ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»

Циклова комісія математичних дисциплін та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «СТВОРЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ»

Виконав: здобувач освіти 4 курсу, групи К121-21
спеціальності 121 – Інженерія
програмного забезпечення
(шифр і спеціальність)
Андрій ЛУЦЕНКО
(ім'я та ПРІЗВИЩЕ)
Керівник Роман КОТОВ
(ім'я та ПРІЗВИЩЕ)
доцент кафедри захисту інформації НУ Запорізька Політехніка, к.т.н., доцент
Рецензент Ганна НЕЛАСА
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ)

Запоріжжя
2025

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ВСП «Економіко-правничий фаховий коледж ЗНУ»

Освітньо-професійний ступінь фаховий молодший бакалавр

Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Голова циклової комісії математичних дисциплін та інформаційних технологій

_____ Тетяна СМОЛЯНКОВА
(підпис)

“ 14 ” червня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ЛУЦЕНКУ Андрію Артемовичу
(ПРІЗВИЩЕ ім’я та по–батькові)

1. Тема роботи «Створення системи автоматизації освітнього процесу»
Керівник роботи КОТОВ Роман Олександрович
(посада, науковий ступінь, вчене звання, ПРІЗВИЩЕ ім’я по-батькові)

Затверджені наказом ВСП ЕПФК ЗНУ від « 26 » листопада 2024 р. № 1852–с

2. Строк подання студентом роботи 13.06.2025

3. Вихідні дані до роботи 1. Реалізувати систему управління та користування розкладом навчального закладу, що включатиме Телеграм-бота для доступу до інформації та вебсайт - редактор для її адміністрування.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз предметної області
2. Концептуальне та логічне проєктування
3. Основні рішення з реалізації компонентів системи

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація до захисту

6. Консультанти розділів роботи

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

7. Дата видачі завдання 01.12.2024

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи:		
	Постановка задачі	Грудень 2024	виконано
2.	Збір вихідних даних, обробка методичних		
	та теоретичних джерел	Січень 2025	виконано
3.	Розробка першого розділу:		
	Огляд інформаційних систем та технологій для їхнього створення	Лютий 2025	виконано
4.	Розробка другого розділу:		
	Розробка проєкту інформаційної системи та вибір технологій	Квітень 2025	виконано
5.	Розробка третього розділу:		
	Розробка застосунку	Травень 2025	виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи та перевірка на плагіат	Червень 2025	виконано
7.	Захист кваліфікаційної роботи	20.06.2025	виконано

Здобувач освіти _____ Андрій ЛУЦЕНКО
(підпис) (ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Роман КОТОВ
(підпис) (ім'я ПРІЗВИЩЕ)

Нормоконтроль пройдено

Нормоконтролер _____ Юлія БОРИСОВСЬКА
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота: 37 сторінок, 1 таблиця, 16 ілюстрацій, 10 позицій у переліку посилань.

Об’єкт дослідження – методи, моделі та програмні засоби для розробки систем автоматизації освітнього процесу в навчальних закладах.

Предмет дослідження – інформаційна система автоматизації освітнього процесу, що включає модуль управління розкладом з доступом через Телеграм-бот та вебінтерфейс для адміністрування.

Мета дослідження – розробка та реалізація інформаційної системи для автоматизації ключових аспектів освітнього процесу, зокрема управління розкладом, з використанням сучасних вебтехнологій та месенджерів.

У кваліфікаційній роботі, спираючись на теоретичне дослідження автоматизації освітніх процесів та аналіз існуючих рішень, розроблено проєкт та реалізовано ключовий модуль інформаційної системи – управління розкладом. Цей модуль використовує СУБД PostgreSQL для зберігання даних розкладу (у форматі JSON). Для зручного доступу студентів та викладачів до актуального розкладу створено Телеграм-бота на платформі Node.js з використанням бібліотеки node-telegram-bot-api. Адміністрування та оновлення розкладу здійснюється через спеціалізований вебредактор, розроблений за допомогою бібліотеки ReactJS. Уся розробка компонентів системи велася мовою JavaScript у середовищі Visual Studio Code, демонструючи ефективний підхід до автоматизації, який може бути масштабований для інших аспектів освітнього процесу.

АВТОМАТИЗАЦІЯ ОСВІТНЬОГО ПРОЦЕСУ, ІНФОРМАЦІЙНА СИСТЕМА, СИСТЕМА УПРАВЛІННЯ РОЗКЛАДОМ, ТЕЛЕГРАМ-БОТ, ВЕБРЕДАКТОР, POSTGRESQL, NODE.JS, REACTJS, JAVASCRIPT, JSON, NODE-TELEGRAM-BOT-API.

SUMMARY

Thesis: 37 pages, 1 table, 16 illustrations, 10 positions in the list of links.

The objects of research are methods, models, and software tools for the development of information systems for automating the educational process in educational institutions.

The purpose of the study is the development and implementation of an information system to automate key aspects of the educational process, particularly schedule management, using modern web technologies and messengers.

In the diploma project, the theoretical aspects of automating educational processes are investigated, and existing approaches and instrumental tools for creating complex information systems in the field of education are analyzed. Based on the conducted analysis, a project was developed and a prototype of an information system was implemented that automates one of the key components of the educational process – the management of class schedules. The system includes a database designed in PostgreSQL DBMS, where schedule information is stored in a structured format (JSON). User access to the schedule is provided through a Telegram bot developed on the Node.js platform using the node-telegram-bot-api library. The administrative part of the system, which allows for schedule management, is implemented as a web editor based on the ReactJS library. The interaction of system components and development was carried out in the JavaScript language within the Visual Studio Code environment. The developed module demonstrates an approach to automation that can be expanded to cover other educational tasks.

EDUCATIONAL PROCESS AUTOMATION, INFORMATION SYSTEM, SCHEDULE MANAGEMENT SYSTEM, TELEGRAM BOT, WEB EDITOR, POSTGRESQL, NODE.JS, REACTJS, JAVASCRIPT, JSON, NODE-TELEGRAM-BOT-API.

ЗМІСТ

Завдання на кваліфікаційну роботу студенту 2
Реферат 4
Summary 5
Вступ 7
1 Аналіз предметної області 10
1.1 Сучасний стан та тенденції автоматизації в освітньому середовищі 10
1.2 Проблематика управління розкладом навчальних занять та необхідність її автоматизації 12
1.3 Принципи організації та обробки даних в сучасних освітніх інформаційних системах 14
1.4 Роль месенджерів в автоматизації комунікації в освітньому процесі 15
1.5 Огляд сучасних фронтенд-фреймворків для розробки вебінтерфейсів освітніх систем 16
2 Розробка проєкту інформаційної системи 20
2.1 Технічне завдання 20
2.2 Етапи створення системи управління розкладом 22
2.3 Діаграма прецедентів 23
2.4 Схема бази даних 24
2.5 Проєктування інтерфейсу 26
3 Програмна реалізація інформаційної системи 28
3.1 База даних застосунку 28
3.2 Створення інтерфейсу 30
3.3 Створення запитів 32
Висновки 34
Перелік використаних джерел 36

ВСТУП

У сучасному світі інформаційні технології (ІТ) відіграють ключову роль у трансформації та оптимізації процесів у різноманітних сферах людської діяльності, і освітня галузь не є винятком. Впровадження систем автоматизації в освітній процес дозволяє не лише підвищити ефективність управління навчальними закладами, але й суттєво покращити якість та доступність освітніх послуг для студентів та викладачів. Автоматизовані інформаційні системи надають навчальним закладам та їх учасникам низку переваг: спрощують адміністративні завдання, забезпечують оперативний доступ до актуальної інформації, сприяють кращій комунікації та оптимізації навчальних планів і ресурсів.

Одним із важливих аспектів освітнього процесу, що потребує ефективної організації, є управління розкладом занять. Традиційні методи формування та поширення розкладів часто є незручними, можуть призводити до помилок та не забезпечують гнучкості й оперативності, яких вимагає динамічне освітнє середовище. Необхідність впровадження сучасних, стандартизованих систем управління розкладом, що підтримують мобільний доступ та інтегруються з популярними серед користувачів платформами, стає все більш нагальною.

