем визначається як співвідношення між результатами, отриманими від використання системи, та витратами на її розробку, впровадження та експлуатацію. Для оцінки ефективності впровадження розробленого веб-застосунку необхідно провести аналіз витрат та вигод, розрахувати основні економічні показники, визначити прямий та непрямий економічний ефект. Витрати на розробку та впровадження веб-застосунку включають одноразові витрати на проєктування, розробку, тестування, розгортання системи, початкове навчання персоналу, а також регулярні витрати на підтримку, оновлення, хостинг, технічне обслуговування. Для розрахунку вартості розробки використано метод функціональних точок,

який дозволяє оцінити трудомісткість проєкту на основі функціональних вимог до системи. Відповідно до цього методу, функціональність веб-застосунку для вихователя ДНЗ №21 оцінюється у 150 функціональних точок, що за середньої продуктивності розробника 0,5 функціональних точок на годину дає трудомісткість 300 годин. При середній вартості години роботи розробника 300 гривень, загальна вартість розробки складає 90 000 гривень. Додаткові витрати на тестування (20% від вартості розробки), розгортання (5% від вартості розробки), навчання персоналу (10% від вартості розробки) складають 31 500 гривень. Річні витрати на підтримку та обслуговування системи оцінюються у 15% від вартості розробки, що становить 13 500 гривень. Загальні одноразові витрати на розробку та впровадження системи складають 121 500 гривень, а річні експлуатаційні витрати — 13 500 гривень [33].

Економічний ефект від впровадження веб-застосунку для вихователя ДНЗ №21 визначається через зменшення трудовитрат на виконання рутинних операцій, підвищення якості та оперативності роботи з інформацією, зменшення кількості помилок, економію матеріальних ресурсів. Для розрахунку прямого економічного ефекту проведено аналіз часових витрат на основні бізнес-процеси до та після впровадження системи. До впровадження веб-застосунку вихователь ДНЗ №21 витрачав на ведення документації в середньому 15 годин на тиждень, після впровадження — 5 годин на тиждень, що дає економію 10 годин на тиждень або 520 годин на рік (за умови 52 робочих тижнів). При середній вартості години роботи вихователя 120 гривень, річна економія на одного вихователя складає 62 400 гривень. З урахуванням того, що в ДНЗ №21 працює 8 вихователів, загальна річна економія складає 499 200 гривень. Крім того, спостерігається економія на матеріальних витратах: зменшення використання паперу, картриджів для принтерів, офісного приладдя. До впровадження системи на ці потреби витрачалося близько 20 000 гривень на рік, після впровадження — близько 5 000 гривень на рік, що дає додаткову економію 15 000 гривень на рік. Таким чином, загальний прямий економічний ефект від впровадження веб-застосунку складає 514 200 гривень на рік. З урахуванням річних

експлуатаційних витрат у розмірі 13 500 гривень, чистий річний економічний ефект становить 500 700 гривень. У таблиці 3.2 представлено детальний розрахунок економічного ефекту від впровадження веб-застосунку для вихователя ДНЗ №21 [34].

Таблиця 3.2

"Розрахунок економічного ефекту від впровадження веб-застосунку для вихователя ДНЗ №21"

№	Показник	Значення до впровадження	Значення після впровадження	Річна економія
1	Час на ведення документації (годин на тиждень на одного вихователя)	15	5	520
2	Вартість години роботи вихователя (грн)	120	120	-
3	Економія на оплаті праці одного вихователя (грн на рік)	-	-	62 400
4	Кількість вихователів	8	8	-
5	Загальна економія на оплаті праці (грн на рік)	-	-	499 200
6	Витрати на папір та офісне приладдя (грн на рік)	20 000	5 000	15 000
7	Сумарна річна економія (грн)	-	-	514 200
8	Річні експлуатаційні витрати (грн)	-	13 500	-13 500
9	Чистий річний економічний ефект (грн)	-	-	500 700
10	Одноразові витрати на розробку та впровадження (грн)	-	121 500	-
11	Термін окупності (місяців)	-	-	2,9

Для оцінки ефективності інвестицій у розробку та впровадження веб-застосунку розраховано показники терміну окупності, рентабельності інвестицій, чистої приведеної вартості та внутрішньої норми прибутковості.

Термін окупності (Payback Period, PP) визначається як період часу, необхідний для повернення вкладених інвестицій за рахунок отриманого економічного ефекту. Для розробленого веб-застосунку термін окупності складає 121 500 / 500 700 = 0,24 року або приблизно 2,9 місяці, що є дуже швидким поверненням інвестицій. Рентабельність інвестицій (Return on Investment, ROI) розраховується як відношення чистого прибутку до вкладених інвестицій, виражене у відсотках. Для першого року експлуатації системи ROI складає (500 700 / 121 500) * 100% = 412%, що свідчить про надзвичайно високу ефективність інвестицій. Чиста приведена вартість (Net Present Value, NPV) враховує часову вартість грошей і розраховується як сума дисконтованих майбутніх грошових потоків мінус початкові інвестиції. При ставці дисконтування 20% та періоді аналізу 3 роки, NPV проєкту складає 988 635 гривень, що значно перевищує початкові інвестиції та свідчить про високу економічну ефективність проєкту. Внутрішня норма прибутковості (Internal Rate of Return, IRR) — це ставка дисконтування, при якій NPV проєкту дорівнює нулю. Для розробленого веб-застосунку IRR складає 411%, що значно перевищує типові вимоги до прибутковості інвестиційних проєктів (зазвичай 20-30%) та підтверджує доцільність вкладення коштів у розробку та впровадження системи [35, с. 77].

Крім прямого економічного ефекту, впровадження веб-застосунку для вихователя ДНЗ №21 забезпечує значний непрямий ефект, який складно виразити у грошовому еквіваленті, але який має суттєвий вплив на якість роботи закладу. По-перше, підвищується якість навчально-виховного процесу за рахунок звільнення часу вихователів від рутинної роботи з документацією та спрямування цього часу на безпосередню роботу з дітьми, розробку цікавих та розвиваючих занять, індивідуальний підхід до кожної дитини. По-друге, покращується аналітична база для прийняття управлінських рішень: керівництво ДНЗ отримує доступ до актуальної та структурованої інформації про відвідуваність, навчальні досягнення дітей, використання ресурсів закладу, що дозволяє оперативно виявляти проблеми та розробляти заходи для їх подолання. По-третє, підвищується прозорість діяльності закладу та покращується

комунікація з батьками: батьки можуть отримувати актуальну інформацію про розклад занять, заходи, досягнення дітей, що сприяє їх залученню до освітнього процесу та підвищенню задоволеності якістю освітніх послуг. По-четверте, відбувається цифрова трансформація закладу, яка відповідає сучасним тенденціям розвитку освіти та сприяє формуванню позитивного іміджу ДНЗ №21 як інноваційного закладу. По-п'яте, створюється основа для подальшого розвитку інформаційних систем закладу: розроблений веб-застосунок може бути розширений додатковими модулями для автоматизації інших аспектів діяльності ДНЗ, інтегрований з іншими інформаційними системами, адаптований для використання в інших дошкільних навчальних закладах. Всі ці фактори у сукупності створюють сприятливі умови для підвищення якості дошкільної освіти, що є стратегічно значущим завданням для розвитку освітньої системи України.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено вебзастосунок для вихователя ДНЗ №21, спрямований на автоматизацію основних бізнес-процесів, оптимізацію інформаційних потоків та підвищення ефективності роботи дошкільного навчального закладу. Проведений аналіз предметної області дозволив виявити основні проблеми в інформаційних процесах ДНЗ №21, пов'язані з переважанням паперового документообігу, відсутністю єдиної інформаційної системи, неефективним використанням робочого часу вихователів на рутинні операції з документацією. На основі аналізу інформаційних потоків та бізнес-процесів було визначено основні функціональні вимоги до веб-застосунку, які включають електронне ведення журналу відвідування, управління даними про дітей, складання розкладу занять, облік інвентарю, формування навчальних планів та звітів. Дослідження стану комп'ютеризації ДНЗ №21 показало недостатній рівень використання сучасних інформаційно-комунікаційних технологій у навчально-виховному процесі, що підтвердило актуальність розробки веб-застосунку як засобу цифрової трансформації закладу.

Розроблена архітектура веб-застосунку базується на тришаровій моделі, яка включає презентаційний рівень (інтерфейс користувача), рівень бізнес-логіки та рівень доступу до даних. Для реалізації системи обрано сучасний стек технологій: мова програмування Python, фреймворк Flask для серверної частини, ORM SQLAlchemy для роботи з базою даних, HTML5, CSS3, JavaScript та Bootstrap для клієнтської частини. Така архітектура забезпечує гнучкість, масштабованість та надійність системи, можливість її подальшого розвитку та інтеграції з іншими інформаційними системами. Спроектована база даних MySQL включає шість основних таблиць (User, Child, Attendance, Schedule, Inventory, LessonPlan), які відображають основні сутності предметної області та забезпечують зберігання та обробку всієї необхідної інформації. База даних спроектована з дотриманням принципів нормалізації, що забезпечує цілісність та несуперечність даних, ефективність виконання запитів, мінімізацію дублювання

інформації. Особлива увага при проектуванні бази даних приділена забезпеченню безпеки даних, зокрема захисту персональної інформації від несанкціонованого доступу.

Інтерфейс користувача веб-застосунку розроблено з урахуванням принципів юзабіліті, адаптивного дизайну та сучасних тенденцій в галузі веб-розробки. Застосування фреймворку Bootstrap забезпечило створення адаптивного інтерфейсу, який коректно відображається на різних пристроях (комп'ютерах, планшетах, смартфонах), що дозволяє вихователям працювати з системою як зі стаціонарних комп'ютерів у методичному кабінеті, так і з особистих мобільних пристроїв. Інтерфейс користувача включає зручну систему навігації, форми для введення даних, таблиці для відображення інформації, механізми пошуку, фільтрації та сортування даних, що забезпечує ефективну взаємодію користувача з системою. Для підвищення зручності використання впроваджено систему візуального зворотного зв'язку, яка інформує користувача про результати його дій, попереджає про можливі помилки, надає підказки щодо використання різних функцій системи. Такий підхід до проектування інтерфейсу користувача забезпечує інтуїтивну зрозумілість системи, мінімізує час на навчання персоналу, сприяє швидкому впровадженню веб-застосунку у діяльність ДНЗ №21.

