МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет економіки, підприємництва та інформаційних технологій
Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2026 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ МОНІТОРИНГУ, КОМПЛЕКСНОЇ ОЦІНКИ ТА ПІДТРИМКИ РОЗВИТКУ ДИТИНИ

Виконав - студент 4 курсу, групи 41-кн
освітньої програми «Комп'ютерні науки»
спеціальності F3 Комп'ютерні науки
Сиваченко Ліна Олегівна

Науковий керівник - д.т.н., професор
Кулаков Павло Ігорович

Умань - 2026

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ……..... 7
ВСТУП…… 8
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ…… 10
1.1. Об’єкт та предмет дослідження…… 10
1.2. Порівняльний аналіз аналогів розробки ……. 17
1.3. Постановка задачі..... 20
РОЗДІЛ 2. РОЗРОБКА МЕТОДУ, МОДЕЛІ ТА СТРУКТУРИ ВЕБ-ДОДАТКУ …….. 22
2.1. Розробка структури веб-додатку ……..…….. 22
2.2. Аналіз варіантів використання веб-додатку …….. 26
2.3. Розробка методу і моделі роботи веб-додатку..... 28
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ …… 32
3.1. Варіантний аналіз та обгрунтування вибору засобів реалізації …… 32
3.2. Розробка бази даних …… 35
3.3. Розробка механізмів авторизації, додавання і показу новин …… 39
ВИСНОВОК…… 45
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ…… 46
ДОДАТОК А. Лістинг програми……... 48

Вступ

У теперішній час, як логічна відповідь на виклики інформаційного перевантаження суспільства, виник та почав активно розвиватися окремий технологічний напрям - системи автоматизації та керування даними. Ці програмно-апаратні комплекси стали незамінним інструментом, який кардинально змінив принципи взаємодії людини з інформаційними потоками. На сьогоднішній день такі системи виконують комплексні завдання, що включають: глибоке структурування різнорідних масивів інформації, забезпечення безпечного та довготривалого зберігання відомостей, чітке розмежування прав доступу на основі ролевих моделей користувачів, налагодження швидкої та захищеної комунікації всередині інформаційного середовища. Функціонування будь-якого підприємства, державної установи чи громадської організації вже неможливо уявити без впровадження внутрішніх клієнтських додатків або спеціалізованих корпоративних систем (ERP, CRM, внутрішніх баз знань). Паралельно з внутрішньою автоматизацією відбувався стрімкий розвиток глобальної мережі. Сучасні веб-ресурси перетворилися на універсальне середовище, яке дає змогу не лише надійно зберігати інформаційні активи, а й ефективно транслювати та презентувати їх необмеженому колу віддалених користувачів, долаючи географічні та часові бар'єри [1, 2]. Сфера вітчизняної освіти також підпорядковується цим тенденціям. Сайти сучасних академічних інституцій, шкіл та установ дошкільного виховання перестали бути просто цифровими візитівками. Специфіка таких ресурсів полягає в тому, що інформаційне наповнення повинно мати подвійну орієнтацію. Для дорослої аудиторії (батьків, педагогів, контролюючих органів): надання офіційної документації, методологічних матеріалів, звітів та нормативно-правової бази. Для дитячої аудиторії (вихованців, учнів): інтеграція гейміфікованих елементів, яскравого візуального контенту та інтутивно зрозумілої навігації, що здатна утримати увагу дитини та стимулювати її інтерес до навчання [3].

Актуальність

Проектування та програмна реалізація веб-додатку, призначення якого полягає в оцінці поточного стану та інформаційній підтримці гармонійного розвитку дитини, як високотехнологічної веб-системи, є актуальною науково-технічною задачею. Такий ресурс має виходити за рамки звичайного інформаційного порталу, пропонуючи інструментарій для комплексної оцінки психоемоційного та інтелектуального стану дитини, а також забезпечуючи повноцінну цифрову підтримку її гармонійного розвитку.

Об'єкт дослідження - процеси створення веб-додатку для комплексної оцінки та підтримки гармонійного розвитку дитини.

Предмет дослідження - програмні засоби, необхідні для реалізації веб-додатку, орієнтованого на використання у закладах дошкільної освіти.

Мета дослідження

Метою та завданням кваліфікаційної роботи бакалавра є розробка веб-додатку для моніторингу, комплексної оцінки та підтримки розвитку дитини.

Для досягнення зазначеної мети передбачається вирішення наступних задач:

- розробити метод оцінки стану та підтримки розвитку дитини у закладах дошкільної освіти з застосуванням спеціалізованого веб-додатку;

- розробити модель функціонування веб-додатку;

- розробити програмні засоби, необхідні для реалізації веб-додатку, орієнтованого на підтримку розвитку дитини;

- розробити зручний та інтуітивний графічний інтерфейс взаємодії веб-додатку з користувачем;

- здійснити тестування розробленого веб-додатку у відповідності до встановлених завдань.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Об’єкт та предмет дослідження

Сучасний веб-додаток становить собою складну інженерну сукупність взаємопов'язаних цифрових сторінок та різноманітного мультимедійного вмісту, які об'єднані під єдиним унікальним доменним ім’ям і функціонують щонайменше на одному потужному веб-сервері. Історично відомими прикладами таких масштабних систем є загальнодоступні платформи на зразок Вікіпедії, пошукового гіганта Google чи комерційного ресурсу Amazon, які разом формують глобальний простір Internet. Водночас існують і закриті, приватні корпоративні веб-додатки, розроблені виключно для внутрішнього використання співробітниками конкретних організацій [4].

Традиційно такі інструменти створюються навколо чітко визначеної тематики або конкретної суспільної мети, будь то поширення новин, електронна комерція чи розваги. Проте сьогодні особливої ваги набуває розробка спеціалізованих веб-додатків, орієнтованих на освітню та соціальну сфери, зокрема на безперервний моніторинг, комплексну оцінку та всебічну підтримку гармонійного розвитку дитини на різних етапах її дорослішання. Завдяки інтуїтивно зрозумілій навігації та логічно побудованій системі гіперпосилань, які зазвичай беруть свій початок із головної сторінки, користувачі отримують миттєвий доступ до глибокої аналітики, медичних та педагогічних карт. Сучасні батьки, вихователі та педіатри можуть легко взаємодіяти з такими платформами за допомогою різноманітних пристроїв, включаючи настільні комп'ютери, ноутбуки, планшети та smartphones, використовуючи для цього звичайні веб-браузери [5, 6].

З моменту створення Internet в 1990 році британським фізиком ЦЕРН Тімом Бернерсом-Лі та її подальшого безкоштовного відкриття для людства у 1993 році, технології пройшли колосальну еволюцію. Старі протоколи на кшталт FTP або gopher пропонували лише примітивну структуру каталогів без

можливості динамічного відстеження інформації, а документи подавалися у вигляді звичайного неформатованого тексту.

Поява протоколу HTTP кардинально змінила ситуацію, дозволивши створювати інтерактивні веб-додатки, які сьогодні є незамінними для детальної оцінки дитячих психоемоційних та фізичних показників. У контексті підтримки дитячого розвитку такий сучасний веб-додаток дозволяє накопичувати та систематизувати щоденні спостереження, формуючи цілісну картину формування особистості [7]. Користувачі можуть бачити чітку динаміку змін, фіксувати поведінкові реакції та вчасно реагувати на будь-які відхилення від норми. Оскільки кожен веб-додаток може містити посилання на суміжні медичні чи логопедичні бази даних, межі між різними платформами стають гнучкими, забезпечуючи комплексний підхід до виховання. При цьому доступ до конфіденційної інформації про дитину, її медичних тестів та індивідуального плану навчання вимагає обов’язкової захищеної реєстрації або оформлення підписки, що є стандартною практикою для багатьох сучасних академічних, комерційних та соціальних платформ. Сама назва цього цифрового інструменту пройшла шлях від початкового написання з великої літери до загальноприйнятого терміну «веб-додаток», що офіційно зафіксовано у провідних світових посібниках зі стилістики [8].

З точки зору внутрішньої архітектури, класичні статичні веб-додатки, кодовані мовами HTML та CSS, відображають фіксований і однаковий вміст для всіх відвідувачів, подібного до друкованої брошури. Вони хоч і можуть містити аудіо чи відео, але позбавлені інтерактивності, а будь-яке оновлення інформації про дитячі досягнення вимагало б ручного редагування коду адміністратором. На противагу цьому, для якісного моніторингу та комплексної оцінки розвитку дітей критично важливим є саме динамічний веб-додаток [9, 10].

Така система автоматично адаптується під потреби конкретного користувача, створюючи сторінки в режимі реального часу за допомогою

серверного комп'ютерного коду та сучасних мов програмування, таких як PHP, Python, Ruby або Java.

Використання спеціалізованих фреймворків дозволяє легко розгортати складні аналітичні модулі, які здатні відстежувати зміну обставин у реальному часі, зберігати файли cookie браузера та виводити кастомізовані графіки розвитку дитини на основі введених батьками чи вихователями даних. Впровадження технологій JavaScript, динамічного HTML та моделі об'єктів документа (DOM) дозволяє батькам взаємодіяти з інтерактивними формами оцінювання навичок дитини, а використання сучасного інструменту WebGL допомагає візуалізувати тривимірні навчальні та діагностичні ігри в максимально інтуїтивно зрозумілій формі.

Завдяки концепції адаптивного дизайну, яка стала стандартом, такий веб-додаток гарантує бездоганне відображення діагностичних панелей як на великих моніторах, так і на екранах мобільних телефонів батьків, забезпечуючи найвищий рівень зручності та залученості в процес виховання.

Хоча багато сучасних цифрових платформ орієнтовані на монетизацію через контекстну рекламу, електронну комерцію чи модель Freemium, веб-додатки у сфері дошкільної освіти та педіатрії ставлять у пріоритет саме соціальну цінність та точність аналізу даних. Масштабні світові сервіси використовують колосальні серверні потужності та системи балансування навантаження для обслуговування мільйонів відвідувачів, і подібна надійність архітектури є надзвичайно важливою для освітніх платформ, де бази даних постійно поповнюються щоденними звітами про розвиток тисяч дітей [11].

Веб-розробка у цьому напрямку охоплює не лише написання коду, а й глибоку веб-інженерію, налаштування безпеки для захисту персональних даних дитини та інтеграцію систем управління вмістом (CMS). Це безпосередньо стосується сучасного інституту дитячого садка, який історично розвинувся наприкінці вісімнадцятого століття в Баварії та Ельзасі як простір для ігор, співу та соціальної взаємодії за методикою Фрідріха Фрьобеля.

Ефективне розв’язання актуальних завдань, що стоять перед сучасними закладами дошкільної освіти, безпосередньо вимагає спроєктувати та впровадити спеціалізований веб-додаток дитячого садочка. Такий веб-додаток є складною, формальною соціотехнічною організаційною структурою, яка цілеспрямовано створюється задля оперативного збору, надійного зберігання, глибокої аналітичної обробки та конфіденційного поширення важливих відомостей. Розглядаючи цей цифровий інструмент крізь призму соціально-технічного підходу, можна виділити чотири ключові взаємопов’язані елементи, які забезпечують життєдіяльність системи, а саме: конкретні виховні та адміністративні завдання, люди, тобто педагоги, батьки та керівництво, організаційна структура із чіткими ролями, а також безпосередньо технологічна база. Будь-який сучасний профільний веб-додаток інтегрує в собі безліч внутрішніх модулів для автоматизованої агрегації та збереження даних, які згодом трансформуються в інформаційні панелі, здатні розширювати знання персоналу про стан виховного процесу та значно полегшувати ухвалення важливих управлінських рішень менеджерами освіти. Спеціалізований веб-додаток поєднує в єдиному цифровому просторі обчислювальну техніку та людей, які щодня інтерпретують отримані показники розвитку дітей, хоча в багатьох випадках цим терміном позначають безпосередньо саме програмне забезпечення, розгорнуте на серверах установи.

