

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій
обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Горецький Олександр Вікторович
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.738.5:378.1

КВАЛІФІКАЦІЙНА РОБОТА

ВЕБ-ЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ РОЗКЛАДОМ
В ЗАКЛАДАХ ВИЩОЇ ОСВІТИ

(тема роботи)

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

О.В. Горецький

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи

Топольницький П.П.

(прізвище, ім'я, по батькові)

Кандидат технічних наук, доцент

(науковий ступінь, вчене звання)

Висновок кафедри _____
за результатами попереднього захисту: _____

Протокол засідання кафедри _____
№ _____ від « _____ » _____ 20____ р.

Завідувач кафедри _____

(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)
« _____ » _____ 20____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____
за шкалою ECTS _____
за національною шкалою _____

Секретар ЕК

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

АНОТАЦІЯ

Горецький О.В. Веб-застосунок для управління розкладом в закладах вищої освіти. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня магістра за спеціальністю 122 – комп'ютерні науки. – Поліський національний університет, Житомир, 2024.

Об'єктом розробки є веб-застосунок для управління розкладом в закладах вищої освіти. Метою розробки є створення веб-застосунку, який зможе оптимізувати та полегшити начальний процес у закладах вищої освіти. Додаток є чудовим рішенням для студентів та викладачів закладу вищої освіти, оскільки вони отримують автоматизований інструмент для швидкого управління, що дозволяє миттєво оброблювати великі обсяги даних.

Ключові слова: веб-застосунок, розклад, управління розкладом, заклад вищої освіти

SUMMARY

Goretskiy O. V. Web Application for Schedule Management in Higher Education Institutions. – Qualification work as a manuscript.

Qualification work for the degree of Master in Computer Science, specialty 122. – Polissia National University, Zhytomyr, 2024.

The object of development is a web application for schedule management in higher education institutions. The purpose of development is to create a web application that can optimize and facilitate the initial process in higher education institutions. The application is an excellent solution for students and teachers of higher education institutions, as they will receive an automated tool for quick management, allowing instant processing of large amounts of data.

Keywords: web application, schedule, schedule management, higher education institution

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Обґрунтування актуальності задачі	7
1.2 Огляд існуючих аналогів.....	8
1.3 Опис вимог до інформаційної системи.....	15
Висновки до розділу 1	17
РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ УПРАВЛІННЯ РОЗКЛАДОМ В ЗВО	18
2.1 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	18
2.2 Визначення варіантів використання та структури	20
Висновки до розділу 2	22
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ УПРАВЛІННЯ РОЗКЛАДОМ В ЗВО.....	23
3.1 Програмна реалізація веб-застосунку	23
3.2 Керівництво користувача веб-застосунку	27
Висновки до розділу 3	29
ВИСНОВКИ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	32

ВСТУП

Планування навчального процесу відіграє ключову роль у забезпеченні концентрації, результативності та успішності навчання студентів та роботи викладачів у закладах вищої освіти (ЗВО). Поширена та динамічна природа навчального процесу вимагає впровадження та вдосконалення сучасних технологій управління, де розклад є необхідною складовою, яка інтегрується з освітнім планом, науковим та творчим життям учасників навчання.

Виходячи з вищезазначеного, електронний розклад як інструмент для покращення комунікації і зменшення витрат часу на відновлення втраченої інформації про розклад може бути розглянутий як обов'язковий об'єкт, що є у кожному закладі вищої освіти.

Актуальність теми роботи. Оскільки задача управління розкладом складається з ряду факторів і обмежень, які складно врахувати людині (виходячи з кількості ітерацій, які застосовують популярні алгоритми), постає необхідність у розробці системи, яка мала б підтримувати всі основні обмеження на складання розкладу, а також, надавати можливість закладам з особливими обмеженнями щодо складання розкладів додавати власні модулі, та/або обмеження, які б застосовувались для гнучкого налаштування розкладів, що будуть згенеровані з використанням системи.

Хоча ряд систем, які задовольняють вищезазначеним вимогам уже існує, кожна із них має недоліки, які необхідно усунути для того, щоб ці системи можна було використати будь-яким закладом освіти без необхідності в ручному редагуванні отриманих розкладів, а також публікувати отримані розклади на власні онлайн-ресурси, щоб доступ до розкладів міг бути наданий користувачам (студентам, викладачам) у онлайн-режимі. Саме тому розробка веб-застосунків для управління розкладом занять в закладах вищої освіти (ЗВО) є актуальною та важливою задачею сьогодення.

Об'єктом дослідження є процес управління навчальним розкладом, а **предметом** – веб-застосунок для цієї мети.

Метою даної кваліфікаційної роботи є розробка веб-застосунку, який надає студентам та викладачам зручний та ефективний інструмент для управління розкладом занять.

Для досягнення мети необхідно розв'язати наступні **задачі**:

- провести аналіз предметної області;
- спроектувати веб-застосунок для управління розкладом занять у ЗВО;
- реалізувати веб-застосунок для управління розкладом занять у ЗВО.

У роботі використовуються такі **методи**, як аналіз літературних джерел та аналогів, проектування і реалізація веб-застосунку.

Практична значимість дослідження полягає у можливості покращення організації навчального процесу в закладах вищої освіти, забезпеченні зручності та ефективності користування розкладом занять для студентів та викладачів, а також у використанні сучасних технологій для підвищення якості освіти.

Структура й обсяг роботи. Кваліфікаційна робота складається з вступу, трьох розділів, висновків та списку використаних джерел, який включає 27 найменувань. Повний обсяг роботи складає 27 сторінок. Робота містить 8 рисунків та 17 додатків.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обґрунтування актуальності задачі

У сучасних закладах вищої освіти ефективне управління розкладом є ключовим елементом, що забезпечує організованість навчального процесу, оптимальне використання ресурсів та задоволення потреб як студентів, так і викладачів. З розвитком інформаційних технологій та зростанням кількості студентів виникає необхідність у створенні більш гнучких, надійних та автоматизованих систем для управління розкладом. Традиційні методи, що базуються на паперових документах або простих електронних таблицях, вже не відповідають вимогам сучасності. Це зумовлює актуальність розробки веб-застосунку для управління розкладом, який дозволяє автоматизувати і оптимізувати цей процес.

Традиційні методи управління розкладом, такі як паперові документи або електронні таблиці, мають низку недоліків, серед яких:

- зміни у розкладі можуть вимагати значних зусиль та часу для оновлення всіх документів та інформування зацікавлених осіб.
- ручне введення даних часто призводить до помилок, що можуть викликати плутанину та дезорганізацію.
- ручне управління розкладом ускладнює оптимальне використання аудиторій і викладацького складу та спричиняє наявність накладок.

Веб-застосунок для управління розкладом вирішує багато з вищезазначених проблем та надає низку переваг. Веб-застосунок дозволяє автоматизувати складання, коригування та поширення розкладу, що знижує ризик помилок та скорочує час на управління розкладом. Завдяки веб-інтерфейсу, користувачі можуть отримувати доступ до актуального розкладу з будь-якого пристрою, підключеного до Інтернету, у будь-який час, оскільки всі дані зберігаються у єдиній базі даних, що забезпечує їх цілісність та узгодженість. Застосунок також дозволяє ефективно планувати використання аудиторій та робочого часу викладачів, що

сприяє кращій організації навчального процесу. Веб-застосунок надає можливість інтеграції з іншими інформаційними системами закладу (наприклад, системами управління студентськими даними) полегшує обмін інформацією та підвищує ефективність управління.

Впровадження веб-застосунку для управління розкладом має важливе значення для закладів вищої освіти. Воно сприяє підвищенню ефективності організаційних процесів, покращенню якості освіти та задоволенню потреб усіх учасників навчального процесу. Студенти отримують можливість швидко та зручно дізнаватися про зміни у розкладі, а викладачі можуть ефективніше планувати свій робочий час. Адміністрація закладу, в свою чергу, отримує потужний інструмент для управління ресурсами та оптимізації навчального процесу.

Отже, актуальність розробки веб-застосунку для управління розкладом у закладах вищої освіти зумовлена необхідністю автоматизації, оптимізації та підвищення ефективності управління навчальним процесом. Веб-застосунок надає численні переваги порівняно з традиційними методами, сприяючи кращій організації роботи, зниженню ризику помилок та підвищенню задоволеності студентів та викладачів. Це робить його важливим інструментом для сучасних освітніх установ, які прагнуть підвищити якість своєї діяльності та відповідати вимогам часу.

1.2 Огляд існуючих аналогів

Станом на сьогодні в Україні функціонує близько трьохсот закладів вищої освіти. Кожен із цих закладів самостійно обирає спосіб відображення розкладу навчальних занять. Існує три основні методи відображення розкладу: електронні таблиці, друковані таблиці на ватмані та Автоматизовані системи управління навчальним закладом (АСУНЗ).

Перший метод передбачає використання програм типу Microsoft Excel

або Google Sheets для створення та управління розкладом. Він дозволяє зручно редагувати і зберігати інформацію, проте його обмеженням є складність у розповсюдженні та оновленні розкладу серед великої кількості користувачів.

Друковані таблиці на ватмані - це по суті роздруковані версії електронних таблиць, які розміщуються на інформаційних дошках у навчальних закладах. Такий метод має обмежену гнучкість, оскільки будь-які зміни потребують повторного друку і розміщення оновлених версій, що є трудомістким і неефективним.

Автоматизовані системи управління навчальним закладом (АСУНЗ) забезпечують автоматизацію процесу складання розкладу та його відображення на вебсайті навчального закладу або у мобільному застосунку. Вони дозволяють користувачам зручно переглядати розклад як студентів, так і викладачів, забезпечуючи актуальність та доступність інформації в режимі реального часу.

АСУНЗ мають значні переваги перед іншими методами, оскільки вони автоматизують і спрощують процес складання розкладу, забезпечують зручний доступ до нього та можливість швидкого внесення змін. Вони дозволяють навчальним закладам ефективніше використовувати свої ресурси і краще задовольняти потреби студентів та викладачів.

Однак, не всі заклади вищої освіти в Україні використовують автоматизовані системи управління розкладом. Так наприклад, Запорізький національний університет публікує Excel таблиці з розкладом на сайті [3]. Студент має завантажити потрібний файл з розкладом та відкрити через програму зчитування таблиць формату *.xls*. На рисунку 1.1 приведений знімок розділу з розкладом на їх офіційному сайті. Сам розклад має структуру, яка показана на рисунку 1.2. На рисунку приведена його частина, бо вся таблиця не поміститься у формат Word документу. Зверху розташовані коди груп, зліва день тижня та номер пари. На їх перетині знаходяться самі клітини з парами, в яких зазначені ім'я, тип пари, та ПІБ викладача.

Ще один вищий навчальний заклад – Київський Європейський Університет [4]. Його система відображення розкладу також полягає в публікації Excel таблиць на офіційному сайті з допомогою Google Docs. Сам вид таблиці приведений на

рисунку 1.3. Це розклад занять першого курсу економічного факультету.