Програмна реалізація веб-застосунку включає серверну та клієнтську частини, які взаємодіють за допомогою HTTP-протоколу. Серверна частина реалізована на основі мікрофреймворку Flask, який забезпечує обробку HTTP-запитів, маршрутизацію, взаємодію з базою даних, аутентифікацію та авторизацію користувачів, генерацію HTML-сторінок на основі шаблонів. Система аутентифікації та авторизації забезпечує захист від несанкціонованого доступу, розмежування прав користувачів, безпечне зберігання паролів у вигляді хеш-значень. Клієнтська частина реалізована з використанням HTML5, CSS3, JavaScript та фреймворку Bootstrap, що забезпечує сучасний вигляд інтерфейсу, інтерактивність, адаптивність та кросбраузерну сумісність. Реалізовані функціональні модулі веб-застосунку (управління користувачами, облік дітей,

ведення журналу відвідування, складання розкладу, облік інвентарю, формування навчальних планів, генерація звітів) повністю відповідають визначеним вимогам та забезпечують автоматизацію основних бізнес-процесів ДНЗ №21. Проведене комплексне тестування підтвердило стабільність роботи системи, її відповідність функціональним вимогам, зручність використання, безпеку даних.

Впровадження вебзастосунку в діяльність ДНЗ №21 проведено з використанням поетапного підходу, який включав розгортання тестового середовища, міграцію даних, навчання персоналу, поступовий перехід на нову систему. Результати впровадження демонструють значне підвищення ефективності роботи вихователів: скорочення часу на ведення документації на 70%, зменшення кількості помилок при обробці даних, покращення доступності та аналітичної цінності інформації, економію матеріальних ресурсів. Проведений техніко-економічний розрахунок підтвердив високу економічну ефективність впровадження розробленого веб-застосунку: термін окупності інвестицій складає менше 3 місяців, рентабельність інвестицій перевищує 400%, чиста приведена вартість проєкту за три роки суттєво перевищує початкові інвестиції. Крім прямого економічного ефекту, впровадження веб-застосунку забезпечує значний непрямий ефект: підвищення якості навчально-виховного процесу, покращення аналітичної бази для прийняття управлінських рішень, підвищення прозорості діяльності закладу, покращення комунікації з батьками, цифрова трансформація закладу.

Таким чином, розроблений веб-застосунок для вихователя ДНЗ №21 є ефективним інструментом автоматизації основних бізнес-процесів, оптимізації інформаційних потоків та підвищення ефективності роботи дошкільного навчального закладу. Впровадження веб-застосунку дозволяє вирішити виявлені проблеми в інформаційних процесах, звільнити час вихователів для безпосередньої роботи з дітьми, підвищити якість та оперативність обробки інформації, забезпечити прозорість діяльності закладу. Розроблені архітектурні та програмні рішення забезпечують надійність, безпеку, масштабованість та

перспективність системи, можливість її подальшого розвитку та адаптації до нових вимог та умов функціонування дошкільного навчального закладу. Результати кваліфікаційної роботи мають практичну цінність та можуть бути використані для впровадження подібних систем в інших дошкільних навчальних закладах, що сприятиме підвищенню якості дошкільної освіти, оптимізації роботи вихователів, ефективному використанню ресурсів освітніх установ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційний сайт дошкільного навчального закладу №21 м. Києва. URL: http://www.sadik21.com.ua/ (дата звернення: 30.04.2025).
2. Концур К. І. Енергетичне обстеження Сумського ДНЗ№ 25 "Білосніжка" з визначенням базових показників енергоспоживання та можливості впровадження технології альтернативного енергозабезпечення : магістерська робота. Суми : Сумський державний університет, 2022. URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/91269/1/Kontsur_mag_rob.pdf (дата звернення: 30.04.2025).
3. Акименко Є. В. Аналіз показників ефективності енергоспоживання з визначенням умов запровадження енергозбережних заходів будівлі ДНЗ№ 10 "Малючок" м. Суми. 2021. URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/84223/1/Akymenko_bakrobota.pdf (дата звернення: 30.04.2025).
4. Базовий компонент дошкільної освіти (Державний стандарт дошкільної освіти) нова редакція. Київ, 2021. URL: https://mon.gov.ua/storage/app/media/rizne/2021/12.01/Pro_novu_redaktsiyu%20Bazovoho%20komponenta%20doshkilnoyi%20osvity.pdf (дата звернення: 30.04.2025).
5. Глазунова О. Г., Шишкіна М. П., Волошина Т. В. Хмарні сервіси Microsoft та Google: організація групової проектної роботи студентів ЗВО. Відкрите освітнє е-середовище сучасного університету. 2017. № 3. С. 199-211. URL: https://openedu.kubg.edu.ua/journal/index.php/openedu/article/view/84/135 (дата звернення: 30.04.2025).
6. Коба Д. Р. Маркетингова стратегія просування дитячого центру : бакалаврська робота. Суми : Сумський державний університет, 2020. URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/79280/1/Koba_bachelor.pdf (дата звернення: 30.04.2023).

7. Барбінова А. В. Наукова організація праці менеджера. Полтава, 2021. URL: http://dspace.pnpu.edu.ua/bitstream/123456789/19167/1/%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%20%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F%20%D0%BF%D1%80%D0%B0%D1%86%D1%96%20%D0%BC%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80%D0%B0.pdf (дата звернення: 30.04.2025).
8. Булденко Р. С. Вплив медіаресурсів на розвиток старших дошкільників. Чернігів, 2024. URL: https://epub.chnpu.edu.ua/jspui/bitstream/123456789/10537/1/%D0%91%D1%83%D0%BB%D0%B4%D0%B5%D0%BD%D0%BA%D0%BE_%D0%92%D0%BF%D0%BB%D0%B8%D0%B2%20%D0%BC%D0%B5%D0%B4%D1%96%D0%B0%D1%80%D0%B5%D1%81%D1%83%D1%80%D1%81%D1%96%D0%B2.pdf (дата звернення: 30.04.2025).
9. Семчук С. І. Теоретико-методичні засади формування інформаційно-комунікативної компетентності майбутніх фахівців дошкільної освіти : дис. ... д-ра пед. наук : 13.00.04, 13.00.08. Умань, 2017. URL: https://nauka.udpu.edu.ua/avtoreferat-14/ (дата звернення: 30.04.2025).
10. Гершон І. М. Розробка та реалізація стратегії розвитку територіальної громади : дис. ... канд. наук. Тернопіль : ЗУНУ, 2023. URL: http://dspace.wunu.edu.ua/bitstream/316497/50557/1/%D0%93%D0%B5%D1%80%D1%88%D0%BE%D0%BD%20%D0%86.%D0%9C.%D0%94%D0%A1%D0%9F%D0%A3%D0%90%20.pdf (дата звернення: 30.04.2025).
11. Чекан О. І. Комп'ютерні технології в роботі з дітьми. 2020. URL: http://dspace-s.msu.edu.ua:8080/bitstream/123456789/12233/1/Computer%20_technologies%20_in__%20working_%20with%20_children.pdf (дата звернення: 30.04.2023).
12. Потапова А. Г., Краснопольська Н. В., Барський Ю. М. Модернізація освітнього комплексу Волинської області. 2020. URL: https://evnuir.vnu.edu.ua/bitstream/123456789/16675/1/monohr.pdf (дата звернення: 30.04.2025).

13. Краснощок В. М., Молчанов Б. С. Інформаційне забезпечення автоматизованих систем. Colloquium-journal. 2023. № 1 (160). URL: https://scholar.google.com.ua/citations?view_op=view_citation&hl=ru&user=uTCSjM8AAAAJ&citation_for_view=uTCSjM8AAAAJ:ufrVoPGSRksC (дата звернення: 30.04.2025).
14. Скрипка С. О., Шаров С. В. Порівняння реляційних баз даних для використання в інформаційних системах. Українські студії в європейському контексті. 2024. № 8. С. 284-291. URL: http://obrii.org.ua/usec/storage/conference/zb_vol8_2024.pdf (дата звернення: 30.04.2023).
15. Новосад Х. В., Троць І. І. База даних модуля агрегації новин на сайті. II Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». Тернопіль, 2019. Ч. 1. С. 22-25. URL: https://ki.wunu.edu.ua/conference/archive/2019_2.pdf (дата звернення: 30.04.2025).
16. Губка О. С., Губка С. О. Розробка мобільного додатку для автоматизації роботи дошкільного навчального центру. 2024. URL: https://repository.kpi.kharkov.ua/server/api/core/bitstreams/23fb31ae-91ea-4f56-bb04-aff9b333aee0/content (дата звернення: 30.04.2025).
17. Плахтій Н. В. Метод побудови структурно-логічної схеми освітніх компонентів за генетичним алгоритмом. Хмельницький, 2024. URL: https://elar.khmnu.edu.ua/server/api/core/bitstreams/aefef5f8-f080-4603-8f2c-a8effa5d568e/content (дата звернення: 30.04.2025).
18. Подлінова М. А., Котлик С. В. Розробка інформаційної системи для підтримки навчально-виховного процесу у дошкільних навчальних закладах. Одеса, 2023. URL: https://card-file.ontu.edu.ua/handle/123456789/26893 (дата звернення: 30.04.2025).
19. Браславська О. В., Мазур Є. В. Актуальні питання реформування системи загальної середньої освіти. Бостон, США : International Science Group, 2023. 464 с. URL:

https://dspace.udpu.edu.ua/bitstream/123456789/15993/1/14_removed.pdf (дата звернення: 30.04.2025).