Актуальність даної роботи полягає у розробці такого підходу до автоматизації управління навчальним розкладом, який би використовував переваги сучасних технологій, зокрема месенджерів, таких як Telegram, для зручного доступу студентів та викладачів, та вебінтерфейсів для ефективного адміністрування. Таке рішення дозволяє зробити процес створення, редагування та поширення розкладів більш ефективним, прозорим та зручним для всіх учасників освітнього процесу.

Мета дослідження – розробка та реалізація інформаційної системи для автоматизації ключових аспектів освітнього процесу, зокрема управління

розкладом, з використанням сучасних вебтехнологій та месенджерів. Для досягнення мети було сформульовано наступні завдання:

1. Проаналізувати сучасні підходи до автоматизації освітніх процесів та дослідити існуючі інформаційні системи управління розкладами, їх переваги та недоліки.

2. Визначити функціональні та нефункціональні вимоги до розроблюваної системи автоматизації, що включає модуль управління розкладом.

3. Розробити архітектуру інформаційної системи, включаючи проєктування бази даних (PostgreSQL), визначення серверної логіки (Node.js) та проєктування користувацьких інтерфейсів (Телеграм-бот, вебредактор на ReactJS).

4. Здійснити програмну реалізацію основних компонентів системи: модуля зберігання та обробки даних розкладу, Телеграм-бота для взаємодії з користувачами та вебсайту-редактора для адміністрування розкладу.

5. Провести тестування розробленого прототипу системи та підготувати рекомендації щодо її можливого впровадження та подальшого розвитку.

Об’єкт дослідження – методи, моделі та програмні засоби для розробки систем автоматизації освітнього процесу в навчальних закладах.

Предмет дослідження – інформаційна система автоматизації освітнього процесу, що включає модуль управління розкладом з доступом через Телеграм-бот та вебінтерфейс для адміністрування.

Для виконання поставлених завдань використовувалися методи системного аналізу, об'єктно-орієнтованого проєктування, сучасні мови програмування та технології, зокрема JavaScript (Node.js, ReactJS), СУБД PostgreSQL, HTML. Розробка програмних компонентів велася в інтегрованому середовищі розробки Visual Studio Code.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та переліку посилань.

Перший розділ містить основні відомості про інформаційні системи, призначені для підприємств, а саме: визначення, призначення та класифікацію таких систем. Наводиться три приклади актуальних розповсюджених програмних продуктів. Також в розділі описуються технології, які можуть бути використані для реалізації сучасної інформаційної системи обліку.

Другий розділ містить технічне завдання, технічні вимоги до програмного продукту, діаграму варіантів використання, ER-діаграму бази даних, опис структур в базі даних, етапи розробки інформаційної системи та макети застосунку.

Третій розділ містить особливості реалізації кожної частини застосунку – бази даних, сервера та клієнта. Надана схема API. Також виконаний опис застосунку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Сучасний стан та тенденції автоматизації в освітньому середовищі

Сучасний етап розвитку суспільства характеризується глибоким проникненням інформаційних технологій у всі сфери людської діяльності, і освітня галузь не є винятком. Автоматизація освітніх процесів стає не просто трендом, а нагальною потребою, зумовленою зростанням обсягів інформації, вимогами до підвищення якості та доступності освіти, а також необхідністю ефективного управління навчальними закладами. В останні десятиліття спостерігається значний прогрес у розробці та впровадженні різноманітних програмних та апаратних рішень, спрямованих на оптимізацію як адміністративних, так і дидактичних аспектів освітньої діяльності [1].

Сьогодні автоматизація охоплює широкий спектр завдань в освітньому середовищі: від адміністративного управління (ведення баз даних студентів та викладачів, облік успішності, фінансові операції, кадровий облік) до безпосередньої організації навчального процесу. Останнє яскраво ілюструється широким використанням систем управління навчанням (LMS), таких як Moodle, Canvas або Blackboard, які надають інструменти для розміщення навчальних матеріалів, організації онлайн-курсів, проведення тестування та взаємодії між студентами та викладачами [1]. Популярними стали також хмарні платформи, наприклад, Google Classroom або Microsoft Teams for Education, що полегшують створення віртуальних класів, обмін завданнями та спільну роботу над проєктами [2]. Важливим напрямком є також автоматизація комунікаційних процесів між усіма учасниками освітнього процесу та управління ресурсами, включаючи оптимізацію розкладів занять, використання аудиторного фонду та лабораторного обладнання.

Впровадження систем автоматизації дозволяє навчальним закладам оптимізувати використання наявних ресурсів, зменшити рутинне навантаження на адміністративний та викладацький склад, підвищити прозорість та об'єктивність процесів контролю та оцінювання, а також надати студентам гнучкіші та більш персоніфіковані освітні траєкторії. Окрім того, автоматизовані системи сприяють збору та аналізу великих обсягів даних (Big Data) про освітній процес, що відкриває можливості для прийняття обґрунтованих управлінських рішень та прогнозування освітніх результатів [1, 2].

Водночас, процес автоматизації освітнього середовища супроводжується певними викликами. До них належать значна вартість впровадження та підтримки сучасних IT-рішень, необхідність забезпечення належного рівня підготовки персоналу для роботи з новими системами, питання інформаційної безпеки та захисту персональних даних, а також ризик поглиблення «цифрового розриву» серед учасників освітнього процесу [3].

Незважаючи на зазначені виклики, загальні тенденції свідчать про подальше поглиблення рівня автоматизації в освіті. Перспективними напрямками вважаються розвиток технологій штучного інтелекту для створення інтелектуальних навчальних систем та персоналізованих освітніх асистентів, використання хмарних технологій для забезпечення доступності освітніх ресурсів, а також інтеграція різноманітних платформ та сервісів, включаючи спеціалізовані університетські портали та мобільні додатки, у єдиний освітній інформаційний простір. Ефективність та надійність усіх цих систем значною мірою залежить від того, наскільки грамотно та раціонально організовано збір, зберігання, обробка та представлення даних, що робить питання принципів організації даних в освітніх інформаційних системах надзвичайно актуальним і логічно веде до наступного підрозділу [3].

1.2 Проблематика управління розкладом навчальних занять та необхідність її автоматизації

Управління розкладом навчальних занять є одним із ключових та водночас найбільш складних організаційних завдань у будь-якому навчальному закладі. Ефективно складений та своєчасно донесений до всіх учасників освітнього процесу розклад є запорукою планомірної та продуктивної навчальної діяльності. Однак, традиційні, переважно ручні або частково автоматизовані методи управління розкладами, які все ще поширені в багатьох освітніх установах, пов'язані з низкою суттєвих проблем [3].

По-перше, складність та трудомісткість процесу складання розкладу є значною перешкодою. Диспетчерам або відповідальним особам доводиться враховувати величезну кількість взаємопов'язаних факторів: наявність та завантаженість аудиторного фонду, доступність викладачів, навчальні плани різних груп та спеціальностей, індивідуальні потреби студентів, санітарні норми та багато іншого. Ручне узгодження всіх цих параметрів часто призводить до тривалого процесу, високої ймовірності помилок, накладок та неоптимального використання ресурсів.

По-друге, низька гнучкість та оперативність внесення змін у вже затверджений розклад є ще однією серйозною проблемою. Необхідність заміни викладача, перенесення заняття через непередбачені обставини або зміна доступності аудиторії вимагає швидкої реакції. В умовах ручного управління це може призвести до затримок в інформуванні студентів та викладачів, плутанини та зриву навчального процесу.

По-третє, проблема доступності та актуальності інформації про розклад для кінцевих користувачів – студентів та викладачів – часто залишається невирішеною. Розклади, розміщені на дошках оголошень або у вигляді статичних файлів на вебсайтах, не завжди оперативно оновлюються, можуть бути незручними для перегляду з мобільних пристроїв та не надають можливості швидкого пошуку потрібної інформації. Це призводить до

непорозумінь, запізнень та загального зниження комфорту навчального процесу.

По-четверте, відсутність стандартизації та єдиного формату подання розкладів у різних підрозділах або навіть для різних груп ускладнює їх сприйняття та інтеграцію з іншими інформаційними системами навчального закладу. Це також ускладнює аналіз завантаженості ресурсів та планування освітньої діяльності на перспективу.