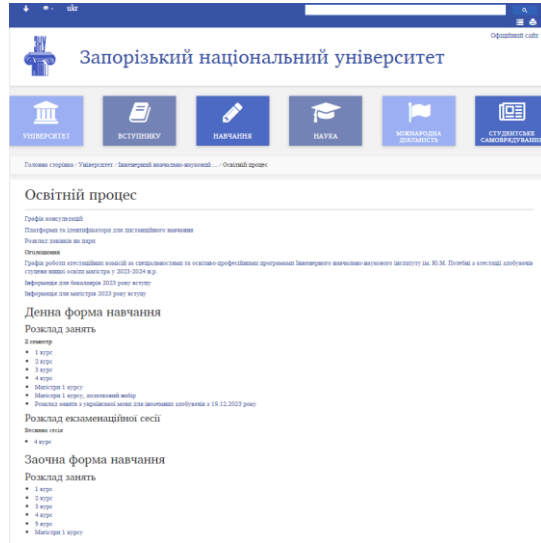


Рисунок. 1.1 – Сторінка з розкладом занять на офіційному сайті ЗНУ

		6.1923-мбг			6.1923-пнб-1			6.1923-пнб-2		
ІНЖЕНЕРСТВО	1									
	2	Будівельна та комп'ютерна графіка	лекція	Гребенюк І.В.	Будівельна та комп'ютерна графіка	лекція	Гребенюк І.В.	Будівельна та комп'ютерна графіка	лекція	Гребенюк І.В.
		Вступ до будівельної справи	лекція	Демченко Н.О.	Вступ до будівельної справи	лекція	Демченко Н.О.	Вступ до будівельної справи	лекція	Демченко Н.О.
	3	Інженерна геодезія	лекція	Салченко Р.В.	Інженерна геодезія	лекція	Салченко Р.В.	Інженерна геодезія	лекція	Салченко Р.В.
		Будівельна техніка і транспорт	лаб.р.	Пастухова С.В.	Інженерна геодезія	лаб.р.	Салченко Р.В.	Інженерна геодезія	лаб.р.	Арутюнян С.Е.
4	Будівельна техніка і транспорт	лекція	Пастухова С.В.	Будівельна техніка і транспорт	лекція	Пастухова С.В.	Будівельна техніка і транспорт	лекція	Пастухова С.В.	

Рисунок. 1.2 – Частина розкладу занять інженерного факультету ЗНУ

дні	час заняття	група 111 "Економіка"			група 112 "Маркетинг: Інтернет-маркетинг"			група 113 "Фінанси, банківська справа та страхування"			група 114 "Окремі галузі економіки"		
		лекція	лаб.	Ауд.	лекція	лаб.	Ауд.	лекція	лаб.	Ауд.	лекція	лаб.	Ауд.
Понеділок	10:15 - 11:15	Інформаційні технології (робота в google classroom)		classroom	Інформаційні технології (робота в google classroom)		classroom	Інформаційні технології (робота в google classroom)		classroom	Інформаційні технології (робота в google classroom)		classroom
	11:45 - 12:45	Управління внутрішніми справами. Дідуцька С.В.	л	302	Маркетинг: відносини з клієнтами. Філіппов В.В.	пр	409	Гроші та кредити. Колотилова Н.М.	пр	google meet	Маркетинг: лог. Сабуров Р.Г.	л	
	13:00 - 14:00	Організація операційної діяльності підприємства. Дідуцька С.В.	л	302	Венчурна жетя та основні фінансові показники. Романовська О.Ю.	пр	318	Фінанси: основні поняття (робота в google classroom)		classroom	Основні поняття та операційні показники. Мельничук В.В.	л	
	14:15 - 15:15												
П'ятниця	09:00 - 10:00												
	10:15 - 11:15	Венчурна жетя та основні фінансові показники (робота в google classroom)		classroom	Венчурна жетя та основні фінансові показники (робота в google classroom)		classroom	Венчурна жетя та основні фінансові показники (робота в google classroom)		classroom	Венчурна жетя та основні фінансові показники (робота в google classroom)		classroom
	11:45 - 12:45	Інформаційні технології	л	306	Інформаційні технології	л	306	Інформаційні технології	л	306	Інформаційні технології	л	306
	13:00 - 14:00	Інноваційне підприємство та управління стартап проектами. Калашнікова О.О.	л	306	Інноваційне підприємство та управління стартап проектами. Калашнікова О.О.	л	306	Інноваційне підприємство та управління стартап проектами. Калашнікова О.О.	л	306	Інноваційне підприємство та управління стартап проектами. Калашнікова О.О.	л	306
14:15 - 15:15													
Субота	09:00 - 10:00												
	10:15 - 11:15	Інноваційне підприємство та управління стартап проектами (робота в google classroom)		classroom	Інноваційне підприємство та управління стартап проектами (робота в google classroom)		classroom	Інноваційне підприємство та управління стартап проектами (робота в google classroom)		classroom	Інноваційне підприємство та управління стартап проектами (робота в google classroom)		classroom

Рисунок. 1.3 – Приклад розкладу Київського Європейського Університету

Одним з найбільш популярних програмних рішень, які призначені для автоматизації процесу складання розкладу є «Деканат» від ПП "Політек-СОФТ". Дана компанія працює над створенням та впровадженням програмних продуктів

для вищих навчальних закладів України з 1998 року. За цей період компанія співпрацювала з понад 250 вищих навчальних закладів від I до IV рівнів акредитації. [13]

Деканат - пакет програм, що призначений для автоматизації планування та обліку навчального процесу, тобто окрім безпосередньо роботи з розкладом навчальних занять, цей пакет програм дозволяє формувати дані щодо структури навчального процесу, всіх викладачів та їх планового навантаження, розклад їх роботи, а також наявності корпусів та аудиторій, їх заповнення.

Проте не зважаючи на такі очевидні переваги, «Деканат» страждає від деяких технічних недоліків, зокрема застарілого та непрактичного інтерфейсу. Тобто, інтерфейс є неефективним з точки зору розміщення інтерактивних елементів, що ускладнює користування платформою. В надмірно насиченому інтерфейсі може бути важко орієнтуватись, оскільки всі необхідні функції і опції можуть бути розташовані без зрозумілої логіки.

Такий нагромаджений інтерфейс може призводити до зниження продуктивності користувачів, адже вони витрачають більше часу на пошук необхідних функцій і команд.

Такі проблеми інтерфейсу можуть бути критичними для використання платформи "Деканат" у вищих навчальних закладах, де вимагається зручний, інтуїтивно зрозумілий інтерфейс для різних категорій користувачів — від адміністраторів до викладачів і студентів.

Рисунок 1.4 демонструє інтерфейс редактора розкладу програми «Деканат».

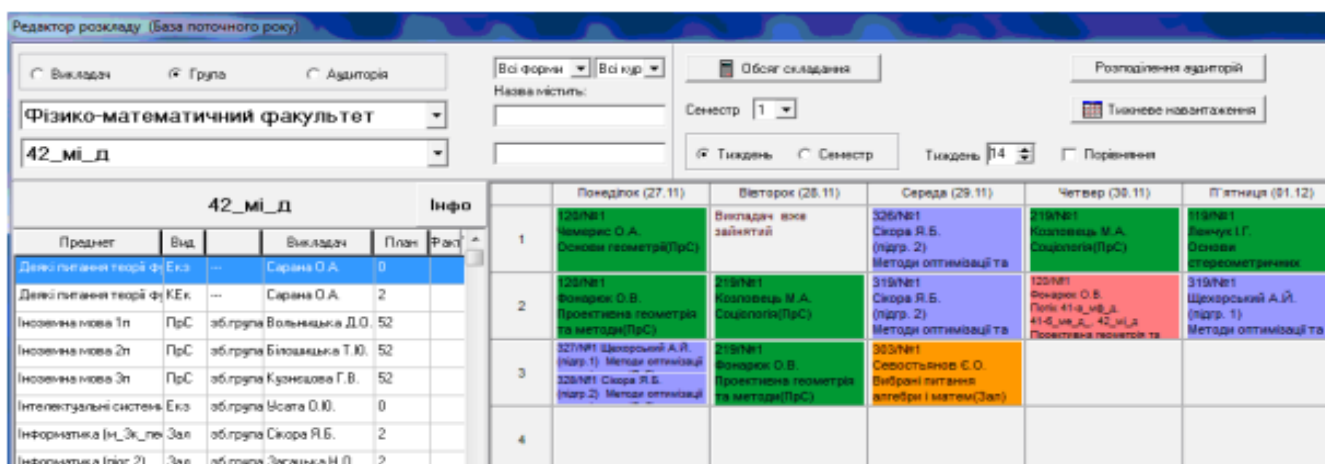


Рисунок 1.4 робоче вікно «Редактор розкладу» програми «Деканат» [13]

Ще однією платформою для керування розкладом є ЛКЛАУД. (рис.1.5)

ЛКЛАУД - це інноваційна хмарна CRM-система, спеціально розроблена для вищих навчальних закладів, яка забезпечує комплекс функціональних можливостей для оптимізації навчального процесу. [14]

Ця платформа дозволяє автоматизувати створення та коригування розкладу занять для груп і викладачів, що значно спрощує процес управління навчальними графіками. Крім того, вона забезпечує централізований доступ до навчально-методичних матеріалів, що спрощує планування і проведення занять. [14]

Однією з головних переваг є спрощення та автоматизація адміністративних процесів, централізація і стандартизація інформації, що призводить до підвищення прозорості та оперативності управління навчальним процесом. Проте є деякі недоліки, такі як можливість технічних збоїв, залежність від доступу до Інтернету, відсутність можливості копіювання розкладу, що в свою чергу зобов'язує створювати розклад занять на кожен день навчального року в ручному режимі, не зручний інтерфейс редактора розкладу.

У підсумку, ЛКЛАУД виступає як потужний інструмент для оптимізації управління навчальним процесом у вищих навчальних закладах, сприяючи їхній інформатизації та адаптації до вимог сучасного цифрового середовища. Однак зазначені технічні недоліки не дозволяють даній платформі повністю використовувати свій потенціал і можуть вплинути на надійність та ефективність її функціонування в умовах високих навантажень і вимог користувачів.

Рисунок 1.5 робоче вікно «Редактор розкладу» програми «ЛКЛАУД» [14]

Далі розглянемо використання системи АСУНЗ на прикладі Національного медичного університету імені О.О. Богомольця [5]. В лівому меню можна вибрати тип розкладу – студента, викладача, аудиторії, тощо. На рисунку 1.6 можна побачити приклад відображення розкладу студентів певного курсу та певної групи, завдяки цієї системи. Також опціонально можна вибрати діапазон дат та інші параметри.

Остання система, яку можна привести – це Telegram чат-бот для перегляду розкладу від Національного Університету «Полтавська Політехніка Імені Юрія Кондратюка» [6]. Приклад роботи з ним видно на рисунку 1.7. Бот надає можливість перегляду розкладу різних груп. Також він має цікаву функцію нагадування про початок пари заздалегідь.

Також, багато навчальних закладів мають систему авторизації, тому оцінити

їх систему публікації розкладу сторонньому спостерігачу неможливо.