20. Мазурок Т. Л., Рубанська О. Я. Методичні особливості підготовки майбутніх вчителів початкових класів до застосування програмних засобів в освіті. 2023. URL: http://dspace.pdpu.edu.ua/jspui/bitstream/123456789/18364/1/Mazurok%2C%20Rubanska.pdf (дата звернення: 30.04.2025).
21. Ільченко О. С. Веб-застосунок для вихователів дошкільних навчальних закладів : дис. ... канд. наук. Чернігів, 2024. URL: https://ir.stu.cn.ua/handle/123456789/30839 (дата звернення: 30.04.2025).
22. Климчук О. М., Собко О. І. Комп'ютерно-ігровий комплекс в ДНЗ. URL: https://informatika-ikt.at.ua/_fr/2/0570807.pdf (дата звернення: 30.04.2023).
23. Шкуренко О. В. Застосування технології веб-квестів при підготовці вчителя початкової школи. 2021. С. 381-384. URL: https://elibrary.kubg.edu.ua/id/eprint/36459/1/O_Shkurenko_ZTVKPPWPK_2021.pdf (дата звернення: 30.04.2025).
24. Веремієнко В. Можливості застосування WEB-додатків в роботі вчителя початкових класів. Берегово, 2024. URL: https://dspace.kmf.uz.ua/jspui/handle/123456789/3908 (дата звернення: 30.04.2025).
25. Сенченко О. І. Основні переваги сервісів Веб 2.0 у сучасному навчальному процесі. Харків, 2024. URL: https://repo.btu.kharkov.ua/bitstream/123456789/58870/1/conf-18-10-24-materialy-130.pdf (дата звернення: 30.04.2025).
26. Кулімова Ю. Використання веб-технологій у процесі психолого-педагогічної підготовки майбутніх учителів початкової школи. Відкрите освітнє е-середовище сучасного університету. 2020. № 8. С. 34-41. URL: https://openedu.kubg.edu.ua/journal/index.php/openedu/article/view/292 (дата звернення: 30.04.2025).

27. Білик Т. А. Використання електронних навчальних посібників в освітній діяльності з дошкільниками. 2023. С. 39-45. URL: https://lib.iitta.gov.ua/id/eprint/739274/1/Zbirnyk-23.11.2023.pdf (дата звернення: 30.04.2025).
28. Лущинська О. Перспективи використання інтернет-технологій у навчально-виховному процесі початкової школи. Науковий вісник Мелітопольського державного педагогічного університету. Серія: Педагогіка. 2017. № 18. С. 257-266. URL: https://magazine.mdpu.org.ua/index.php/nv/article/view/1919 (дата звернення: 30.04.2025).
29. Смолянко Ю., Валентьєва Т. Технологія розроблення електронного навчального посібника з дисципліни «Педагогіка дошкільна». NewInception. 2022. № 1-2 (7-8). С. 70-77. URL: https://newinception.com.ua/index.php/newinception/article/view/66 (дата звернення: 30.04.2025).
30. Кулалаєва Н. В., Романова Г. М. Веб-квести у підвищенні кваліфікації педагогічних працівників закладів професійної (професійно-технічної) освіти. 2018. С. 256. URL: http://eprints.zu.edu.ua/27789/1/%D0%97%D0%B1%D0%B8%D1%80%D0%BD%D0%B8%D0%BA%20%D1%81%D0%B5%D0%BC%D0%B8%D0%BD%D0%B0%D1%80%D1%83.pdf#page=256 (дата звернення: 30.04.2025).
31. Wirfs-Brock A., Eich B. JavaScript: the first 20 years. Proceedings of the ACM on Programming Languages. 2020. Vol. 4 (HOPL). P. 1-189. URL: https://dl.acm.org/doi/doi/pdf/10.1145/3386327 (дата звернення: 30.04.2025).
32. Пахомов І. В. Інформаційний супровід підвищення кваліфіекації майстрів виробничого навчання закладів професійної (професійно-технічної) освіти. 2023. С. 35-38. URL: https://lib.iitta.gov.ua/id/eprint/735994/ (дата звернення: 30.04.2025).
33. Карапузова І. В., Бурсова С. С. Інформатизація освітнього середовища закладу дошкільної освіти. Полтава, 2022. URL:

http://dspace.pnpu.edu.ua/handle/123456789/21236 (дата звернення: 30.04.2025).

34. Дячук А. О. Інформаційна система забезпечення навчально-виховної діяльності катедри. 2024. URL: https://ela.kpi.ua/server/api/core/bitstreams/511a0390-b052-4650-b63f-e4763922e002/content (дата звернення: 30.04.2025)

ДОДАТКИ

Додаток А

```
#Структура бази даних
from . import db
from flask_login import UserMixin
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(150), nullable=False)
    email = db.Column(db.String(150), unique=True, nullable=False)
    password_hash = db.Column(db.String(256), nullable=False)
class Child(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    full_name = db.Column(db.String(150), nullable=False)
    birth_date = db.Column(db.Date, nullable=False)
    group_name = db.Column(db.String(100), nullable=False)
    parent_name = db.Column(db.String(150), nullable=False)
    parent_phone = db.Column(db.String(100), nullable=False)

class Attendance(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    child_id = db.Column(db.Integer, db.ForeignKey('child.id'), nullable=False)
    date = db.Column(db.Date, nullable=False)
    time_in = db.Column(db.Time, nullable=True)
    time_out = db.Column(db.Time, nullable=True)

    child = db.relationship('Child', backref=db.backref('attendances', lazy=True))

class Schedule(db.Model):
```

```
    id = db.Column(db.Integer, primary_key=True)
    day_of_week = db.Column(db.String(20), nullable=False)
    time = db.Column(db.Time, nullable=False)
    activity = db.Column(db.String(150), nullable=False)

class Inventory(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(150), nullable=False)
    condition = db.Column(db.String(100), nullable=False)
    quantity = db.Column(db.Integer, nullable=False)

class LessonPlan(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(150), nullable=False)
    date = db.Column(db.Date, nullable=False)
    description = db.Column(db.Text, nullable=False)
    file_path = db.Column(db.Text(200), nullable=True)
```

Додаток Б

```
#Ініціалізація додатку
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager

db = SQLAlchemy()
login_manager = LoginManager()

def create_app():
    app = Flask(__name__)
    app.config['SECRET_KEY'] = 'your_secret_key_here'
    app.config['SQLALCHEMY_DATABASE_URI'] =
'sqlite:///../instance/dnz.sqlite3'
    app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

    db.init_app(app)
    login_manager.init_app(app)
    login_manager.login_view = 'login'

    from .routes import main
    app.register_blueprint(main)

    return app

from .models import User

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))
```

Додаток В

```
#Ключові маршрути веб-додатку
import os
from flask import Blueprint, render_template, redirect, url_for, flash, request, current_app
from flask_login import login_user, logout_user, login_required, current_user
from werkzeug.security import generate_password_hash, check_password_hash
from werkzeug.utils import secure_filename
from .models import User
from . import db

main = Blueprint('main', __name__)

@main.route('/')
def home():
    return redirect(url_for('main.login'))

@main.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        user = User.query.filter_by(email=email).first()
        if user and check_password_hash(user.password_hash, password):
            login_user(user)
            flash('Вхід успішний!', 'success')
            return redirect(url_for('main.dashboard'))
        else:
```

```
            flash('Неправильний email або пароль', 'danger')
    return render_template('login.html')

@main.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        password = request.form['password']

        existing_user = User.query.filter_by(email=email).first()
        if existing_user:
            flash('Користувач з такою поштою вже існує.', 'danger')
            return redirect(url_for('main.register'))

        new_user = User(
            name=name,
            email=email,
            password_hash=generate_password_hash(password)
        )
        db.session.add(new_user)
        db.session.commit()

        flash('Реєстрація успішна! Тепер увійдіть.', 'success')
        return redirect(url_for('main.login'))

    return render_template('register.html')

@main.route('/dashboard')
@login_required
```

```
def dashboard():
    return render_template('dashboard.html')

@main.route('/children')
@login_required
def children():
    from .models import Child
    children = Child.query.all()
    return render_template('children.html', children=children)

from datetime import datetime

@main.route('/delete_child/<int:id>', methods=['POST'])
@login_required
def delete_child(id):
    from .models import Child
    child = Child.query.get(id)

    if child:
        db.session.delete(child)
        db.session.commit()
        flash('Дитину успішно видалено!', 'success')
    else:
        flash('Дитина не знайдена!', 'danger')

    return redirect(url_for('main.children'))

@main.route('/children/sort_name')
@login_required
def sort_children_by_name():
```

```
    from .models import Child
    children = Child.query.order_by(Child.full_name.asc()).all()
    return render_template('children.html', children=children)

@main.route('/add_child', methods=['GET', 'POST'])
@login_required
def add_child():
    from .models import Child
    if request.method == 'POST':
        full_name = request.form['full_name']
        birth_date_str = request.form['birth_date']
        group_name = request.form['group_name']
        parent_name = request.form['parent_name']
        parent_phone = request.form['parent_phone']

        # Перетворення строки у об'єкт дати
        birth_date = datetime.strptime(birth_date_str, '%Y-%m-%d').date()

        new_child = Child(
            full_name=full_name,
            birth_date=birth_date,
            group_name=group_name,
            parent_name=parent_name,
            parent_phone=parent_phone
        )
        db.session.add(new_child)
        db.session.commit()

        flash('Дитину додано успішно!', 'success')
        return redirect(url_for('main.children'))
```

```
return render_template('add_child.html')
```

Додаток Г

Шаблони веб-інтерфейсу

#Базовий шаблон

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>{% block title %}  Застосунок для вихователя ДНЗ№21
{% endblock %}</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
    <div class="container">
      <a  class="navbar-brand"  href="{{  url_for('main.dashboard')  }}">
Застосунок для вихователя ДНЗ№21
</a>
      <button  class="navbar-toggler"  type="button"  data-bs-
toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          {% if current_user.is_authenticated %}
          <li class="nav-item">
            <a  class="nav-link"  href="{{  url_for('main.children')
}}">Діти</a>
```