Крім суто практичного застосування, розробка та вивчення таких систем виступає самостійним академічним напрямком, що досліджує мережеве комп'ютерне обладнання та алгоритми, за допомогою яких заклади здійснюють фільтрацію, створення та дистрибуцію важливого контенту. При цьому особлива увага приділяється системам із чітко окресленими межами, конкретними групами користувачів, обчислювальними мережами та каналами комунікації. У багатьох сучасних освітніх об'єднаннях підрозділи, що відповідають за працездатність цих платформ, офіційно називають інформаційними службами. Будь-який такий інноваційний веб-додаток покликаний всебічно підтримувати

повсякденні операції, полегшувати рутинне управління групами та забезпечувати наочність результатів для батьків. Даний веб-додаток базується на передових інформаційно-комунікаційних технологіях та оптимізує процеси щоденної взаємодії користувачів із цифровим середовищем, підвищуючи загальну продуктивність праці адміністрації. Ряд авторитетних науковців чітко розмежовують апаратну частину, бізнес-процеси та безпосередньо прикладне програмне забезпечення. Спеціалізований веб-додаток концентрується саме на кінцевому, корисному використанні інформаційних технологій, допомагаючи здійснювати безперервний моніторинг та контроль ефективності роботи кожного вихователя й закладу загалом.

Можна розглядати такий веб-додаток як особливий різновид робочої системи, де персонал і технічні засоби взаємодіють задля створення якісних освітніх послуг для клієнтів. У цьому контексті діяльність платформи повністю підпорядкована фіксації, передачі, накопиченню, вилученню, гнучкому маніпулюванню та візуальному відображенню даних. Створюваний веб-додаток тісно пов'язаний із потоками інформації, з одного боку, та реальними діями вихователів, з іншого. Він стає сучасним інструментом комунікації та формою цифрової соціальної пам'яті закладу, функціонуючи як напівформальна мова для підтримки педагогічних рішень, і виступає важливою сферою досліджень організаційної інформатики. Розвиток, проєктування та практичне впровадження таких платформ у дошкільних закладах часто здійснюється на основі системної інженерії, зокрема класичного життєвого циклу розробки систем (SDLC). Цей процес послідовно охоплює етапи глобального планування, ретельного системного аналізу і формування технічних вимог, архітектурного проєктування, написання коду, комплексної інтеграції та тестування модулів, фінального впровадження, експлуатації та подальшого технічного обслуговування. Сучасні тенденції спрямовані на те, щоб залучати до розвитку та наповнення платформи весь колектив дитячого садка, а сам веб-додаток може створюватися як силами власних IT-спеціалістів, так і передаватися на

аутсорсинг стороннім розробникам повністю або частково. Іноді команди розробників можуть бути географічно розподілені.

Історичні класичні погляди на структурування програмного забезпечення, популярні у вісімдесятих роках минулого століття, описували ієрархічну піраміду, де внизу перебували системи обробки рутинних транзакцій, вище - інструменти моніторингу та управління, а на самій вершині - виконавчі комплекси для керівництва. Проте з появою нових технологій архітектура сучасних систем суттєво трансформувалася. Сьогодні спеціалізований веб-додаток для дитячого садка може включати елементи систем підтримки прийняття рішень, соціальних платформ для батьківської спільноти, інтелектуальних модулів оцінки навичок, систем управління процесами, сховищ аналітичних даних, інструментів планування ресурсів, експертних та пошукових систем, мультимедійних комплексів та засобів автоматизації офісної роботи. Комп'ютеризований веб-додаток використовує передові обчислювальні технології для виконання поставлених завдань, а його ключовий фундамент, відомий як обчислювальна платформа, складається з чотирьох базових компонентів: апаратного забезпечення (монітори, процесори), програмного забезпечення, реляційних баз даних та мережевих підсистем для спільного використання ресурсів. Спираючись на ці елементи, фахівці створюють IT-послуги, що забезпечують безпеку, захист від кіберзагроз та надійне керування конфіденційними даними дітей [12].

Залежно від масштабів, веб-додаток може обслуговувати як окрему вікову групу, так і велике об'єднання дошкільних установ, де кожна функціональна область має свій набір цифрових інструментів. Спеціалізовані функціональні модулі виступають опорою для загальних систем бізнес-аналітики та керівних панелей інструментів, підтримуючи конкретні напрямки діяльності, такі як бухгалтерський облік, фінанси, управління матеріально-технічним забезпеченням, маркетинг освітніх послуг та менеджмент людських ресурсів. У фінансово-бухгалтерському секторі дитячого садка адміністрація активно

застосовує веб-додаток для точного прогнозування бюджету, розрахунку батьківської плати, оптимізації витрат на харчування та проведення аудитів фінансової звітності. Крім того, в структуру платформи інтегруються модулі обробки транзакцій, системи планування ресурсів, автоматизація діловодства, засоби управління ланцюгами постачання продуктів та елементи електронної комерції для оплати додаткових гуртків. Особливу роль відіграють інтерактивні інформаційні панелі (дашборди), які забезпечують завідувачу садка швидкий і наочний доступ до звітів у реальному часі, а вбудовані експертні системи намагаються дублювати роботу досвідчених методистів і психологів, застосовуючи логічні міркування та накопичену базу знань для вирішення специфічних педагогічних ситуацій.

Сучасні географічні або просторові модифікації програмного софту дозволяють також враховувати територіальне розташування сімей вихованців. Сам процес побудови та розгортання такого комплексного цифрового інструменту для дошкільного закладу обов'язково здійснюється поетапно. На початкових стадіях розробники проводять ретельне розпізнання та детальне уточнення наявних управлінських проблем, здійснюють всебічний збір первинної інформації, формують чітку специфікацію вимог до майбутньої системи, після чого переходять безпосередньо до проєктування інтерфейсів, побудови програмного коду, фінальної реалізації веб-додатка на серверах та його регулярного експлуатаційного обслуговування й оновлення.

У сучасних реаліях дитячий садок потребує потужної цифрової підтримки, і впровадження інноваційного веб-додатка є життєво необхідним кроком для кожної освітньої установи. Такий спеціалізований веб-додаток забезпечує не просто базовий зв'язок між адміністрацією, вихователями та батьками, а виступає глобальним інструментом для фіксації щоденних успіхів малюка, проведення комплексних психологічних оцінок, моніторингу етапів соціалізації та залучення потенційних інвесторів для покращення освітнього простору, що робить його проектування та впровадження максимально актуальним і

стратегічно виправданим завданням для розвитку закладу дошкільної освти та майбутнього покоління.

1.2. Порівняльний аналіз аналогів розробки

У теперішніх реаліях, що характеризуються пануванням передових цифрових рішень та мережевих комунікацій, наявність віртуального ресурсу є не просто додатковим бонусом, а критично важливим, обов'язковим та фундаментальним фактором успішного довгострокового існування, позиціонування та повноцінної життєдіяльності будь-якої сучасної освітньої установи. Проте детальний аналіз, глибокий моніторинг та систематизація результатів пошукової видачі за відповідними тематичними запитами у всесвітньому павутинні наочно свідчать про те, що на сьогоднішній день реальна кількість дійсно якісних, функціональних, ергономічних та змістовно наповнених ресурсів є критично малою та незадовільною.

Більшість існуючих рішень не відповідають високим сучасним стандартам веброзробки та не здатні повноцінно задовольнити запити цільової аудиторії, що підтверджує наявність серйозної технологічної проблеми у цьому секторі освітніх послуг. Для детальнішого розуміння поточної ситуації на ринку доцільно детально розглянути та критично оцінити кілька конкретних прикладів існуючих вебсторінок, які належать різним дитячим виховним центрам.

Якщо звернути увагу на офіційну сторінку приватного міні-садка під назвою «Смайл» (Рис. 2.1), то при першому поверхневому ознайомленні з її інтерфейсом складається враження, що розробники реалізували доволі вдалу, яскраву та гармонійну дизайнерську концепцію, яка з перших секунд викликає позитивні емоції та формує приємне загальне враження у звичайного відвідувача.

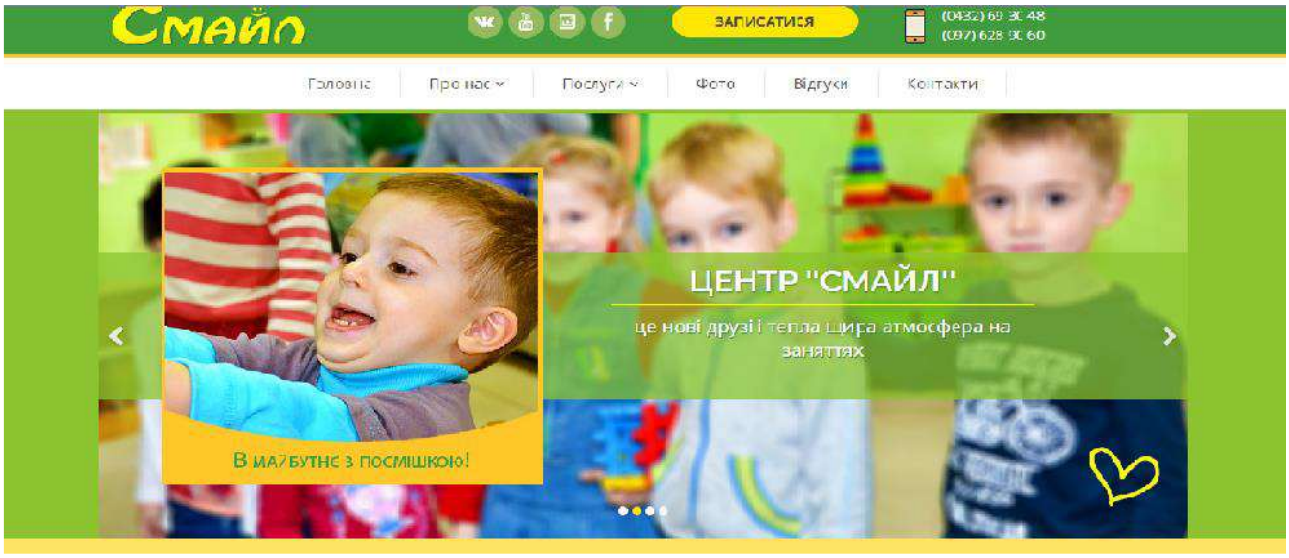


Рисунок 1.1 - Дитячий садочок "Смайл"

Однак під час глибшого аналізу внутрішнього наповнення та архітектури сайту виявляється суттєвий, вагомий та концептуальний недолік, який полягає у повній відсутності спеціалізованого інформаційного або інтерактивного розділу, орієнтованого безпосередньо на всебічне навчання, ментальний розвиток, виконання корисних вправ та виховання маленьких дітей. Схожі структурні проблеми демонструє і веб-ресурс дитячої установи «Пролісок» (Рис. 1.2), де за зовнішньою привабливістю ховається серйозний брак корисного функціоналу.



