Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Розробка та розгортання Discord-бота для відстеження розкладу занять у ХНУРЕ
(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-2
Дмитро ЯКУЩЕНКО
(власне ім’я, прізвище)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп’ютерна інженерія
(повна назва освітньої програми)

Керівник: ас. Ігор МИХАЙЛІЧЕНКО
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім’я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Якущенку Дмитру Олексійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Розробка та розгортання Discord-бота для відстеження розкладу занять у ХНУРЕ

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи

1) документація JS
2) документація Node.js
3) документація discord.js
4) документація SQL і СУБД SQLite
5) редактор вихідного коду Visual Studio Code

4. Перелік питань, що потрібно опрацювати у роботі

1) аналіз предметної області;
2) аналіз використовуваних технологій;
3) програмна реалізація;
4) інструкція користувача;
5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Слайд-презентація – 11 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	27.05.25-28.05.25	
2	Вибір технології розробки та	29.05.25-31.05.25	
	інструментальних засобів		
3	Розробка програмних модулів	01.06.25-03.06.25	
4	Відлагодження програмних модулів	04.06.25-08.06.25	
5	Оформлення матеріалів кваліфікаційної роботи	09.06.25-12.06.25	
6	Подання кваліфікаційної роботи керівникові	13.06.25-15.06.25	
	та її попередній захист		
7	Подання кваліфікаційної роботи на рецензу-	16.06.25-17.06.25	
	вання		

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Ігор МИХАЙЛІЧЕНКО
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 75 с., 19 рис., 5 табл., 2 дод., 17 джерел.

JAVASCRIPT, SQL, DISCORD BOT, NODE.JS, SQLITE.

Метою кваліфікаційної роботи є створення discord бота для перегляду розкладу. Основними особливостями розробленого бота є вибір групи студента, власна база даних з розкладом, нагадування про заняття та створення нотаток.

У ході виконання кваліфікаційної роботи були проаналізовані існуючі програмні рішення для перегляду розкладу. На висновках аналізу їх недоліків та переваг було сформовано усі необхідні функції, які повинні бути у бота.

Для реалізації поставленого завдання використано Visual Studio Code для редагування коду, мову програмування JavaScript (Node.js), бібліотеку discord.js для взаємодії з Discord API, SQLite як систему управління базами даних, а також бібліотеку puppeteer для обробки веб-сторінок і збору даних.

ABSTRACT

Bachelor’s thesis: 75 pages, 19 figures, 5 tables, 2 appendices, 17 sources.

JAVASCRIPT, SQL, DISCORD BOT, NODE.JS, SQLITE.

The major goal of this thesis is creating a discord bot for viewing the schedule. The main features of the developed bot are the selection of a student's group, its own database with the schedule, class reminders, and creating notes.

During the qualification work, existing software solutions for viewing the schedule were analyzed. Based on the conclusions of the analysis of their shortcomings and advantages, all the necessary functions that the bot should have were formed.

In order to complete the task Visual Studio Code was used for code editing, the JavaScript programming language (Node.js), the discord.js library for interacting with the Discord API, SQLite as a database management system, and the puppeteer library for processing web pages and collecting data.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – База даних

ВНЗ – Вищий навчальний заклад

ПЗ – Програмне забезпечення

СУБД – Система управління базами даних

API – Application Programming Interface

IDE – Integrated Development Environment

JSON – JavaScript Object Notation

npm – Node Package Manager

WWW – World Wide Web

ВСТУП

Сучасний світ набуває все більше нових способів комунікації та обміну інформацією. Однією з таких популярних систем є Discord, платформа із сервісу для геймерів еволюціонувала у багатофункціональний інструмент для спілкування, організації спільнот за інтересами та навчання або роботи. Розвиток інформаційних технологій та його проникнення в усі сфери життя, навіть у освітній процес, відкриває нові можливості для розвитку таких платформ. Зростаюча популярність Discord зумовлює необхідність розширення його функціональних можливостей шляхом інтеграції різноманітних ботів. Наразі боти можуть автоматизуюти рутинні завдання або вводити системи рівнів.

Усім викладачам та студентам важливо, щоб кожен в групі мав під рукою актуальний розклад. Раніше розклад публікували в групах в соціальних мережах, надсилали електронною поштою або розміщювали на вебсайтах, але це було незручно або могло бути пропущено користувачами. З цього приводу розробка Discord бота, здатного швидко надавати актуальний розклад за запитом користувача та попереджати про наступаючі заняття, знизила б відсоток пропуску занять та здавання робіт після дедлайну.

Актуальність теми кваліфікаційної роботи зумовлена зростаючою потребою в ефективних інструментах для управління інформацією в онлайн-середовищах, і Discord якнайбільш підходить для середовища, що стрімко розвивається, тому розробка бота з функціоналом розкладу не лише розкриє потенціал системи, а й дозволить оптимізувати процес доступу студентів до інформації про заняття.

Мета кваліфікаційної роботи – розробка та дослідження функціональності Discord бота, який буде показувати студентам актуальний розклад занять та нагадувати їм про їх відвідування.

1 АНАЛІЗ ПРОБЛЕМИ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1. Платформа Discord та її актуальність

Discord — це багатоплатформовий сервіс для голосового чату та текстового спілкування, він дає можливість користучавам створювати сервери для комунікації в режимі реального часу [1]. За допомогою безкоштовні, зручності та різномантним інтеграціїям, спочатку розроблений для геймерської спільноти, швидко вийшов за межі лише ігрового середовища та став повсякденно використовуватися для навчання, роботи, організації заходів та обміну інформацією.

На зображені (рисунок 1.1) показане стандартне вікно клієнта Discord з операційної системи Windows. У лівій частині вікна знаходиться панель серверів — вертикальна колонка з іконками серверів. Нижче розміщена кнопка для створення або приєднання до нового сервера.

Праворуч від панелі серверів знаходиться панель особистих повідомлень, тут відображаються друзі, активні чати та вкладки "Друзі", "Nitro", "Крамниця". У цій частині також знаходятся користувачі, яким було відправлено або отримано останні повідомлення.

Центральна частина вікна відповідає за основний список контактів. Тут показано активних друзів, зокрема, відображається їхній статус ("В голосовому каналі", "Онлайн" тощо) та можливість перейти до бесіди.

У правій частині вікна розміщено блок активних контактів, цей блок дозволяє бачити, що і хто з друзів користувача наразі робить(слухає музику, проводить потік, тощо).

У нижньому лівому куті інтерфейсу міститься панель керування додатком. Тут знаходиться ім’я профілю, кнопки мікрофона, навушників та налаштувань, які дозволяють швидко вмикати/вимикати звук або переходити до параметрів облікового запису.

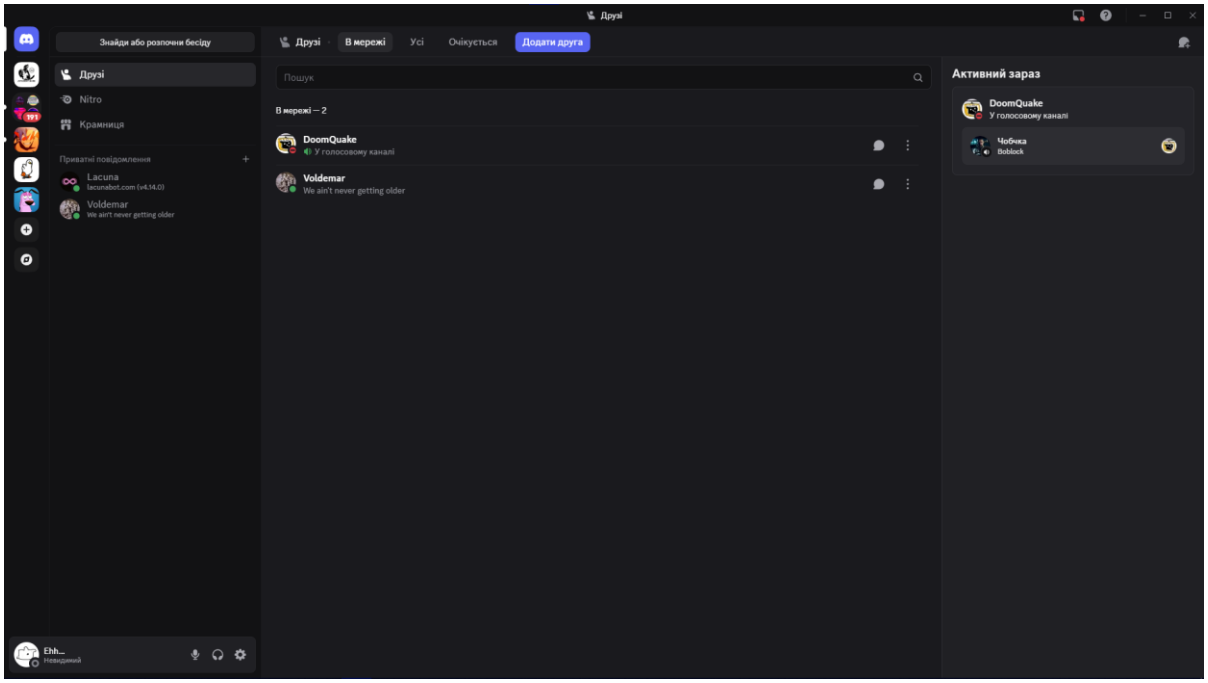


Рисунок 1.1 – Графічний інтерфейс користувача Discord

Платформа активно використовується студентами різних українських університетів, зокрема ХНУРЕ. Студенти ХНУРЕ створили власний Discord-сервер та досягли вже декількох тисяч учасників. На цьому сервері організовано тематичні канали для кожного факультету, а також секції для технічної підтримки та спілкування на дозвіллі.

1.2 Існуючі рішення інших ВНЗ

Сучасні вищі навчальні заклади повинні забезпечувати студентам зручний та своєчасний доступ до навчального розкладу. Наприклад, Національний університет "Полтавська політехніка імені Юрія Кондратюка" успішно вирішив цю проблему, впровадивши одразу кілька рішень для електронного доступу до розкладу.