Усі перелічені проблеми безпосередньо впливають на якість освітнього процесу, ефективність використання ресурсів навчального закладу та задоволеність студентів і викладачів. Саме тому необхідність автоматизації управління розкладом навчальних занять є очевидною та нагальною.

Впровадження сучасної автоматизованої системи управління розкладом дозволяє:

- Значно спростити та прискорити процес складання розкладу завдяки використанню алгоритмів оптимізації та автоматизованому врахуванню обмежень.
- Підвищити гнучкість та оперативність управління змінами, забезпечуючи миттєве оновлення інформації для всіх користувачів.
- Забезпечити зручний та швидкий доступ до актуального розкладу через різні канали, включаючи вебінтерфейси та мобільні додатки, зокрема популярні месенджери.
- Стандартизувати подання інформації та забезпечити її інтеграцію з іншими компонентами загальної системи автоматизації освітнього процесу.
- Оптимізувати використання аудиторного фонду та робочого часу викладачів, що сприяє економії ресурсів.

Таким чином, автоматизація управління розкладом є не просто технічним удосконаленням, а стратегічно важливим кроком на шляху до підвищення ефективності та модернізації всього освітнього процесу, що робить розробку відповідних програмних рішень актуальним та затребуваним завданням.

1.3 Принципи організації та обробки даних в сучасних освітніх інформаційних системах

Ефективність будь-якої системи автоматизації освітнього процесу безпосередньо залежить від того, наскільки грамотно організовано збір, зберігання, обробку та представлення даних. Освітні інформаційні системи оперують великими масивами різнорідної інформації, що включає персональні дані студентів та викладачів, навчальні плани, розклади занять, дані про успішність, методичні матеріали та багато іншого. Відтак, вибір належних принципів організації даних та відповідних систем управління базами даних (СУБД) є критично важливим завданням на етапі проєктування [4, 5].

Ключовими вимогами до організації даних в освітніх інформаційних системах є забезпечення їх цілісності, узгодженості, безпеки, доступності та можливості масштабування системи. Для задоволення цих вимог традиційно та успішно застосовуються реляційні бази даних, які забезпечують структурований та організований спосіб зберігання інформації, заснований на моделі таблиць, рядків та стовпців. Реляційна модель дозволяє чітко визначати зв'язки між різними сутностями (наприклад, "студент" – "курс" – "оцінка"), підтримувати цілісність даних за допомогою ключів та обмежень, а також виконувати складні запити для вибірки та аналізу інформації [7].

Серед реляційних СУБД однією з найпотужніших та найпопулярніших систем з відкритим вихідним кодом є PostgreSQL. Ця СУБД вирізняється своєю надійністю, функціональністю та високим рівнем відповідності стандартам SQL. PostgreSQL підтримує широкий спектр функцій, критично важливих для освітніх систем, таких як транзакції для забезпечення атомарності операцій (наприклад, при зарахуванні на курс чи виставленні оцінки), збережені процедури та тригери для реалізації складної бізнес-логіки на рівні бази даних, а також розширені механізми забезпечення цілісності даних. Її здатність працювати з великими обсягами даних та високі показники

продуктивності роблять її придатною для освітніх установ різного масштабу. Крім того, PostgreSQL має активну спільноту розробників та користувачів, що забезпечує постійний розвиток системи, наявність великої кількості документації та інструментів для роботи [9].

Хоча останнім часом активно розвиваються і NoSQL бази даних, що пропонують більшу гнучкість для роботи з неструктурованими або напівструктурованими даними, реляційні системи, такі як PostgreSQL, залишаються незамінними для зберігання основних, структурованих даних освітнього процесу, де важливі чіткість схеми та гарантії цілісності. Для таких завдань, як управління розкладом, облік студентів та їхньої успішності, саме реляційний підхід часто є найбільш виправданим [4].

Важливим аспектом обробки даних є також забезпечення можливості їх аналізу для прийняття управлінських рішень та моніторингу якості освітнього процесу. Сучасні СУБД, включаючи PostgreSQL, надають інструменти для виконання аналітичних запитів, агрегації даних та побудови звітів, що є цінним для адміністрації навчальних закладів [7, 8].

Таким чином, вибір принципів організації даних та конкретної СУБД, як-от PostgreSQL, що поєднує надійність, гнучкість та потужний функціонал, є фундаментальним кроком у створенні ефективної та стійкої системи автоматизації освітнього процесу, здатної задовольнити сучасні вимоги до управління освітньою діяльністю.

1.4 Роль месенджерів в автоматизації комунікації в освітньому процесі

Паралельно з розвитком вебплатформ, невід'ємною частиною сучасного комунікаційного простору, особливо серед молоді, стали месенджери. Їхня повсюдність, миттєвість доставки повідомлень та зручність використання на мобільних пристроях відкривають значні можливості для автоматизації та

оптимізації інформаційної взаємодії в освітньому середовищі. Серед популярних месенджерів Telegram вирізняється гнучким API для створення ботів, що робить його привабливою платформою для розробки спеціалізованих освітніх інструментів [10].

Використання Телеграм-бота для доступу до розкладу навчальних занять, як це передбачено в даній роботі, є яскравим прикладом ефективного застосування месенджерів. Такий підхід дозволяє студентам та викладачам отримувати актуальну інформацію про заняття безпосередньо на свої пристрої у звичному інтерфейсі. Бот може забезпечувати не лише перегляд розкладу на поточний день чи тиждень, але й надсилати оперативні сповіщення про зміни, заміни, скасування занять, а також нагадування про майбутні події. Це значно підвищує рівень інформованості, зменшує ймовірність запізнень та пропусків занять через необізнаність. Крім того, Телеграм-боти можуть використовуватися для розсилки навчальних матеріалів, проведення швидких опитувань або збору зворотного зв'язку, тим самим доповнюючи функціонал більш комплексних LMS. Простота розробки та відносно низькі витрати на підтримку таких ботів роблять їх доступним інструментом для багатьох навчальних закладів [5].

1.5 Огляд сучасних фронтенд-фреймворків для розробки вебінтерфейсів освітніх систем

Для створення інтуїтивних та продуктивних користувацьких інтерфейсів розробники сьогодні мають у своєму розпорядженні багатий арсенал фронтенд-технологій. Серед них домінують такі інструменти, як React, Angular та Vue.js, що здобули популярність завдяки своїй продуктивності, зручності в розробці, широкій підтримці спільноти та можливостям масштабування [6].

React, як бібліотека для побудови інтерфейсів, акцентує на компонентній архітектурі та надає гнучкі механізми управління станом, часто за допомогою зовнішніх бібліотек типу Redux або через Context API. Angular, натомість, позиціонується як повноцінний фреймворк, що "з коробки" пропонує функціонал для роботи з формами, навігацією та мережевими запитами. Vue.js, відомий своєю простотою освоєння та гнучкістю, часто стає вибором для команд, що прагнуть швидко розпочати розробку та цінують інтуїтивність [6].

Крім цих лідерів, ринок пропонує й інші рішення, такі як Svelte, Ember.js та Backbone.js, які, хоч і менш популярні, можуть бути оптимальними для певних проєктних сценаріїв завдяки своїм специфічним перевагам [6].

Порівняльний аналіз ключових технологій наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння існуючих вебтехнологій

Технологія	Переваги	Недоліки	Використання
React	Гнучкість, активна спільнота	Потребує додаткових бібліотек	Вебдодатки
Angular	Повний набір інструментів	Висока складність освоєння	Великі проєкти
Vue.js	Простота, зручний API	Менша екосистема	Середні/малі проєкти
Svelte	Легка вага, відсутність залежностей	Новизна, менша підтримка	Специфічні рішення та стартапи

1.6 Постановка задачі на розробку системи автоматизації освітнього процесу з модулем управління розкладом

Проведений аналіз сучасного стану автоматизації в освітньому середовищі (п. 1.1), принципів організації даних (п. 1.2), а також детальний розгляд проблематики управління розкладом навчальних занять (п. 1.3) однозначно вказують на актуальність та необхідність розробки ефективних програмних рішень у цій сфері. Було виявлено, що ручні та частково автоматизовані методи управління розкладами є трудомісткими, негнучкими, часто призводять до помилок та не забезпечують належного рівня доступності актуальної інформації для студентів та викладачів. Водночас, сучасні технології, такі як месенджери (на прикладі Telegram, п. 1.4) для взаємодії з користувачами та потужні фронтенд-фреймворки (зокрема ReactJS, п. 1.5) для створення адміністративних інтерфейсів, надають ефективні інструменти для вирішення цих проблем.