Рисунок 1.2 - Дитячий садок "Пролісок"

Крім того, у процесі дослідження ринку нерідко трапляються випадки, коли розробники припускаються грубих елементарних помилок ще на етапі проектування архітектури, вибору кодування інформації або оптимізації текстового контенту, що призводить до некоректного відображення символів або повної втрати інформативності. Наочним прикладом такої невдалої реалізації є офіційний сайт дошкільного закладу «Пролісок», де повністю відсутній будь-який унікальний, корисний чи змістовний контент, здатний зацікавити потенційного клієнта, утримати увагу батьків або принести практичну користь вихователям, за винятком сухих контактних даних, точної юридичної адреси та кількох лаконічних загальних речень про історію створення та роботу самого садка. Зважаючи на виявлені серйозні прогалини та технічні недоліки існуючих аналогів, головним стратегічним завданням даної кваліфікаційної випускної роботи бакалавра є комплексна інженерна розробка та повноцінна програмна реалізація сучасного інтерактивного вебресурсу дитячого садочка, який буде оснащений спеціальними навчальними модулями, елементами розвиваючих вправ, корисними методичними матеріалами та логічними іграми для малечі. До цього переліку належить теоретичне обґрунтування та створення унікального методу комплексної оцінки, фіксації та підтримки гармонійного розвитку дитини в умовах дошкільної установи із залученням потенціалу спеціалізованої вебсистеми, а також побудова детальної математичної та логічної моделі функціонування розроблюваної мережевої платформи.

Наступними важливими кроками є здійснення всебічного критичного аналізу інформаційного забезпечення сайту, безпосереднє проектування та написання коду для ефективних програмних засобів реалізації системи комплексної оцінки та підтримки дитячого розвитку, а також створення максимально зручного, ергономічного, інтуїтивно зрозумілого та сучасного графічного інтерфейсу користувача. Завершальним етапом роботи є проведення серії тестувань готового програмного продукту відповідно до раніше визначених критеріїв для перевірки його стабільності та відповідності всім поставленим

інженерним задачам. Інтерфейс повинен містити модуль для оперативного відображення свіжих новин установи, стрічку важливих оголошень, окремий блок із корисними гіперпосиланнями в спеціалізованому дитячому розділі, а також зручні інструменти для швидкого додавання, редагування чи видалення текстових новин та зовнішніх посилань адміністратором. Окрему увагу в архітектурі платформи приділено надійним механізмам авторизації та первинної реєстрації нових членів спільноти, розробці привабливої головної сторінки, побудові чіткого та логічного навігаційного меню для швидкого переміщення між розділами, наповненню сторінки «Про нас» актуальними відомостями про колектив та досягнення садка, а головне - інтеграції великого блоку з різноманітними розвиваючими матеріалами, інноваційними інтерактивними завданнями, психологічними тестами, логічними ребусами та повчальними комп'ютерними іграми, призначеними для ефективного стимулювання інтелектуального зростання вихованців закладу.

1.3. Постановка задачі

Метою кваліфікаційної роботи бакалавра є розробка веб-додатку для моніторингу, комплексної оцінки та підтримки розвитку дитини. Розроблений веб-додаток повинен бути зручним і практичним рішенням, яке забезпечує оцінку стану та підтримку розвитку дитини. Його особливістю є наявність інтерактивних елементів, спрямованих на когнітивний та психомоторний розвиток дітей молодшого віку.

Завданням кваліфікаційної роботи бакалавра є дослідження об’єкту та предмету розробки, особливостей веб-додатків для закладів дошкільної освіти, обгрунтування та вибір технологій, аналіз існуючих технологій створення веб-додатків, розробка програмного забезпечення веб-додатку для підтримки розвитку дитини.

Для вирішення поставлених завдань необхідно:

- розробити метод оцінки стану та підтримки розвитку дитини у закладах

дошкільної освіти з застосуванням спеціалізованого веб-додатку;

- розробити модель функціонування веб-додатку;

- розробити програмні засоби, необхідні для реалізації веб-додатку, орієнтованого на підтримку розвитку дитини;

- розробити зручний та інтуітивний графічний інтерфейс взаємодії веб-додатку з користувачем;

- здійснити тестування розробленого веб-додатку у відповідності до встановлених завдань.

РОЗДІЛ 2. РОЗРОБКА МЕТОДУ, МОДЕЛІ ТА СТРУКТУРИ ВЕБ-ДОДАТКУ

2.1. Розробка структури веб-додатку

Протягом проектування веб-додатку всі процеси, маршрутизація та обробка інформаційних потоків, реалізовані завдяки інструментарію фреймворку Spring Boot. Ця система дозволяє абстрагуватися від низькорівневої обробки HTTP-протоколів за допомогою гнучких засобів, заснованих на системі мережевих реквестів та метаданих у вигляді Java-анотацій. Усі параметри або ідентифікатори інкапсулюються в тіло HTTP-запиту або структуру URL-посилання, після чого перехоплюються мапінгами, проходять етап валідації, типізації та автоматично форматуються у готові до обробки програмні об'єкти.

Під час проектування графічного інтерфейсу користувача створено максимально чистий простір, не перевантажений зайвими декоративними елементами, строкатими палітрами кольорів чи надлишковим контентом, що розпорошує увагу. Стартова сторінка інформаційного порталу слугує візитною карткою закладу і містить лише лаконічне навігаційне меню для переходу між ключовими розділами та декілька якісних тематичних світлин, які презентують повсякденне життя та затишну атмосферу дитячого садочка (рис. 2.1).

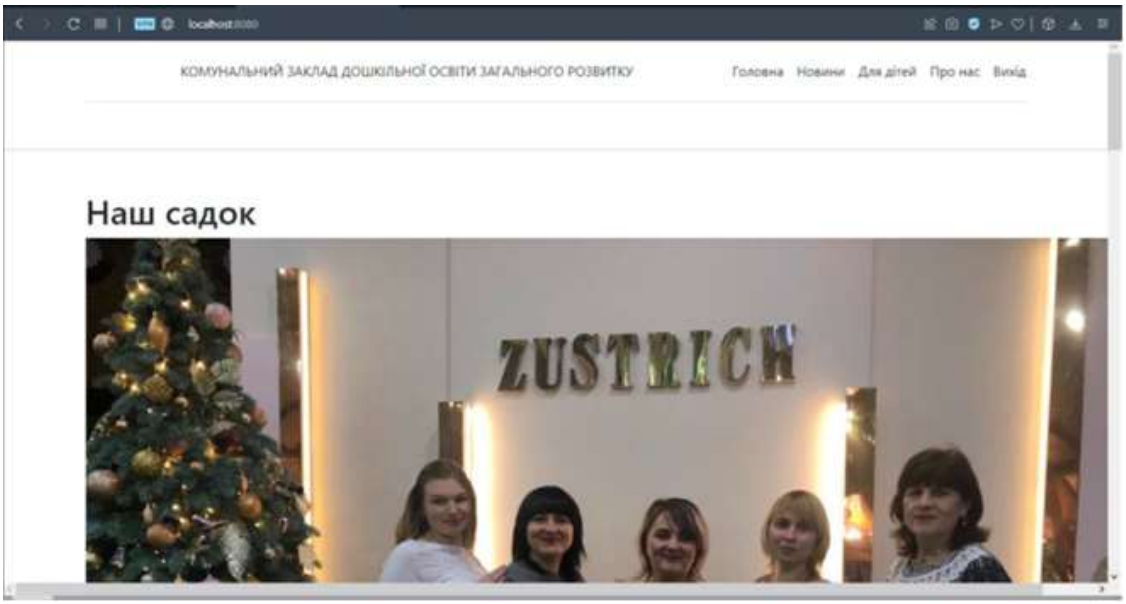


Рисунок 2.1 - Головна сторінка

Розділ новин і важливих оголошень спроектований за принципом стриманості, де кожне інформаційне повідомлення оформлене у вигляді окремого ізольованого блоку з чіткою структурою тексту та датами. При цьому інтерфейс цієї сторінки динамічно адаптується під рольову модель доступу: звичайні відвідувачі бачать лише чисту стрічку публікацій, тоді як авторизований адміністратор отримує додаткові інтерактивні елементи керування для модерації. На рисунку 2.2 зображено сторінку новин з однією тестовою новиною з боку адміністратора, оскільки гості та звичайні користувачі повністю позбавлені технічної можливості самостійно генерувати, редагувати чи публікувати будь-які записи в загальну стрічку.

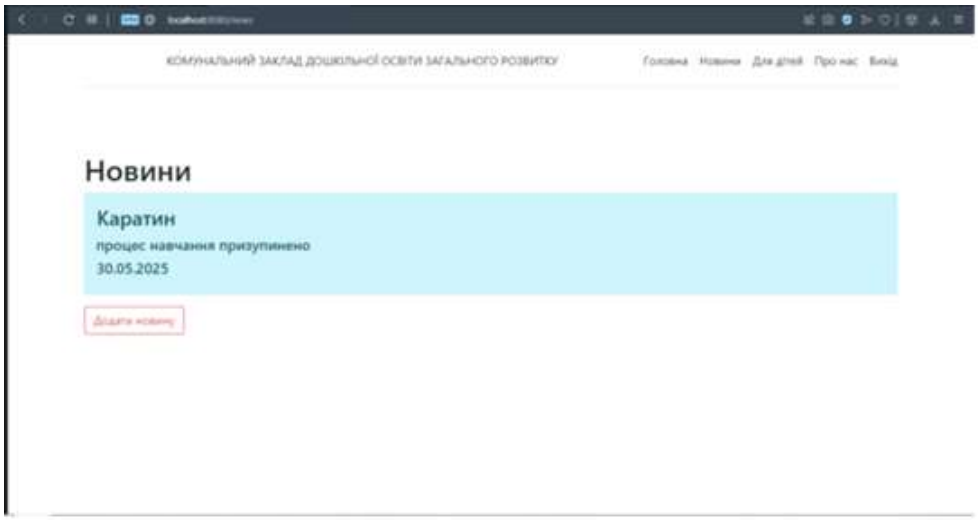


Рисунок 2.2 - Сторінка новин

Дитяча сторінка з корисними матеріалами функціонує за спорідненою логікою компонування, проте має суттєву функціональну відмінність, яка полягає в тому, що заголовки її підрозділів виступають безпосередніми інтерактивними гіперпосиланнями на зовнішні чи внутрішні ресурси, будь то навчальні інтерактивні ігри, розвиваючі завдання чи пізнавальні відеоролики на популярних світових відеохостингах (рис. 2.3.).

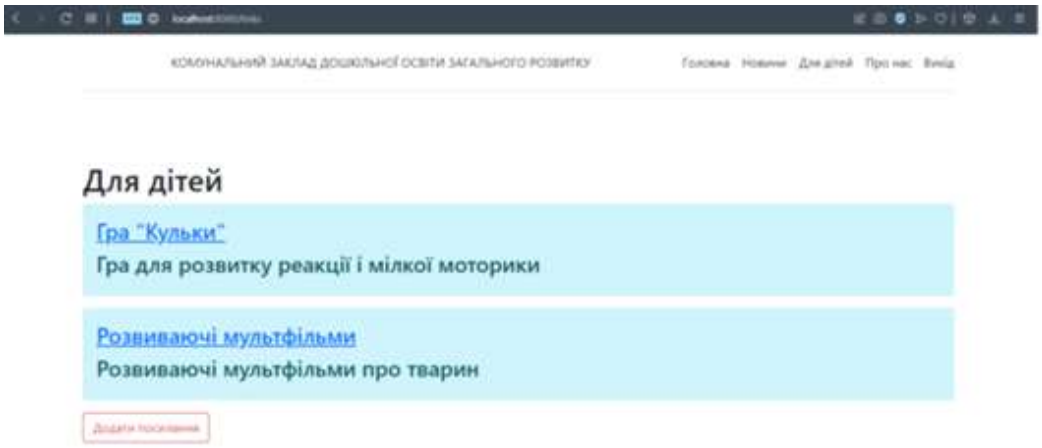


Рисунок 2.3 - Сторінка «Для дітей»

Інформаційний блок, що розповідає про історію та команду закладу, також позбавлений візуального шуму і пропонує користувачеві релевантні та юридично необхідні відомості, серед яких офіційна поштова та фізична адреса установи, контактні номери телефонів, актуальний e-mail для листування, а також повне ім’я чинного керівника дошкільного закладу (рис. 2.4.).