По-перше, студентам доступний мобільний додаток "eUniversity" для iOS [2] та Android [3]. Ці додатки (рисунок 1.2, 1.3) надають зручний інтерфейс для перегляду розкладу.

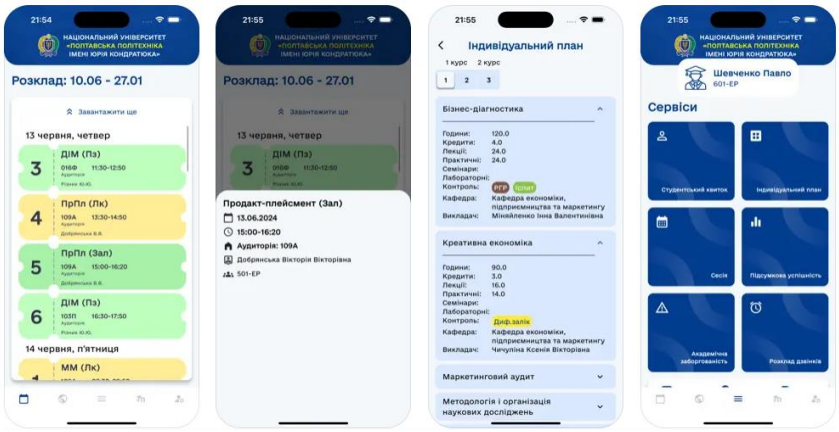


Рисунок 1.2 – Мобільний додаток "eUniversity" для iOS

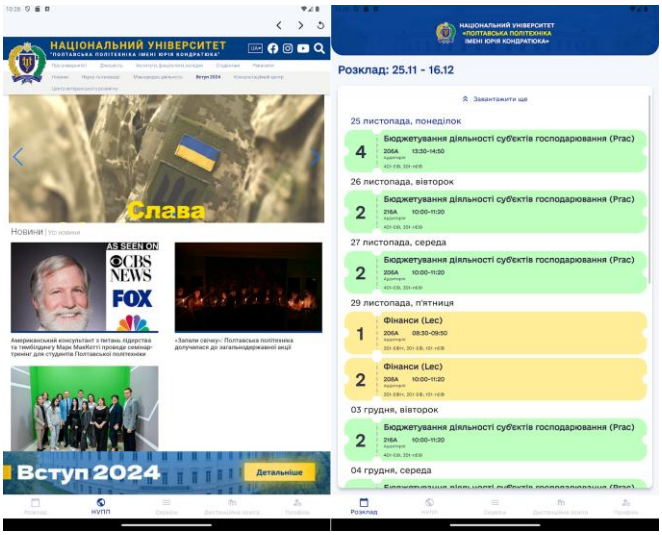


Рисунок 1.3 – Мобільний додаток "eUniversity" для Android

По-друге, був створений офіційний Telegram-бот "@NuppBot" [4]. Цей бот (рисунок 1.4) надає актуальний розклад, дозволяє підписуватись на оновлення та використовує зручний інтерфейс для пошуку потрібної інформації.

Виходячи з того, що студенти ХНУРЕ масово використовують Discord, то впровадження схожої системи у платформу також актуальне і перспективне рішення. Інтеграція бота з інформуванням про розклад безпосередньо в Discord стане ефективним кроком до підвищення інформованості та дисциплінованості в освітньому процесі.

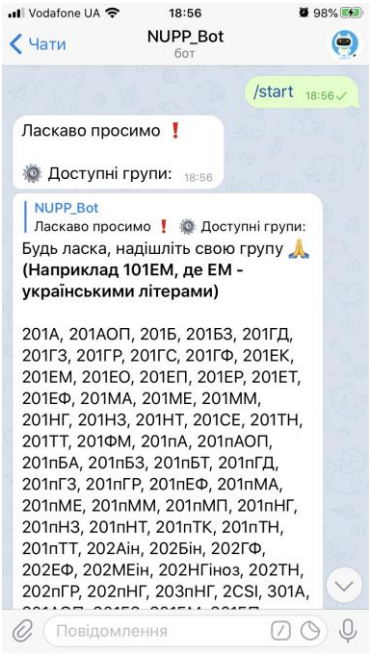


Рисунок 1.4 – Telegram-бот "@NuppBot"

1.3 Аналіз існуючих засобів перегляду розкладу

Як і Полтавський ВНЗ, що був приведений до прикладу вище, ХНУРЕ надає кілька інструментів для забезпечення студентів актуальною інформацією про розклад занять. До них належать мобільні додатки для iOS та Android, а також офіційний вебсайт університетської інформаційної системи CIST.

1.3.1 ПЗ KNURE Timetable (iOS)

KNURE Timetable — мобільний додаток для користувачів пристроїв на базі iOS, він надає змогу переглядати розклад занять студентів, викладачів та кафедр ХНУРЕ. Додаток безкоштовний і доступний у магазині App Store (рисунок 1.5).

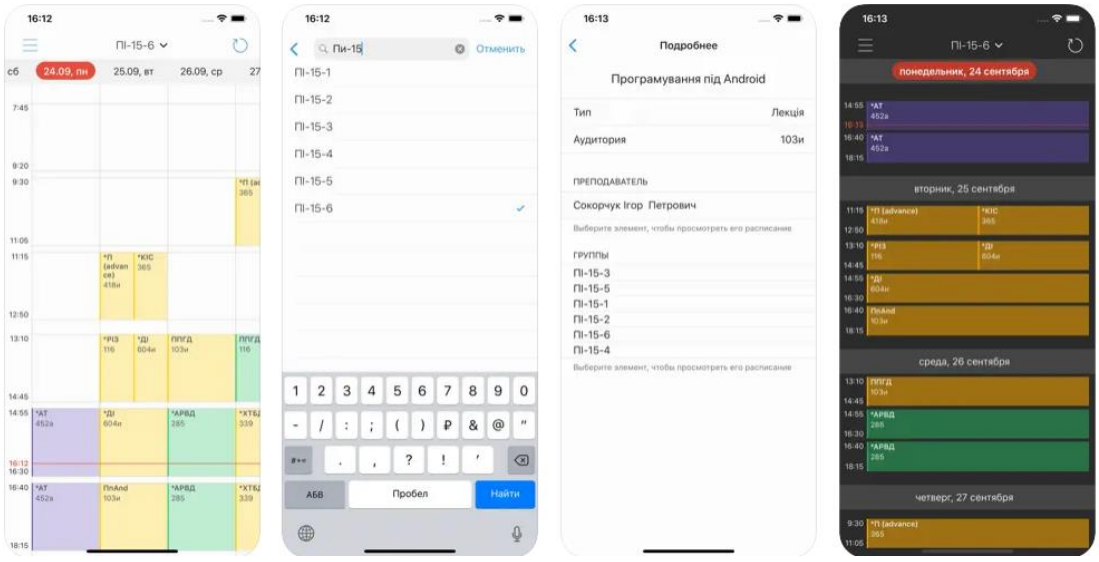


Рисунок 1.5 — Інтерфейс мобільного додатку KNURE Timetable на iOS

Основною перевагою додатку є підключення до бази даних, яка постійно оновлюється. Інтерфейс програми інтуїтивно зрозумілий, дозволяє швидко обрати групу, кафедру або викладача, а також переглядати розклад як на поточний тиждень, так і на інші дні. Зберігти час також допомагає можливість збереження останніх обраних налаштувань.

Серед недоліків можна відзначити відсутність push-сповіщень про заняття або зміни в розкладі, що підвищує ризик пропущення занять. Додаток не підтримує інтеграцію з месенджерами, календарями чи іншими сервісами, якими активно користуються студенти. Найважливішим недоліком додатку можна назвати відсутність української мови. Для виключно українськомовних студентів такий недолік може завадити освітньому процесу. Окрім цього, іноді виникає затримка з оновленням інформації через кешування старих даних.

1.3.2. ПЗ Розклад ХНУРЕ (Android)

Додаток Розклад ХНУРЕ розроблений для пристроїв Android і дозволяє зручно переглядати розклад занять за групою, викладачем, кафедрою або аудиторією. Його можна завантажити з Google Play (рисунок 1.6).

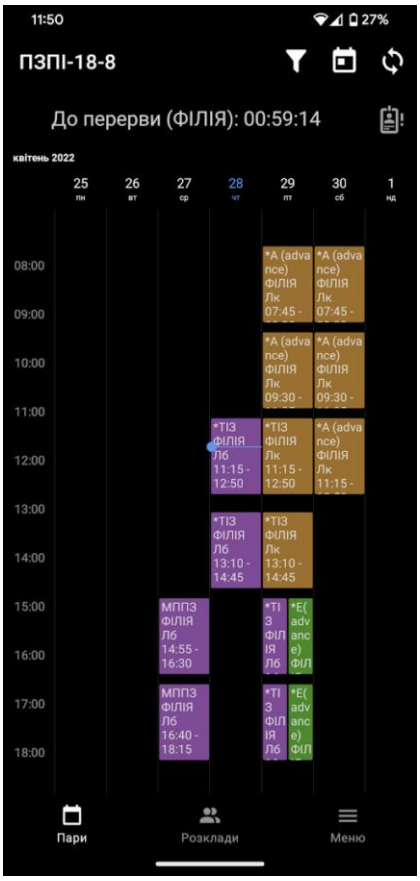


Рисунок 1.6 — Інтерфейс мобільного додатку Розклад ХНУРЕ на Android

Основна перевага застосунку - простий та швидкий інтерфейс, що дозволяє студентам швидко знайти необхідний розклад. ПЗ також підтримує збереження розкладу для офлайн-доступу, має темну тему для комфортного перегляду в темну пору доби, і автоматично оновлює дані без втручання користувача.