```
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('main.attendance')
}}">Відвідування</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('main.schedule')
}}">Розклад</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('main.inventory')
}}">Інвентар</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('main.lesson_plans')
}}">Плани</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('main.reports')
}}">Звітність</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('main.logout')
}}">Вихід</a>
                    </li>
                    {% else %}
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('main.login')
}}">Вхід</a>
                    </li>
```

```
                {% endif %}
            </ul>
        </div>
    </div>
</nav>


<div class="container">
    {% with messages = get_flashed_messages() %}
      {% if messages %}
        {% for message in messages %}
          <div class="alert alert-info">{{ message }}</div>
        {% endfor %}
      {% endif %}
    {% endwith %}

    {% block content %}{% endblock %}
</div>
</body>
</html>4
```

Додаток Д

```
#Відвідування
{% extends "base.html" %}

{% block title %}Журнал відвідування{% endblock %}

{% block content %}
<h2>Журнал відвідування</h2>
<div class="d-flex justify-content-between align-items-center mb-3">
    <a href="{{ url_for('main.children') }}" class="btn btn-secondary">До списку дітей</a>
                        <button class="btn btn-secondary" onclick="printAttendance()">Роздрукувати</button>
    <form action="{{ url_for('main.clear_attendance') }}" method="POST" style="display:inline;">
                <button type="submit" class="btn btn-danger" onclick="return confirm('Ви впевнені, що хочете очистити журнал відвідування?')">Очистити журнал</button>
    </form>
</div>
<table class="table table-bordered" id="attendanceTable">
    <thead>
        <tr>
            <th>Ім'я дитини</th>
            <th>Дата</th>
            <th>Час приходу</th>
            <th>Час виходу</th>
        </tr>
    </thead>
```

```
            <tbody>
                {% for record in records %}
                <tr>
                    <td>{{ record.child.full_name }}</td>
                    <td>{{ record.date }}</td>
                     <td>{{ record.time_in.strftime('%H:%M:%S') if record.time_in else "Не зазначено" }}</td>
                      <td>{{ record.time_out.strftime('%H:%M:%S') if record.time_out else "Не зазначено" }}</td>
                </tr>
                {% endfor %}
            </tbody>
        </table>

        <h2>Відмітити відвідування</h2>
        <ul class="list-group">
            {% for child in children %}

             <li class="list-group-item d-flex justify-content-between align-items-center">
                {{ child.full_name }}
                  <a href="{{ url_for('main.mark_attendance', child_id=child.id) }}" class="btn btn-sm btn-primary">Відмітити</a>
            </li>
            {% endfor %}
        </ul>
        <script>
            function printAttendance() {
                console.log("Функція printAttendance викликана");
                const tableBody = document.querySelector('#attendanceTable tbody');
```

```
if (!tableBody) {
    console.log("Тіло таблиці не знайдено!");
    return;
}

// Копируем содержимое tbody
const clonedBody = tableBody.cloneNode(true);

const newWindow = window.open('', '', 'width=800, height=600');

newWindow.document.write(`
    <html>
        <head>
            <title>Роздрукувати журнал відвідування</title>
                                                    <link       rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css">
        </head>
        <body>
            <h2 class="mb-3">Журнал відвідування</h2>
            <table class="table table-bordered">
                <thead>
                    <tr>
                        <th>Ім'я дитини</th>
                        <th>Дата</th>
                        <th>Час приходу</th>
                        <th>Час виходу</th>
                    </tr>
                </thead>
                ${clonedBody.outerHTML}
```

```
                </table>
            </body>
        </html>
    `);