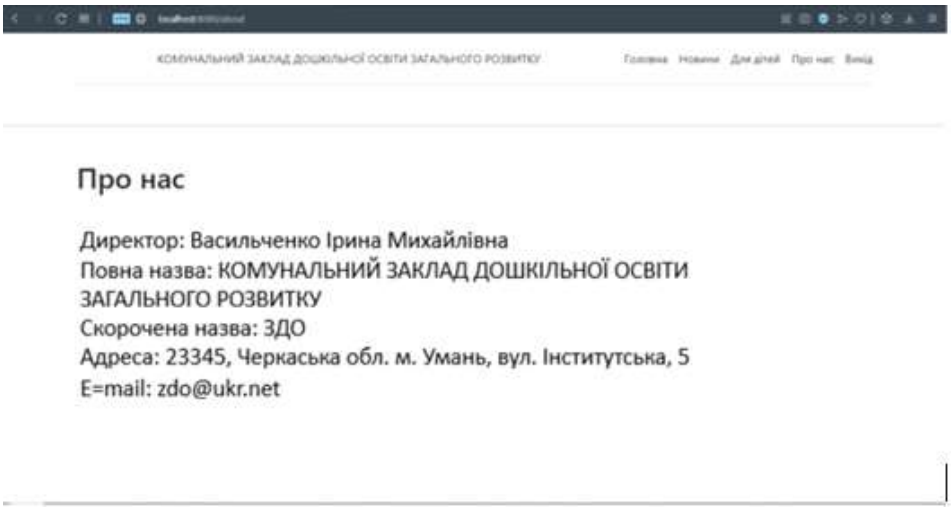


Рисунок 2.4 - Сторінка «Про нас»

Окрім публічного сегменту, веб-додаток оснащений двома спеціалізованими закритими сторінками, які надійно захищені конфігурацією безпеки Spring Security і доступні для відвідування виключно користувачам із правами суперадміністратора. Перша з цих адмін-панелей призначена для швидкого додавання та редагування свіжих текстових новин у загальну стрічку,

що відображено на рис. 2.5, де користувач може побачити структуру форми створення контенту.

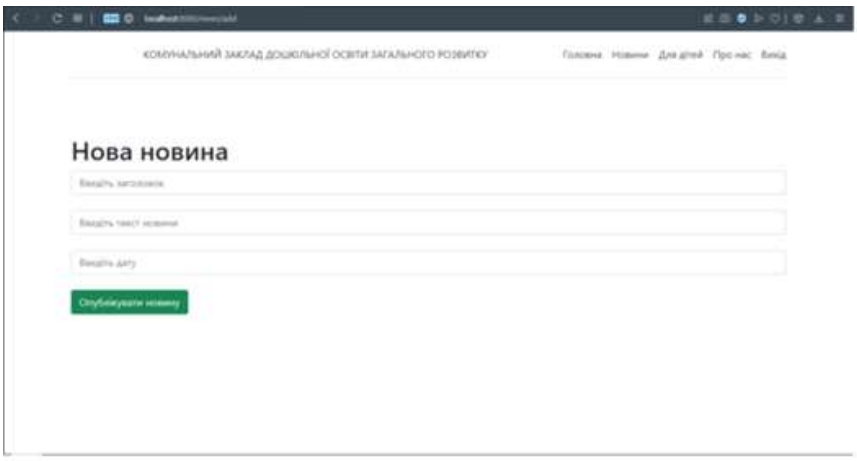


Рисунок 2.5 - Сторінка додавання новини.

Друга закрита сторінка забезпечує інтерфейс для генерації, валідації та публікації нових медіа-посилань, вигляд якої представлений на рис. 2.6.

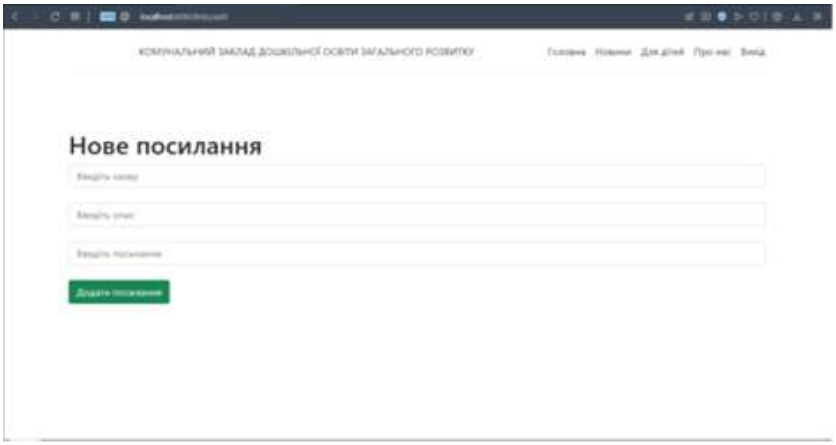


Рисунок 2.6 - Сторінка створення посилання.

Замикають архітектуру веб-додатку дві загальнодоступні форми, які відповідають за менеджмент користувачів та контроль рівнів доступу. Це форма авторизації, наведена на рис. 2.7, де раніше зареєстровані особи можуть безпечно ввести свої персональні логіни та паролі для входу в систему, а також сторінка первинної реєстрації, наведена на рис. 2.8, яка дозволяє новим членам спільноти створити свій обліковий запис шляхом внесення анкетних даних до бази.

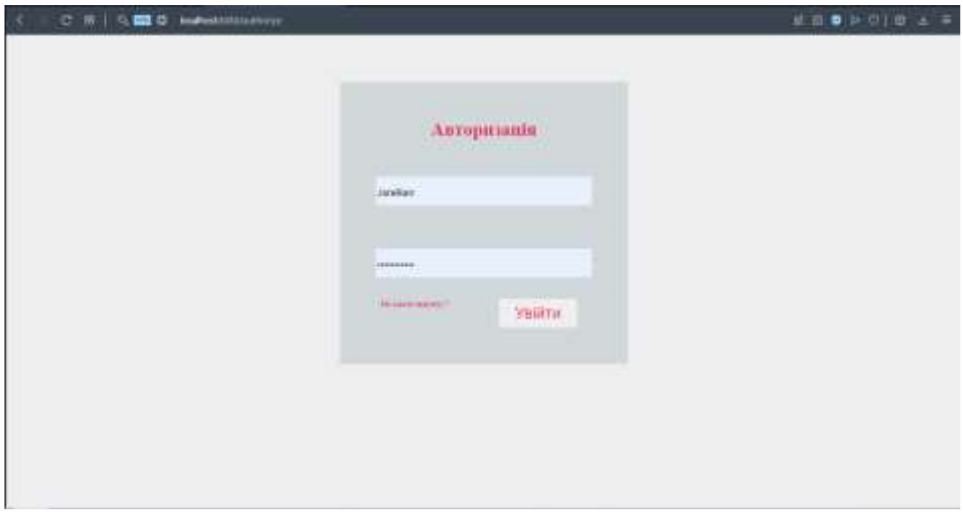


Рисунок 2.7 - Сторінка авторизації.

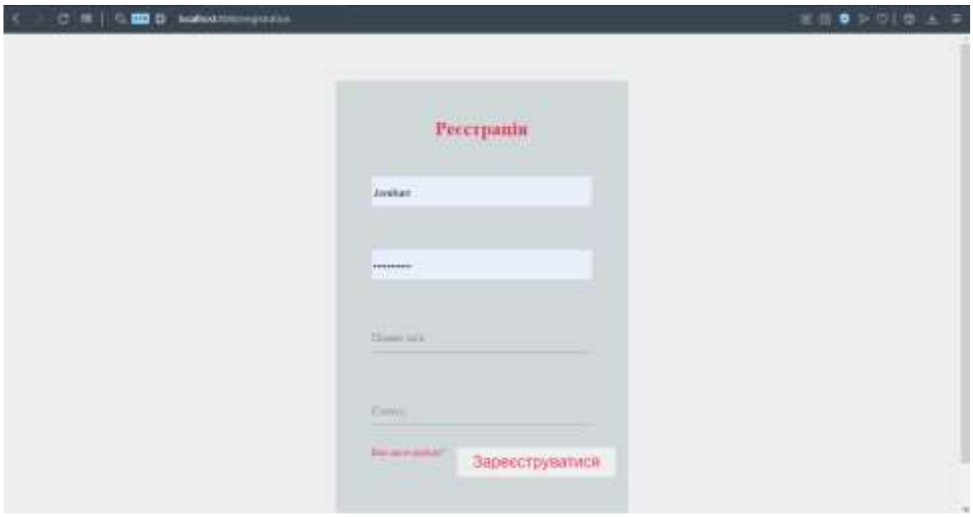


Рисунок 2.8 - Сторінка реєстрації.

2.2. Аналіз варіантів використання веб-додатку

Проектування сучасного веб-додатку для дошкільних навчальних закладів вимагає створення гнучких архітектурних рішень, здатних ефективно автоматизувати процеси контролю та аналізу динаміки вихованців. Центральне місце у технічній документації такого веб-додатка посідає діаграма варіантів використання, яка наочно декомпозує функціональні можливості системи та визначає ролі ключових суб'єктів взаємодії (акторів). Ця діаграма виступає сполучною ланкою між вимогами кінцевих користувачів та програмною реалізацією на базі сучасних фреймворків, забезпечуючи чітку структуру

інтеракції між клієнтською частиною (фронтендом) та серверною логікою (бекендом) [13].

У розробленому веб-додатку диференціюються дві основні категорії акторів, кожна з яких володіє унікальним набором прав доступу та функціональних обов'язків: **Користувач** (батьки, вихователі або методисти) та **Адміністратор** (керівник закладу або технічний персонал). Рольова модель доступу, реалізована на рівні конфігурації безпеки веб-додатка, гарантує, що кожен актор взаємодіє виключно зі своїм пулом прецедентів, захищаючи конфіденційні дані про вихованців від несанкціонованого втручання.

Головна гілка варіантів використання для базового користувача зосереджена навколо щоденного моніторингу та всебічної оцінки розвитку дитини. Первинним прецедентом є «Авторизація/Реєстрація у системі», яка є обов'язковим шлюзом для доступу до приватного контенту. Після успішної автентифікації користувачеві відкривається доступ до прецеденту «Перегляд головної сторінки», яка функціонує як інформаційний хаб і забезпечує транзит до спеціалізованих модулів. Ключовим вектором діяльності вихователя чи батьків є прецедент «Взаємодія з розділом "Для дітей"». Цей варіант використання містить у собі розгалужену систему лінків та інтерактивних засобів, спрямованих на розвиток дитини, включаючи запуск навчальних ігор та перехід до розвиваючих медіа-файлів на зовнішніх хостингах. Наступним важливим прецедентом є «Ознайомлення з новинами», що дозволяє користувачам відстежувати події в житті дитячого садочка, та «Перегляд сторінки "Про нас"», який забезпечує швидкий доступ до верифікованих контактних даних та адміністративної інформації.

З іншого боку, актор «Адміністратор» оперує прецедентами вищого рівня, які безпосередньо впливають на контентне наповнення бази даних. Діаграма відображає специфічні варіанти використання, такі як «Додавання нової публікації (новини)» та «Створення/Модерація зовнішніх посилань». Ці прецеденти мають вирішальне значення для підтримки актуальності навчального

середовища, адже адміністратор є єдиним суб'єктом, що володіє технічною можливістю динамічно модифікувати стрічку новин та оновлювати ігрові чи відео-ресурси для дітей.