Виходячи з цього, основною проблемою, на вирішення якої спрямована дана кваліфікаційна робота, є відсутність або недостатня ефективність існуючих у багатьох навчальних закладах інструментів для автоматизованого, гнучкого та зручного управління розкладом навчальних занять, що негативно впливає на якість освітнього процесу та ефективність використання ресурсів.

Відповідно**,** метою даної кваліфікаційної роботи є розробка інформаційної системи для автоматизації освітнього процесу, що включає ключовий модуль – систему управління та користування розкладом навчального закладу. Цей модуль має забезпечити ефективне адміністрування розкладу через вебінтерфейс та зручний доступ до актуальної інформації для студентів і викладачів за допомогою Телеграм-бота.

Для досягнення поставленої мети необхідно вирішити наступні основні завдання:

1. Спроєктувати архітектуру модуля управління розкладом, визначивши його основні компоненти (база даних, серверна логіка API, Телеграм-бот, вебредактор) та принципи їх взаємодії.

2. Розробити структуру бази даних (з використанням СУБД PostgreSQL), що забезпечить ефективне зберігання та управління даними про розклад занять, навчальні групи, викладачів та інші пов'язані сутності.

3. Реалізувати серверну частину системи (на платформі Node.js), що включатиме API для взаємодії з базою даних та клієнтськими додатками, а також логіку роботи Телеграм-бота (з використанням node-telegram-bot-api).

4. Розробити клієнтську частину – вебсайт-редактор (з використанням бібліотеки ReactJS) для адміністрування розкладу, що дозволить уповноваженим особам створювати, редагувати, публікувати та оновлювати розклад.

5. Забезпечити функціонал Телеграм-бота для надання користувачам (студентам, викладачам) можливості перегляду актуального розкладу, отримання сповіщень про зміни та виконання інших пов'язаних запитів.

6. Провести тестування розробленого модуля для перевірки його функціональності, надійності та зручності використання.

Очікується, що розроблена система дозволить значно підвищити ефективність управління розкладом, покращити інформованість учасників освітнього процесу та створить основу для подальшої автоматизації інших аспектів освітньої діяльності навчального закладу.

2 РОЗРОБКА ПРОЄКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Технічне завдання

2.1.1 Найменування і область застосування

Програмний продукт, що розробляється, отримує найменування: «Система управління розкладом навчального закладу».

Програма призначена для автоматизації доступу до інформації про розклад через телеграм бота для студентів та викладачів навчального закладу та генерації відповідного розкладу.

2.1.2 Підстава для розробки

Програма розробляється на підставі наказу № 1852-с «Про затвердження тем кваліфікаційних робіт студентів освітньо-професійного рівня ступеня фаховий молодший бакалавр денної форми здобуття освіти ВСП «Економіко-правничий фаховий коледж ЗНУ» від 26.11.2024 року.

2.1.3 Призначення розробки

Даний проєкт призначений для вирішення наступних завдань:

1. Автоматизація процесу адміністрування розкладу.
2. Забезпечення оперативного та зручного доступу до актуального розкладу.
3. Підвищення ефективності використання навчальних та ресурсів.

4. Покращення рівня інформованості та комунікації всіх учасників освітнього процесу.

5. Створення програмної основи для подальшої інтеграції та розширення системи.

2.1.4 Технічні вимоги до програмного продукту

На початку роботи над системою були сформульовані наступні загальні системні вимоги:

1. Розробити клієнт-серверне програмне рішення для управління та відображення розкладу навчальних занять.

2. Система повинна складатися з серверної частини (API), бази даних, клієнтської частини для адміністрування (вебредактор) та інтерфейсу для користувачів (Телеграм-бот).

Вимоги до бази даних:

1. Забезпечити зберігання даних про розклад (день, час, предмет, ПІБ викладача, ID/пароль/посилання на конференцію).

2. Дані розкладу для груп/днів мають зберігатися у структурованому вигляді (наприклад, JSON у відповідному полі таблиці).

3. Реалізувати таблиці для зберігання інформації про групи, користувачів (з ролями) та, за необхідності, інші пов'язані сутності.

Вимоги до серверної частини:

1. Реалізувати API для обробки запитів від Телеграм-бота та вебредактора.

2. API повинен забезпечувати CRUD-операції (створення, читання, оновлення, видалення) для даних розкладу.

3. Забезпечити валідацію вхідних даних від клієнтів.

Вимоги до вебредактора:

1. Надати адміністраторам інтерфейс для створення нових розкладів.

2. Забезпечити можливість редагування існуючих розкладів.

3. Реалізувати функцію збереження/публікації розкладу в базі даних через API.

4. Забезпечити можливість завантаження/вивантаження розкладу у форматі JSON.

2.2 Етапи створення системи управління розкладом

Процес розробки автоматизованої інформаційної системи обліку для крамниці музичних інструментів можна розділити на декілька етапів:

1. Детальне визначення вимог та планування (дослідницький етап).
2. проєктування системи (архітектурний етап).
3. Розробка (реалізація) програмних компонентів (конструкторський етап).
4. Інтеграція та тестування системи (випробувальний етап).
5. Підготовка документації та завершення проєкту.

Під час детального визначення вимог та планування слід провести аналіз предметної області та потреб користувачів, на основі чого чітко сформулювати функціональні (що система має робити) та нефункціональні (яким критеріям відповідати: продуктивність, безпека) вимоги. Важливою частиною є обґрунтований вибір технологічного стеку (наприклад, PostgreSQL, Node.js, ReactJS, node-telegram-bot-api) та завершення етапу складанням плану робіт.

Під час проєктування системи необхідно розробити загальну архітектуру системи, визначивши її основні компоненти (база даних, серверний API, Телеграм-бот, вебредактор) та схеми їх взаємодії. Ключова увага приділяється проєктуванню бази даних (концептуальна, логічна з ER-діаграмою, фізична моделі для PostgreSQL), проєктування API (ендпоінти,

формати запитів/відповідей) та проєктуванню користувацьких інтерфейсів (макети та сценарії взаємодії для бота і вебредактора).

Цей етап передбачає написання коду. Доцільно розпочати зі створення бази даних, потім перейти до розробки серверної частини на Node.js (реалізація API, логіки бота, взаємодії з БД). Паралельно або послідовно може вестися розробка вебредактора на ReactJS (створення UI, взаємодія з API) та функціоналу Телеграм-бота (обробка команд, формування відповідей).

Після розробки окремих частин настає етап їх об'єднання та всебічної перевірки. Рекомендується проводити модульне (окремі функції), інтеграційне (взаємодія компонентів) та системне (комплексна перевірка) тестування для виявлення та виправлення помилок і забезпечення відповідності вимогам.

Заключний етап повинен включати систематизацію інформації: підготовку технічної документації (опис архітектури, функціоналу) та звітної документації. Залежно від цілей, цей етап може також включати планування впровадження системи.

2.3 Діаграма прецедентів

Діаграма прецедентів системи управління розкладом представлена на рисунку 2.1.