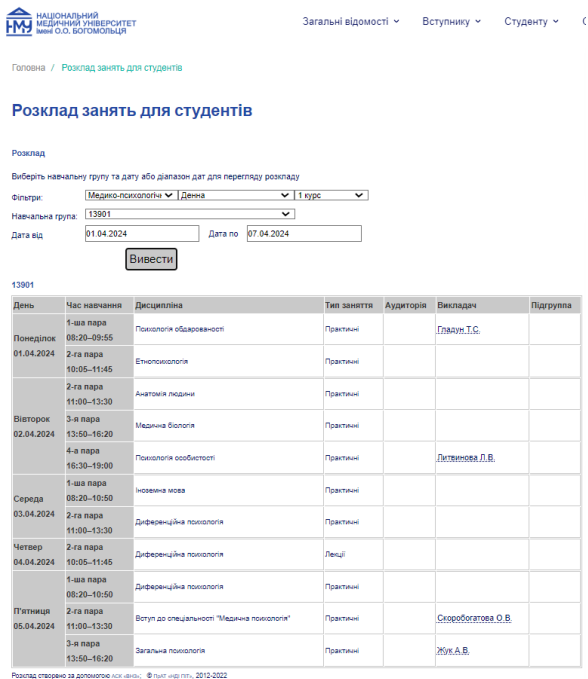


Рисунок. 1.6 – Приклад відображення розкладу студентів

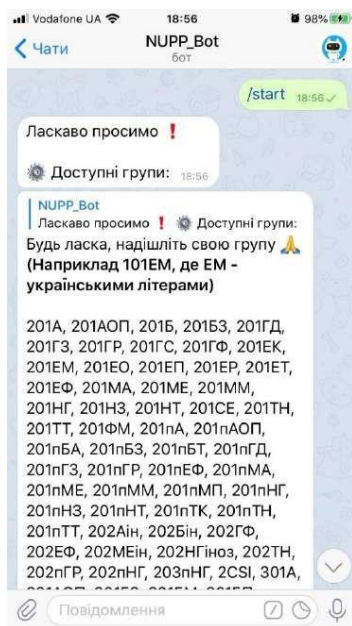


Рисунок 1.7 – Чат-бот від НУПП

На підставі проведеного аналізу можна зробити висновок, що існуючі інформаційні системи мають свої переваги, однак вони також супроводжуються певними недоліками, які потребують уваги та подальшого вдосконалення.

З урахуванням цього, виникає актуальна потреба у розробці нової веб-орієнтованої системи для управління розкладом. Нова система має на меті вдосконалити доступність та ефективність взаємодії між різними елементами навчального процесу, забезпечуючи зручний і сучасний інтерфейс.

1.3 Опис вимог до інформаційної системи

Проаналізувавши існуючі аналоги, можна побачити, що проблема недостатньої автоматизації та не повної відповідності систем до базових вимог присутня і не дозволяє організувати начальний процес в цілому.

На основі цього і буде розроблено проєкт і створено сервіс «Веб-застосунок для управління розкладом в закладах вищої освіти», призначенням якого є:

1. Покращення досвіду користування при перегляді розкладу користувачами. Веб-застосунок забезпечить інтуїтивно зрозумілий та зручний інтерфейс, що дозволить студентам та викладачам легко і швидко знаходити потрібну інформацію про розклад.

2. Полегшення роботи адміністраторів із розкладом, його заповненням та редагуванням. Автоматизовані інструменти для створення та коригування розкладу зменшать навантаження на адміністративний персонал та допоможуть швидко та ефективно створювати розклад, враховуючи всі необхідні обмеження та вимоги.

3. Веб-застосунок буде доступний з будь-якого пристрою, підключеного до Інтернету, включаючи комп'ютери, планшети та смартфони, що забезпечить максимальну доступність та зручність для користувачів.

Отже, основною метою кваліфікаційної роботи є створення веб-застосунку для управління розкладом у закладах вищої освіти, який включатиме основні переваги описаних існуючих аналогів. Процес реалізації цього застосунку охоплює такі чотири етапи:

1. Проєктування веб-застосунку;
2. Розробка зручного інтерфейсу та системи навігації;

3. Розробка функціоналу редагування розкладу адміністраторами системи;
4. Розробка способів відображення розкладу для користувачів.

Проектування веб-застосунку передбачає розробку загальної архітектури системи, включаючи визначення основних компонентів, їх взаємодії та технічних вимог.

На етапі розробки інтерфейсу та навігаційної системи буде створено зручний та інтуїтивно зрозумілий інтерфейс, що забезпечить користувачам легкий доступ до всіх функцій застосунку. Навігаційна система буде логічною і дозволить швидко переміщатися між різними розділами веб-застосунку.

Розробка базового функціоналу редагування розкладу адміністраторами системи. Тут йдеться про створення інструментів для адміністраторів, що дозволяють ефективно керувати та редагувати розклад. Це включає можливість додавання, видалення та зміни часу проведення занять, а також врахування специфічних потреб і обмежень.

На останньому етапі буде відбуватись розробка відображення розкладу для користувачів, тобто буде здійснюватись реалізація функцій, що забезпечать зручний перегляд розкладу для кінцевих користувачів, таких як студенти та викладачі. Це включає візуалізацію розкладу у різних форматах, можливість пошуку та фільтрації занять.

Таким чином, створення веб-застосунку для управління розкладом передбачає комплексний підхід, що включає проектування, розробку інтерфейсу, реалізацію адміністративних функцій та забезпечення зручного відображення інформації для користувачів. Ця платформа заснована на простій концепції і охоплює всі основні аспекти процесу створення розкладу. Проте вона має значний потенціал розвинути в повноцінну систему.

Висновки до розділу 1

У даному розділі було проведено аналіз предметної області, зокрема, зроблено обґрунтування актуальності задачі, проведено огляд існуючих аналогів, та складено опис вимог до розробки.

Актуальність проблеми управління розкладом у ЗВО не підлягає сумніву. Розклад занять є важливою складовою навчального процесу, а ефективне його управління має ключове значення для якості навчання та організації робочого часу.

Аналіз існуючих аналогів показав, що хоча деякі ЗВО використовують сучасні функціональні системи, багато з них залишаються відсталими у використанні застарілих або незручних методів.

Розробка веб-застосунку для управління розкладом у ЗВО є актуальною та перспективною задачею. Оптимізація розкладу через використання веб-застосунку може мати значні переваги, такі як зручність для користувачів, оптимізація робочого часу, сповіщення та нагадування, а також можливість аналізу даних для покращення навчального процесу.

Отже, подальше дослідження та розробка веб-застосунку для управління розкладом у ЗВО є важливим напрямом для покращення якості навчання та організації робочого процесу.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ УПРАВЛІННЯ РОЗКЛАДОМ В ЗВО

2.1 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для розробки програмного модуля, відповідального за управління розкладом в закладах вищої освіти, для досягнення максимальної ефективності та успішної реалізації проекту буде використано наступний інструментарій:

Система управління базами даних (СУБД) PostgreSQL: для зберігання та ефективного оброблення інформації про розклади, викладачів, студентів та аудиторії.

Мова програмування: TypeScript як для фронтенд-розробки, так і для бекенд-розробки, що забезпечить надійну та продуктивну роботу системи за рахунок статичної типізації, яка відсутня в JavaScript, це дозволяє виявляти помилки на етапі компіляції, а не під час виконання. Що сприяє підвищенню якості коду і зменшенню кількості багів.

Фреймворки та бібліотеки: React для побудови інтерактивного користувацького інтерфейсу, забезпечення швидкої і ефективної роботи з компонентами; MUI (Material-UI): для створення стильного і послідовного користувацького інтерфейсу з готовими компонентами; Redux для управління станом додатка, що спрощує передавання даних між компонентами і покращує передбачуваність поведінки додатка; Nestjs для розробки серверної частини, забезпечуючи модульну архітектуру, легку масштабованість та інтеграцію з іншими інструментами і бібліотеками.

Інструменти для тестування та створення документації API: Swagger для інтерактивного доступу до бази даних у реальному часі через його візуальний інтерфейс, автоматизації процесу створення документації для API та тестування API з інтерфейсу Swagger.

Хмарні сервіси: для застосування СУБД PostgreSQL була розгорнута хмарна

база даних ElephantSQL, її спеціалізація полягає в підтримці PostgreSQL. Це рішення дозволяє створювати та керувати базами даних PostgreSQL без потреби в установці та налаштуванні на локальних серверах. Завдяки цьому, можна почати розробку проекту, обходячи складнощі налаштування бази даних.

Системи контролю версій: Git для відстеження змін у коді та координації роботи команди розробників.

Редактори коду: Visual Studio Code для зручного написання, відлагодження та підтримки кодової бази проекту. Програма VS Code підтримує, всі необхідні для розробки мови: TypeScript, JavaScript, JSON, HTML, CSS.

Розроблено структуру бази даних для інформаційної системи, яка представлена на додатку В. Основні таблиці описано нижче:

- таблиця «користувачі»: зберігає дані про користувачів системи, включаючи ім'я, адресу електронної пошти, рівень доступу та пароль в захешованому вигляді.

- таблиця «категорії груп» або ж іншими словами факультети: зберігає інформацію про існуючі факультети навчального закладу, включає такі поля як назва факультету та список груп.

- таблиця «групи»: зберігає дані про навчальні групи, включаючи назву групи, курс навчання, рік вступу, кількість студентів, форма навчання (денна або заочна), навантаження та ID факультету до якого група відноситься.

- таблиця «навантаження групи»: відображає інформацію про розподіл навчального навантаження, тобто які дисципліни будуть вивчати студенти групи, дана таблиця включає такі поля: назва дисципліни, семестр в якому дисципліна вивчається, тип навантаження (лекції, практичні заняття, лабораторні, семінари або екзамен), години, кількість студентів, що відвідують дисципліну, ID групи та ID викладача, який буде вести дисципліну.

- таблиця «категорії викладачів»: зберігає інформацію про кафедри навчального закладу, включає поля: назва кафедри та список викладачів кафедри.

- таблиця «викладачі»: містить інформацію про викладачів, включаючи ПІБ та ID кафедри, до якої викладач відноситься;

– таблиця «категорії аудиторій»: охоплює інформацію про навчальні корпуси освітнього закладу, дана таблиця має такі поля як назва та список аудиторій.

– таблиця «аудиторії»: містить інформацію про аудиторії, зокрема їх номери, кількість місць та ID корпусу до якого вони відносяться.

– таблиця «розклад уроків»: зберігає дані щодо проведення уроків, включаючи назву дисципліни, дату проведення, номер уроку, семестер, кількість студентів, години тип навантаження, ID групи, аудиторії та викладача.

– таблиця «розклад уроків»: берігач налаштування системи, що використовуються для управління її функціональністю, а саме управління користувачами системи, графік освітнього процесу та розклад дзвінків.

2.2 Визначення варіантів використання та структури

Розробка UML-діаграм є необхідною складовою процесу розробки програмного забезпечення, оскільки UML надає стандартну мову моделювання для візуалізації, специфікації, конструювання та документування складних систем. Цей огляд допомагає покращити якість та ефективність розробки, забезпечує узгодженість у проєкті та зменшує ризик виникнення помилок під час розробки системи.