На рисунку 2.9 наведено діаграму варіантів діяльності веб-додатку для моніторингу, комплексної оцінки та підтримки розвитку дитини.

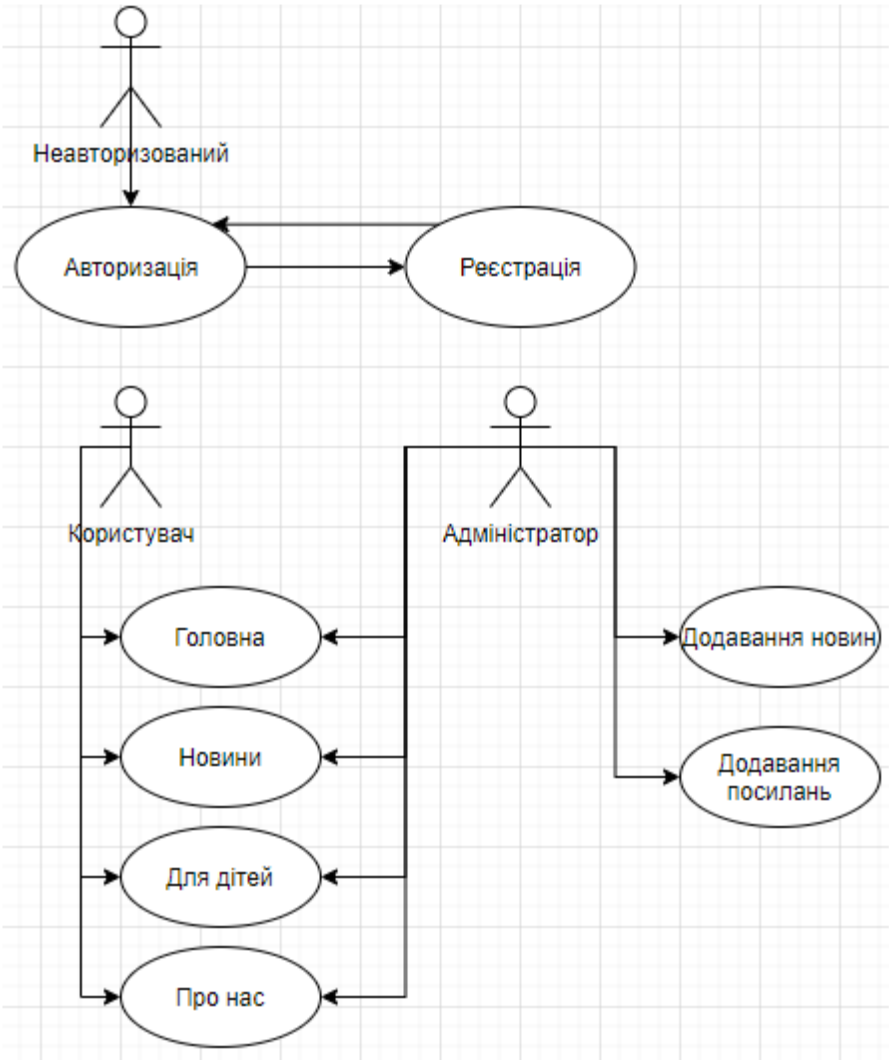


Рисунок 2.9 - Діаграма варіантів використання

2.3. Розробка методу і моделі роботи веб-додатку

Протягом виконання роботи розроблено модель веб-додатку з використанням UML-діаграми діяльності (рис. 2.10), яка об’єднує базові модулі системи і представляє систему взаємозв’язків між ними [14, 15].

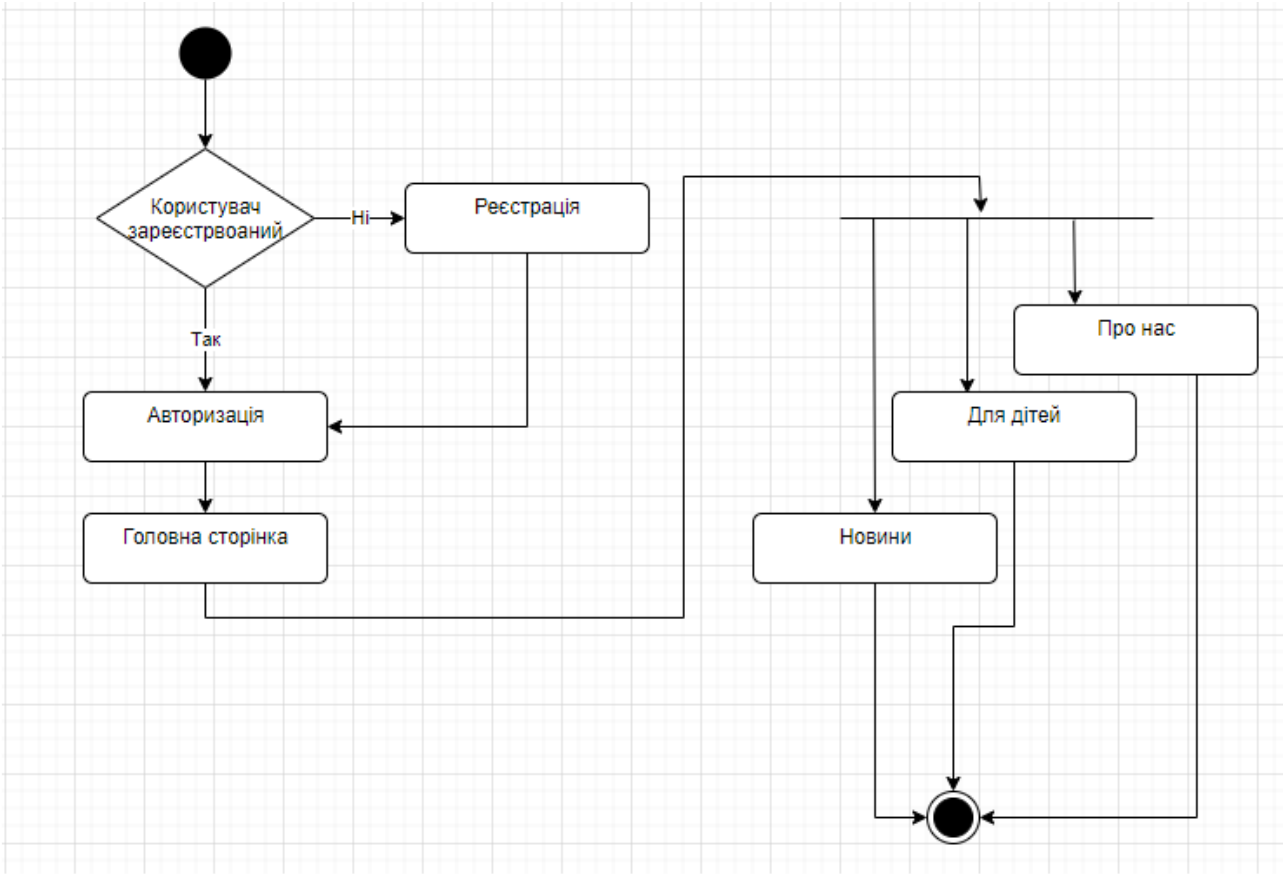


Рисунок 2.10 - Діаграма діяльності

Розроблений веб-додаток орієнтований на підвищення загальної ефективності процесів багатофакторного аналізу, діагностики та стимулювання інтелектуального і психоемоційного зростання вихованців. Головним практичним інструментом для досягнення цих педагогічних цілей є інтеграція у внутрішню архітектуру спеціальних гнучких дидактичних модулів, корисних цифрових матеріалів та інтерактивних засобів навчання, зосереджених у межах окремого загальнодоступного блоку інтерфейсу, що носить назву розділу для дитячого контенту. Специфіка методології моніторингу та супроводу вихованців дитячих садочків, яка реалізована за допомогою даного програмного середовища, полягає в залученні гейміфікованих методик, електронних логічних тренажерів та візуальних посібників, адаптованих під специфіку сприйняття інформації малюками. Цей комплексний підхід базується на скоординованій послідовності взаємопов’язаних операцій, кроків та процедур, які виконуються різними категоріями користувачів системи.

Усе починається з обов’язкового проходження верифікаційного шлюзу, що передбачає введення ідентифікаційних даних та успішну авторизацію у веб-додатку як звичайних відвідувачів, наприклад, батьків та вихователів групи, так і технічного керівника проекту з правами глобального суперадміністратора. Слідом за цим відбувається безпечне перенаправлення авторизованої особи до її персонального цифрового кабінету, який виступає індивідуальним центром управління та аналітики.

Наступна фаза взаємодії полягає в ознайомчому моніторингу загальних публічних сторінок сайту, читанні новинної стрічки та вивченні організаційних оголошень адміністрації закладу. Орієнтуючись на кінцеву мету занять, користувач переходить до спеціалізованої дитячої вкладки, де активовано функціонал інтерактивного пошуку, фільтрації та підбору наочних засобів, які розроблені у доступній, привабливій та ігровій формі з широким використанням анімації, звукового супроводу та елементів зворотного зв'язку. У межах цього ж цільового простору клієнтам та їхнім дітям відкривається прямий безперешкодний доступ до бібліотеки розвиваючих ігор, математичних квестів, завдань на розвиток пам'яті, дрібної моторики, просторової уяви та базового логічного мислення, які підходять для вікових стандартів дітей дошкільного періоду.

Необхідна постійна робота адміністратора, який здійснює регулярний аудит ресурсу, оновлює бази даних, завантажує свіжі дидактичні матеріали та оперативно корегує інформаційне наповнення веб-платформи для підтримки її актуальності.

Впровадження та постійна експлуатація таких навчальних компонентів та розважально-освітнього контенту безпосередньо на офіційному мережевому порталі дитячого садка приносить користь усім учасникам освітнього процесу, дозволяючи підростаючому поколінню у невимушеній, розважальній та легкій формі засвоювати базові знання, корисні навички та необхідні інтелектуальні компетенції. Окрім педагогічного ефекту, цей інструмент виконує важливу

функцію первинної соціалізації та адаптації до грамотного, безпечного та продуктивного використання сучасних комп'ютерних систем, мобільних пристроїв та глобальних інтернет-технологій, трансформуючи звичайне пасивне споглядання екрана на корисну та контрольовану інтелектуальну діяльність.

Виконуючи різноманітні захоплюючі завдання логічного, мовного чи художнього спрямування, діти не просто проводять час за монітором, а автоматично формують свій власний цифровий профіль, при цьому система ретельно веде та відображає персональний рейтинг активності кожного користувача.

Своєю чергою, для самого дошкільного навчального закладу наявність такого високотехнологічного функціоналу стає далекоглядним та ефективним маркетинговим інструментом. Наявність інтерактивного кабінету розвитку дитини суттєво виділяє дитячий садок серед конкурентів на ринку освітніх послуг, виступає чудовим рекламним засобом в інформаційному просторі, приваблює велику кількість нових зацікавлених відвідувачів, утримує увагу постійних користувачів та стабільно підвищує загальні показники відвідуваності та популярності веб-ресурсу в мережі.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ

3.1. Варіантний аналіз та обгрунтування вибору засобів реалізації

Мова програмування Java - це технологія з об'єктно-центрованим інструментом розробки вищого рівня, архітектура якого мінімізує апаратні прив'язки та системні залежності. Цей інструментарій функціонує за принципом кросплатформеності, що дозволяє створеному програмному коду виконуватися на будь-яких цифрових вузлах без необхідності повторного збирання проєкту. Процес обробки вихідних скриптів передбачає їх трансформацію у проміжний байт-код, який безперешкодно інтерпретується ізольованим середовищем виконання без прив'язки до специфіки конкретного комп'ютерного заліза [16].