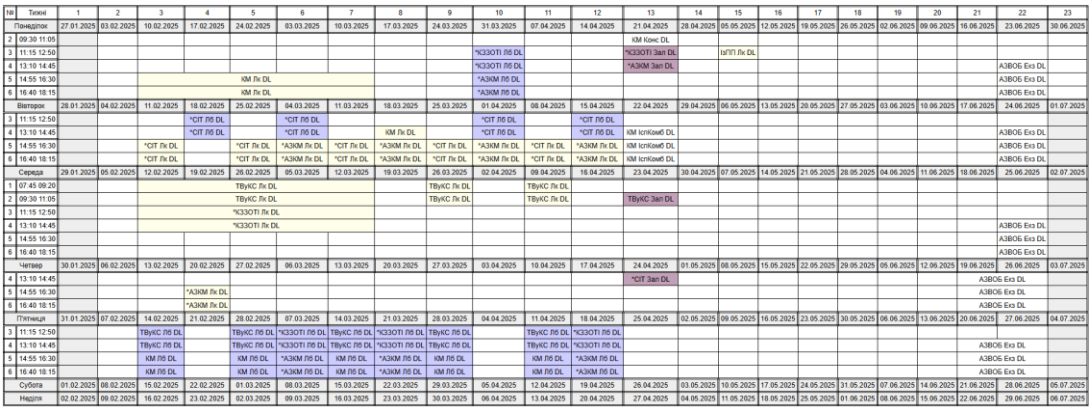
Як і у IOS застосунку, серед недоліків - відсутність нагадувань про заняття або зміни в розкладі. Також, додаток не підтримує інтеграцію з месенджерами, календарями чи іншими сервісами, якими активно користуються студенти.

1.3.3. Вебсайт cist.nure.ua

CIST (cist.nure.ua) — головний вебсайт, який забезпечує доступ до

актуального розкладу занять у ХНУРЕ (рисунок 1.7). Його основна функція — надання інформації про розклад для всіх користувачів: студентів, викладачів, кафедр і адміністрації (рисунок 1.8). Усі мобільні застосунки використовують сайт для оновлення своїх баз даних. Таким чином, його стабільна робота критично важлива для всієї інформаційної інфраструктури університету.

Рисунок 1.7 — Головна сторінка сайту cist.nure.ua



№	Тиждні	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	Понеділок	27.01.2025	03.02.2025	10.02.2025	17.02.2025	24.02.2025	03.03.2025	10.03.2025	17.03.2025	24.03.2025	31.03.2025	07.04.2025	14.04.2025	21.04.2025	28.04.2025	05.05.2025	12.05.2025	19.05.2025	26.05.2025	02.06.2025	09.06.2025	16.06.2025	23.06.2025	30.06.2025
2	09:30 11:05													КМ Конс DL										
3	11:15 12:50										*КЗЗОТІ Лб DL			*КЗЗОТІ Зал DL		taПП Лк DL								
4	13:10 14:45										*КЗЗОТІ Лб DL			*АЗКМ Зал DL									АЗВОБ Екз DL	
5	14:55 16:30					КМ Лк DL					*АЗКМ Лб DL												АЗВОБ Екз DL	
6	16:40 18:15					КМ Лк DL					*АЗКМ Лб DL												АЗВОБ Екз DL	
	Вівторок	28.01.2025	04.02.2025	11.02.2025	18.02.2025	25.02.2025	04.03.2025	11.03.2025	18.03.2025	25.03.2025	01.04.2025	08.04.2025	15.04.2025	22.04.2025	29.04.2025	06.05.2025	13.05.2025	20.05.2025	27.05.2025	03.06.2025	10.06.2025	17.06.2025	24.06.2025	01.07.2025
3	11:15 12:50				*СІТ Лб DL		*СІТ Лб DL				*СІТ Лб DL		*СІТ Лб DL											
4	13:10 14:45				*СІТ Лб DL		*СІТ Лб DL		КМ Лк DL		*СІТ Лб DL		*СІТ Лб DL	КМ IcnКомб DL									АЗВОБ Екз DL	
5	14:55 16:30			*СІТ Лк DL		*СІТ Лк DL	*АЗКМ Лк DL	*СІТ Лк DL	*АЗКМ Лк DL	*СІТ Лк DL	*АЗКМ Лк DL	*СІТ Лк DL	*АЗКМ Лк DL	КМ IcnКомб DL									АЗВОБ Екз DL	
6	16:40 18:15			*СІТ Лк DL		*СІТ Лк DL	*АЗКМ Лк DL	*СІТ Лк DL	*АЗКМ Лк DL	*СІТ Лк DL	*АЗКМ Лк DL	*СІТ Лк DL	*АЗКМ Лк DL	КМ IcnКомб DL									АЗВОБ Екз DL	
	Середа	29.01.2025	05.02.2025	12.02.2025	19.02.2025	26.02.2025	05.03.2025	12.03.2025	19.03.2025	26.03.2025	02.04.2025	09.04.2025	16.04.2025	23.04.2025	30.04.2025	07.05.2025	14.05.2025	21.05.2025	28.05.2025	04.06.2025	11.06.2025	18.06.2025	25.06.2025	02.07.2025
1	07.45 09.20					ТВуКС Лк DL				ТВуКС Лк DL		ТВуКС Лк DL												
2	09:30 11:05					ТВуКС Лк DL				ТВуКС Лк DL		ТВуКС Лк DL		ТВуКС Зал DL										
3	11:15 12:50					*КЗЗОТІ Лк DL																		
4	13:10 14:45					*КЗЗОТІ Лк DL																	АЗВОБ Екз DL	
5	14:55 16:30																						АЗВОБ Екз DL	
6	16:40 18:15																						АЗВОБ Екз DL	
	Четвер	30.01.2025	06.02.2025	13.02.2025	20.02.2025	27.02.2025	06.03.2025	13.03.2025	20.03.2025	27.03.2025	03.04.2025	10.04.2025	17.04.2025	24.04.2025	01.05.2025	08.05.2025	15.05.2025	22.05.2025	29.05.2025	05.06.2025	12.06.2025	19.06.2025	26.06.2025	03.07.2025
4	13:10 14:45													*СІТ Зал DL								АЗВОБ Екз DL		
5	14:55 16:30				*АЗКМ Лк DL																	АЗВОБ Екз DL		
6	16:40 18:15				*АЗКМ Лк DL																	АЗВОБ Екз DL		
	П'ятниця	31.01.2025	07.02.2025	14.02.2025	21.02.2025	28.02.2025	07.03.2025	14.03.2025	21.03.2025	28.03.2025	04.04.2025	11.04.2025	18.04.2025	25.04.2025	02.05.2025	09.05.2025	16.05.2025	23.05.2025	30.05.2025	06.06.2025	13.06.2025	20.06.2025	27.06.2025	04.07.2025
3	11:15 12:50			ТВуКС Лб DL		ТВуКС Лб DL	*КЗЗОТІ Лб DL	ТВуКС Лб DL	*КЗЗОТІ Лб DL	ТВуКС Лб DL		ТВуКС Лб DL	*КЗЗОТІ Лб DL											
4	13:10 14:45			ТВуКС Лб DL		ТВуКС Лб DL	*КЗЗОТІ Лб DL	ТВуКС Лб DL	*КЗЗОТІ Лб DL	ТВуКС Лб DL		ТВуКС Лб DL	*КЗЗОТІ Лб DL									АЗВОБ Екз DL		
5	14:55 16:30			КМ Лб DL		КМ Лб DL	*АЗКМ Лб DL	КМ Лб DL	*АЗКМ Лб DL	КМ Лб DL		КМ Лб DL	*АЗКМ Лб DL									АЗВОБ Екз DL		
6	16:40 18:15			КМ Лб DL		КМ Лб DL	*АЗКМ Лб DL	КМ Лб DL	*АЗКМ Лб DL	КМ Лб DL		КМ Лб DL	*АЗКМ Лб DL									АЗВОБ Екз DL		
	Субота	01.02.2025	08.02.2025	15.02.2025	22.02.2025	01.03.2025	08.03.2025	15.03.2025	22.03.2025	29.03.2025	05.04.2025	12.04.2025	19.04.2025	26.04.2025	03.05.2025	10.05.2025	17.05.2025	24.05.2025	31.05.2025	07.06.2025	14.06.2025	21.06.2025	28.06.2025	05.07.2025
	Неділя	02.02.2025	09.02.2025	16.02.2025	23.02.2025	02.03.2025	09.03.2025	16.03.2025	23.03.2025	30.03.2025	06.04.2025	13.04.2025	20.04.2025	27.04.2025	04.05.2025	11.05.2025	18.05.2025	25.05.2025	01.06.2025	08.06.2025	15.06.2025	22.06.2025	29.06.2025	06.07.2025

*СІТ — *Cloud-технології : Лк (20) - КІУКІ-21-1,2,3;КІУКІу-22-1, Ні.Я. С.; Лб (16) - КІУКІ-21-2, Ні.Я. С., Четуріна І. С.; Зал (2) - КІУКІ-21-2, Ні.Я. С.:
*АЗКМ — *Апаратні засоби комп'ютерних мереж : Лк (20) - КІУКІ-21-1,2,3;КІУКІу-22-1, Єрошенко О. А.; Лб (16) - КІУКІ-21-2, Єрошенко О. А., Тимошенко Д. О.; Зал (2) - КІУКІ-21-2, Єрошенко О. А.:
АЗВОБ — Атестації здобувачів вищої освіти ступеню БАКАЛАВР : Екз (42) - КІУКІ-21-2,4,6:
taПП — Інструктаж з практичної підготовки : Лк (2) - КІУКІ-21-1,2,3,4,5,6, Спіцина Т. Є.:
*КЗЗОТІ — *Комп'ютерні засоби збору та обробки технологічної інформації : Лк (20) - КІУКІ-21-1,2,3;КІУКІу-22-1, Партика С. О.; Лб (16) - КІУКІ-21-2, Партика С. О., Майстренко Г. В.; Зал (2) - КІУКІ-21-2, Партика С. О.:
КМ — Комп'ютерні мережі : Лк (22) - КІУКІ-21-1,2,3,4,5,6, Коваленко А. А.; Лб (20) - КІУКІ-21-2, Порошенко А. І., Партика С. О.; Конс (2) - КІУКІ-21-2, Коваленко А. А.; IcnКомб (6) - КІУКІ-21-2, Коваленко А. А.:
ТВуКС — Технології віртуалізації у комп'ютерних системах : Лк (28) - КІУКІ-21-1,2,3,4,5,6, Ткачов В. М.; Лб (28) - КІУКІ-21-2, Ткачов В. М., Четуріна І. С.; Зал (2) - КІУКІ-21-2, Ткачов В. М.:

Рисунок 1.8 — Таблиця розкладу групи КІУКІ-21-2

Сайт надає широкі можливості для фільтрації та пошуку інформації: можна знайти розклад за групою, викладачем, аудиторією чи кафедрою. Надає доступ до повного семестрового розкладу та користувач має

можливість його експорту або роздруку.