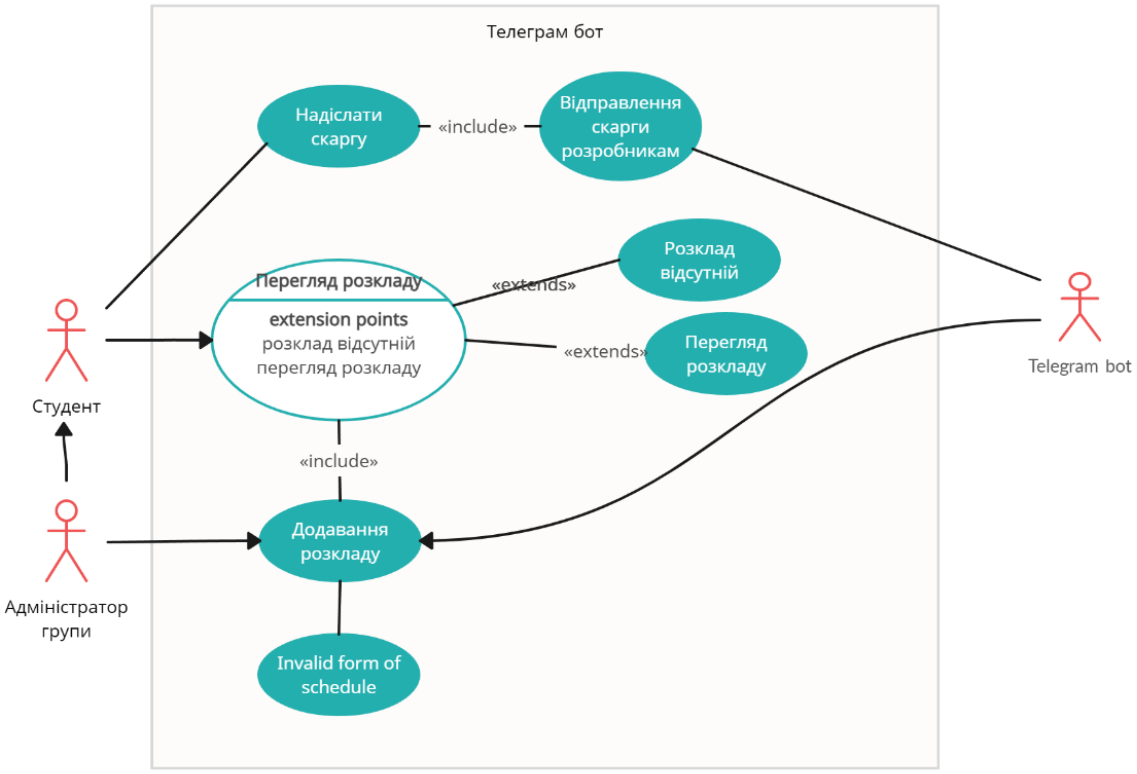


Рисунок 2.1 – Діаграма прецедентів

2.4 Схема бази даних

На основі аналізу предметної області та функціональних вимог до системи управління та користування розкладом навчального закладу я визначив ключові сутності та їх атрибути. Центральною сутністю є "File", яка представляє файли, що зберігаються в системі. Ця сутність має атрибути id (первинний ключ), name_of_json (назва JSON-файлу), content (вміст файлу у форматі BLOB), createdAt (дата створення) та updatedAt (дата оновлення). Крім того, я виділив додаткові сутності, які існують всередині json файлу для зберігання інформації розкладу, необхідні для відображення даних про студентів, викладачів, курси та розклад занять відповідно.

Сутність "File" (Файл) має атрибути:

- id (первинний ключ)
- name_of_json (назва JSON-файлу)

- content (вміст файлу у форматі BLOB)
- createdAt (дата створення)
- updatedAt (дата оновлення)

Сутність “Lesson” (Розклад), що зберігається у json та є у атрибуті content:

- time (період тривалості заняття)
- lesson-name (назва предмету)
- teacher (ім’я викладача)
- link (посилання)
- identifier (ідентифікатор (опціонально))
- code (код (опціонально))

Сутність “Schedule” зберігає масив днів розкладу, у дні зберігається масив сутностей “Lesson”.

Використовуючи описану структуру, можемо мати таку ER-модель, яка зображена на рисунку 2.2.

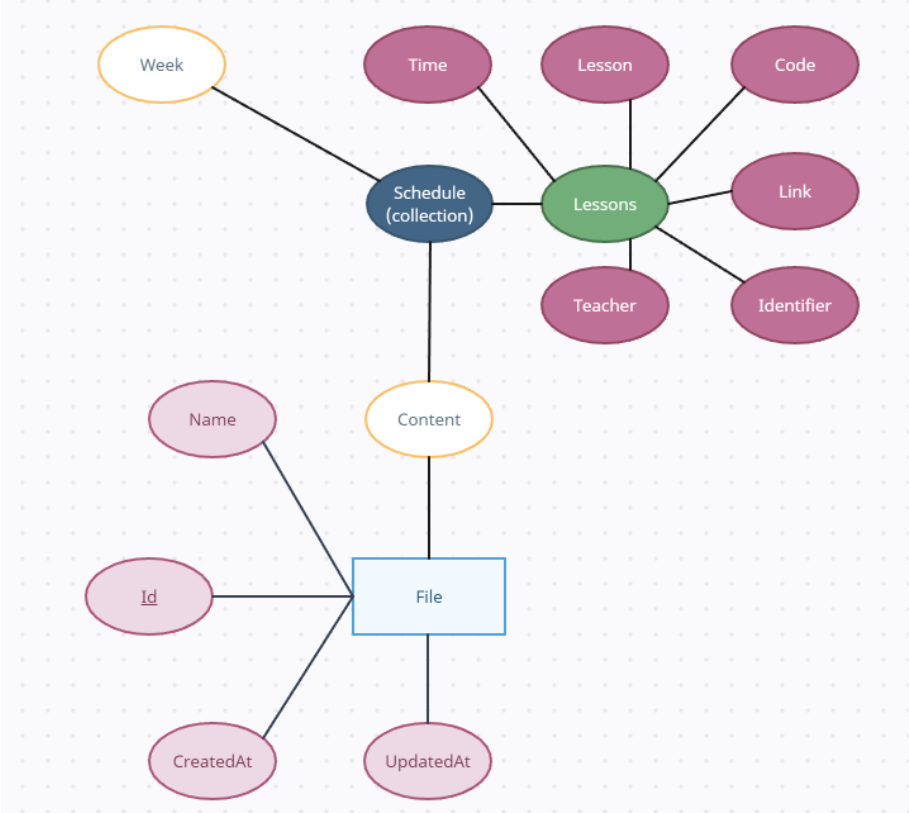


Рисунок 2.2 – ER діаграма

2.5 Проєктування інтерфейсу

На основі технічного завдання був с проєктований інтерфейс застосунку. На рисунку 2.3 наведений макет головної сторінки.

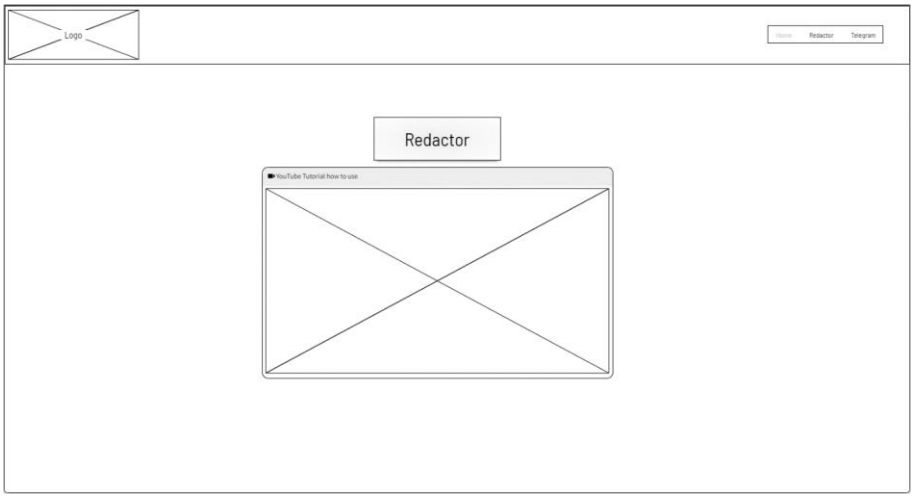


Рисунок 2.3 – Макет головної сторінки

На рисунку 2.4 наведений макет сторінки редактору самого розкладу. Сторінка містить список шість елементів розкладу на кожен день тижня.