    newWindow.document.close();
    newWindow.print();
    newWindow.close();
}

function clearAttendance() {
    fetch("{{ url_for('main.clear_attendance') }}", {
        method: "POST"
    }).then(response => {
        if (response.ok) {
            location.reload();
        } else {
            alert("Помилка очищення журналу!");
        }
    });
}
</script>
{% endblock %}
```

Додаток Е. Форми

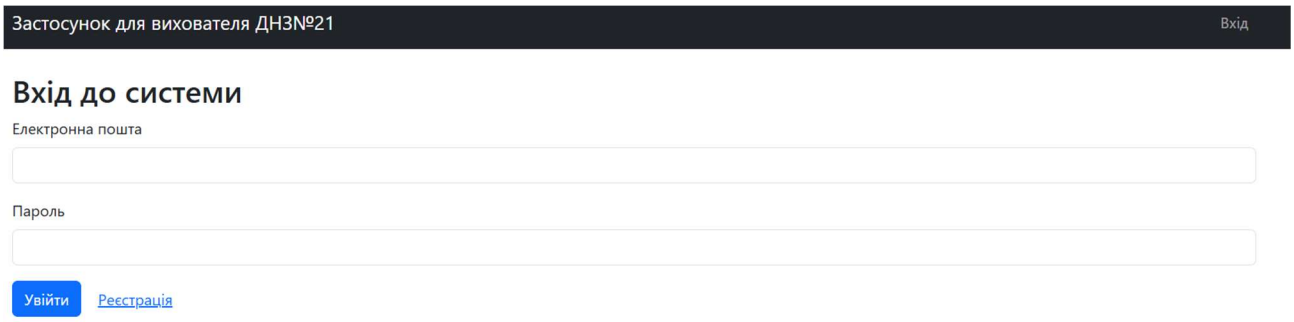

Рисунок Е.1 – «Вхід до системи»

Рисунок Е.2 – «Реєстрація»