Проте вебсайт має і ряд суттєвих недоліків. Його інтерфейс не адаптований до мобільних пристроїв, через що користувачам незручно дивитись розклад на смартфонах. Крім того, навігація може бути незручною через велику кількість параметрів, особливо для нових користувачів. Ще одна проблема - часті тимчасові втрати доступу до сайту, які трапляються в умовах військової агресії та енергетичних проблем в Україні з боку країни агресора, зокрема через вимкнення електроенергії під час масованих обстрілів. Це суттєво знижує надійність ресурсу як постійного засобу доступу до розкладу.

Нижче представлена таблиця 1.1 з підведеними підсумками.

Таблиця 1.1 – Порівняння функціональності програмних засобів

Критерій	KNURE Timetable (iOS)	Розклад ХНУРЕ (Android)	Сайт cist.nure.ua
Операційна система	iOS	Android	Будь-яка
Мова інтерфейсу	Англійська, російська	Українська, англійська	Українська
Push-сповіщення	Немає	Немає	Немає
Інтеграція з іншими сервісами	Немає	Немає	Частково (експорт)

Продовження таблиці 1.1

1	2	3	4
Персоналізація інтерфейсу	Немає	Є	Немає
Мобільна адаптація інтерфейсу	Так	Так	Немає
Стабільність доступу	Висока	Висока	Бувають збої

Мобільні застосунки хоча і мають багато переваг в порівнянні з вебсайтом, а також, незважаючи на те, що заняття парсяться з самого сайту, все ще показують останні оновлення, навіть якщо сайт не працює, але все одно мають такі сутеві недоліки, як неможливість завантажити на інші ОС(Windows, Linux, MacOS) та відсутність повідомлень.

2 АНАЛІЗ ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ

2.1 Node.js

У сучасному світі веб-розробки Node.js займає одне з провідних місць, він надає розробникам потужний інструмент для створення масштабованих та високопродуктивних серверних додатків. Його популярність значною мірою зумовлена використанням JavaScript, як основної мови програмування, що відкриває широкі можливості для уніфікації розробки на фронтенді та бекенді.

Node.js - кросплатформове середовище виконання JavaScript з відкритим вихідним кодом, побудоване на рушії V8 JavaScript від Google Chrome [5]. Воно дозволяє розробникам запускати JavaScript код поза браузером, що традиційно було прерогативою серверних мов, таких як PHP, Python або Ruby. Завдяки асинхронній, керованій подіями архітектурі, Node.js ідеальний вибір для побудови мережевих додатків, які вимагають високої швидкості та ефективності, таких як чати в реальному часі, потокові сервіси та API. Пакетний менеджер npm - найбільший у світі репозиторій програмного забезпечення з відкритим вихідним кодом, пропонуючий тисячі бібліотек та фреймворків, які значно прискорюють процес розробки [6].

JavaScript - це високорівнева, інтерпретована мова програмування, яка є однією з трьох основних технологій WWW (разом з HTML і CSS) [7]. Спочатку розроблена Netscape для створення інтерактивних веб-сторінок на стороні клієнта, а з появою Node.js вона вийшла за межі браузера і стала повноцінною мовою для розробки як фронтенду, так і бекенду. JavaScript динамічна та типізована мова програмування з підтримкою об'єктно-орієнтованого, імперативного та функціонального стилів програмування. Її гнучкість, постійний розвиток (через стандарти ECMAScript) та величезна спільнота роблять її надзвичайно привабливою для широкого спектру

застосувань.

2.1.1 JavaScript та Node.js

Уніфікація стеку технологій одна з найвагоміших переваг. Завдяки Node.js, розробники можуть використовувати JavaScript як для клієнтської (браузер) частини, так і для серверної (Node.js) частини додатку. Це усуває необхідність вивчення та підтримки різних мов програмування, що значно спрощує процес розробки, прискорює навчання нових членів команди та покращує взаємодію між фронтенд та бекенд розробниками. Один і той же формат даних (JSON) та одні й ті ж інструменти можуть бути використані по всьому стеку.

Висока продуктивність та масштабованість Node.js зумовлена його архітектурою. Він побудований на рушії V8, який є дуже швидким у виконанні JavaScript коду. Крім того, його асинхронна, неблокуюча модель вводу/виводу дозволяє обробляти велику кількість одночасних запитів без необхідності створення окремих потоків для кожного запиту. Це означає, що Node.js може ефективно обробляти великі обсяги даних і підтримувати значну кількість користувачів, роблячи його ідеальним для додатків, які потребують високої масштабованості, таких як API, сервіси реального часу та мікросервіси [8].

Велика та активна спільнота JavaScript сприяє його широкому впровадженню. JavaScript має одну з найбільших спільнот розробників у світі. Люди по всьому світу створюють величезну кількость ресурсів, навчальних матеріалів, готових рішень та надають активну підтримку кожному розробнику. Пакетний менеджер npm, як вже зазначалося, є найбільшим репозиторієм бібліотек, який дозволяє швидко інтегрувати готові функціональності та значно скорочувати час розробки.

Легкість вивчення та низький поріг входу робить мову зручною для початківців. Для тих, хто вже знайомий з JavaScript для фронтенд розробки,

перехід на Node.js є відносно легким. Перехід зменшує криву навчання та дозволяє командам швидше розпочинати роботу над проектами.

Завдяки своїй легкості та ефективності, Node.js чудово підходить для розробки мікросервісів. Мікросервіси дозволяють розбивати великі додатки на невеликі, незалежні та легкі в обслуговуванні сервіси, така розробка спрощує розробку, розгортання та масштабування.

JSON як нативний формат даних є значною зручністю. JavaScript і JSON тісно пов'язані. JSON легкий формат обміну даними, який широко використовується у веб-розробці. Оскільки JavaScript легко працює з JSON, обробка даних між клієнтом та сервером спрощується, зменшуючи необхідність у додаткових трансформаціях.

2.1.2 Порівняння з Bun.js

На ринку інструментів для JavaScript-розробки постійно з'являються нові гравці, які прагнуть покращити продуктивність та досвід розробки. Одним з таких нових конкурентів є Bun.js.

Bun.js - це швидкий універсальний JavaScript runtime, бандлер, транспілятор і менеджер пакетів, розроблений для підвищення швидкості та ефективності розробки. Він побудований на рушії JavaScriptCore і написаний мовою програмування Zig[9]. Bun.js позиціонується як єдиний інструмент, який може замінити Node.js, Webpack, Babel, Jest та інші інструменти для розробки.

Нижче представлена таблиця 2.1 з порівнянням Node.js та Bun.js.

Таблиця 2.1 – Порівняння Node.js та Bun.js

Характеристика	Node.js	Bun.js
Рушій JavaScript	V8 (від Google Chrome)	JavaScriptCore (від Apple Safari)
Мова реалізації	C++, JavaScript	Zig
Менеджер пакетів	npm (інтегрований, але зовнішній). Також підтримуються Yarn, pnpm.	Вбудований менеджер, сумісний із npm/yarn.
Швидкість	Висока, але може поступатися при запуску й установці пакетів.	Дуже швидкий запуск, транспіляція та установка пакетів завдяки вбудованим інструментам.
Екосистема	Зріла, з величезною кількістю пакетів і фреймворків. Активна спільнота.	Молодша, але активно розвивається. Деякі пакети можуть мати проблеми з сумісністю.
Вбудовані інструменти	Потребує зовнішніх інструментів для бандлінгу (Webpack), транспіляції (Babel), тестування (Jest).	Має вбудовані бандлер, транспілятор (TypeScript, JSX), тестовий раннер.

Продовження таблиці 2.1

1	2	3
Сумісність з API	Повноцінна підтримка Node.js API.	Стремиться до сумісності з Node.js API та Web API. Можливі незначні відмінності.
Цільове призначення	Виконання серверних додатків, CLI-інструментів, десктопних програм.	Універсальний інструмент для JS-розробки — рантайм, бандлер, транспілятор, тестування, менеджер пакетів. Основна перевага — швидкість і простота використання.

Bun.js розроблен з акцентом на продуктивність. Його оптимізований вбудований менеджер пакетів та використання мови Zig дозволяють йому значно швидше запускати додатки, встановлювати залежності та виконувати збірку проекту порівняно з Node.js. По-перше, на відміну від Node.js, який часто потребує додаткових інструментів, Bun.js інтегрує ці функціональності безпосередньо, спрощуючи таким чином налаштування проекту і зменшуючи кількість залежностей. По-друге, він зручніший для сучасних фронтенд-проектів, бо він може транспілювати TypeScript та JSX без додаткових конфігурацій. Але, незважаючи на усі переваги Bun.js, для реалізації бота буде достатньо і Node.js.

2.1.3 Порівняння з Deno

Ще одним помітним конкурентом у світі JavaScript-рантаймів є Deno. Deno створив Райан Дал(розробник Node.js)[10]. Основною метою Deno було виправлення "помилок проектування" Node.js та пропонування більш безпечного середовища виконання. Він побудований на рушії V8 та написан на Rust [11].