UML діаграма прецедентів (додаток А) відображає користувачів системи та можливі дії цих користувачів. Вона демонструє, що система складається з двох компонентів: програмного модуля та акторів (супер адміністратор, адміністратор та не авторизований користувач).

Розглянемо акторів більш детально:

– роль адміністратор передбачає можливість авторизації в системі, створення всіх базових елементів навчального процесу, таких як: навчальні групи та їх навантаження, викладачі, аудиторії та розклад занять.

– роль супер адміністратора включає в себе всі можливості адміністратора, а також функціонал, що пов'язаний зі створенням нових адміністраторів системи,

встановлення термінів навчання (початок та кінець кожного семестру) та розклад дзвінків.

– роль користувач передбачає лише перегляд розкладу, до таких користувачів відносяться викладачі та студенти, які є кінцевими споживачами веб-орієнтованого застосунку.

Для відображення послідовності дій, які відбуваються в процесі складання розкладу створено діаграму послідовності (Додаток Б). Дана діграма складається з 6 дій:

1. Адміністратор авторизується на платформі, під час авторизації адміністратор повинен надати свої особисті дані, такі як адреса електронної пошти та пароль, що вказувались при реєстрації. Реєстрація нових користувачів може здійснюватись лише авторизованими адміністраторами з сторінки налаштувань веб-застосунку;

2. Перед складанням розкладу потрібно обов'язково заповнити базову інформацію навчальні групи, викладачів та аудиторії;

3. Навчальне навантаження може бути сформоване лише після створення навчальної групи, оскільки кожна дисципліна прив'язується до конкретної групи;

4. Далі потрібно вказати які педагоги які навчальні дисципліни будуть викладати, це можна зробити на сторінці «Розподіл навантаження»;

5. Після виконання всіх зазначених кроків, можна приступати до створення розкладу. Системою передбачено 2 варіанти складання розкладу: в ручному режимі та в режимі копіювання вже наявного розкладу;

6. Після складання розкладу користувачі можуть бути ознайомлені з ним на відповідній сторінці.

Така структура розробленої системи забезпечує ефективний процес управління розкладом в закладах вищої освіти, що відповідає поставленим завданням та враховує потреби студентів та викладачів.

Загалом, розроблюваний застосунок є не лише корисним інструментом для управління розкладом, але й потужним засобом підвищення якості освіти та доступності до навчальних ресурсів.

Висновки до розділу 2

Даний розділ присвячений інструментальним засобам та вимогам до апаратного забезпечення, а також визначенню варіантів використання та структури в розроблювальному продукті «Веб-застосунок для управління розкладом в закладах вищої освіти».

Серед зазначених вище інструментів були описані такі, як TypeScript, React, Redux, Material-UI, Nestjs, PostgreSQL, Swagger, ElephantSQL та VS Code. Кожен інструмент було детально проаналізовано та описано, відзначено особливості кожного, переваги та недоліки. Обрані інструменти та технології було обрані попередньо вивчивши доцільність використання, актуальність інструментів, зрозумілість та легкість експлуатації, аби в майбутньому не виникало складнощів у налаштуванні та відлагодженні.

Розроблено структуру бази даних, яка відповідає поставленим вимогам. Структура бази даних включає таблиці для зберігання інформації про користувачів, факультети, групи, навчальне навантаження, кафедри, викладачів, аудиторії та розклад уроків.

З метою покращення якості та ефективності розробки та зменшення ризику виникнення помилок було розроблено UML діаграми прецедентів та послідовності. Перша відображає акторів та прецедентів (варіантів використання веб-застосунку) обмежених системою, відношення серед прецедентами, та відношення між акторами. На другій зображено процес складання розкладу, діаграма включає послідовні дії від авторизації до заповнення базової інформації та створення розкладу, з можливістю ручного складання або копіювання вже наявного розкладу.

Загалом, розроблений застосунок є ефективним інструментом для управління розкладом у закладах вищої освіти, що підвищує якість освіти та доступність навчальних ресурсів.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ УПРАВЛІННЯ РОЗКЛАДОМ В ЗВО

3.1 Програмна реалізація веб-застосунку

Система побудована з використанням архітектури клієнт-сервер, що забезпечує не лише масштабованість і високу продуктивність, а також гнучкість і надійність в роботі з різними типами запитів і обміном даними. Ця архітектура дозволяє системі ефективно пристосовуватися до зростаючого навантаження та забезпечує оптимальне використання ресурсів сервера, що є критично важливим для забезпечення стабільної роботи і високої відповідності.

Для реалізації клієнтської частини було створено проєкт React за допомогою інструмента збірки фронтенда Vite, який забезпечує швидке оточення розробки. Vite дозволяє створити новий проєкт на основі шаблонів популярних фреймворків, в тому числі React. Лістинг коду наведено у додатку Р. Організація структури проєкту представлена у додатку Т.

Спочатку було проведено проєктування інтерфейсу користувача, для цього використовувався онлайн-сервіс Figma, яка дозволяє створювати макети інтерфейсів. Після завершення дизайну розпочалася розробка компонентів інтерфейсу. Використовувалися бібліотеки ReactJS та MUI для ефективної реалізації різноманітних компонентів інтерфейсу. Веб-застосунок має декілька сторінок, кожна з яких буде розглянута далі.

Сторінки GroupsPage, AuditoriesPage та TeachersPage мають досить схожу структуру інтерфейсу та функціональність. На сторінці GroupsPage представлений інтерфейс для керування факультетами та групами. Компонент має стани для управління взаємодією з користувачем, включаючи список факультетів та таблицю груп для активної категорії. Кожен факультет можна вибрати для перегляду пов'язаних груп, а таблиця груп показує інформацію та доступні дії для кожної групи.

На сторінці `AuditoriesPage` доступний інтерфейс для управління аудиторіями та корпусами. Використовується хук `useEffect` для отримання списку корпусів при завантаженні компонента. Аудиторії групуються за корпусами і відображаються у вигляді аккордеона. Корпуси та аудиторії можна додавати, редагувати або видаляти через відповідні функції, які передаються як пропси у компонент.

На сторінці `TeachersPage` впроваджено інтерфейс для управління викладачами та їх кафедрами. Використовується `Redux` для отримання стану, що включає дані про категорії викладачів та їх статус завантаження. Інтерфейс дозволяє додавати нових викладачів, редагувати існуючих та управляти категоріями за допомогою модальних вікон і форм.

На сторінці `LoadPage` реалізовано інтерфейс для управління навчальним навантаженням груп. Інтерфейс сторінки включає список факультетів, кожен з яких містить свій список груп, що розкривається при кліку. Також присутні дві панелі для додавання навантаження: ліва панель для вибору групи та права панель для відображення та редагування навантаження обраної групи. Для додавання нового навантаження використовується компонент `CreateLessonForm`. Інтерфейс сторінки дозволяє користувачеві зручно керувати навчальним навантаженням груп, додавати нові уроки та редагувати існуючі, а також обирати групи для перегляду їх навантаження.

На сторінці `DistributionPage` реалізовано розподіл навчального навантаження між групами та викладачами. Інтерфейс сторінки складається з трьох основних частин: В першій частині є список груп, який можна вибрати за допомогою компонента `SelectGroupModal`. Після вибору групи відображається інформація про неї; якщо група не обрана, користувачеві пропонується обрати її.

Друга частина містить таблицю розподілу навантаження для обраної групи, що відображається через компонент `DistributionLessonsTable`. При завантаженні даних показується спінер, а якщо дані відсутні, відображається пуста карточка.

Третя частина показує список викладачів для розподілу навантаження. При завантаженні даних також відображається спінер, і якщо дані відсутні, показується пуста карточка.

Цей інтерфейс дозволяє користувачам зручно вибирати групи, переглядати їх навантаження та розподіляти його між викладачами, спрощуючи управління навчальним процесом у навчальному закладі.

Компонент `TimetablePage` відображає сторінку редактора розкладу. Компонент використовує хук `useEffect` для автоматизованого отримання даних з сервера при завантаженні компонента. Він здійснює HTTP GET-запит за допомогою бібліотеки `Axios`, отримує відповідь від сервера і обробляє її для створення масиву об'єктів. Ці дані використовуються для відображення розкладу занять. Загалом, `TimetablePage` забезпечує повний функціонал управління розкладом занять, здійснюючи взаємодію з користувачем через різні інтерактивні елементи та використання власних станів для зберігання і обробки даних.

Наступним кроком була розробка маршрутизації. Маршрутизація в React забезпечує навігацію та управління відображенням різних компонентів в залежності від URL. Для цього використовується бібліотека `react-router-dom`, яка надає потужні інструменти для створення динамічних односторінкових додатків (Single Page Applications, SPA).

Маршрутизація в React з використанням `react-router-dom` базується на кількох ключових компонентах. Головним з них є `BrowserRouter`, який встановлює контекст маршрутизації для додатку. Компонент `Route` визначає, які компоненти потрібно відображати в залежності від поточного URL, використовуючи атрибути `path` та `element`. Для навігації між сторінками використовується компонент `Link`, який дозволяє здійснювати переходи без перезавантаження сторінки.

```
return (
  <BrowserRouter>
    <Routes>
      <Route element={MainLayout} />
      <Route element={GroupsPage} path="/groups" />
      <Route element={FullGroupPage} path="/groups/:id" />
      <Route element={FullGroupPage} path="/groups/create/:categoryId" />
      <Route element={TeachersPage} path="/teachers" />
      <Route element={AuditoriesPage} path="/auditories" />
      <Route element={LoadPage} path="/load" />
      <Route element={DistributionPage} path="/distribution" />
      <Route element={TimetablePage} path="/timetable" />
      <Route element={SettingsPage} path="/settings" />
    </Routes>
    <Route element={MinimalLayout} />
    <Route element={DefaultPage} path="/" />
    <Route element={AuthPage} path="/auth" />
    <Route element={ViewTimetablePage} path="/view-schedule" />
  </Routes>
</BrowserRouter>
)
```

Рисунок 3.1 – реалізація маршрутизації за допомогою `react-router-dom`

Для реалізації серверної частини було створено проєкт NestJS. Лістинг коду наведено у додатку С. Організація структури проєкту представлена у додатку У. Цей процес є досить структурованим та включає кілька ключових кроків.

Першим кроком була інсталяція NestJS CLI за допомогою npm (Node Package Manager). CLI спрощує створення нових проєктів і генерацію компонентів. Після успішної інсталяції був створений новий проєкт командою `nest new webSchedule`. Ця команда автоматично створила структуру проєкту NestJS з усіма необхідними файли і каталогами.

Стандартна структура проєкту містить папку `src`, де зазвичай знаходяться вихідні файли коду, `test` для тестів, а також `node_modules` для залежностей npm. У `src` знаходяться основні елементи проєкту, такі як модулі, контролери, сервіси і провайдери.

Модулі є основною будівельною одиницею NestJS, які групують пов'язані за функціональністю елементи, такі як контролери, сервіси і провайдери. Контролери обробляють вхідні HTTP-запити і взаємодіють зі службами, які містять бізнес-логіку. Провайдери використовуються для ін'єкції залежностей і взаємодії з іншими компонентами додатку.