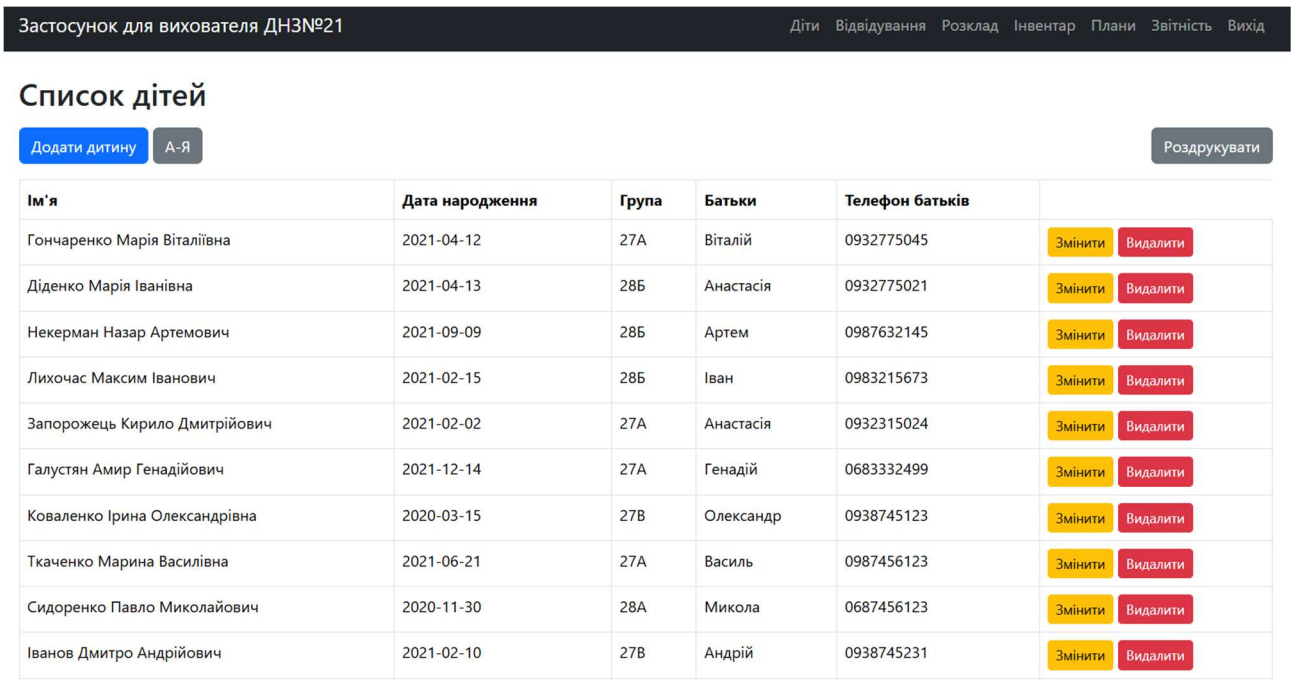

Рисунок Е.3 – «Список дітей»

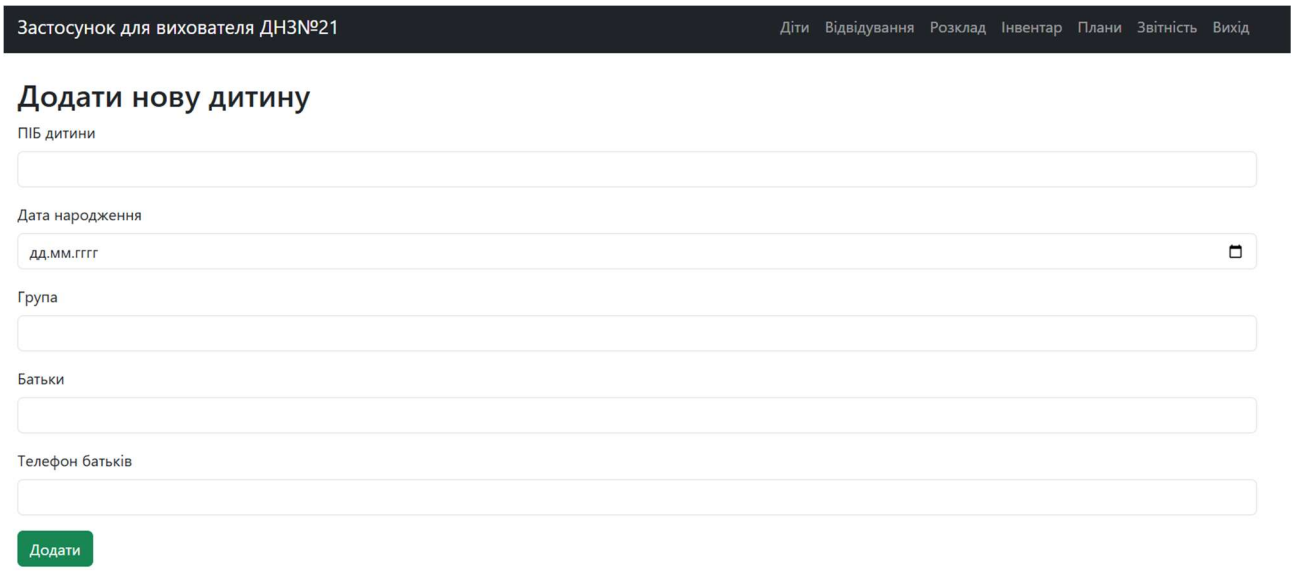

Рисунок Е.4 – «Додавання нової дитини»

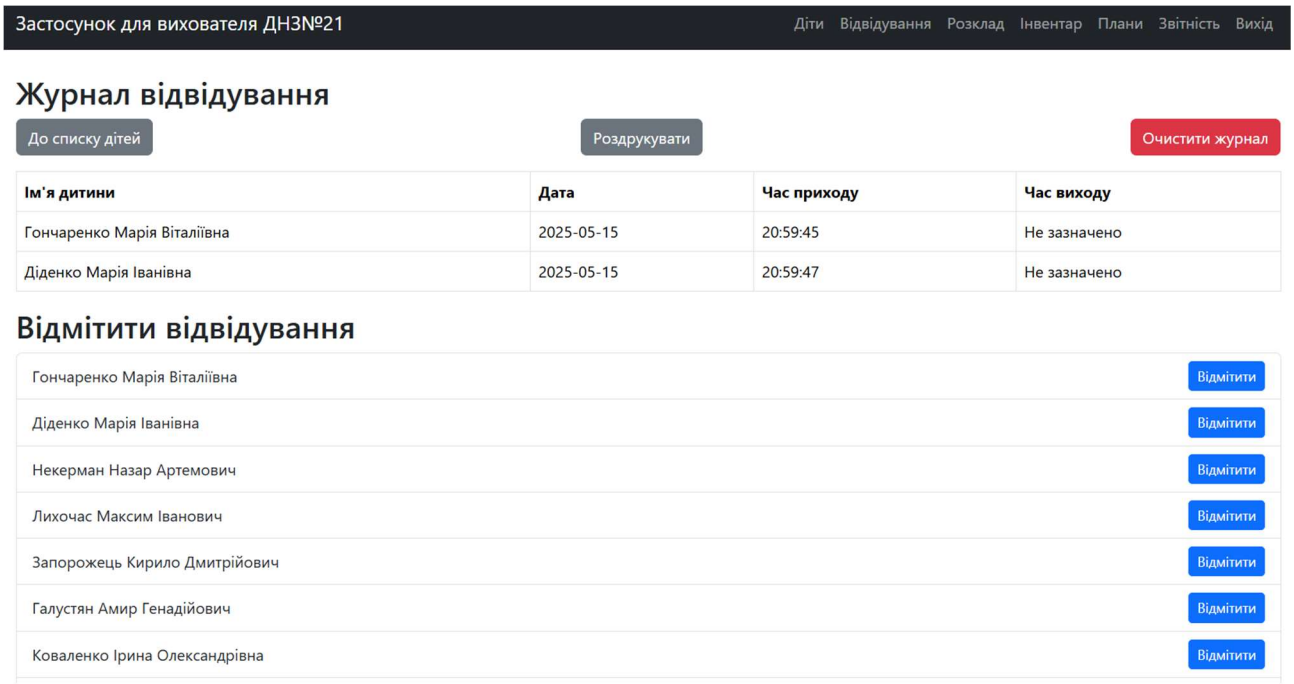

Рисунок Е.5 – «Журнал відвідування»

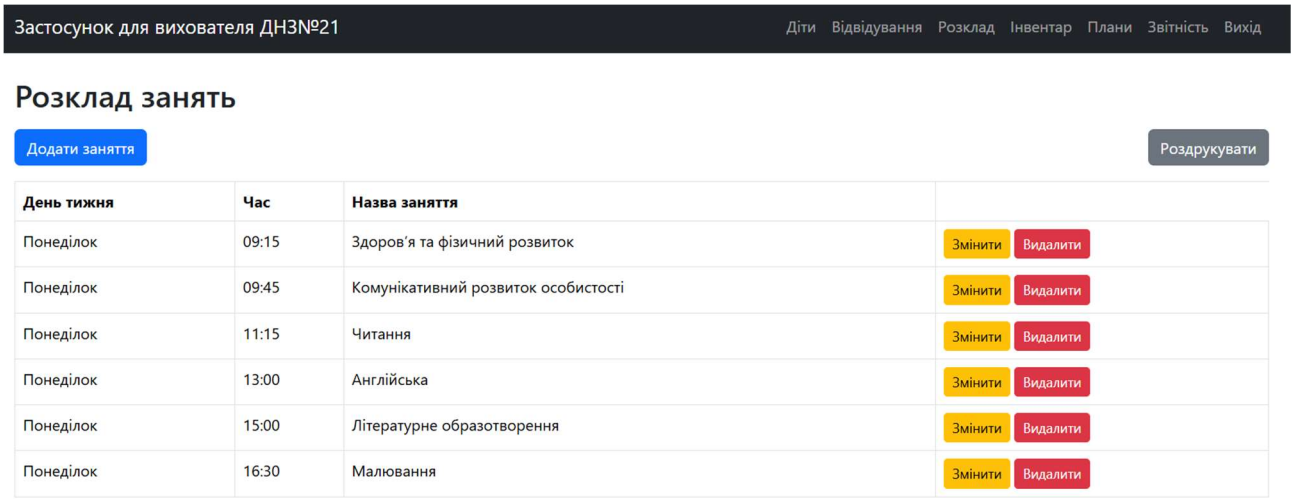

Рисунок Е.6 – «Розклад занять»

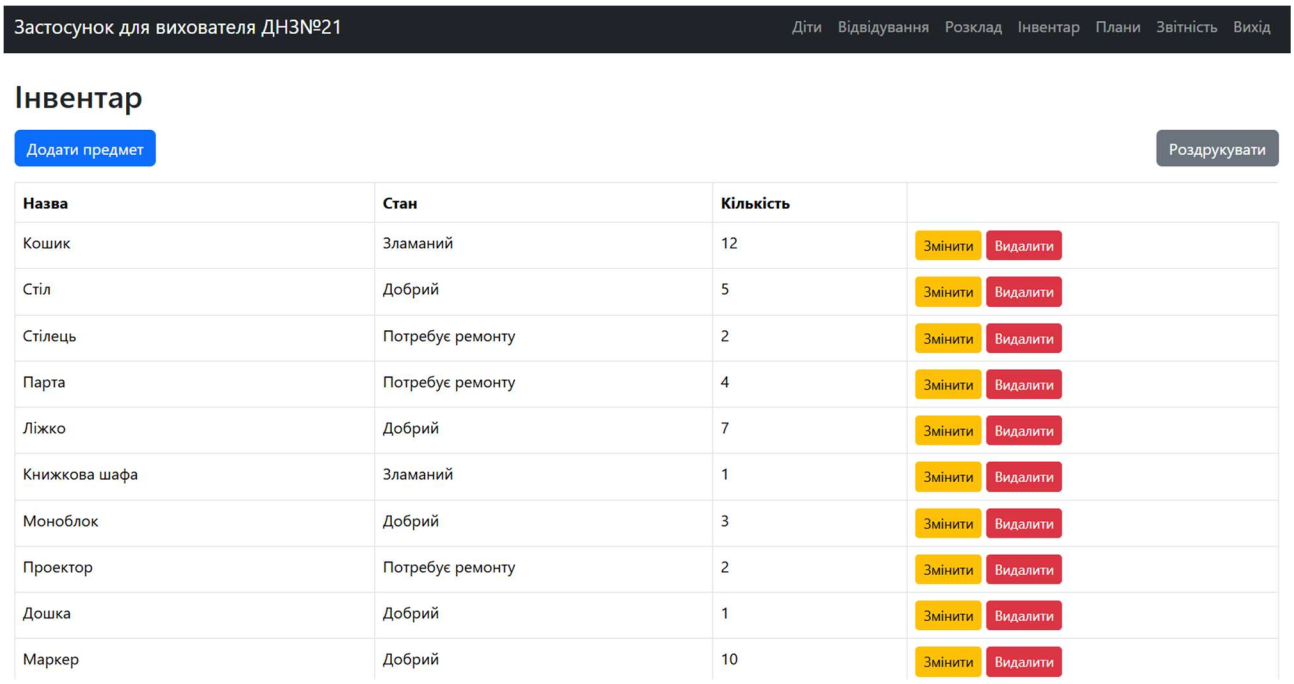

Рисунок Е.7 – «Інвентар»

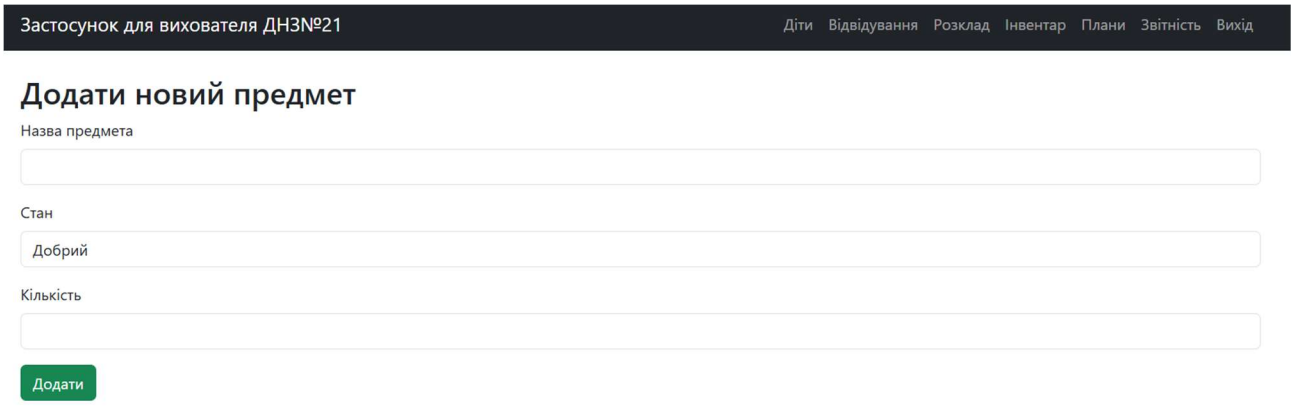

Рисунок Е.8 – «Додавання предмету»

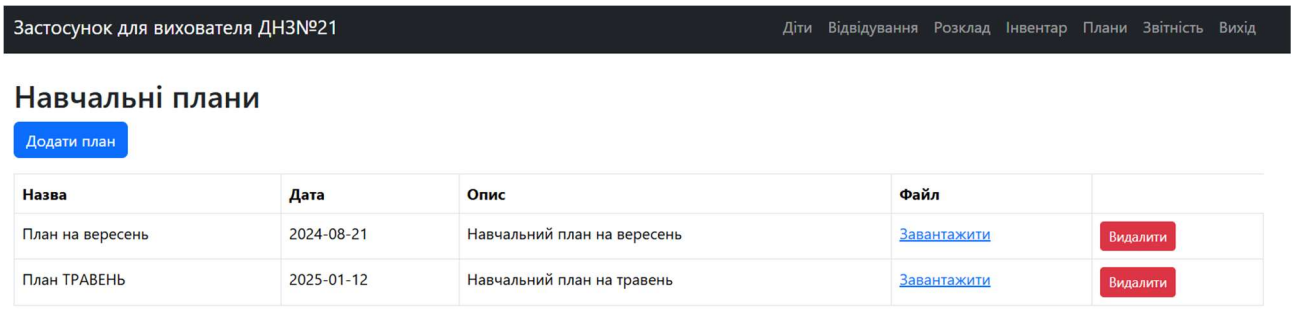