Нижче представлена таблиця 2.2 з порівнянням Node.js та Deno.

Таблиця 2.2 – Порівняння Node.js та Deno

Характеристика	Node.js	Deno
Рушій JavaScript	V8	V8
Мова реалізації	C++, JavaScript	Rust, TypeScript
Безпека	Повний доступ до файлової системи, мережі та змінних середовища за замовчуванням. Обмеження потребують зовнішніх бібліотек.	Безпечне середовище за замовчуванням. Доступ до ресурсів (файлів, мережі, змінних середовища) потребує явного дозволу через флаги (--allow-read, --allow-net тощо).
Модулі	За замовчуванням використовує CommonJS (require()). ES-модулі підтримуються, але з обмеженнями. Потрібні node_modules та npm.	Нативно використовує ES Modules (import/export). Підтримує імпорт модулів з URL (HTTP/HTTPS). node_modules не потрібен.
Підтримка TypeScript	Потребує окремого налаштування або зовнішніх інструментів (наприклад, Babel, ts-node).	Має вбудовану підтримку TypeScript — код можна запускати напряму без додаткової конфігурації.

Продовження таблиці 2.2

1	2	3
Менеджер пакетів	npm	Відсутній централізований менеджер. Модулі імпортуються напряму з URL. Є вбудовані інструменти (deno run, deno fmt, deno lint, deno check).
API	Node.js API (fs, http та інші).	Орієнтований на сумісність із Web API (наприклад, fetch, localStorage) + власні специфічні API для Deno.
Сумісність з npm	Повна нативна підтримка.	Спочатку npm не підтримувався. Зараз підтримується через npm:-специфікатори (npm:lodash).

Хоча Deno вирішує багато проблем, притаманних Node.js, його екосистема все ще менш зріла. Node.js має перевагу в величезній кількості готових бібліотек та фреймворків. Однак Deno постійно розвивається, і його підтримка npm-пакетів через npm:specifier значно зменшує цей розрив.

2.2 Бібліотеки для роботи з Discord API

Для розробки ботів для Discord необхідне використання спеціалізованих бібліотек. Вони спрощують процес взаємодії з Discord API, надаючи зручні абстракції для надсилання та отримання даних, обробки подій та керування функціоналом боту.

На 2025 рік існують основні бібліотеки discord.js та discord.py.

2.2.1 Discord.js

Discord.js розроблений з використанням об'єктно-орієнтованого підходу, що дозволяє легко керувати різними сутностями Discord, такими як користувачі, канали, ролі, повідомлення та багато іншого[12]. Завдяки асинхронній природі JavaScript та Node.js, discord.js можна створювати високопродуктивних та масштабованих ботів, здатних обробляти велику кількість одночасних подій. Бібліотека забезпечує повну підтримку Discord API, включаючи взаємодії (interactions) — такі як слеш-команди, контекстні меню та компоненти — кнопки та випадаючі списки. Це дозволяє розробникам створювати складні та інтерактивні інтерфейси для своїх ботів.

Бібліотека надає інтуїтивно зрозумілі об'єкти для кожної сутності Discord, наприклад, Client, Guild, Channel, Message, User. Якщо назва інших об'єків каже сама за себе, то Guild - це об'єкт відповідальний за сервера.

Discord.js побудований на архітектурі, керованій подіями. Бот реагує на події, що відбуваються в Discord, дозволяючи розробникам легко реалізувати логіку обробки цих подій.

Розробники можуть створювати власні команди, обробники подій та плагіни, щоб додати унікальну функціональність своїм ботам. Велика кількість допоміжних бібліотек та фреймворків, побудованих поверї discord.js ще більше розширюють можливості.

Також, Discord.js має вбудовану систему кешування, яка зберігає дані про сутності Discord (користувачів, канали, гільдії) у пам'яті. Це значно прискорює доступ до цих даних та зменшує кількість запитів до Discord API, підвищуючи продуктивність боту.

Бібліотека має дуже детальну та зрозумілу документацію [12]. Крім того, існує велика та активна спільнота користувачів та розробників, які готові надати допомогу та відповісти на запитання. Боти на discord.js можуть бути розгорнуті на різних платформах, включаючи власні сервери, VPS або хмарні сервіси.

2.2.2 Discord.py

Для порівняння функціоналу та підходів до розробки ботів Discord, доцільно розглянути бібліотеку discord.py, яка є популярним вибором для розробників, що працюють з мовою Python.

Discord.py - це сучасна, легка та масштабована бібліотека Python для розробки ботів Discord [13]. Вона також базується на асинхронній моделі програмування та пропонує схожий функціонал для взаємодії з Discord API.

Нижче представлена таблиця 2.3 з порівнянням discord.js та discord.py.

Таблиця 2.3 – Порівняння discord.js та discord.py

Характеристика	discord.js (JavaScript)	discord.py (Python)
Мова програмування	JavaScript (на Node.js)	Python
Асинхронність	Вбудована через Promise та async/await	Використовує модуль asyncio для асинхронності
Продуктивність	Висока завдяки рушію V8 та неблокуючому вводу/виводу	Гарна продуктивність, хоча в складних сценаріях може поступатися через GIL. Для більшості ботів цього достатньо
Екосистема	Величезна екосистема npm для фронтенду, бекенду, інструментів	Широка екосистема PyPI: обробка даних, ML, веб-розробка тощо

Продовження таблиці 2.3

1	2	3
Легкість вивчення	Добре підходить для тих, хто вже знайомий з JavaScript. Менш читабельний синтаксис для початківців	Простий і читабельний синтаксис. Часто перша мова для новачків
Документація та спільнота	Розгорнута офіційна документація, велика активна спільнота	Добре написана документація, активна спільнота
Типізація	Динамічна. Можна використовувати TypeScript для статичної типізації	Динамічна. Є підтримка type hints для покращення читабельності та аналізу коду
Взаємодії (Interactions)	Повна підтримка інтеракцій через модулі	Повна підтримка інтеракцій
Підтримка API Discord	Швидко оновлюється завдяки великій кількості контриб'юторів	Також підтримується активно, але часом оновлення дещо запізнюються
Сумісність з іншими частинами проєкту	Ідеально інтегрується у проєкти на JavaScript/Node.js, забезпечує єдиний стек для веб-серверів, інтерфейсів тощо	Добре підходить, якщо весь проєкт написаний на Python. Якщо основа — Node.js, інтеграція може бути складнішою

З огляду на те, що для роботи вже було обрано Node.js для розробки, discord.js є очевидним та найкращим вибором. Використання discord.js дозволить зберегти єдиний стек технологій на JavaScript для всього проекту. Уніфікація мови програмування та середовища виконання мінімізує

необхідність перемикання контексту між різними мовами та екосистемами, забезпечуючи більш злагоджений та ефективний процес розробки. Завдяки цьому можна максимально використовувати переваги вже обраного стеку, такі як npm для управління залежностями та асинхронні можливості Node.js для обробки подій Discord.

2.3 СУБД SQLite

Зручне збереження та управління даними є критично важливим аспектом розробки будь-якого застосунку, включаючи Discord-ботів. Обрана СУБД значно впливає на продуктивність, масштабованість, надійність та складність розробки проекту.

SQLite - це СУБД з відкритим вихідним кодом, яка відрізняється своєю легковаговістю, високою продуктивністю та вбудованістю. Вона не є окремим процесом, який запускається як сервер, а інтегрується безпосередньо в додаток[14]. Це робить SQLite ідеальним рішенням для проектів, де потрібна автономна, швидка та проста у розгортанні база даних.

База даних SQLite зберігається як єдиний файл на диску. Це усуває необхідність у складному налаштуванні сервера, управлінні мережевими з'єднаннями або іншими операціями, характерними для клієнт-серверних СУБД.

SQLite є легковаговою та має невеликий розмір, що робить її ідеальною для вбудованих систем, мобільних додатків та програм, які не вимагають окремого серверного процесу бази даних.

Висока продуктивність SQLite забезпечується завдяки тому, що вона не має накладних витрат на мережеві комунікації та керування клієнтськими з'єднаннями, як у традиційних клієнт-серверних СУБД. Побачити її продуктивність можна при виконанні локальних операцій читання та запису даних.

СУБД підтримує стандартний SQL, включаючи транзакції, індекси,

представлення, тригери та інші можливості реляційних баз даних [14]. Мова дає змогу використовувати вже знайомі інструменти та запити для управління даними, якщо розробник не користувався раніше SQLite. Бібліотека є транзакційною, тому вона гаранутє надійність даних навіть у разі збоїв системи. Транзакційність забезпечує узгодженість, ізольованість та довговічність операцій [14].

SQLite має широку сумісність з різними операційними системами та мовами програмування, для яких існують офіційні або сторонні драйвери. Саме для Node.js існує стабільна та популярна бібліотека sqlite3, яка дозволяє легко взаємодіяти з базами даних SQLite [15].

2.4 Порівняння SQLite з PostgreSQL/MySQL

PostgreSQL та MySQL є двома з найпопулярніших клієнт-серверних реляційних баз даних, які часто використовуються у великих та складних додатках.

PostgreSQL - це потужна, з відкритим вихідним кодом об'єктно-реляційна СУБД, відома своєю надійністю, розширюваністю та відповідністю стандартам SQL [16]. Вона пропонує широкий спектр функцій, включаючи підтримку складних типів даних, транзакцій, збережених процедур, а також потужну систему розширення.

MySQL - це ще одна широко використовувана реляційна СУБД з відкритим вихідним кодом [17]. Вона відома своєю швидкістю і легкістю у використанні та популярністю серед веб-розробників. MySQL також пропонує різні механізми зберігання, що дозволяє оптимізувати продуктивність для різних сценаріїв використання.

Нижче представлена таблиця 2.4 з порівнянням SQLite, PostgreSQL та MySQL.