Після створення загальної структури проєкту NestJS, наступним кроком було створення декількох ресурсів, таких як `auditories`, `groups`, `teachers`, `load`, `schedule-lessons` та інші. Для кожного з цих ресурсів було створено CRUD операції, що дозволяють управляти всіма елементами освітнього процесу.

Таким чином, процес створення NestJS застосунку був структурованим і ефективним завдяки чіткому розподілу відповідальностей між модулями, контролерами і сервісами.

3.2 Керівництво користувача веб-застосунку

Для забезпечення зручного та ефективного використання розробленого веб-застосунку для управління розкладом у закладах вищої освіти, було розроблено керівництво користувача. Одразу як користувач відкриє головну сторінку, його зустрине графічний інтерфейс де йому буде запропоновано один з двох варіантів, перший – авторизація (для адміністраторів системи), та другий це перегляд розкладу. Інтерфейс головної сторінки відображено в додатку Г. Даний веб-застосунок вимагає авторизації (додаток Д) перед доступом до функцій управління розкладом занять. Для використання додатку, якщо у вас вже є обліковий запис адміністратора, потрібно ввести свою електронну пошту та пароль. При цьому якщо користувач бажає просто переглянути розклад занять (додаток Е), йому авторизацію здійснювати не потрібно.

Головною функцією веб-застосунку є складання розкладу, цей процес починається з побудови основної структури університету. Першим кроком є створення груп, факультетів, кафедр, викладачів, корпусів та аудиторій навчального закладу.

Почнемо з академічних груп. На кожному факультеті необхідно визначити, які академічні групи існують. Ці групи формуються на основі спеціалізацій, курсу навчання та кількості студентів. Інтерфейс сторінки відображено в додатку Ж

Після цього визначаємо кафедри та викладачів. Кожен факультет має кілька кафедр, які відповідають за викладання певних дисциплін. Наприклад, факультет фізики може мати кафедри теоретичної фізики, експериментальної фізики тощо. Для кожної кафедри необхідно створити записи із зазначенням їхньої назви. Наступним кроком є внесення даних про викладачів. Потрібно додати інформацію про кожного викладача, включаючи їхні імена, прізвища та кафедри, на яких вони працюють. Це дозволить надалі правильно розподіляти навчальне навантаження. Інтерфейс сторінки відображено в додатку И

Не менш важливим є створення корпусів та аудиторій. Визначаємо кількість навчальних корпусів та додаємо інформацію про кожну аудиторію: її номер та

місткість. Це допоможе в подальшому правильно розподілити заняття по аудиторіях, враховуючи їхню місткість та оснащення (додаток К).

Після того, як структура університету створена, переходимо до визначення навчального навантаження. Це можна зробити на спеціальній сторінці "Навчальне навантаження". Спочатку вибираємо конкретну академічну групу, для якої будемо визначати навчальне навантаження. Для кожної групи вводимо назви дисциплін, які будуть викладатися. Важливо зазначити вид навантаження для кожної дисципліни – це може бути лекція, практичне заняття, лабораторна робота тощо. Далі визначаємо семестр, в якому буде викладатися дисципліна, та кількість годин, відведених на неї. Це забезпечить чітке розуміння, які предмети і в якій кількості будуть викладатися протягом навчального року. Інтерфейс сторінки відображено в додатку Л.

Наступний крок – розподіл навчального навантаження між викладачами. Для цього спочатку вибираємо групу та дисципліну, яку потрібно розподілити. Потім призначаємо викладача, який буде викладати цю дисципліну. Важливо враховувати спеціалізацію викладача, його досвід та наявність вільного часу в його розкладі. Це дозволить ефективно використовувати кадровий потенціал університету та забезпечити високий рівень викладання. Інтерфейс сторінки відображено в додатку М.

Коли навчальне навантаження розподілено, необхідно встановити терміни навчання відповідно до графіку освітнього процесу. Це включає визначення дат початку та закінчення кожного семестру. Окрім того, може виникнути потреба коригувати розклад дзвінків, враховуючи специфіку навчального процесу та потреби студентів і викладачів. Наприклад, можна змінити час початку занять, тривалість пар та перерви між ними. Також є функція налаштування кольорів, яка дозволяє адміністраторам вибирати кольори для окремих видів навантаження, це дозволяє персоналізувати свій розклад занять згідно з власними уподобаннями та потребами. Це не лише полегшує їхнє визначення на розкладі, але й створює більш організований та персоналізований досвід використання застосунку. Сторінка налаштувань показана в додатку П.

Останній етап – безпосереднє складання розкладу. Для цього необхідно вибрати конкретну групу або викладача, для якого складається розклад. Потім вибираємо дисципліну, яка викладається в даній групі або викладачем у конкретному семестрі. Після цього у календарі розміщуємо заняття (лекції, практичні заняття, лабораторні роботи) на конкретні дати та часи. Користувачі можуть додавати нові заняття, редагувати існуючі та видаляти їх за потреби. Важливо враховувати наявність вільних аудиторій та розклад інших занять, щоб уникнути накладок і забезпечити оптимальне використання ресурсів університету. Сторінка редактора розкладу зображена на додатку Н.

Таким чином, процес складання розкладу є багатоступеневим і потребує систематичного підходу. Від створення структури університету до розподілу навчального навантаження та складання розкладу, кожен етап є важливим для забезпечення ефективного та організованого навчального процесу. Це керівництво надає користувачам всю необхідну інформацію для ефективного використання веб-застосунку для управління розкладом у ЗВО.

Висновки до розділу 3

У цьому розділі було розглянуто процес реалізації веб-застосунку для управління розкладом у закладах вищої освіти. Цей застосунок побудований з використанням архітектури клієнт-сервер. Для реалізації клієнтської частини системи було створено проект на основі фреймворку React з використанням інструмента збірки фронтенда Vite. Дизайн інтерфейсу був розроблений у Figma.

Усі основні функціональні сторінки веб-застосунку, були розроблені з використання бібліотек React та MUI, з урахуванням потреб користувачів та вимог до управління навчальним процесом.

На серверній частині було реалізовано проект на базі фреймворку NestJS, що забезпечило структурування коду та чіткий розподіл відповідальностей між модулями, контролерами і сервісами. Це дозволило створити ефективну серверну

архітектуру з CRUD операціями для управління всіма елементами освітнього процесу.

Керівництво користувача, яке було розроблено для веб-застосунку, надає детальну інформацію про всі етапи роботи з системою, від авторизації до складання розкладу занять. Це забезпечує зручне та ефективне використання розробленої системи для управління розкладом занять у закладах вищої освіти.

Розроблена система дозволяє значно спростити процес складання та управління розкладом занять, забезпечуючи високу продуктивність, гнучкість та надійність роботи. Вона відповідає сучасним вимогам до веб-застосунків, що використовуються в освітніх установах, і може бути легко адаптована для різних типів навчальних закладів.

ВИСНОВКИ

У ході написання кваліфікаційної роботи був розроблений веб-застосунок для управління розкладом у ЗВО. Було проведено аналіз предметної області, обґрунтована актуальність задачі розробки веб-застосунку для управління розкладом у ЗВО. Розглянуто існуючі аналоги та описані вимоги до розробки даного застосунку.

Проектування веб-застосунку було проведено з урахуванням обраного інструментарію та вимог до апаратного забезпечення. Було розглянуто та досліджено платформи для реалізації поставленої задачі. Досліджено переваги даних платформ та тонкощі. Визначено варіанти використання та структуру застосунку.

Реалізація веб-застосунку була здійснена з урахуванням структури інтерфейсу, що була ретельно розроблена. Надано керівництво користувача, яке дозволяє ефективно користуватися застосунком. Також проведено тестування роботи застосунку для перевірки його функціональності та коректності роботи.

В результаті дослідження та розробки було створено веб-застосунок, який забезпечує студентам та викладачам зручний і ефективний інструмент для управління розкладом занять в закладах вищої освіти. Цей застосунок дозволяє ефективно організувати навчальний процес та забезпечує безпеку та зручність користування ним.

Розроблений веб-застосунок для управління розкладом у ЗВО може бути впровадженою в рамках широкого спектру закладів вищої освіти, оскільки покриває особливості функціонування більшої частини з них. Надалі планується вдосконалення алгоритму та розширення можливостей більш точного і більш повного налаштування параметрів для управління розкладом у ЗВО.

Отже, виконання кваліфікаційної роботи підтвердило актуальність теми та продемонструвало можливості використання сучасних веб-технологій у сфері освіти.