Рисунок Е.9 – «Навчальні плани»

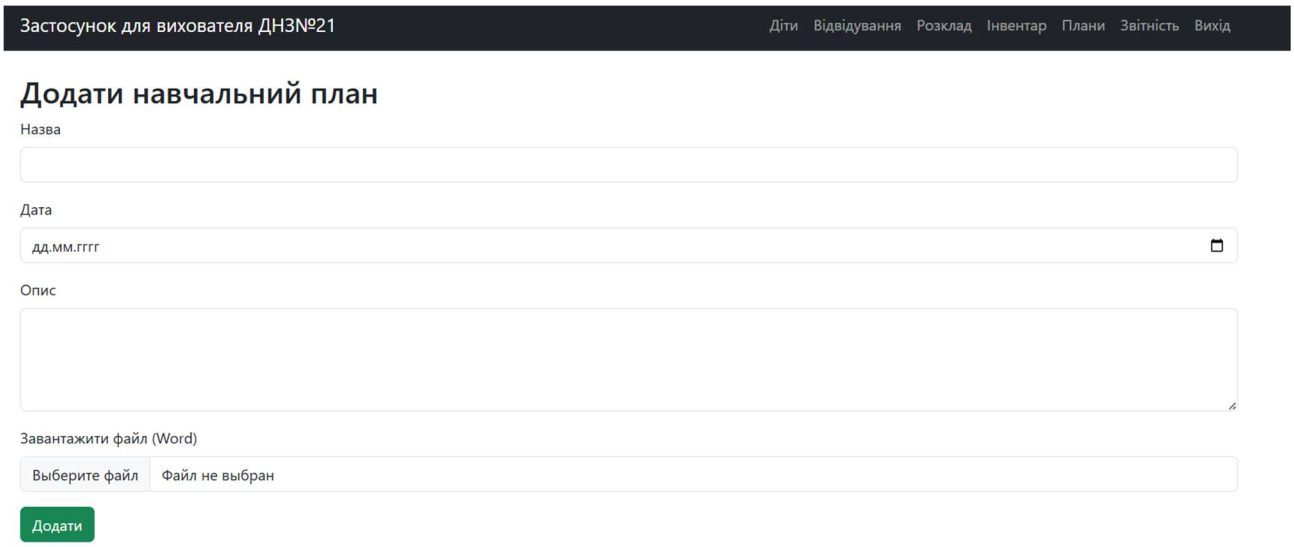

Рисунок Е.10 – «Додавання плану»

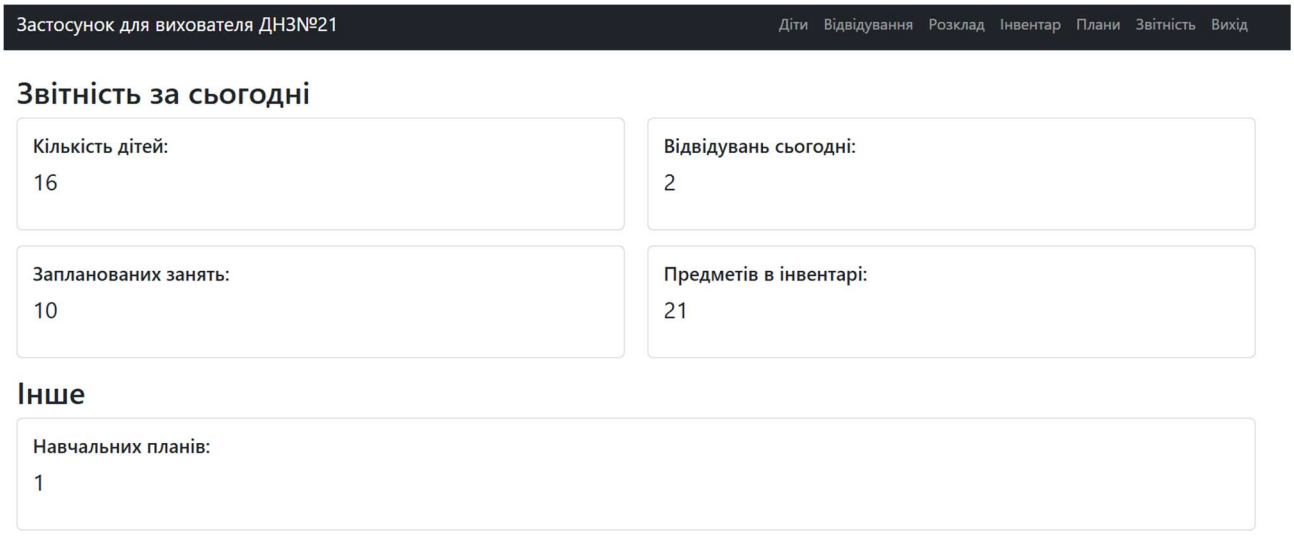

Рисунок Е.11 – «Загальна звітність»

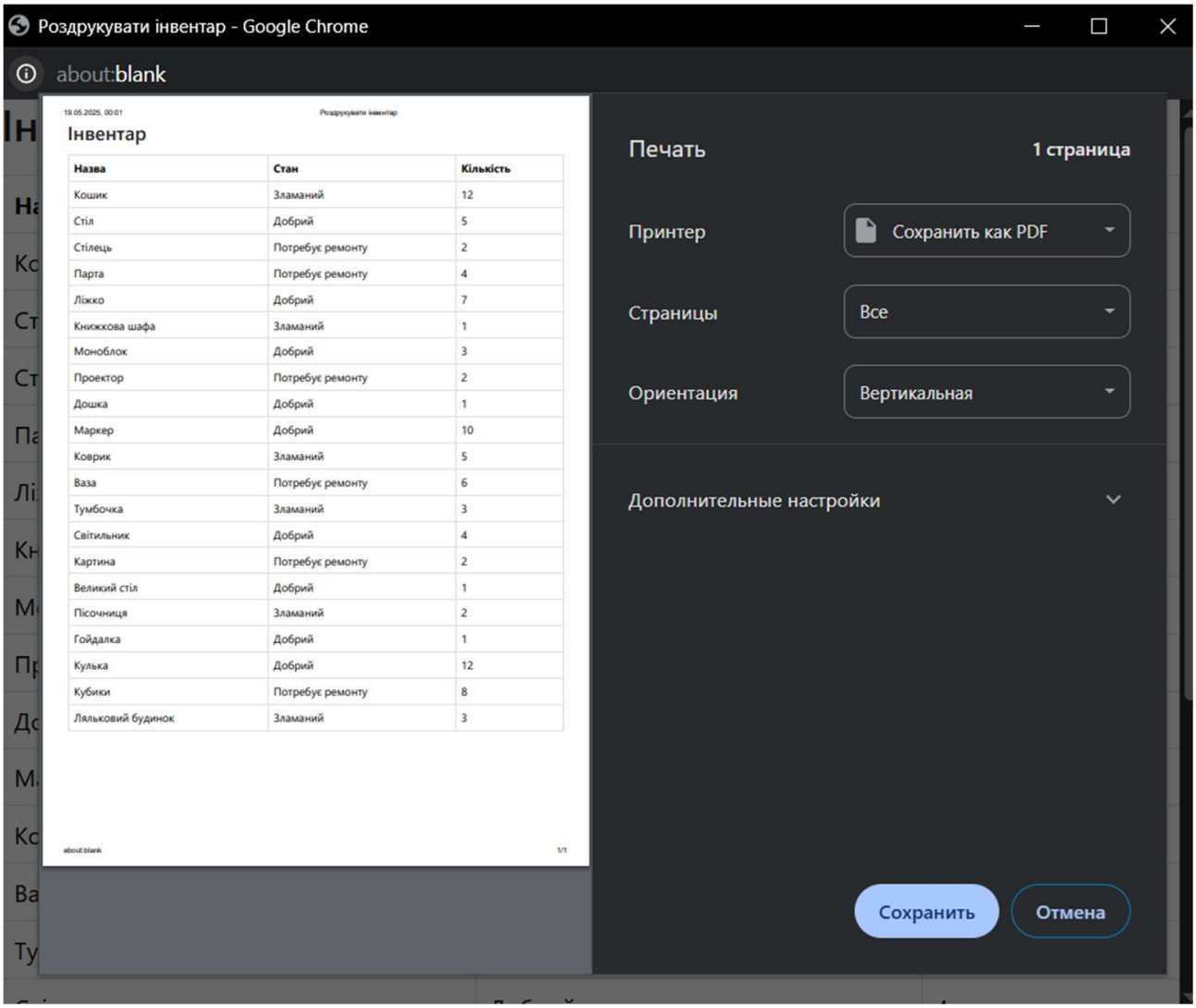

Рисунок Е.12 – «Приклад звітності»

Додаток З. Організцаційно-функціональна схема

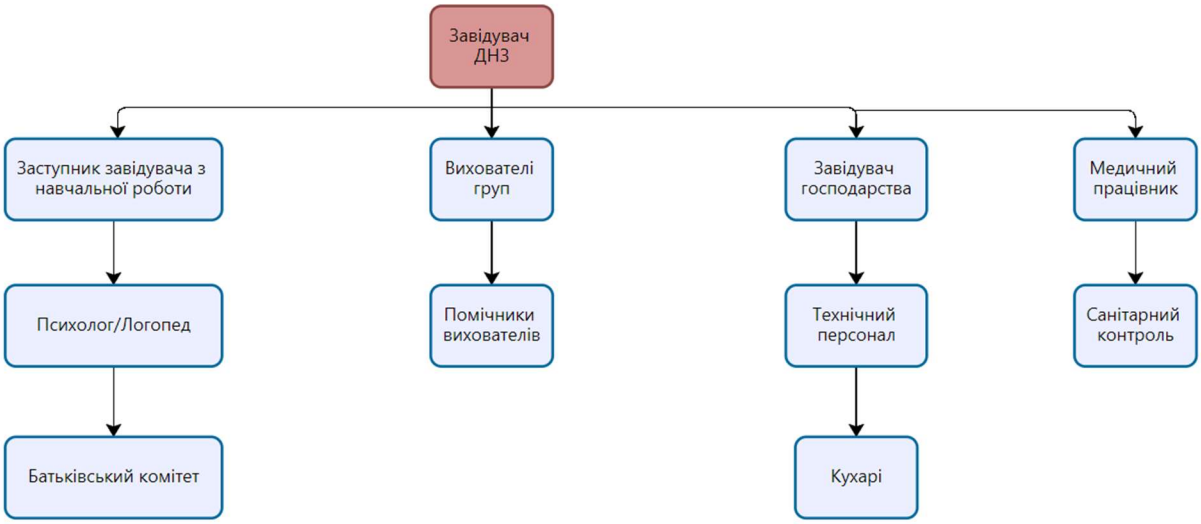

Рисунок З.1 – «Організцаційно-функціональна схема ДНЗ»

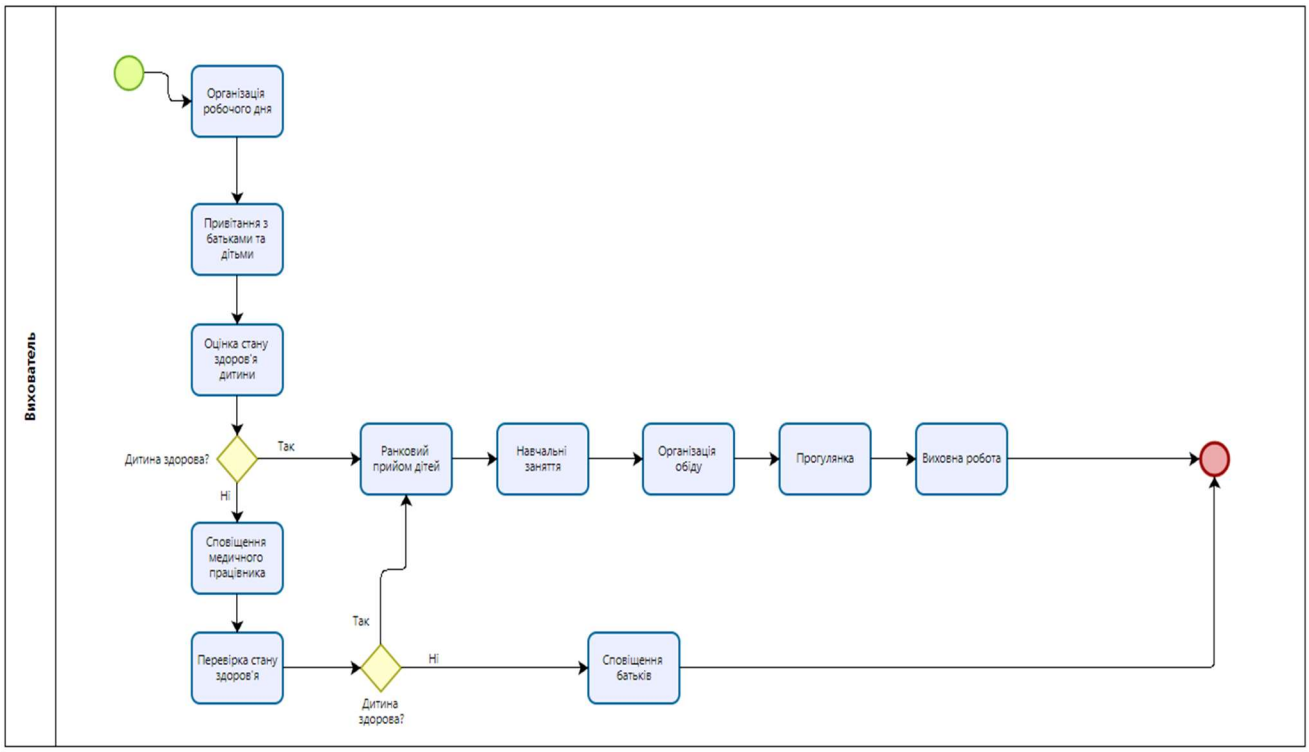

Рисунок З.2 – «Бізнес-процеси роботи вихователя»