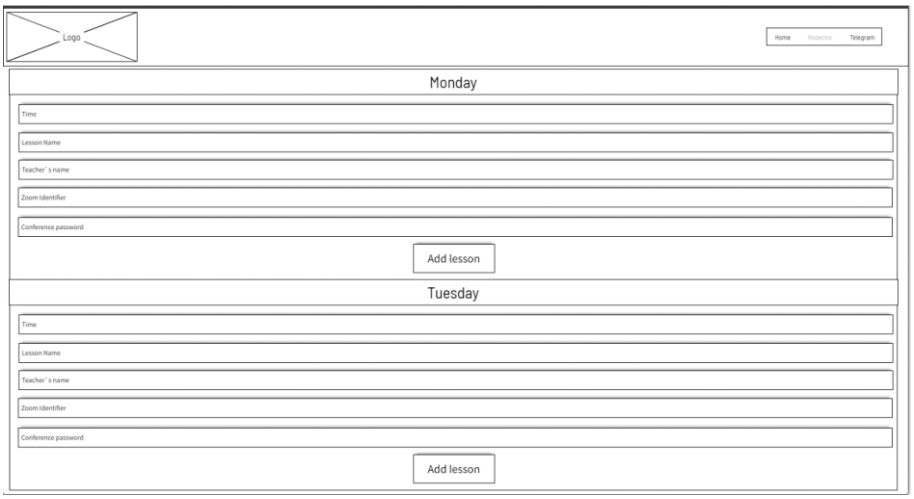


Рисунок 2.4 – Макет сторінки редактору розкладу

На рисунку 2.5 наведений макет сторінки переходу на бот у Telegram.

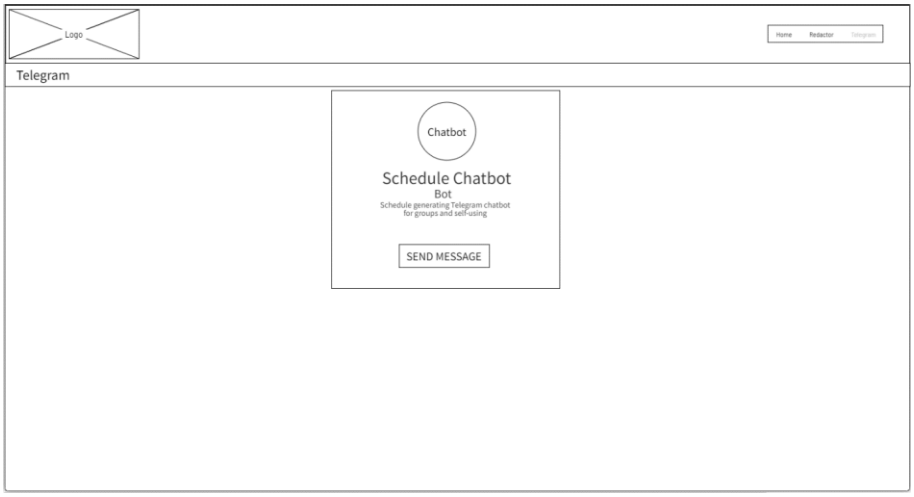


Рисунок 2.5 – Макет відображення категорії

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

На етапі проєктування інформаційної системи було вирішено створити систему засобами React, node-telegram-bot, PostgreSQL. Під час розробки використовувалася система контролю версій Git: були створені окремі репозиторії для серверу та клієнту.

3.1 База даних застосунку

Було створено таку основну таблицю, як «Files» (файли або розклади), зображену на рисунках 3.1 та 3.2, в якій міститься інформація про з'єднані зв'язком телеграм переписки (чати) та розклади, які відповідають підлюченому чату, має таку інформацію у собі:

- Id (тип даних: лічильник, розмір поля: ціле);
- Name (тип даних: текстовий, розмір поля: 255);
- Content (тип даних: blob, розмір поля: не більше 1 GB);
- CreatedAt (тип даних: дата/час, формат поля: довгий формат дати, маска ввода: 00\->L<LL\-0000;0;_);
- UpdatedAt (тип даних: дата/час, формат поля: довгий формат дати, маска ввода: 00\->L<LL\-0000;0;_).

	id [PK] integer	name character varying (255)	content bytea	createdAt timestamp with time zone	updatedAt timestamp with time zone
1	1	765307114.json	[binary dat...	2024-04-18 01:05:04.886+03	2024-04-18 01:05:04.886+03

Рисунок 3.1 – Таблиця «Files»

Таблиця «Files» в режимі конструктора представлена на рисунку 3.2.

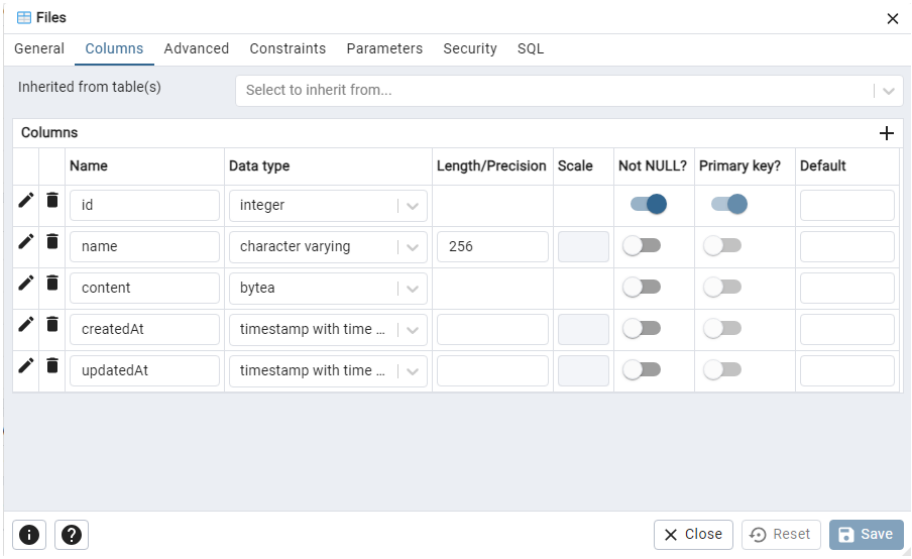


Рисунок 3.2 – Таблиця «Files» в режимі конструктора

Потрібно зазначити про те, що у пункті Content зберігається відформатований JSON файл, який зберігає у собі розклад. Цей розклад складається зі списку днів тижня, у якому вже зберігається список занять, показаний на рисунку 3.3. Одна сутність заняття має таку структуру:

- time (час початку та кінця
- lesson-name
- teacher
- link
- identifier
- code.


```
{
    "week": "неділя",
    "lessons": []
},
{
    "week": "понеділок",
    "lessons": [
        {
            "time": "11:30 - 11:45",
            "lesson": "Математика",
            "teacher": "Петренко І. О.",
            "link": "https://www.internet.com/",
            "identifier": "22222",
            "code": "22222"
        }
    ]
}
```


Рисунок 3.3 – Приклад розкладу, який зберігається у базі даних

3.2 Створення інтерфейсу

Для роботи з інтерфейсом у телеграм боті використовується html. Для роботи з конструктором розкладу використовується ReactJS.

Був створений сайт для конструктування розкладу.

Була створена головна сторінка сайту-конструктора розкладу. Зображена на рисунку 3.4.

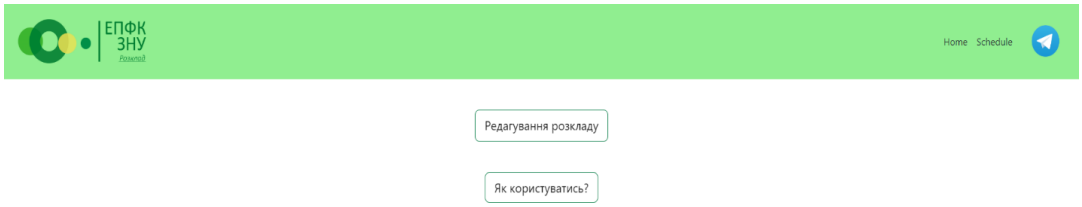


Рисунок 3.4 – Головна сторінка сайту-конструктора

Форма для редагування і створення розкладу, який оброблюється у телеграм боті, зображена на рисунку 3.5.