Таблиця 2.4 – Порівняння SQLite, PostgreSQL та MySQL

Характеристика	SQLite	PostgreSQL	MySQL
Архітектура	Вбудована, безсерверна.	Клієнт-серверна.	Клієнт-серверна.
Розгортання	Дуже просте. Не потребує сервера чи складної конфігурації.	Потребує встановлення серверного процесу та налаштування.	Потребує встановлення та конфігурації.
Масштабованість	Підходить для невеликих або середніх проєктів. Обмежена при високому навантаженні.	Висока масштабованість, підтримка кластерів, реплікації, розподілених систем.	Добра масштабованість. Підтримує реплікацію, менш гнучкий ніж PostgreSQL.
Паралелізм	Обмежений. Під час запису блокується вся база.	Високий. Паралельна обробка запитів на читання/запис.	Високий. Ефективна робота з багатьма одночасними з’єднаннями.

Продовження таблиці 2.4

1	2	2	3
Надійність	Висока. ACID-транзакції. Добре підходить для локального зберігання.	Дуже висока. Стандартизоване ACID-виконання, репутація однієї з найнадійніших СУБД.	Висока, але залежить від типу таблиць (наприклад, MyISAM не підтримує транзакції).
Функціонал	Базовий. Вистачає для багатьох задач, але немає розширених можливостей.	Потужний: підтримка JSONB, CTE, розширень, повнотекстовий пошук, зберігані процедури, власні типи.	Потужний, але дещо простіший. Підтримка JSON, розширень, кілька механізмів зберігання (InnoDB, MyISAM).
Використання	Мобільні додатки, вбудовані системи, настільні програми, легкі веб-проєкти.	Веб-додатки, ERP/CRM, аналітика, складні бекенди, великі платформи.	Веб-додатки, CMS (WordPress, Joomla), магазини, типові веб-проєкти.

Продовження таблиці 2.4

1	2	2	3
Node.js-сумісність	Відмінна (через бібліотеки, наприклад sqlite3, better-sqlite3).	Відмінна (pg, knex, sequelize).	Відмінна (mysql2, knex, sequelize).
Складність адміністрування	Мінімальна. Все в одному файлі, не потребує налаштувань.	Складніша: потребує глибших знань, особливо при масштабуванні.	Потребує налаштування та моніторингу, але простіший за PostgreSQL в багатьох кейсах.

Вибір між SQLite, PostgreSQL та MySQL залежить від конкретних потреб проекту. З огляду на те, що для роботи розробляється Discord-бот, який спочатку не передбачає екстремально високого навантаження та потреби у розподілених базах даних, SQLite є ліпшим вибором. Вона пропонує простоту розгортання, високу швидкість для локальних операцій та легку інтеграцію в Node.js. Використання SQLite дозволить зосередитися на функціональності бота, не обтяжуючись складним адмініструванням бази даних, при цьому зберігаючи можливість перейти до PostgreSQL/MySQL у майбутньому, якщо у цьому виникне потреба.

2.5 Середовище розробки

Для ефективної розробки програмного забезпечення критично важливо обрати відповідне середовище розробки або редактор коду. Вони надають функціонал, який значно спрощує написання, налагодження та тестування

коду. Правильний вибір середовища розробки може суттєво вплинути на продуктивність розробки та якість кінцевого продукту. IDE та редактори коду надають такі можливості, як підсвічування синтаксису, автодоповнення, інтеграція з системами контролю версій та налагоджувачі.

На ринку існує широкий спектр середовищ розробки, від легких текстових редакторів до повноцінних IDE. Кожен з них має свої переваги та недоліки, пов'язані зі споживанням системних ресурсів, функціональністю та гнучкістю.

2.5.1 Visual Studio Code

Visual Studio Code — безкоштовний багатоплатформний редактор коду з відкритим вихідним кодом, розроблений Microsoft. Він працює швидше, ніж Visual Studio адже він не є середовищем розробки, але пропонує багато їхніх переваг, зберігаючи при цьому мінімальне споживання системних ресурсів.

Visual Studio Code підтримує широкий спектр мов програмування, включаючи JavaScript, TypeScript, Python, C++, Java та інші, надаючи такі функції як підсвічування синтаксису, автодоповнення коду, налагодження, вбудований термінал та інтеграцію з системами контролю версій, зокрема Git.

2.5.2 Розширення

Функціонал редактора можна значно розширити за допомогою величезної бібліотеки розширень, доступних через вбудований Marketplace. Розширення можуть додавати підтримку нових мов, інструментів для роботи з базами даних, інтеграції з хмарними сервісами та багато іншого.

Для роботи з БД SQLite, існує розширення з відповідною назвою "SQLite" або подібні до нього. Воно дозволяє безпосередньо переглядати

структуру БД, включаючи таблиці, стовпці та індекси, виконувати SQL-запити до бази даних, переглядати та редагувати дані в таблицях.

Використання розширень значно спрощує процес розробки, оскільки для цього не потрібно перемикатись між різними інструментами для взаємодії з БД, все доступно в єдиному середовищі Visual Studio Code.

2.5.3 Порівняння з IDE

На відміну від повноцінних інтегрованих середовищ розробки, таких як Visual Studio, IntelliJ IDEA або Eclipse, Visual Studio Code позиціонується як легкий та швидкий редактор коду. Як вже було зазначено вище, на відміну від IDE, редактор займає менше ресурсів комп’ютера, і хоча він уступає в моментах, але все ще має деякі переваги, як високий ступінь гнучкості та кастомізації через систему розширень і швидкість запуску.

Головним недоліком можна назвати зосередження на редагуванні коду та базових інструментах розробки, тоді як IDE прагнуть надати комплексне рішення для всього життєвого циклу розробки програмного забезпечення.

Вибір був зроблений у перевагу Visual Studio Code за його легковагість, високу швидкість роботи та відмінну підтримку JavaScript/Node.js, редактор коду ідеально відповідає вимогам розробки Discord-бота без надмірних витрат системних ресурсів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Архітектурний огляд

Архітектура бота модульна та ієрархічна. Бота можна легко розширювати створюючи новий файл у директорії /commands.

Файл Bot.js слугує центральною точкою запуску програми. Він відповідає за ініціалізацію клієнта, завантаження всіх доступних команд бота та запуск фонових циклічних перевірок для сповіщень.

Також реалізовано окремі модулі для збору та оновлення даних до БД. Модуль schedule.js відповідає за парсинг та оновлення розкладу занять з офіційного веб-порталу до БД. Бібліотека Puppeteer симулює браузер та дозволяє автоматизувати взаємодію з веб-сторінками. Модуль groups_parser.js займається завантаженням актуального списку факультетів та груп до бази даних, також використовуючи Puppeteer.

Уся логіка команд бота реалізована в окремих модулях, розташованих у директорії commands/. Кожен файл відповідає за обробку відповідної до його назви команди.

Для збереження та керування даними використовується реляційна база даних SQLite. Взаємодія з базою даних відбувається через бібліотеки sqlite3 або better-sqlite3. Структура бази даних передбачає такі таблиці: groups для зберігання інформації про групи, users для обліку користувачів бота, schedules для розкладу занять та notes для особистих нотаток користувачів.

3.2 Структура бази даних

База даних бота складається з чотирьох основних таблиць (рисунок 3.1).

Таблиця groups зберігає унікальний список усіх академічних груп, які

були завантажені з вебсайту. Надалі вона буде слугувати довідником груп, дозволяючи користувачам вибирати свою групу за її назвою. Поле id є унікальним ідентифікатором кожної групи, яке також завантажується з сайту, а не записується автоінкрементом, а name містить текстову назву групи, наприклад, "КІУКІ-21-2". Унікальність поля name гарантує відсутність дублюючих записів для однієї і тієї ж групи.

Таблиця users зберігає інформацію про кожного користувача Discord, який взаємодіяв з ботом, слугуючи пам’яттю про їх налаштування у боті. Без цієї таблиці бот не зміг би визначати у програмі, який саме користувач написав йому та довелося б кожен раз перепитувати групу для надання розкладу, а сповіщення взагалі не змогли б працювати. Поле discord_id зберігає id користувача Discord, що дозволяє боту розрізняти користувачів. notifications_enabled визначає, чи бажає користувач отримувати автоматичні сповіщення про заняття або нотатки. Поле group_id встановлює зв'язок між користувачем і його академічною групою.

Таблиця schedules призначена для зберігання детального розкладу занять, які були завантажені з вебсайту. Кожен запис у цій таблиці представляє окреме заняття. Поле id слугує унікальним ідентифікатором кожного запису розкладу та, на відміну від попередніх таблиць, збільшується автоінкрементом. group_id пов'язує конкретне заняття з відповідною академічною групою. Поля name, time, weekday та date містять інформацію про назву предмета, час його проведення, день тижня та точну дату.

Таблиця notes використовується для зберігання особистих нотаток та нагадувань користувачів. Вона дозволяє кожному користувачеві створювати, переглядати та видаляти свої нотатки. Поле id є унікальним ідентифікатором кожної нотатки, яке також збільшується автоінкрементом. discord_id встановлює зв'язок нотатки з її власником — конкретним користувачем Discord. Поле content містить текстове наповнення нотатки. date може зберігати дату та час, якщо користувач бажає встановити нагадування. Поле notified є прапорцем, який вказує, чи було вже надіслано сповіщення про

дану нотатку, щоб уникнути повторних сповіщень.