Синтаксис цієї мови багато в чому наслідує конструкції С-подібних систем, проте розробники позбавили її складних механізмів прямого керування пам'яттю та інших низькорівневих компонентів. Водночас платформа забезпечує гнучку динамізацію процесів, включаючи інтроспекцію та модифікацію структур безпосередньо в ході виконання інструкцій, що є рідкістю для статично компільованих аналогів. Завдяки цьому інструмент утримує лідерські позиції в глобальних репозиторіях і широко застосовується мільйонами інженерів для побудови розподілених архітектур. Ключовим архітектурним орієнтиром технології є абсолютна незалежність від програмно-апаратного оточення, завдяки чому створений продукт функціонує ідентично на будь-яких операційних системах за наявності відповідного рушія. Така універсальність реалізована за допомогою проміжної трансляції вихідних інструкцій у незалежний байт-код, а не у специфічні бінарні команди конкретного процесора.

Як і будь-який Java-софт, код виконується ізольованим середовищем JVM, що гарантує незалежність від операційної системи. Основним призначенням сторінок є віддача HTML або XML, проте через прямі бінарні потоки OutputStream можлива трансляція будь-яких типів медіафайлів. Під час інтерпретації ядро JSP автоматично формує набір системних об'єктів, серед яких

модулі запиту, відповіді, конфігурації, контексту, сесії та обробки помилок. Цей спеціалізований транслятор є системною утилітою, яка аналізує вихідний код JSP та перетворює його на функціональні Java-сервлети. Зазвичай даний інструмент інтегрований безпосередньо в сервер застосунків і спрацьовує автоматично при першому ж зверненні клієнта до сторінки, хоча для підвищення швидкодії системи розробники часто використовують попередню компіляцію або виконують її на етапі збирання проєкту для виявлення синтаксичних помилок [17].

Для створення програмного забезпечення інженери часто використовують IntelliJ IDEA - потужне інтегроване середовище розробки, написане мовою Java компанією JetBrains. Ця IDE представлена як у комерційній версії з платним ліцензуванням, так і у відкритій редакції під егідою Apache 2, причому обидва варіанти офіційно дозволені для комерційного використання. Дебютувавши на початку 2000-х років, ця платформа стала одним із перших інструментів, що запропонували програмістам глибокий аналіз коду, автоматизований рефакторинг та просунуту систему навігації за структурою проєкту.

У профільних галузевих рейтингах, зокрема у звітах аналітиків InfoWorld, дане середовище неодноразово займало лідерські позиції, випереджаючи такі відомі альтернативи, як Eclipse, NetBeans та JDeveloper. Дане середовище надає розробнику повний спектр автоматизованих помічників: інтелектуальне автодоповнення тексту на основі контексту, миттєвий перехід до описів класів та методів, безпечне перейменування чи перебудову внутрішніх структур, а також покрокове налагодження логіки з виправленням архітектурних невідповідностей на основі вбудованих рекомендацій. Додатковою перевагою є глибока інтеграція з утилітами автоматизації збірки типу Gradle чи SBT, системами керування залежностями, а також із популярними сховищами коду, включаючи Git, Mercurial та SVN. Крім того, розширена версія Ultimate дозволяє здійснювати пряме адміністрування та взаємодію з базами даних Microsoft SQL

Server, Oracle, PostgreSQL, SQLite і MySQL безпосередньо через інтегровані модулі DataGrip [18].

Гнучкість цього середовища розробки забезпечується розвиненою архітектурою розширень, які дозволяють інтегрувати додаткові інструменти та модулі в інтерфейс IDE. Користувачі можуть здійснювати інсталяцію додаткових компонентів безпосередньо через офіційний веб-портал лістингів або скориставшись інтегрованим менеджером пошуку у внутрішніх налаштуваннях програми. При цьому кожна окрема генерація середовища має свій каталог сумісних утиліт.

У сфері збереження інформації одним із найвідоміших інструментів є MySQL, яка є реляційним сховищем даних з відкритим кодом. Організація інформаційного простору тут базується на табличних структурах, пов'язаних між собою за допомогою специфічних ключів та відношень, що допомагає впорядкувати великі масиви відомостей. Сама мова SQL слугує інструментом, за допомогою якого інженери здійснюють проектування схем, коригування та вибірку записів, а також регулюють права доступу користувачів. Окрім виконання суто логічних операцій, цей сервер взаємодіє з операційною системою для фізичного розміщення файлів на накопичувачах пристрою, адмініструє облікові записи, підтримує мережеву взаємодію, а також контролює цілісність таблиць і автоматизує створення резервних копій [19].

Для прямого маніпулювання даними сервер комплектується власною консольною утилітою, проте найчастіше він функціонує в тандемі зі сторонніми додатками, де виступає в ролі надійного сховища. Зокрема, цей компонент є базовим елементом класичного серверного стеку LAMP, що об'єднує ОС Linux, веб-сервер Apache та мови програмування PHP чи Python. Сховище лежить в основі популярних систем управління контентом, таких як Drupal, Joomla та WordPress, а також застосовується масштабними світовими платформами, серед яких Facebook, Twitter та YouTube.

Конструктивно сервер створений на базі мов C та C++. Його внутрішній обробник мовних конструкцій базується на інструменті уасс, тоді як лексичний аналізатор розробники написали самостійно. Програмний комплекс адаптований до функціонування на великій кількості системних платформ, включаючи різноманітні дистрибутиви Linux, BSD-системи, macOS, Microsoft Windows, а також спеціалізовані комерційні середовища типу AIX, Solaris чи HP-UX, маючи до того ж окрему збірку для архітектури OpenVMS. Реалізація серверних компонентів MySQL разом із супутніми клієнтськими конекторами базується на принципі дуального ліцензування. Цей підхід передбачає розповсюдження коду або на умовах суспільної угоди GPL другої версії, або в рамках окремих комерційних контрактів, що узгоджуються з правовласником. Програмний комплекс можна скомпілювати власноруч із вихідних кодів, проте, за відсутності специфічних вимог до архітектури, розробники віддають перевагу інсталяції готових бінарних збірок. У більшості операційних систем сімейства Linux цей процес автоматизовано через штатні менеджери пакунків, хоча для виведення системи у робочий режим зазвичай потрібне фінальне ручне коригування конфігураційних файлів з метою оптимізації продуктивності та захисту даних.

3.2. Розробка бази даних

У сучасному світі цифровізація педагогічного супроводу є одним із пріоритетних напрямків розвитку інформаційних технологій. Веб-додатки для моніторингу, комплексної оцінки та підтримки розвитку дитини дозволяють автоматизувати процес збору аналітики, мінімізувати вплив людського фактора під час тестування та забезпечити оперативний зворотний зв'язок для батьків, педіатрів, психологів і педагогів. Основою ефективності такого приклада є надійна, масштабована та гнучка архітектура бази даних. Оскільки дані про розвиток дитини включають медичні показники, результати психологічних тестів, антропометричні виміри та динаміку формування когнітивних чи моторних навичок, проектування бази даних вимагає особливого підходу до

забезпечення цілісності, конфіденційності та швидкодії. Використання Spring Boot Framework як базового технологічного стеку для бекенд-розробки дозволяє максимально спростити та стандартизувати процес взаємодії з базою даних завдяки системі Spring Data JPA, технологіям автоматичної міграції та вбудованим механізмам безпеки. Архітектура включає кілька ключових доменних областей, де спочатку виділяється модуль користувачів та ролей для реалізації контролю доступу. Таблиця користувачів містить облікові дані, такі як ідентифікатор, пошта, хеш пароля та статус акаунта, а таблиця ролей визначає рівні доступу для батьків, лікарів, вихователів та адміністраторів через проміжну таблицю зв'язку багато до багатьох [20].

Наступним важливим елементом є модуль профілю дитини, де ключова таблиця зберігає ім'я, прізвище, дату народження, стать та зовнішній ключ батька для організації зв'язку один до багатьох. Для комплексної оцінки розробляється довідник критеріїв за віковими періодами відповідно до стандартів ВООЗ, де категорії розвитку, такі як фізичний, когнітивний, мовно-комунікативний чи соціально-емоційний, пов'язуються з конкретними очікуваними навичками або маркерами, що містять мінімальний та максимальний вік у місяцях для виконання норми. Безпосередній моніторинг та оцінювання реалізуються через таблиці вимірів антропометричних даних, де фіксуються вага, зріст та обвід голови дитини на певну дату, а також через логі досягнень, де фіксується статус конкретної навички, дата перевірки, нотатки та ідентифікатор спеціаліста, що проводив оцінку. Крім того, додається модуль рекомендацій та планів підтримки, який містить базу знань із порадами, прив'язаними до певного віку чи дефіциту навичок, що дозволяє автоматично або вручну формувати індивідуальні плани розвитку дитини.

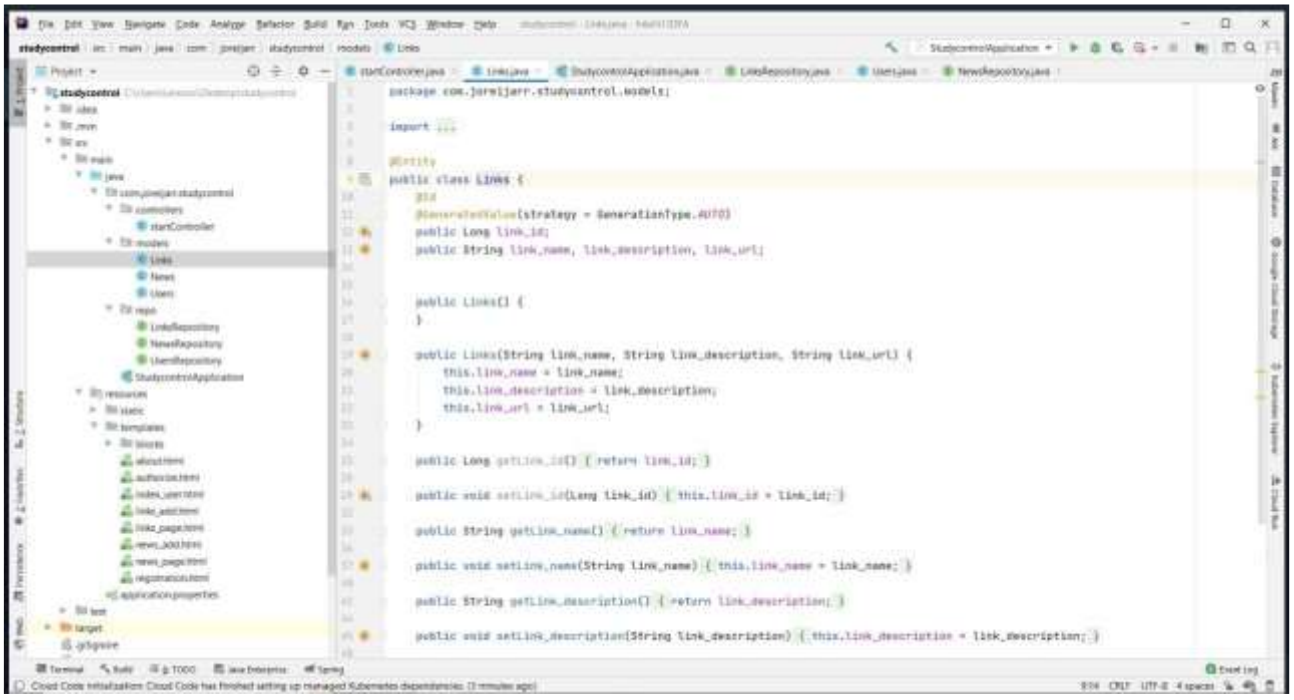
Spring Boot надає концепцію інверсії управління та декларативного керування транзакціями, що дозволяє зосередитися на бізнес-логіці, а не на низькорівневому написанні SQL-запитів, при цьому розробка шару бази даних складається з кількох етапів, починаючи зі створення ORM-сутностей. Кожна

таблиця проектується як Java-клас, анотований як Entity, де завдяки специфікації Jakarta Persistence зв'язки між таблицями трансформуються в об'єктні зв'язки. У класі дитини прописуються всі поля, типи даних, перерахування для статі, а також зв'язок ManyToOne з батьком та зв'язок OneToMany з логами досягнень.