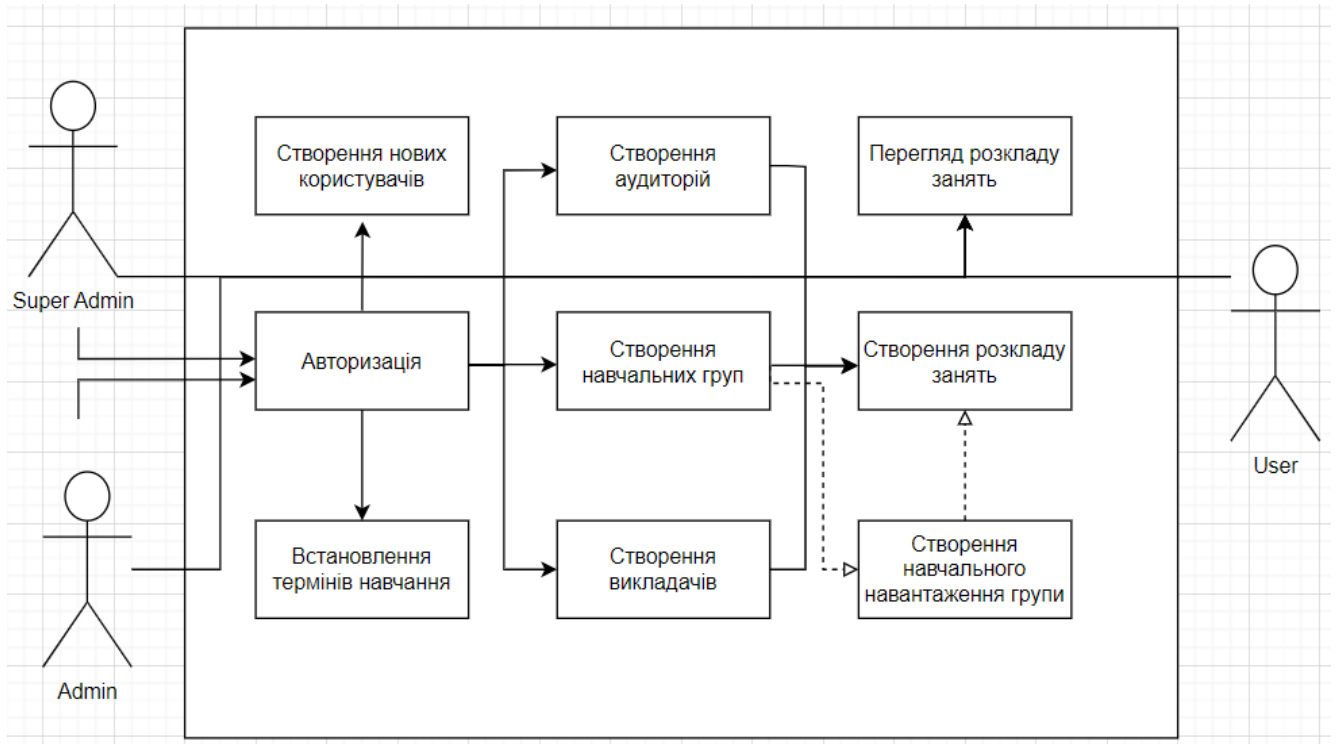
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Державна служба статистики України. [Електронний ресурс]. – Режим доступу: <https://ukrstat.gov.ua> (дата звернення: 26.03.24);
2. ПП «Політек-СОФТ». Веб-сайт. URL: <http://www.politek-soft.kiev.ua/index.php> (дата звернення: 17.04.2023).
3. Навчальна хмара «ЛКЛАУД». Веб-сайт. URL: <https://lcloud.in.ua/> (дата звернення: 18.04.2023).
4. Офіційний сайт Запорізького Національного Університету. [Електронний ресурс]. – Режим доступу: <https://www.znu.edu.ua/> (дата звернення: 17.02.2024)
5. Офіційний сайт Київського Європейського Університету. [Електронний ресурс]. – Режим доступу: <https://e-u.edu.ua/> (дата звернення: 26.03.2024)
6. Офіційний сайт Національного медичного університету імені О.О. Богомольця. [Електронний ресурс]. – Режим доступу: <http://nmuofficial.com/> (дата звернення: 26.03.2024)
7. Чат-бот від Національного Університету «Полтавська Політехніка Імені Юрія Кондратюка». [Електронний ресурс]. – Режим доступу: <https://np.pl.ua/2020/09/poltavsk-kyu-universytet-zapustyv-telegram-chat-bot-dlia-studentiv/> (дата звернення: 26.03.24);
8. Особливості веб-орієнтованих інформаційних систем веб-сайт. URL: <https://studfile.net/preview/5535715/page:3> (дата звернення: 23.03.2023);
9. Основи розробки баз даних. веб-сайт. URL: <https://support.microsoft.com/uk-ua/office/основи-розробки-баз-даних-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5> (дата звернення: 08.01.2024);
10. Марченко А.В. Проєктування інформаційних систем: навч. посіб. Хмельницький. 2015. 375с.
11. Основні етапи створення сайту. веб-сайт. URL: <https://pbb.lviv.ua/statti-inovyny/statti-shchodo-stvorennia-saitu/osnovni-etapy-stvorennia-saitu> (дата звернення: 25.02.2024);
12. Codesido I. What is front-end development? веб-сайт. URL:

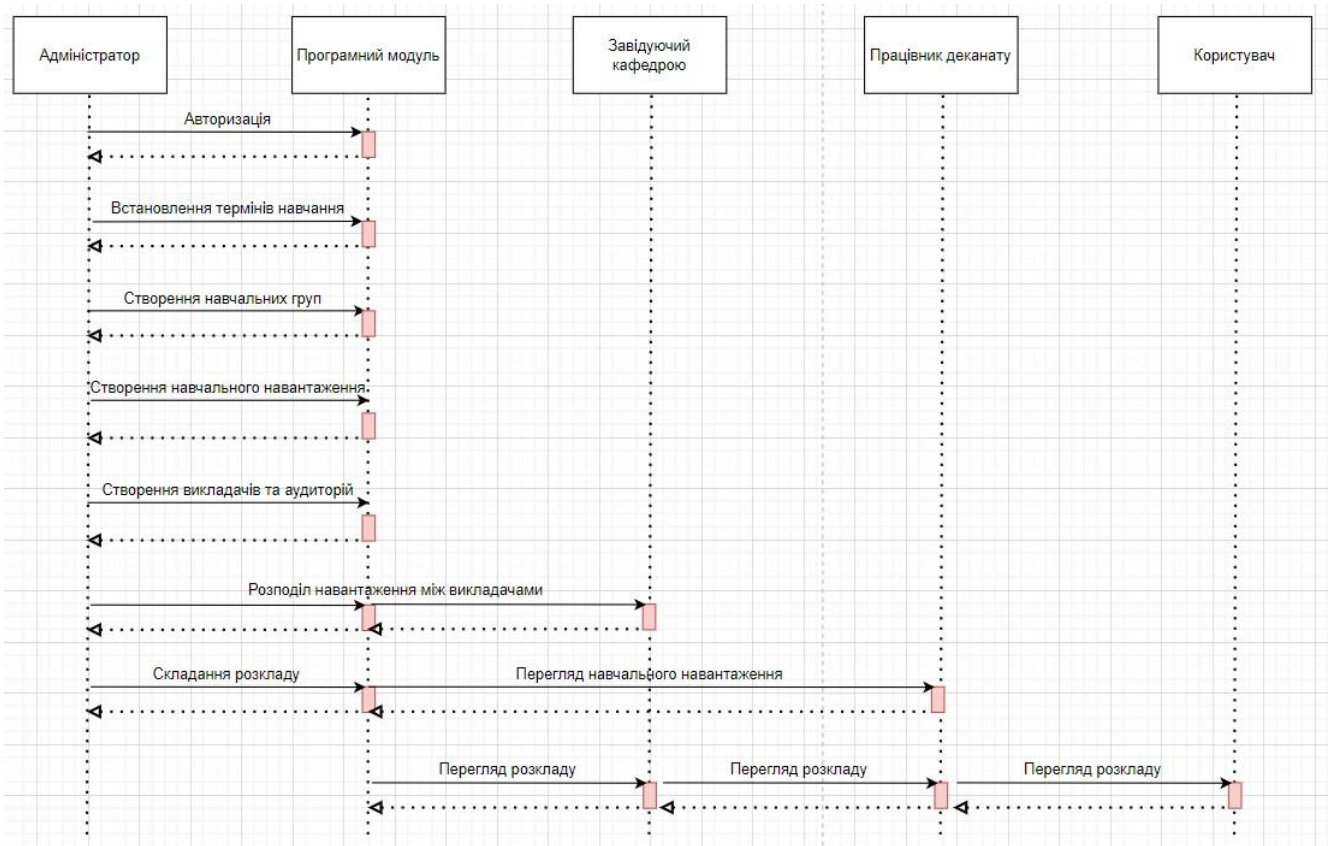
- <https://www.theguardian.com/help/insideguardian/2009/sep/28/blogpost>. (дата звернення: 18.02.2024);
13. User Experience (UX): Process and Methodology. URL: <https://uiuxtrend.com/user-experience-ux-process/> (дата звернення: 06.03.2024);
 14. Stack Overflow - Where Developers Learn, Share, & Build Careers. URL: <https://stackoverflow.com/> (дата звернення: 19.04.2024);
 15. PostgreSQL: The World's Most Advanced Open Source Relational Database веб-сайт. URL: <https://www.elephantsql.com/> (дата звернення: 04.03.2024);
 16. ElephantSQL веб-сайт. URL: <https://www.elephantsql.com/> (дата звернення: 05.03.2024);
 17. Томашевський О.М. Інформаційні технології та моделювання бізнес-процесів: навч. посіб. для студ. ВНЗ / О.М. Томашевський, Г.Г. Цегелик, М.Б. Вітер, В.І. Дубук. — К.: Центр уч. лри, 2022. — 295 с.
 18. Соколовська З.М., Клепікова О.А. Комп'ютерне моделювання складних економічних систем : монографія. Одеса : Астропринт, 2021. 512 с. 488-489
 19. Морзе Н.В. Інформаційні системи. Навч. посіб. Івано-Франківськ, «ЛілеяНВ», – 2019. – 384 с.
 20. Анісімов А. В. Інформаційні системи та бази даних: навч. посіб. для студентів факультету комп'ютерних наук та кібернетики. Київ. 2017. – 110 с.
 21. Swagger веб-сайт. URL: <https://swagger.io/> (дата звернення: 16.03.2024);
 22. React.dev веб-сайт. URL: <https://react.dev/> (дата звернення: 12.03.2024);
 23. React Router веб-сайт. URL: <https://reactrouter.com/en/main> (дата звернення: 30.03.2024);
 24. Axios Docs URL: <https://axios-http.com/ru/docs/intro> (дата звернення: 01.04.2024);
 25. Nestjs веб-сайт. URL: <https://nestjs.com/> (дата звернення: 17.04.2024);
 26. GitHub веб-сайт. URL: <https://github.com/> (дата звернення: 10.10.2023);
 27. Автоматизована система управління навчальним закладом. [Електронний ресурс]. – Режим доступу: <http://mkr.org.ua/> (дата звернення: 18.02.2024)