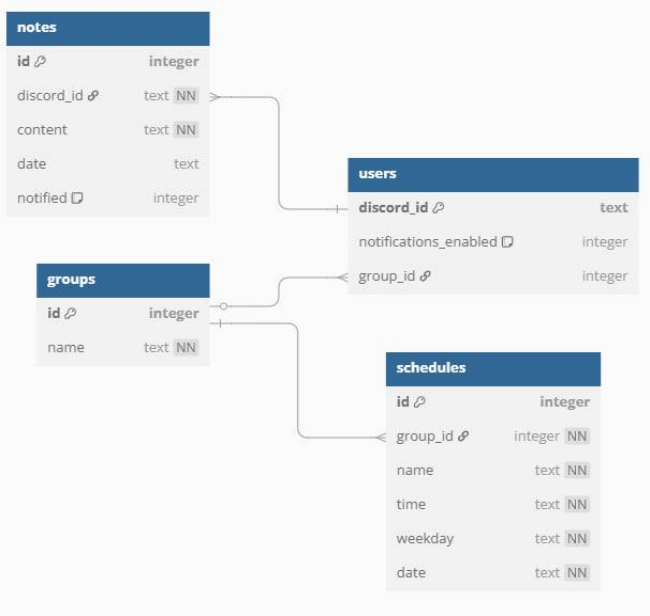


Рисунок 3.1 – ER-діаграма БД

3.3 Опис основних модулів

3.3.1 Ініціалізація й основний цикл (Bot.js)

Bot.js виконує ключові функції при запуску бота. Створюється екземпляр Client з необхідними інтенсами (Guilds, GuildMessages, DirectMessages, MessageContent) та частковими каналами для обробки приватних повідомлень. Команди з директорії commands/ динамічно підвантажуються в колекцію client.commands. Після успішного підключення бота до Discord, при івенті ready, після запуску бота у консоль виводиться повідомлення про готовність. Також запускаються дві фонові функції checkForUpcomingLessons() та checkForNotes(), які виконуються з інтервалом в одну хвилину. Перша функція відповідає за розсилку нагадувань про заняття користувачам, у яких увімкнені повідомлення. Друга функція слідкує за нотатками з часом та датою та надсилає нотатки в вказаний час. Вхідні повідомлення обробляються за подією messageCreate, фільтруючи

повідомлення від інших ботів та ті, що не починаються з префіса !, після чого викликається відповідний модуль команди через метод .execute().

3.3.2 Парсер груп

Модуль groups_parser.js використовує бібліотеку Puppeteer для парсингу інформації про групи. Він відкриває визначену сторінку cist.nure.ua. З отриманих елементів витягуються ідентифікатори та назви факультетів/груп. Отримана інформація записується в таблицю groups бази даних через better-sqlite3, при цьому використовується оператор INSERT OR IGNORE для уникнення дублювання вже існуючих записів.

3.3.3 Парсер розкладу

Модуль schedule.js призначений для збереження розкладу занять до бази даних. Він приймає змінну groupId як аргумент для парсингу розкладу конкретної групи. Модуль розраховує діапазон тижнів від початку минулого до кінця наступного місяця, формуючи масив тижневих проміжків функцією getWeekRanges. Для кожного з цих тижнів Puppeteer відкриває відповідний URL з параметрами P201_FIRST_DATE, P201_LAST_DATE та P201_GROUP. Puppeteer очікує завантаження таблиці MainTT на сторінці, потім парсить її рядки, витягуючи дату, день тижня, час та назву заняття. Перед вставкою нового запису в таблицю виконується перевірка на наявність дублюючих записів.

3.3.4 Командний інтерфейс (commands/)

Усі команди бота реалізовані як окремі js-модулі в директорії commands/. Кожен модуль експортує об'єкт з властивостями name, description, execute.

На момент написання кваліфікаційної роботи було розроблено наступні команди:

1) група.js - команда приймає назву групи як аргумент та шукає відповідний id групи у таблиці groups, а потім записує або оновлює зв'язок між Discord ID користувача та group_id у таблиці users. Після успішного встановлення групи автоматично викликається schedule.js для початкового завантаження розкладу для цієї групи;

2) розклад.js - модуль отримує group_id користувача з бази даних та формує повідомлення з розкладом на поточний тиждень. До повідомлення додаються три кнопки: " стрілка ліворуч" для перегляду попереднього тижня, "знак оновлення" для актуалізації даних та " стрілка праворуч" для переходу до розкладу наступного тижня. При натисканні кнопок "ліворуч" або "праворуч" відображається розклад відповідного тижня. Кнопка "Оновити" повторно викликає schedule.js для парсингу та перезавантаження оновлених даних розкладу;

3) повідомлення.js - команда перемикає прапорець notifications_enabled у таблиці users для поточного користувача. Бот інформує користувача про новий стан повідомлень (ввімкнено/вимкнено);

4) нотатка.js - команда створює новий запис у таблиці notes. Бот автоматично визначає, чи в кінці тексту нотатки вказано дату у форматі РРРР-ММ-ДД ГГ:ХХ. Якщо дата присутня, вона зберігається у відповідний стовпчик date для подальших нагадувань;

5) нотатки.js - команда витягує всі нотатки, що належать поточному користувачеві, з таблиці notes. Нотатки сортуються за датою (нотатки без дати відображаються останніми) та формуються у вигляді нумерованого списку для зручного перегляду;

6) видалити.js - команда приймає індекс нотатки зі списку (отриманого за допомогою !нотатки), перевіряє коректність наданого індексу та видаляє відповідний запис з таблиці notes;

7) допомога.js - команда надсилає користувачу довідкове

повідомлення, яке містить перелік усіх доступних команд бота та їх короткий опис.

3.4 Механізм сповіщень та оновлення розкладу

Фонова функція checkForUpcomingLessons у Bot.js регулярно запитує з бази даних інформацію про всіх користувачів з ввімкненими сповіщеннями та їхній розклад на найближчі хвилини. Використовуючи date-fns, бот порівнює дати та час з точністю до хвилини. При виявленні поточного заняття, бот надсилає приватне повідомлення користувачу через метод client.users.fetch().send().

Інша фонова функція, checkForNotes, фільтрує нотатки, де date IS NOT NULL та notified = 0. Якщо дата та час нотатки збігаються з поточною хвилиною, бот надсилає відповідне приватне повідомлення користувачеві та встановлює параметр notified у 1, щоб уникнути повторних сповіщень.

4 ІНСТРУКЦІЯ КОРИСТУВАЧА

4.1 Встановлення Discord на різні операційні системи

Насамперед, для використання бота, а також інших можилвостей платформи Discord, треба встановити клієнт Discord. Він доступний для широкого спектру операційних систем, включаючи як для мобільних, так і для стаціонарних комп’ютерів і ноутбуків.

Перед завантаженням додатку передусім рекомендується зареєструватись на платформі, якщо користувач цього ще не зробив. Реєстрація швидка та нескладна і виконується на офіційному сайті Discord (рисунок 4.1).

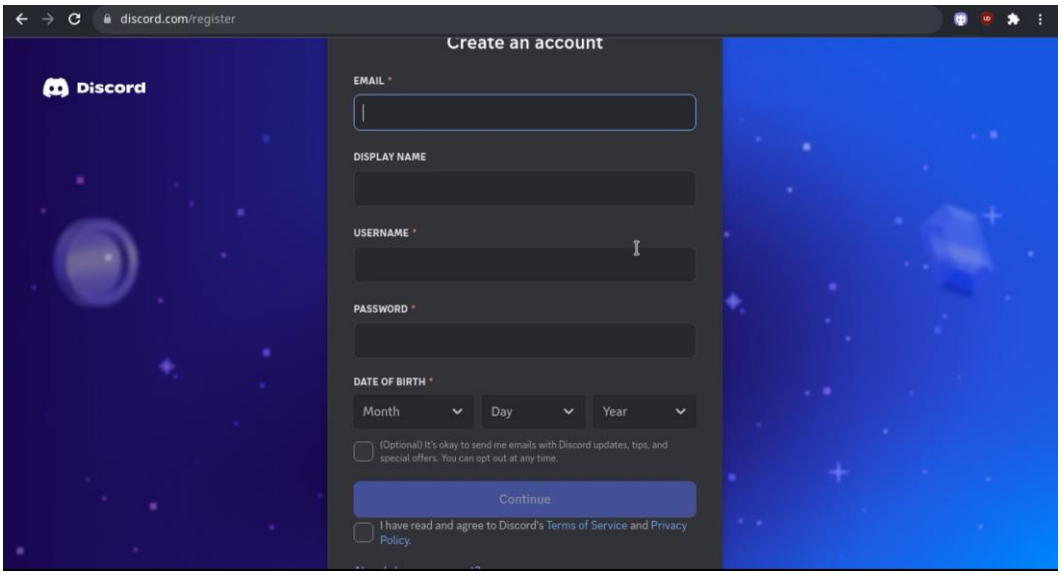


Рисунок 4.1 — Вікно реєстрації на офіційному сайті

4.1.1 Встановлення Discord на комп'ютери

Процес встановлення Discord на настільні операційні системи macOS, Windows та Linux є схожим.

Для встановлення на Windows або macOS, потрібно перейти на

офіційний сайт Discord у веб-браузері. На головній сторінці необхідно натиснути на відповідну кнопку завантаження, яка відрізняється в залежності від операційної системи. Після завантаження інсталяційного файлу необхідно запустити його. Інсталятор автоматично виконає процес встановлення. На macOS може знадобитися перетягнути іконку Discord в папку "Програми". На Linux після завантаження пакету .deb, необхідно відкрити термінал, перейти до каталогу із завантаженим файлом і виконати команду `sudo dpkg -i discord-*.deb`. Після завершення встановлення Discord запуститься, і з'явиться вікно входу або реєстрації.

4.1.2 Встановлення на мобільні пристрої

Встановлення Discord на мобільні пристрої здійснюється через відповідні магазини додатків.

Для встановлення на пристрої з операційною системою Android, треба відкрити Google Play Store. У пошуковому рядку знайти "Discord" і натиснути кнопку "Встановити". Після завершення встановлення іконка додатка з'явиться на головному екрані.

Для встановлення на пристрої з операційною системою iOS, необхідно відкрити App Store. У пошуковому рядку знайти "Discord" і натиснути "Отримати". Після завантаження та встановлення, іконка Discord з'явиться на головному екрані.

4.2 Додавання бота на сервер

Користувач може взаємодіяти з ботом через приватні повідомлення або будучи доданим до сервера. Кожен з варіантів має свої особливості встановлення, але більшість функцій призначена для використовування у приватних повідомленнях.