Після сутностей створюються репозиторії на основі Spring Data JPA, які наслідують інтерфейс JpaRepository для автоматичної генерації CRUD-операцій під час запуску додатку. Механізм Query Methods дозволяє описувати складні вибірки, наприклад пошук логів розвитку за ідентифікатором дитини та назвою категорії або пошук тривожних маркерів із певним статусом затримки, безпосередньо у назві методу інтерфейсу, а для важких аналітичних запитів чи побудови графіків перцентилів за великий період використовується анотація Query з написанням SQL або JPQL. Генерація таблиць за допомогою автоматичного оновлення є неприпустимою для реальних систем, тому для контролю версій схеми бази даних у Spring Boot інтегрується інструмент міграцій Flyway, де всі зміни структури описуються у вигляді послідовних SQL-скриптів, що гарантує ідентичність стану бази даних на всіх серверах.

Проектування та створення бази даних для веб-додатку моніторингу розвитку дитини на базі Spring Boot - це комплексне завдання, що успішно вирішується за допомогою об'єднання реляційної моделі даних та сучасного ORM-інструментарію.

Використання Spring Data JPA мінімізує кількість шаблонного коду, Flyway забезпечує безпечну еволюцію схеми даних, а коректно налаштовані індекси та механізми безпеки гарантують високу швидкість обробки інформації та надійний захист приватних даних користувачів. Такий підхід створює міцний фундамент для побудови інтелектуальних систем підтримки прийняття рішень, здатних вчасно виявляти відхилення у розвитку дитини та пропонувати ефективні коригуючі програми. Нижче подано код розроблених моделей і репозиторіїв (рис 3.1, 3.2, 3.3).



```
package com.jareljarr.studycontrol.models;

import ...

@Entity
public class Links {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public Long link_id;
    public String link_name, link_description, link_url;


    public Links() {
    }

    public Links(String link_name, String link_description, String link_url) {
        this.link_name = link_name;
        this.link_description = link_description;
        this.link_url = link_url;
    }

    public Long getLink_id() { return link_id; }

    public void setLink_id(Long link_id) { this.link_id = link_id; }

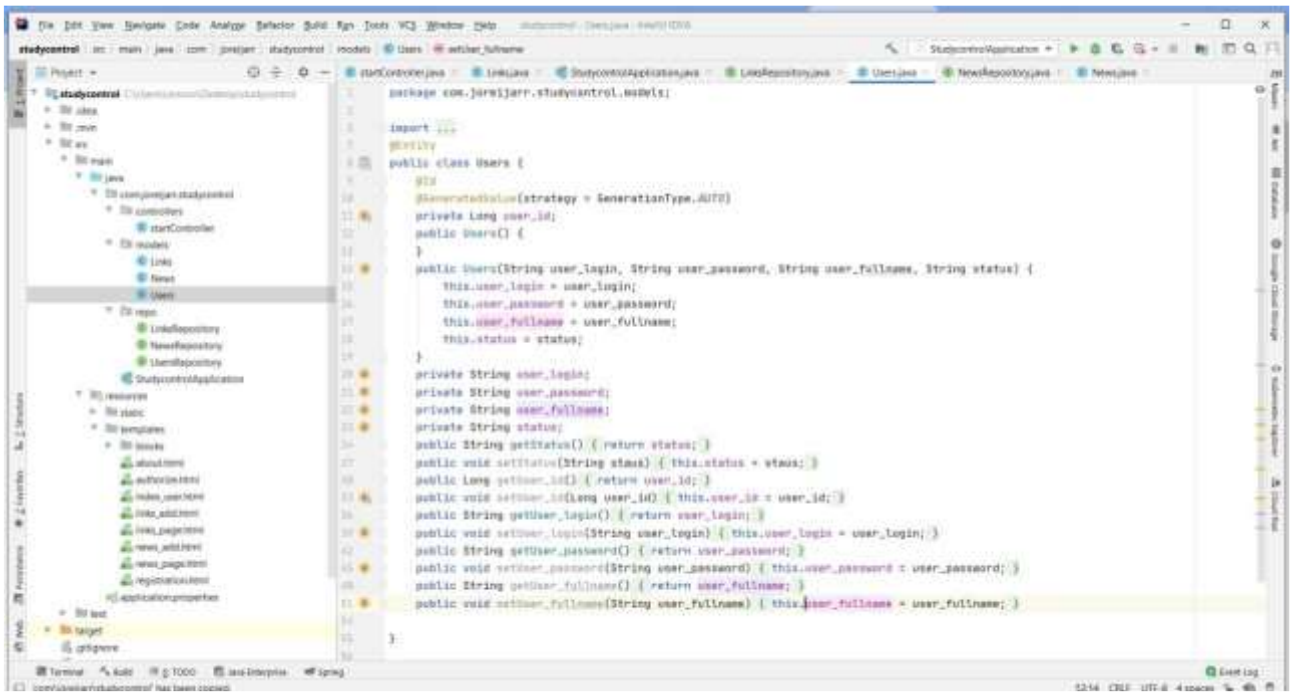
    public String getLink_name() { return link_name; }

    public void setLink_name(String link_name) { this.link_name = link_name; }

    public String getLink_description() { return link_description; }

    public void setLink_description(String link_description) { this.link_description = link_description; }
```

Рисунок 3.1 - Сторінка Links.java



```
package com.jareljarr.studycontrol.models;

import ...
@Entity
public class Users {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long user_id;
    public Users() {
    }
    public Users(String user_login, String user_password, String user_fullname, String status) {
        this.user_login = user_login;
        this.user_password = user_password;
        this.user_fullname = user_fullname;
        this.status = status;
    }
    private String user_login;
    private String user_password;
    private String user_fullname;
    private String status;
    public String getStatus() { return status; }
    public void setStatus(String status) { this.status = status; }
    public Long getUser_id() { return user_id; }
    public void setUser_id(Long user_id) { this.user_id = user_id; }
    public String getUser_login() { return user_login; }
    public void setUser_login(String user_login) { this.user_login = user_login; }
    public String getUser_password() { return user_password; }
    public void setUser_password(String user_password) { this.user_password = user_password; }
    public String getUser_fullname() { return user_fullname; }
    public void setUser_fullname(String user_fullname) { this.user_fullname = user_fullname; }

}
```

Рисунок 3.2 - Сторінка Users.java

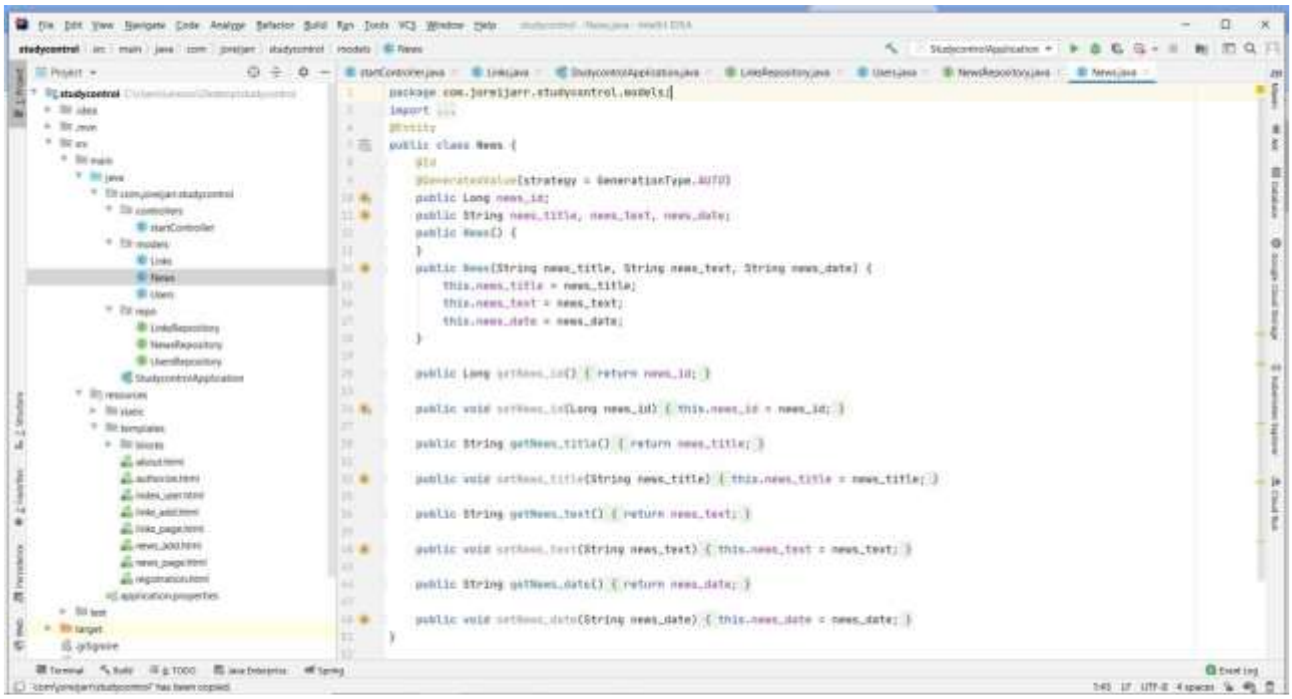


Рисунок 3.3 - Сторінка News.java

Всього створено три сутності: для зберігання користувачів, новин і посиланнь на ресурси для дітей.

3.3. Розробка механізмів авторизації, додавання і показу новин

У сучасному світі цифровізація медичного та педагогічного супроводу є одним із пріоритетних напрямків розвитку інформаційних технологій. Веб-додатки для моніторингу, комплексної оцінки та підтримки розвитку дитини дозволяють автоматизувати процес збору аналітики, мінімізувати вплив людського фактора під час тестування та забезпечити оперативний зворотний зв'язок для батьків, педіатрів, психологів і педагогів. Компонент новин у такій системі - це не просто стрічка загальних повідомлень, а важливий інструмент підтримки, через який адміністрація, профільні лікарі, педіатри та дитячі психологи можуть публікувати методичні рекомендації, сповіщення про нові нормативи тестування, наприклад оновлення шкал ВООЗ, анонси вебінарів та інформацію про роботу центру підтримки. Оскільки доступ до системи мають різні категорії користувачів з кардинально відмінними правами, включаючи

батьків, медичних спеціалістів, психологів, педагогів та адміністраторів, реалізація гнучкого й надійного механізму автентифікації та авторизації є першочерговою умовою.