Додаток П. Функціональна схема

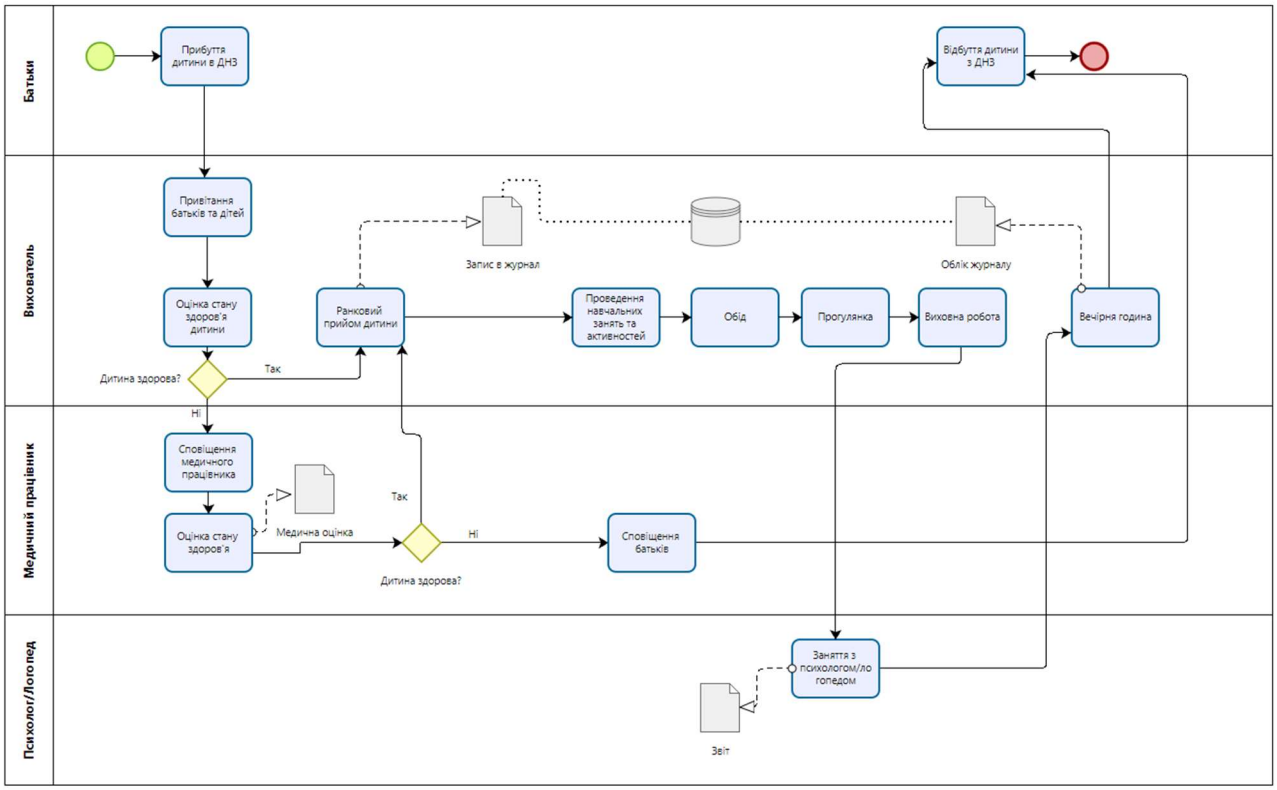

Рисунок П.1 – «Функціональна схема роботи ДНЗ»

Додаток Ж. Схеми бази даних

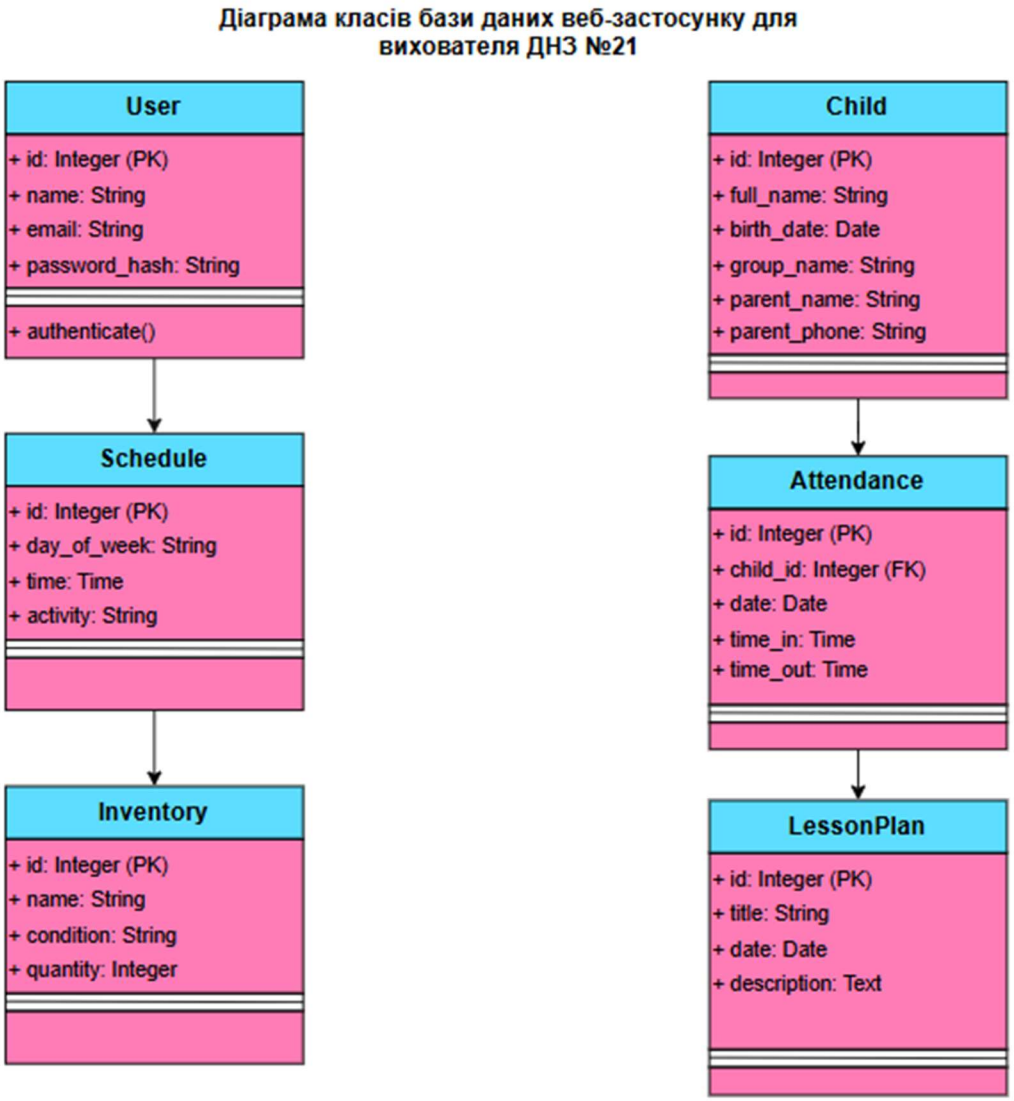

Рисунок Ж.1 — «Діаграма класів бази даних веб-застосунку для вихователя ДНЗ №21»

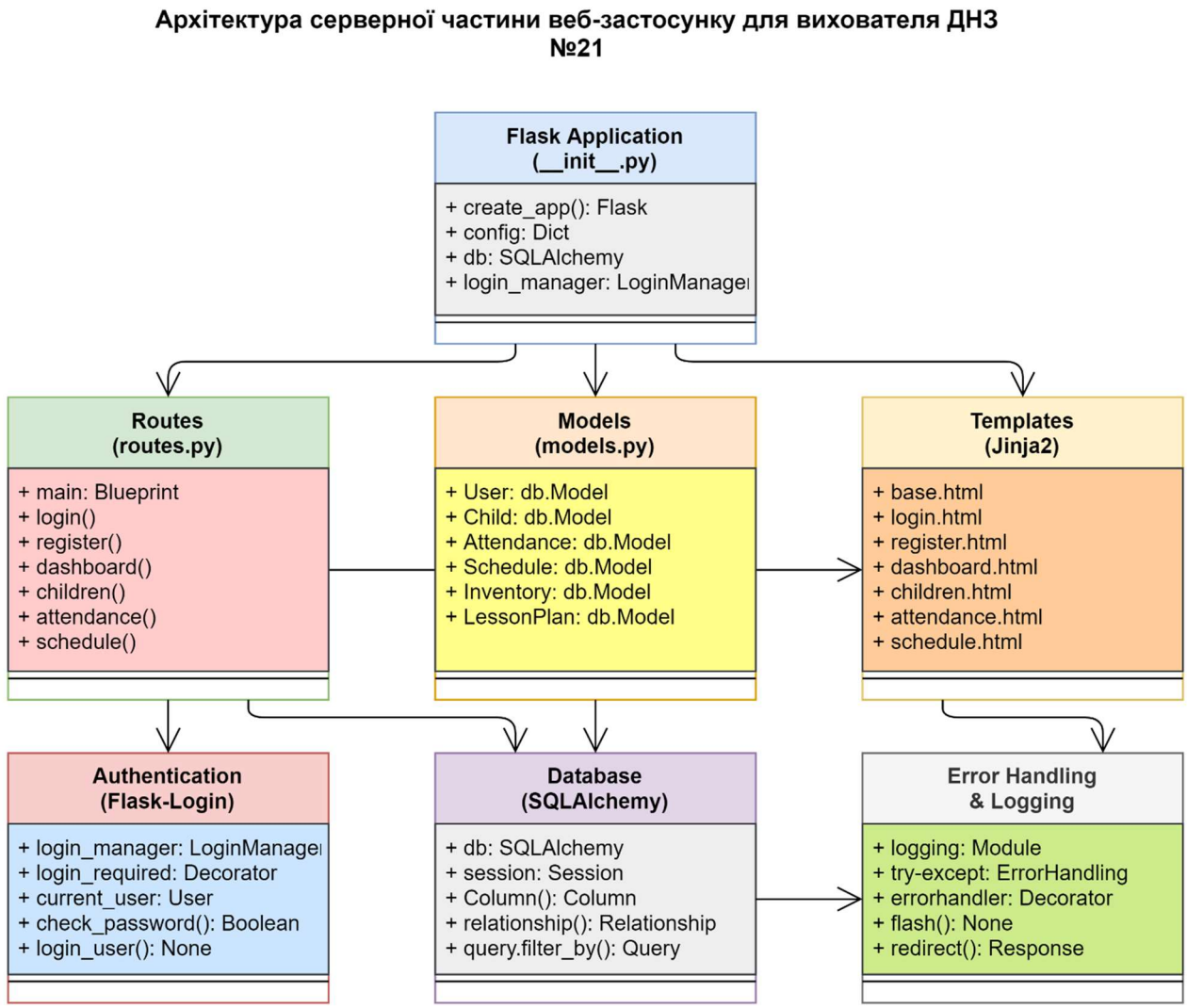

Рисунок Ж.2 — «Архітектура серверної частини веб-застосунку для вихователя ДНЗ №21»

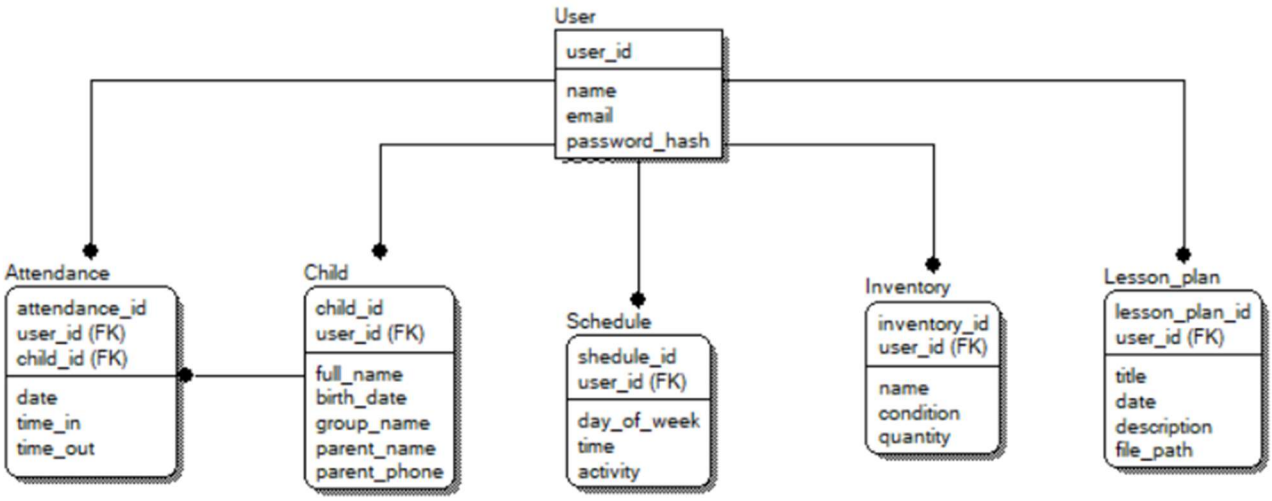

Рисунок Ж.3 — «Логічна модель бази даних»