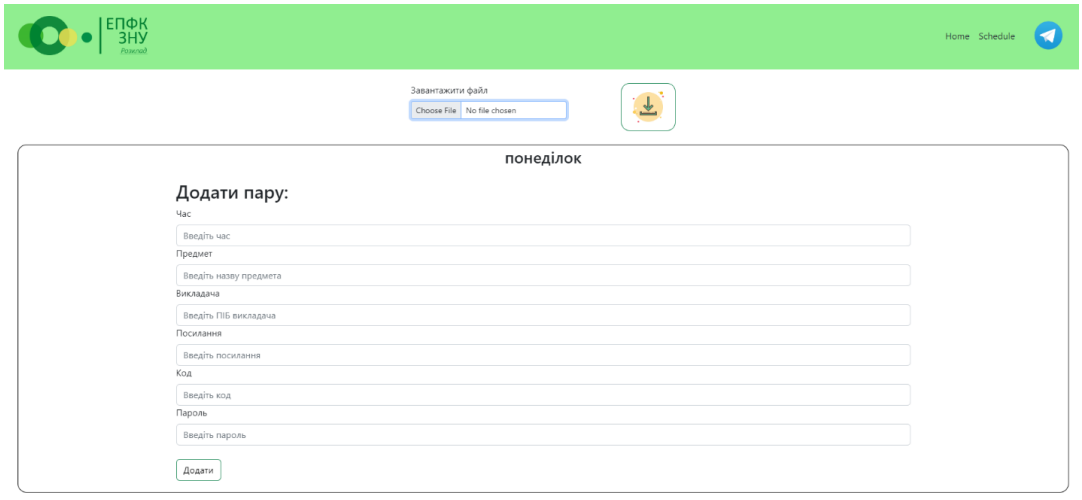


Рисунок 3.5 – Інтерфейс сторінки редагування

Була створена сторінка допомоги до користування телеграм ботом, зображена на рисунках 3.6 та 3.7. Телеграм бот приймає у себе документ з JSON файлом для обробки. У разі некоректного форматування, бот надасть повідомлення користувачу про «зламаний» файл.

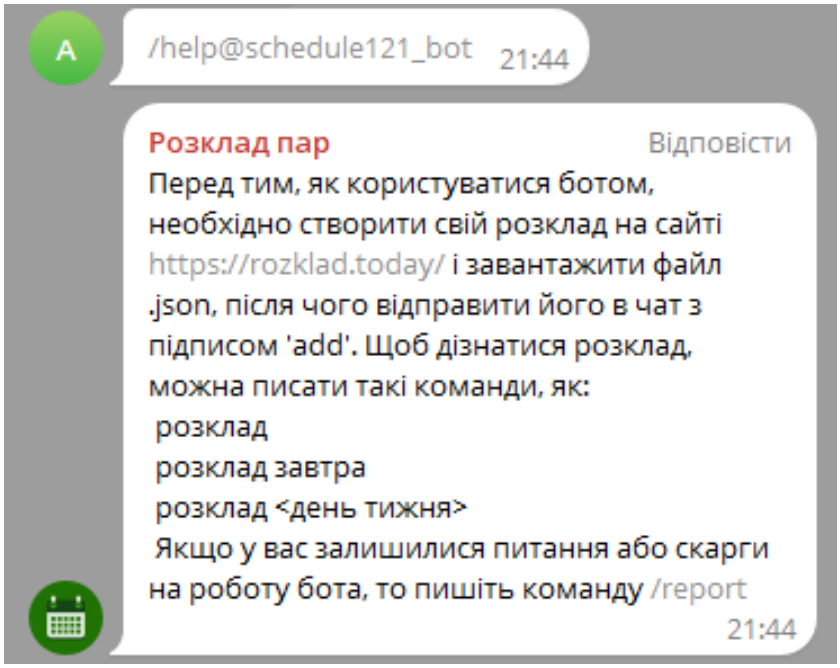


Рисунок 3.6 – Меню допомоги

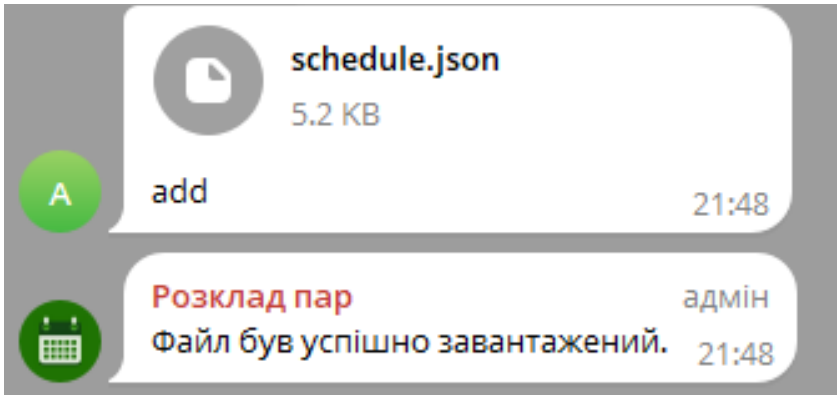


Рисунок 3.7– Меню завантаження файлу

Реалізація представлена на рисунку 3.8.

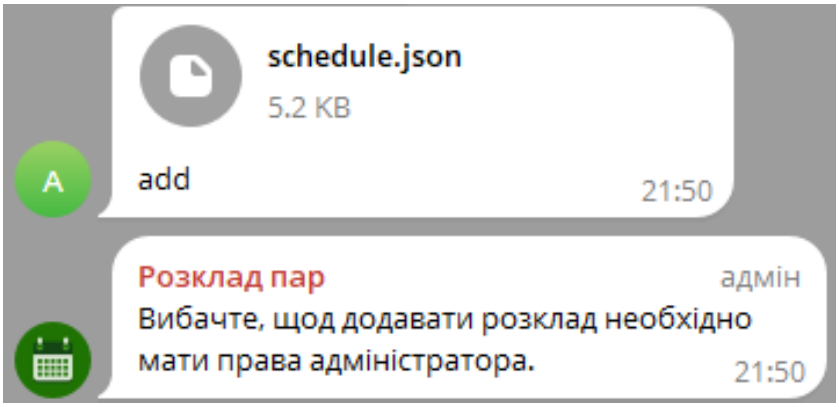


Рисунок 3.8 – Помилка через відсутність у користувача адміністраторських прав

Створено меню за допомогою телеграму, зображена на рисунках 3.9.

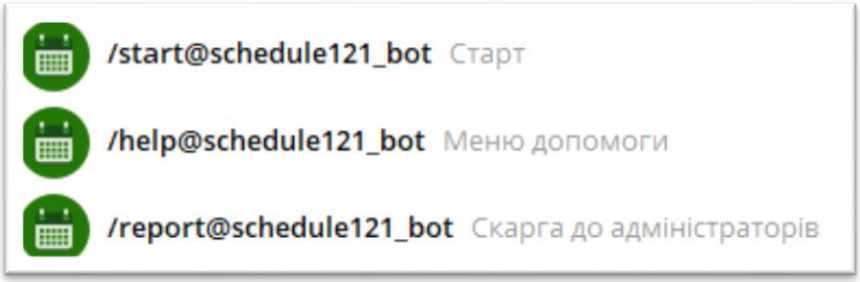


Рисунок 3.9 – Меню команд

3.3 Створення запитів

Форма для введення розкладу та його конструктор. Надана можливість редагування вже існуючого розкладу при його загрузці до сайту. Зображено на рисунку . Конструктор має дві кнопки для завантаження та вивантаження розкладу. Завантаження розкладу надає користувачу JSON файл який він надаватиме телеграм боту для підключення до чату. Вивантаження отримує JSON файл розкладу, який оброблює і у разі некоректного файлу надає користувачу повідомлення про «зламаний» файл.

Оформлення запиту на отримання розкладу, відбувається на прикладі,

зображеному на рисунку 3.10. Реалізована текстова перевірка та операнди на день тижня, дефолтне значення (без операндів) – сьогоднішній день тижня.