Додатки

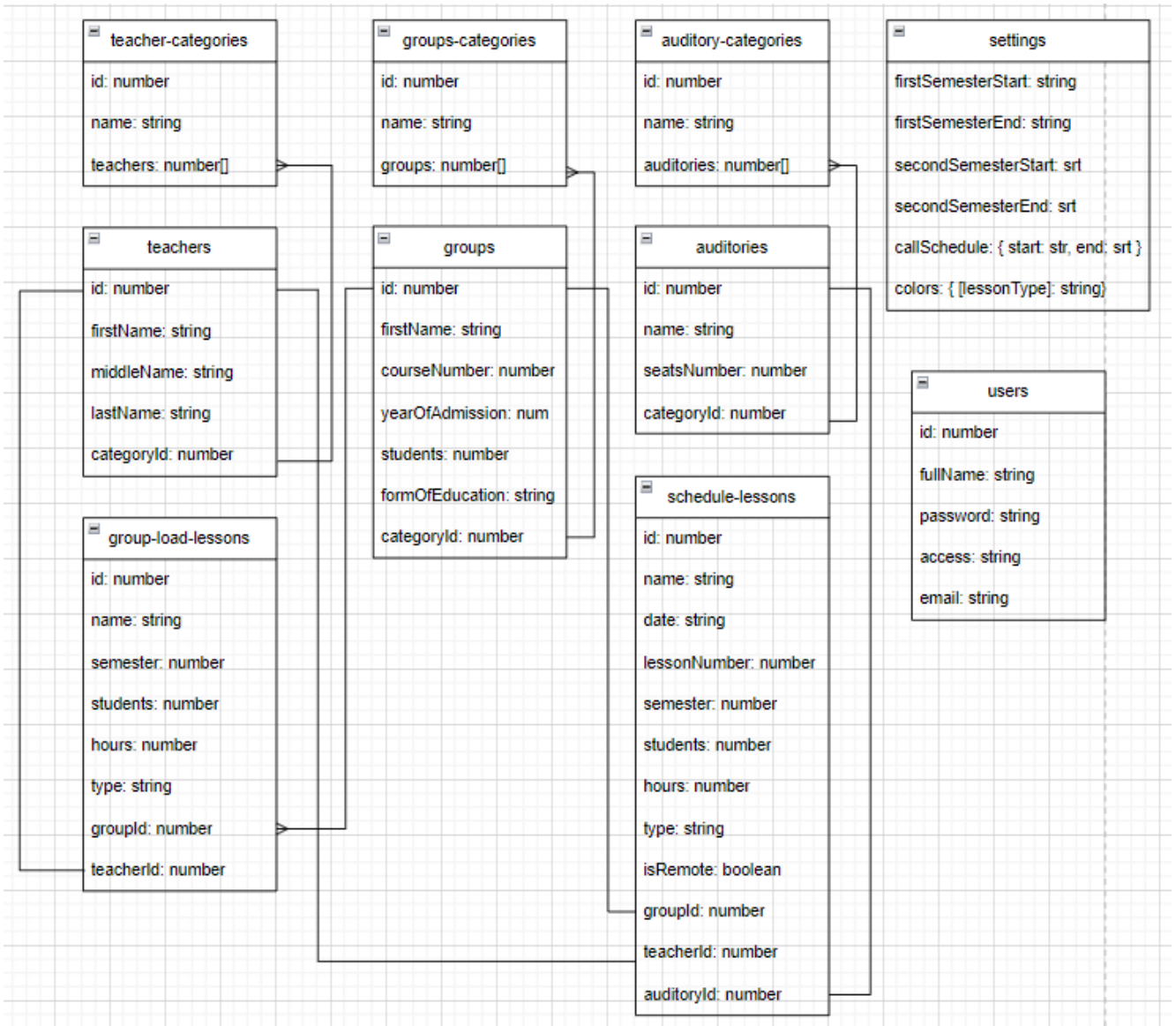
Діаграма прецедентів



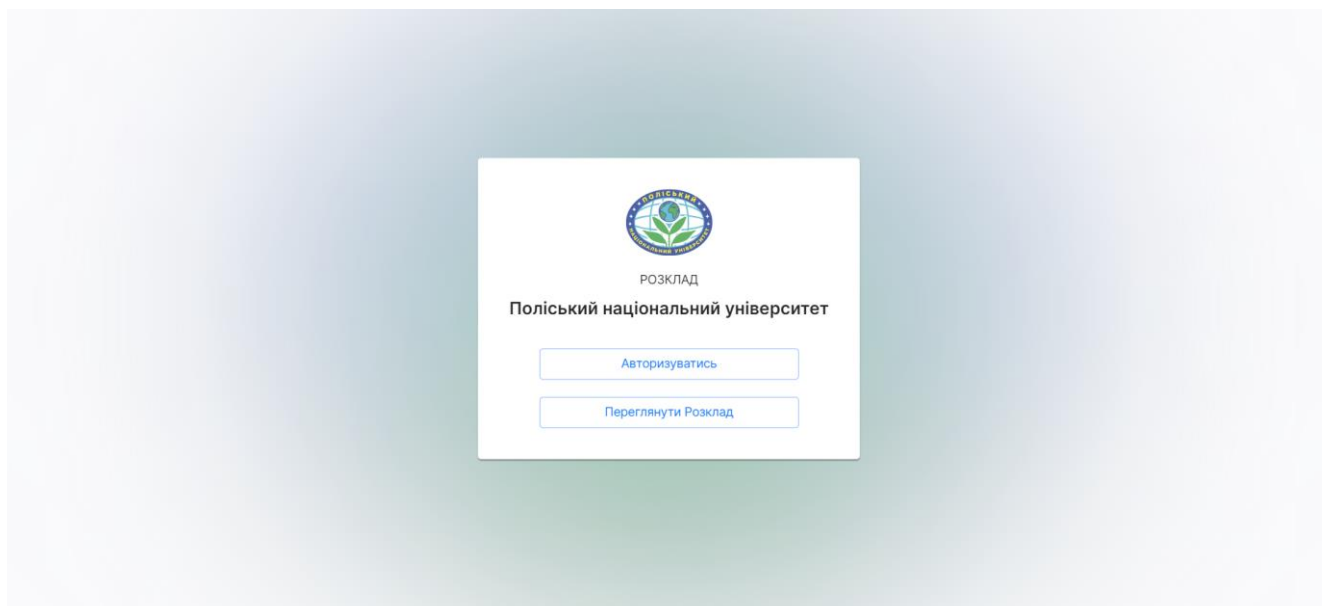
Діаграма послідовності



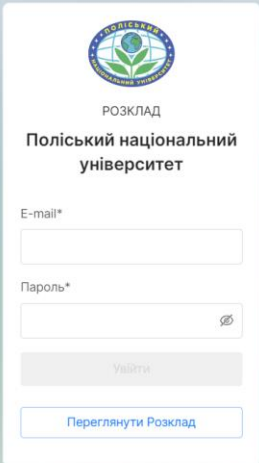
Структура бази даних




Інтерфейс головної сторінки




Інтерфейс сторінки авторизації




РОЗКЛАД
Поліський національний
університет

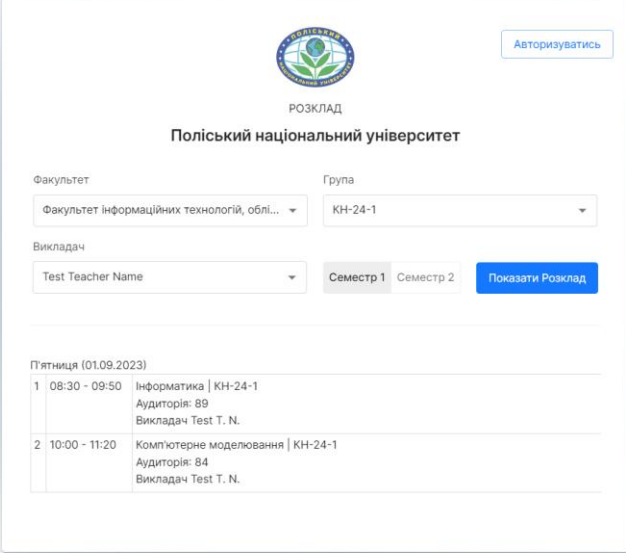
Е-mail*

Пароль*
 

Увійти

[Переглянути Розклад](#)

Інтерфейс сторінки «Перегляд розкладу»



Авторизуватись

РОЗКЛАД
Поліський національний університет

Факультет: Факультет інформаційних технологій, обл...
Група: КН-24-1

Викладач: Test Teacher Name

Семестр 1 Семестр 2 Показати Розклад

П'ятниця (01.09.2023)

1	08:30 - 09:50	Інформатика КН-24-1 Аудиторія: 89 Викладач Test T. N.
2	10:00 - 11:20	Комп'ютерне моделювання КН-24-1 Аудиторія: 84 Викладач Test T. N.

Інтерфейс сторінки «Групи»

Поліський університет
Navigation

Групи

- Викладачі
- Аудиторії
- Навчальне навантаження
- Розподіл навантаження
- Редагування розкладу
- Перегляд розкладу
- Налаштування

Структурні підрозділи

- Факультет інформаційних технологій
- Факультет права та публічного управління
- Факультет обліку та фінансів

Групи

Факультет інформаційних технологій

Назва	Курс	Студентів	Дії
КН-24-1	1	30	
КН-23-1	2	28	
ІСТ-24-1	1	22	
ІСТ-23-1	2	18	

admin

Інтерфейс сторінки «Викладачі»

Поліський університет

Navigation

- Групи
- Викладачі**
- Аудиторії
- Навчальне навантаження
- Розподіл навантаження
- Редагування розкладу
- Перегляд розкладу
- Налаштування

admin

Викладачі

Кафедра: Комп'ютерних технологій і моделювання систем	3			
Скорик Є. Р.				
Андрейчин Т. К.				
Лісничий О. І.				
Кафедра: Іноземних мов	3			
Кафедра: Фінансів і кредиту	2			
Кафедра: Вищої та прикладної математики	2			
Кафедра: Правознавства	2			

ДОДАТИ НОВОГО ВИКЛАДАЧА

Прізвище*

Ім'я*

По батькові*

Кафедра*

Створити

ДОДАТИ НОВУ КАФЕДРУ

Назва*

Створити

Інтерфейс сторінки «Аудиторії»

Поліський університет

Navigation

- Групи
- Викладачі
- Аудиторії**
- Навчальне навантаження
- Розподіл навантаження
- Редагування розкладу
- Перегляд розкладу
- Налаштування

admin

Аудиторії

Корпус 1	3			
Корпус 2	4			
84				
89				
87				
86				
Корпус 3	2			
Корпус 4	2			

ДОДАТИ НОВУ АУДИТОРІЮ

Назва*

Кількість місць*

Корпус*

Створити

ДОДАТИ НОВУ КАТЕГОРІЮ

Назва*

Створити

Інтерфейс сторінки «Навчальне навантаження»

Поліський університет

Navigation

- Групи
- Викладачі
- Аудиторії
- Навчальне навантаження**
- Розподіл навантаження
- Редагування розкладу
- Перегляд розкладу
- Налаштування

admin

Група: КН-24-1

Навчальне навантаження

ГРУПИ

- Факультет інформаційних технологій
 - КН-24-1
 - КН-23-1
 - ІСТ-24-1
 - ІСТ-23-1
- Факультет права та публічного управління
- Факультет обліку та фінансів

ДОДАТИ НАВАНТАЖЕННЯ

Назва*

Вид навантаження*

Семестр*

Кількість годин*

Створити

НАВЧАЛЬНЕ НАВАНТАЖЕННЯ ГРУПИ

№	Назва дисципліни	Тип	Семестр	Години
1	Ділова іноземна мова (B2)	ПЗ	1	36
2	Комп'ютерне моделювання	ЛК	1	12
3	Інформаційні технології	ЛАБ	2	10
4	Інформаційні технології	ЛАБ	2	30
5	Теорія прийняття рішень	ПЗ	1	24
6	Комп'ютерне моделювання	ЛАБ	1	28
7	Інтернет речей	ЛАБ	2	20
8	Інтернет речей	ЛК	2	14

Інтерфейс сторінки «Розподіл навантаження»

Поліський університет

Navigation

- Групи
- Викладачі
- Аудиторії
- Навчальне навантаження
- Розподіл навантаження**
- Редагування розкладу
- Перегляд розкладу
- Налаштування

admin

Розподіл навантаження

НАВАНТАЖЕННЯ ГРУПИ: КН-24-1

Назва дисципліни	Тип	Семестр	Години
Ділова іноземна мова (B2)	ПЗ	1	36
Комп'ютерне моделювання	ЛК	1	12
Інформаційні технології	ЛАБ	2	10
Теорія прийняття рішень	ПЗ	1	24
Інтернет речей	ЛАБ	2	20

ДІЛОВА ІНОЗЕМНА МОВА (B2)

Годин Викладач

ВИКЛАДАЧІ

Комп'ютерних технологій і моделювання систем	3	▼
Іноземних мов	3	▼
Фінансів і кредиту	2	▼
Вищої та прикладної математики	2	▼
Правознавства	2	▼

Інтерфейс сторінки «Налаштування»