Використання Spring Boot Framework у поєднанні з модулем Spring Security дозволяє побудувати централізовану систему розмежування доступу на основі ролей, гарантуючи, що створення та редагування контенту залишатиметься прерогативою авторизованих спеціалістів, тоді як перегляд буде доступний відповідним цільовим групам. Безпека додатку базується на архітектурі Spring Security, де для веб-додатків сучасним стандартом є використання безстановочної авторизації на основі JWT (JSON Web Tokens), що дозволяє масштабувати систему та забезпечує зручну взаємодію з клієнтською частиною у вигляді мобільних додатків або веб-інтерфейсів. Процес автентифікації починається з розробки сутностей користувача та ролі, де клас користувача реалізує інтерфейс UserDetails, а його поля включають електронну пошту, яка виступає унікальним логіном, захешований за допомогою алгоритму BCrypt пароль та набір наданих прав. Конфігурація безпеки описується у спеціальному класі, анотованому як Configuration та EnableWebSecurity, де створюється SecurityFilterChain для визначення правил доступу до різних кінцевих точок API. При цьому ендпоінти автентифікації та публічний перегляд новин робляться доступними для всіх без авторизації, тоді як ендпоінти створення та модифікації новин закриваються для звичайних користувачів. Прив'язка прав доступу реалізується декларативно за допомогою методів перевірки шляхів або через анотації над методами контролерів PreAuthorize, завдяки чому доступ до методів додавання новин обмежується виразом перевірки ролей адміністратора чи спеціаліста. Перевірка кожного запиту здійснюється через кастомний фільтр, який наслідує OncePerRequestFilter, перехоплює HTTP-запит, вилучає токен із заголовка Authorization, валідує його за допомогою секретного ключа, зчитує ім'я користувача і його ролі, після чого поміщає об'єкт автентифікації в контекст безпеки системи SecurityContextHolder, що дозволяє чітко ідентифікувати автора

будь-якої дії в межах сесії. Модуль публікації новин спроектовано з урахуванням необхідності структурування та, за потреби, таргетування контенту, тому в базі даних сутність News описується класом із такими обов'язковими атрибутами, як унікальний ідентифікатор, заголовок, повний текст публікації, посилання на медіафайли або зображення, категорія контенту для розділення на психологію чи фізичний розвиток, дата і час створення, а також зв'язок ManyToOne з сутністю користувача-автора, який створив цей запис. Для передачі даних між клієнтом і сервером використовується патерн DTO (Data Transfer Object), де клас створення новини містить лише ті поля, які заповнює автор, і супроводжується анотаціями валідації Jakarta Bean Validation, що запобігає збереженню порожніх або некоректних новин на рівні сервісу. Рівень доступу до даних реалізується через інтерфейс NewsRepository, який розширює JpaRepository, а безпосередня бізнес-логіка зосереджена в сервісному класі NewsService, де метод додавання новини виконується у транзакційному контексті за допомогою анотації Transactional. Сервіс приймає об'єкт передачі даних, за допомогою контексту SecurityContextHolder визначає поточного автентифікованого спеціаліста, створює новий об'єкт сутності, встановлює поточний час системи та викликає метод збереження репозиторію, що забезпечує повний аудит дій, оскільки в базі даних кожна публікація чітко прив'язана до конкретного експерта. Показ новин у веб-додатку з моніторингу розвитку дитини має бути гнучким та адаптивним, щоб батьки та спеціалісти мали можливість не просто бачити суцільний список публікацій, а фільтрувати контент за тематикою або шукати інформацію за ключовими словами, що реалізується через додаткові методи репозиторію або специфікації JPA для динамічної фільтрації. Критично важливим аспектом розробки є впровадження пагінації, оскільки з часом кількість публікацій може обчислюватися тисячами, і одноразове завантаження всієї стрічки новин призвело б до критичного перевантаження пам'яті сервера Java Heap Space та уповільнення роботи клієнтського інтерфейсу. У Spring Boot це вирішується передачею об'єкта Pageable у метод репозиторію, після чого контролер повертає

клієнту об'єкт класу Page, який крім самого списку новин містить метадані про загальну кількість сторінок, поточну сторінку та ознаку наявності наступної порції даних. REST-контролер надає кінцеву точку для отримання новин, яка є повністю публічною, а для оптимізації продуктивності при високій інтенсивності переглядів сервісний шар інтегрується з механізмом кешування Spring Boot за допомогою анотації Cacheable. Дані новин, які рідко змінюються, зберігаються в оперативній пам'яті за допомогою вбудованого кешу Caffeine або зовнішнього сховища Redis, тому при виклику методу показу новин Spring спочатку перевіряє наявність результату в кеші, і лише в разі його відсутності звертається до бази даних. Під час додавання нової новини через адмін-панель автоматично спрацьовує анотація CacheEvict, яка скидає застарілий кеш і змушує систему оновити стрічку для всіх користувачів. Побудова механізмів автентифікації, додавання та показу новин на базі Spring Boot Framework дозволяє створити захищене та високоефективне інформаційне середовище всередині веб-додатку, де модуль Spring Security забезпечує надійний заслін для конфіденційних операцій, а використання архітектурних рішень Spring Data JPA, пагінації та кешування дозволяє системі демонструвати стабільну швидкодію та низький час відгуку навіть при великій кількості одночасних запитів, що є ключовим фактором для створення якісного та зручного IT-продукту в сфері підтримки розвитку дитини.

Механізм авторизації і реєстрації користувачів є одним з основних механізмів системи. Вся система реалізована механізмами Spring Boot Framework, код наведено нижче.

```
package com.kindergarten.model;
import java.time.LocalDate;
public class Child {
    private Long id;
    private String name;
```

```
    private LocalDate birthDate;
    public Child() {}
    public Child(Long id, String name, LocalDate birthDate) {
        this.id = id;
        this.name = name;
        this.birthDate = birthDate;
    }

    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public LocalDate getBirthDate() { return birthDate; }
    public void setBirthDate(LocalDate birthDate) { this.birthDate = birthDate; }
}

package com.kindergarten.model;
import java.time.LocalDateTime;
import java.util.Map;
public class AssessmentResult {
    private Long id;
    private Long childId;
    private LocalDateTime assessmentTime;

    public Long getId() { return id; }

    public void setId(Long id) { this.id = id; }

    public Long getChildId() { return childId; }

    public void setChildId(Long childId) { this.childId = childId; }
```

Один з основних механізмів - це механізм показу і додавання новин. Код даного механізму наведено на рис 3.4 і 3.5.

```
@GetMapping("/news")
public String newsPage(Model model)
{
    model.addAttribute("title", "Новини");
    Iterable<News> news_it = newsRepository.findAll();
    List<News> news = new ArrayList<>();
    news_it.forEach(news::add);
    model.addAttribute("currUser", current_user);
    model.addAttribute("news", news);
    return "news_page";
}
```

Рисунок 3.4 - Показ новин

```
@PostMapping("/news/add")
public String newsAdd_post(Model model,
                           @RequestParam String news_title,
                           @RequestParam String news_text,
                           @RequestParam String news_date)
{
    News news = new News(news_title, news_text, news_date);
    newsRepository.save(news);
    return "redirect:/news";
}
```

Рисунок 3.5 - Додавання новин

Висновок

Протягом виконання кваліфікаційної роботи бакалавра було розроблено веб-додаток, який призначений для оцінки поточного стану та інформаційній підтримці гармонійного розвитку дитини. Проведено аналіз наявних програмних аналогів, що функціонують у сфері дошкільної освіти та педіатрії. Особливістю розробленого веб-додатку є інтеграція інтерактивних елементів, спрямованих на когнітивний та психомоторний розвиток дітей молодшого віку.

Актуальність теми зумовлена тим, що в умовах сучасного цифрового розвитку, проектування та програмна реалізація веб-додатку, призначеного для оцінки поточного стану та інформаційній підтримці гармонійного розвитку дитини, як високотехнологічної веб-системи, є важливим для діджиталізації сфери дошкільної освіти.

Практична реалізація веб-додатку здійснювалася на базі інтегрованого середовища розробки JetBrains IntelliJ IDEA, яке надало всі необхідні інструменти для ефективного написання коду. На початкових етапах дослідження було проведено аналіз поточного стану розв'язання обраного завдання, що дозволило детально вивчити наявні аналоги. Етап проектування та аналізу технологій завершився детальним аргументуванням вибору мови програмування Java та сучасного каркаса Spring Boot як надійного фундаменту для створення веб-додатка, а для організації безпечного й оптимізованого збереження користувацьких відомостей було залучено реляційну систему керування базами даних MySQL. На основі цих інструментів вдалося розробити оригінальну логічну модель та математичний метод функціонування системи, а також створити зручний користувацький інтерфейс.

Завершальна фаза роботи була присвячена верифікації результатів, де фінальне тестування підтвердило стабільність архітектури, повну працездатність створеного програмного продукту та його відповідність усім пунктам технічного завдання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Types of Software Architecture Patterns [Електронний ресурс]. Режим доступу: https://www.geeksforgeeks.org/types-of-software-architecture-patterns/

2. Bestpetsfoods. Основні мови програмування, які використовуються для додатків та веб-сайтів. URL: https://uk.bestpetsfoods.com/principali-linguaggi-di-programmazione-usati-per-app-e-siti-web (дата звернення: 18.05.2026)

3. Особливості HTML. URL: https://uk.education-wiki.com/1711441-advantages-of-html (дата звернення: 18.05.2026).

4. Gadget-info. Різниця між HTML і CSS. URL: https://uk.gadget-info.com/difference-between-html (дата звернення: 18.05.2026).

5. Дитячий садочок «Смайл» [Електронний ресурс] - Режим доступу до ресурсу: http://www.wizardsmile.com.ua/

6. Дитячий садок «Пролісок» [Електронний ресурс] - Режим доступу до ресурсу: https://dytsadok.info/dytyachyj-sadok/37663376/

7. Types of Software Architecture Patterns [Електронний ресурс]. Режим доступу: https://www.geeksforgeeks.org/types-of-software-architecture-patterns/

8. JavaScript. URL: https://hackit-ukraine.com/627-the-advantages-and-disadvantages-of-javascript (дата звернення: 18.05.2026).

9. Інтегроване середовище розробки [Електронний ресурс] – Режим доступу до ресурсу: https://w3schoolsua.github.io/hyperskill/ide.html#gsc.tab=0

10. Eclipse https://www.eclipse.org/downloads/

11. IntellijIdea [Електронний ресурс] – Режим доступу до ресурсу: https://www.jetbrains.com/idea/

12. Binstock, Andrew (May 20, 2015). "Java's 20 Years of Innovation". Forbes. Archived from the original on March 14, 2016. Retrieved March 18, 2016. [Електронний ресурс] – Режим доступу до ресурсу: https://www.scribd.com/document/484532060/computer-class-notes-4

13. James Gosling, Bill Joy, Guy Steele, The Java Language Specification, Java Se

Edition 7th Revised ed. February 24, 2013, 672 p

14. Java [Електронний ресурс] - Режим доступу до ресурсу: https://www.java.com/

15. "Deep Dive Into the New Java JIT Compiler - Graal | Baeldung". www.baeldung.com. August 6, 2021. Retrieved October 13, 2021. [Електронний ресурс] - Режим доступу до ресурсу: https://zditect.com/code/java/deep-dive-into-the-new-java-jit-compiler.html

16. "2 Changes in MySQL 5.5.62 (2018-10-22, General availability)". docs.oracle.com. Retrieved 28 April 2020. [Електронний ресурс] - Режим доступу до ресурсу: https://dev.mysql.com/doc/relnotes/mysql/5.6/en/

17. Hoydalsvik, Geir (19 April 2018). "What's New in MySQL 8.0? (Generally Available)". Retrieved 10 October 2019. [Електронний ресурс] - Режим доступу до ресурсу: https://dev.mysql.com/blog-archive/whats-new-in-mysql-8-0-generally-available/

18. "JavaOne Review: Java Takes on the Internet of Things". [Електронний ресурс] – Режим доступу до ресурсу: www.oracle.com.

19. "A Conversation with James Gosling - ACM Queue". Queue.acm.org. August 31, 2004. Archived from the original on July 16, 2015. Retrieved June 9, 2010. [Електронний ресурс] - Режим доступу до ресурсу: https://www.scribd.com/document/484532060/computer-class-notes-4

20. Build better apps faster with Jetpack Compose [Електронний ресурс] - Режим доступу до ресурсу: https://developer.android.com/develop/ui/compose