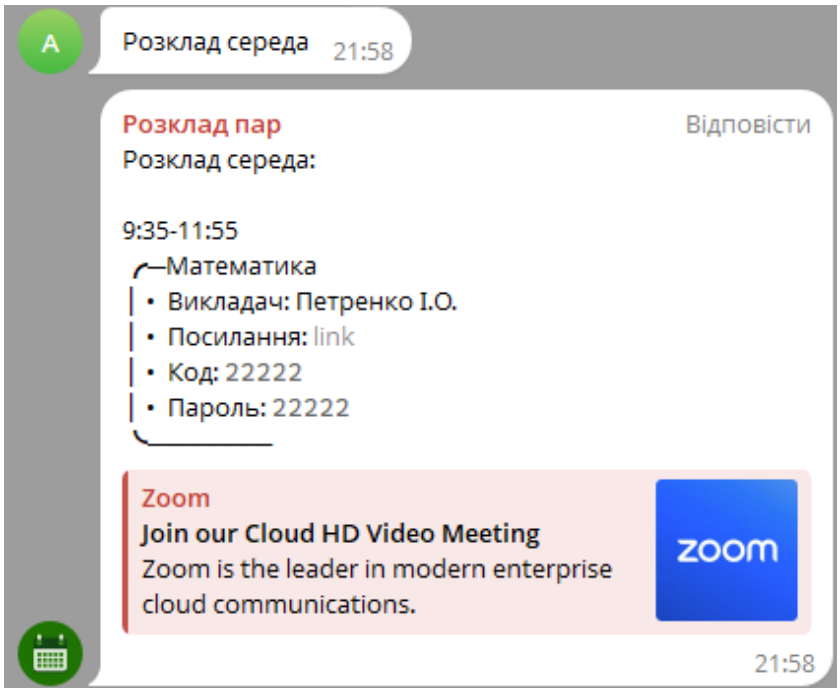


Рисунок 3.10 – Запит на отримання розкладу

Приклад відправки проблеми до розробників боту, зображений на рисунку 3.11.

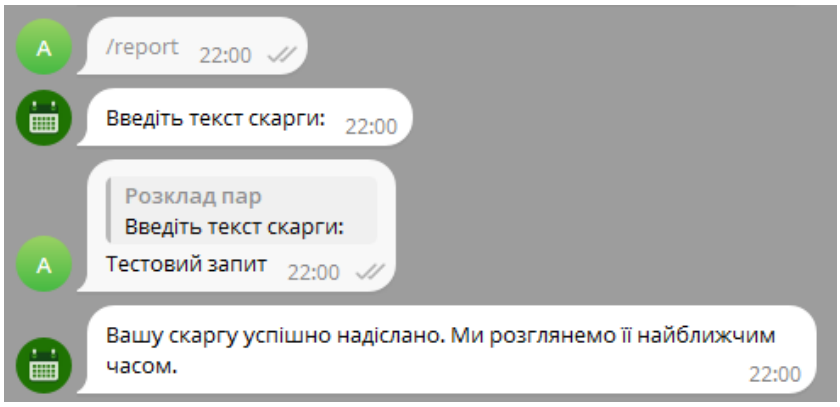


Рисунок 3.11 – Надсилання скарги

ВИСНОВКИ

У кваліфікаційній роботі було розглянуто теоретичні та практичні аспекти створення систем автоматизації освітнього процесу. Було досліджено сучасні вимоги до таких інформаційних систем, проаналізовано засоби для їх розробки та обґрунтовано вибір технологічного стеку. На основі цього було сформульовано технічне завдання та детальні технічні вимоги до розроблюваного ключового модуля – системи управління та користування розкладом навчального закладу. Була створена програмна реалізація даного модуля, що інтегрує Телеграм-бота для доступу користувачів та вебредактор для адміністрування.

У ході виконання кваліфікаційної роботи були виконані такі основні завдання:

1. Проаналізовано сучасний стан та значення автоматизації освітніх процесів, зокрема виявлено проблематику та обґрунтовано необхідність автоматизації управління навчальними розкладами.

2. Проведено огляд існуючих підходів та інструментальних засобів для автоматизації управління розкладами та забезпечення ефективної взаємодії з користувачами в освітньому середовищі.

3. Обґрунтовано вибір та детально описано інструментальні засоби, використані для розробки модуля управління розкладом, а саме: середовище виконання JavaScript – Node.js, бібліотека ReactJS для створення вебінтерфейсу, СУБД PostgreSQL для зберігання даних, та node-telegram-bot-api для розробки Телеграм-бота.

4. Розроблено комплексний проєкт модуля управління розкладом, що включає визначення архітектури системи, проєктування бази даних (ER-модель, реляційна модель), а також проєктування користувацьких інтерфейсів Телеграм-бота та вебредактора.

5. Створено базу даних в системі управління базами даних PostgreSQL на основі розробленої ER-моделі, що забезпечує ефективне зберігання та управління інформацією про розклади.

6. Реалізовано серверну частину модуля на платформі Node.js, яка забезпечує логіку роботи API для взаємодії з клієнтськими застосунками та функціонування Телеграм-бота.

7. Розроблено клієнтську частину у вигляді інтерактивного вебредактора розкладів з використанням бібліотеки ReactJS, що надає адміністраторам інструменти для управління розкладами.

8. Забезпечено функціонал Телеграм-бота для надання студентам та викладачам зручного доступу до актуального розкладу.

9. Проведено тестування функціональності розробленого модуля, включаючи перевірку коректності роботи його компонентів та їх взаємодії.

Розроблений модуль системи управління розкладом відповідає сформульованому технічному завданню та визначеним технічним вимогам, демонструючи практичну реалізацію підходу до автоматизації одного з ключових аспектів освітнього процесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Ботуз В.В. Переваги автоматизації навчального процесу за рахунок освітніх платформ та вебсервісів. Харків: НУ «ХПІ», 2025. 120 с.

2. Ільчук Н.М. Найпопулярніші освітні платформи для організації дистанційного навчання. URL: https://vseosvita.ua/blogs/naipopuliarnishi-osvitni-platformy-dlia-orhanizatsii-dystantsiinoho-navchannia-102669.html (дата звернення: 17.03.2025).

3. Мигасюк Р., Смотр О.О. Розробка telegram боту для автоматизації процесу отримання розкладу в навчальному закладі: Тези. 2021. URL: http://sci.ldubgd.edu.ua:8080/jspui/handle/123456789/9311 (дата звернення: 11.04.2025).

4. Резніков І. Хмарні бази даних. Детальна інструкція, як застосовувати сучасний IT-підхід. URL: https://dou.ua/forums/topic/43820/ (дата звернення: 27.05.2025).

5. Реляційні бази даних: Що це таке і як вони працюють. URL: https://surl.li/cemqtc (дата звернення: 27.05.2025).

6. Хостинги для телеграм бота. Як написати бот на Node.js та як встановити? URL: https://ukrline.com.ua/ua/blog/telegram-bot-07.php (дата звернення: 27.05.2025).

7. Що таке RDBMS (relational database management system / реляційна система управління базами даних)? URL: https://tseivo.com/b/memecode/t/orml9pedyg/shcho-take-rdbms-relational-database-management-system-reliatsiina-systema-upravlinnia-bazamy-danykh (дата звернення: 27.05.2025).

8. Most used web frameworks among developers 2024| Statista. URL: https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/ (дата звернення: 27.05.2025).

9. PostgreSQL. PostgreSQL. URL: https://www.postgresql.org (дата звернення: 27.05.2025).

10. Telegram Bot Platform. Telegram. URL: https://telegram.org/blog/bot-revolution (дата звернення: 27.05.2025).

Декларація

академічної доброчесності

здобувача освіти ВСП «Економіко-правничого фахового коледжу ЗНУ»

Я, Луценко Андрій Артемович, здобувач(-ка) освіти 4 курсу, спеціальності/освітньо-професійної програми Інженерія програмного забезпечення/Розробка програмного забезпечення, групи К121-21, адреса електронної пошти epk.12121lucenko@gmail.com;

- підтверджую, що написана мною кваліфікаційна робота на тему «Створення системи автоматизації освітнього процесу» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений/ознайомлена;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата _____ Підпис _____ Андрій ЛУЦЕНКО

Дата _____ Підпис _____ Роман КОТОВ