Поліський університет

admin

Налаштування

Терміни навчання

Перший семестр

Початок	Кінець
09.01.2023	12.24.2023

Другий семестр

Початок	Кінець
02.01.2024	06.30.2024

Зберегти

Налаштування кольорів

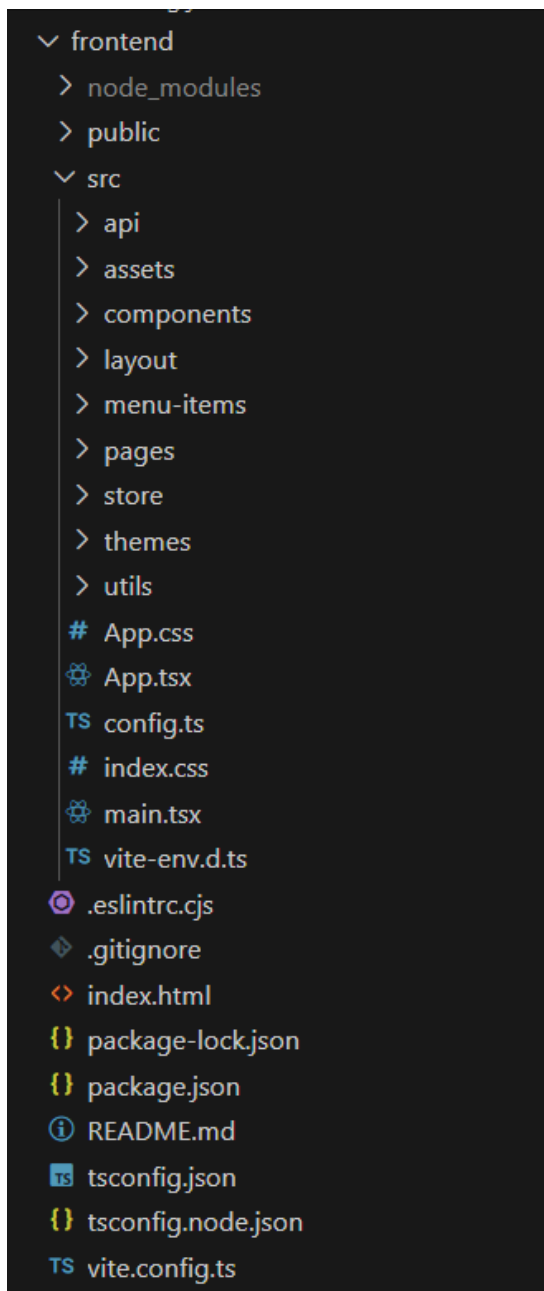
Лекції	rgb(232, 255, 82)
Практичні	rgb(24, 176, 67)
Лабораторні	rgb(43, 163, 185)
Семінари	rgb(159, 107, 243)

Розклад дзвінків

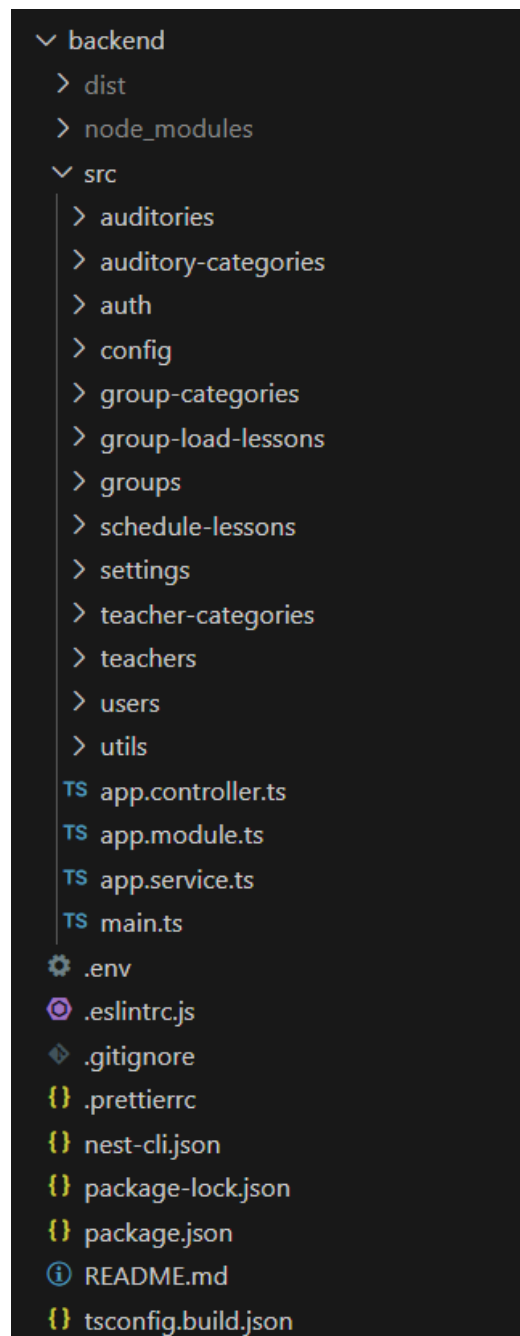
1.	Початок: 08:30	Кінець: 09:50
2.	Початок: 10:00	Кінець: 11:20
3.	Початок: 12:00	Кінець: 13:20
4.	Початок: 13:30	Кінець: 14:50
5.	Початок: 15:00	Кінець: 16:20
6.	Початок: 16:30	Кінець: 17:50
7.	Початок: 18:00	Кінець: 19:20

Зберегти

Організація структури frontend



Організація структури backend



Лістинг програмного коду frontend

```
import React from "react"
import { Route, Routes, useNavigate } from "react-router-dom"

import "./App.css"
import ThemeCustomization from "./themes"
import MainLayout from "./layout/MainLayout"
import ScrollTop from "./components/ScrollTop"
import "react-toastify/dist/ReactToastify.css"
import { useAppDispatch } from "./store/store"
import { LoadPage } from "./pages/Load/LoadPage"
import MinimalLayout from "./layout/MinimalLayout"
import { AuthPage } from "./pages/AuthPage/AuthPage"
import { authMe } from "./store/auth/authAsyncActions"
import { GroupsPage } from "./pages/Groups/GroupsPage"
import FullGroupPage from "./pages/FullGroup/FullGroupPage"
import { TeachersPage } from "./pages/Teachers/TeachersPage"
import { SettingsPage } from "./pages/Settings/SettingsPage"
import { DefaultPage } from "./pages/DefaultPage/DefaultPage"
import { TimetablePage } from "./pages/Timetable/TimetablePage"
import { AuditoriesPage } from "./pages/Auditories/AuditoriesPage"
import { DistributionPage } from "./pages/Distribution/DistributionPage"
import { ViewTimetablePage } from
"./pages/ViewTimetablePage/ViewTimetablePage"

const App = () => {
  const dispatch = useAppDispatch()
```

```
const navigate = useNavigate()
```

```
React.useEffect(() => {
```

```
  const token = window.localStorage.getItem("webSchedule-token")
```

```
  if (token) {
```

```
    dispatch(authMe({ token }))
```

```
  } else {
```

```
    navigate("/")
```

```
  }
```

```
}, [])
```

```
return (
```

```
  <ThemeCustomization>
```

```
  <ScrollTop>
```

```
  <Routes>
```

```
    <Route element={<MainLayout />}>
```

```
      <Route element={<GroupsPage />} path="/groups" />
```

```
      <Route element={<FullGroupPage />} path="/groups/:id" />
```

```
      <Route element={<FullGroupPage />} path="/groups/create/:categoryId" />
```

```
      <Route element={<TeachersPage />} path="/teachers" />
```

```
      <Route element={<AuditoriesPage />} path="/auditories" />
```

```
      <Route element={<LoadPage />} path="/load" />
```

```
      <Route element={<DistributionPage />} path="/distribution" />
```

```
      <Route element={<TimetablePage />} path="/timetable" />
```

```
      <Route element={<SettingsPage />} path="/settings" />
```

```
    </Route>
```

```
    <Route element={<MinimalLayout />}>
```

```
<Route element={<DefaultPage />} path="/" />
<Route element={<AuthPage />} path="/auth" />
<Route element={<ViewTimetablePage />} path="/view-schedule" />
</Route>
</Routes>
</ScrollTop>
</ThemeCustomization>
)
}

export default App
```

Лістинг програмного коду backend

```
import { Module } from '@nestjs/common';

import { AppService } from './app.service';
import { ConfigModule } from '@nestjs/config';
import { TypeOrmModule } from '@nestjs/typeorm';
import { AuthModule } from './auth/auth.module';
import { AppController } from './app.controller';
import { UsersModule } from './users/users.module';
import { GroupsModule } from './groups/groups.module';
import { UserEntity } from './users/entities/user.entity';
import { TeachersModule } from './teachers/teachers.module';
import { SettingsModule } from './settings/settings.module';
import { GroupEntity } from './groups/entities/group.entity';
import { AuditoriesModule } from './auditories/auditories.module';
import { TeacherEntity } from './teachers/entities/teacher.entity';
import { SettingsEntity } from './settings/entities/setting.entity';
import { AuditoryEntity } from './auditories/entities/auditory.entity';
import { GroupCategoriesModule } from './group-categories/group-categories.module';
import { ScheduleLessonsModule } from './schedule-lessons/schedule-lessons.module';
import { GroupCategoryEntity } from './group-categories/entities/group-
category.entity';

import { GroupLoadLessonsModule } from './group-load-lessons/group-load-
lessons.module';

import { TeacherCategoriesModule } from './teacher-categories/teacher-
categories.module';

import { ScheduleLessonsEntity } from './schedule-lessons/entities/schedule-
lesson.entity';
```

```
import { AuditoryCategoriesModule } from './auditory-categories/auditory-  
categories.module';  
  
import { TeacherCategoryEntity } from './teacher-categories/entities/teacher-  
category.entity';  
  
import { GroupLoadLessonEntity } from './group-load-lessons/entities/group-load-  
lesson.entity';  
  
import { AuditoryCategoryEntity } from './auditory-categories/entities/auditory-  
category.entity';
```

```
@Module({  
  imports: [  
    ConfigModule.forRoot(),  
    TypeOrmModule.forRoot({  
      type: 'postgres',  
      host: process.env.DB_HOST,  
      port: Number(process.env.DB_PORT) || 5432,  
      username: process.env.DB_USER,  
      password: process.env.DB_PASSWORD,  
      database: process.env.DB_NAME,  
      entities: [  
        UserEntity,  
        GroupEntity,  
        TeacherEntity,  
        AuditoryEntity,  
        SettingsEntity,  
        GroupCategoryEntity,  
        TeacherCategoryEntity,  
        GroupLoadLessonEntity,  
        ScheduleLessonsEntity,  
        AuditoryCategoryEntity,
```

```
    ],  
    extra: {  
      max: 1, // set pool max size  
    },  
    synchronize: true,  
  }  
),  
AuthModule,  
UsersModule,  
GroupsModule,  
TeachersModule,  
SettingsModule,  
AuditoriesModule,  
GroupCategoriesModule,  
ScheduleLessonsModule,  
GroupLoadLessonsModule,  
TeacherCategoriesModule,  
AuditoryCategoriesModule,  
],  
controllers: [AppController],  
providers: [AppService],  
})  
export class AppModule { }
```