Додавання бота на сервер дозволяє усім учасникам взаємодіяти з

ботом. Щоб додати його на сервер необхідно перейти за посиланням з додаванням боту. Після переходу за посиланням відкриється вікно авторизації (рисунок 4.2). У цьому вікні потрібно буде вибрати сервер зі списку, на якому користувач має права адміністратора. Потім підтвердіть дозволи, які бот запитує (надсилання повідомлень, створення команд і.т.д.). Після підтвердження бот з'явиться у списку учасників сервера в офлайн або онлайн статусі, і усі учасники сервера зможуть взаємодіяти з ним у текстових каналах сервера. Також рекомендується створити окремий текстовий канал для користування ботом, якщо сервер розрахований на велику кількість учасників, щоб уникнути зловживання ботом та спамом у інших каналах.

Не варто надавати боту надмірні дозволи, якщо вони не є необхідними для його роботи.

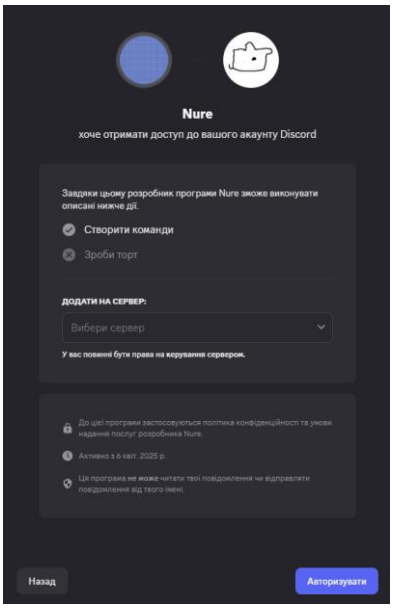


Рисунок 4.2 – Вікно авторизації бота до серверів

4.3 Додавання бота до приватних повідомлень

Додавання бота до приватних повідомлень дозволяє користувачеві взаємодіяти з ботом безпосередньо. Якщо користувач має нотатки або не хоче щоб хтось, крім нього, бачив розклад, який він викликав, то таке

використання підходить користувачу якнайкраще. До того ж, бот надсилає нагадування про заняття та нотатки лише до приватних повідомлень. Щоб додати бота до приватних повідомлень, необхідно проробити усі вище вказані дії на будь-якому сервері. Після чого можна буде виділити бота у списку учасників сервера (рисунок 4.3) та написати йому до приватних повідомлень.

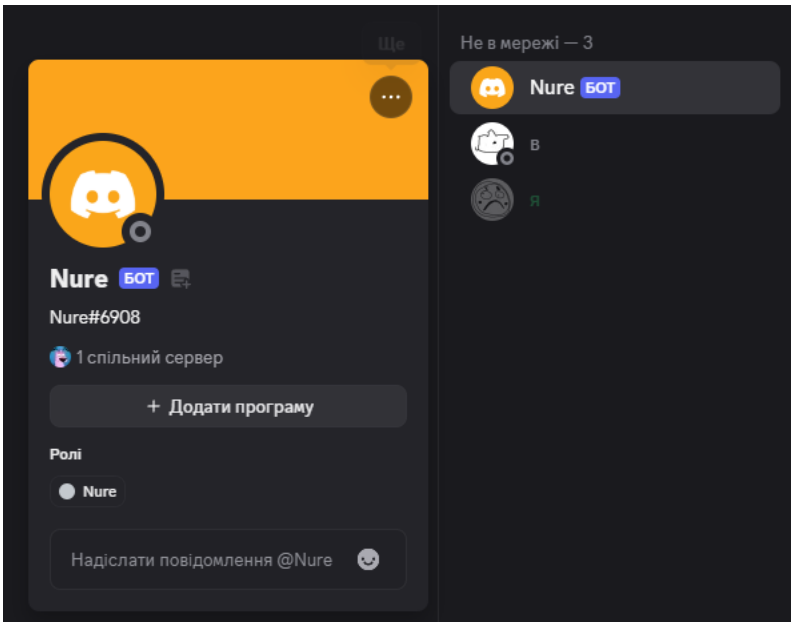


Рисунок 4.3 – Бот у списках учасників серверу

4.4 Використання бота та команди

Після успішного додавання бота до Discord, користувач може розпочати взаємодію з ним за допомогою спеціальних команд. Бот розроблений для спрощення доступу до розкладу занять та управління нотатками, завдяки набору команд. Усі команди активуються за допомогою префікса "!".

!група [назва групи] - команда дозволяє користувачу встановити або оновити свою академічну групу. Після введення команди, бот зв'язує Discord ID користувача з ID обраної групи у БД. При подальшому використанні команди !розклад бот автоматично показує розклад для групи, яку

користувач встановив раніше. Наприклад, щоб встановити групу "КІУКІ-21-2", необхідно ввести у чат або канал з ботом: "!група КІУКІ-21-2" (рисунок 4.4).

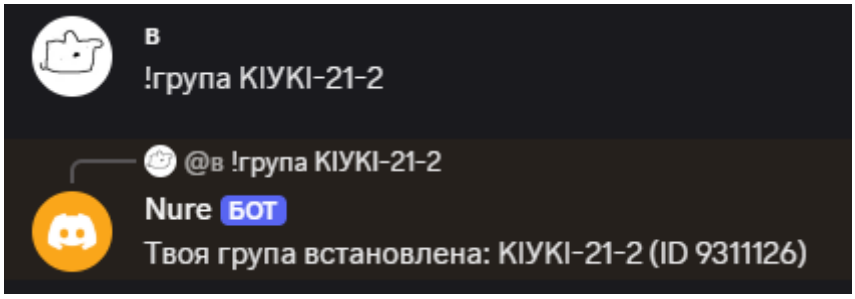


Рисунок 4.4 – Приклад виконання команди !група

!розклад - використання команди дозволяє переглянути розклад занять для обраної групи. Бот надсилає розклад на поточний тиждень у вигляді повідомлення, яке містить інтерактивні кнопки для навігації. Кнопка "ліворуч" дозволяє переглянути розклад попереднього тижня, а кнопка "праворуч" — наступного тижня. Також присутня кнопка "оновлення", яка запускає процес парсингу актуального розкладу з cist.nure.ua та його збереження до бази даних, після чого відображається оновлений розклад. Приклад виконання команди для групи КІУКІ-21-2 на тиждень 05.05.2025-11.05.2025 наведений на рисунку (рисунок 4.5).

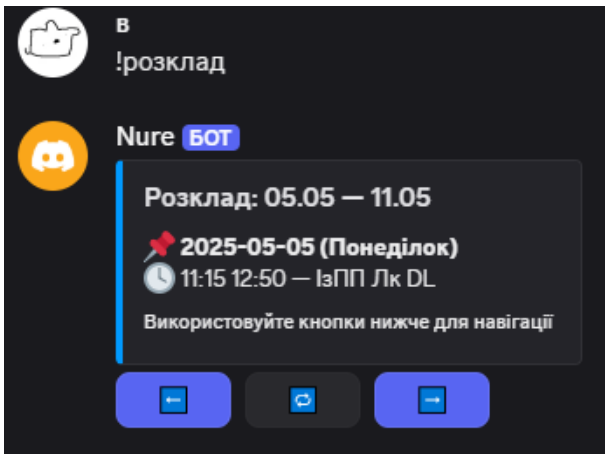


Рисунок 4.5 – Приклад виконання команди !розклад

!повідомлення - ця команда призначена для керування сповіщеннями.

Використовуючи її, користувач може увімкнути або вимкнути отримання автоматичних сповіщень про початок занять або нагадування про нотатки. Приклад виконання наведений нижче (рисунок 4.6).

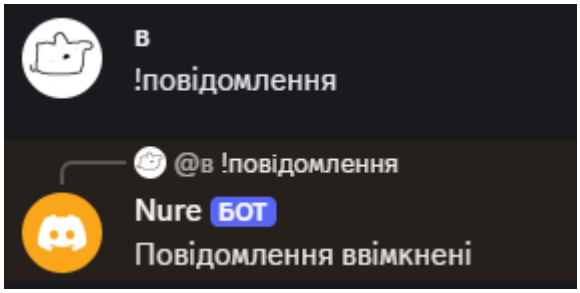


Рисунок 4.6 – Приклад виконання команди !повідомлення

!нотатка [текст] [дата у форматі YYYY-MM-DD HH:mm (необов'язково)] - команда використовується для створення нової нотатки. Після префікса вказується текст нотатки. Якщо потрібно, щоб бот нагадав про неї, можна додати дату та час у форматі РРРР-ММ-ДД ГГ:ХХ. Наприклад, !нотатка додаткові заняття 2025-05-07 18:00 (рисунок 4.7). Нотатки зберігаються у базі даних та прив'язані до діскорд айді користувача. Якщо користувач ввімкнув повідомлення, тоді бот буде повідомляти користувача про нотатки з датою в приватні повідомлення (рисунок 4.8).

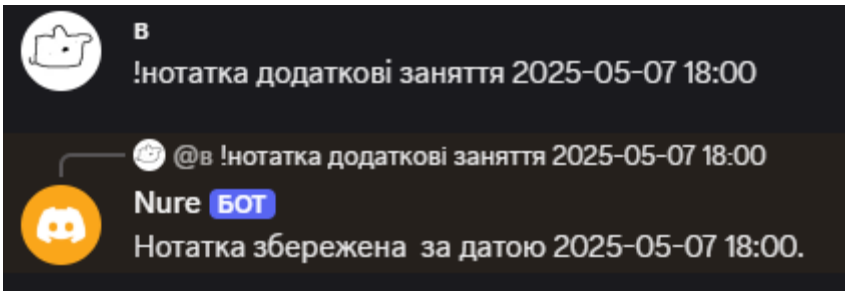


Рисунок 4.7 – Приклад виконання команди !нотатка

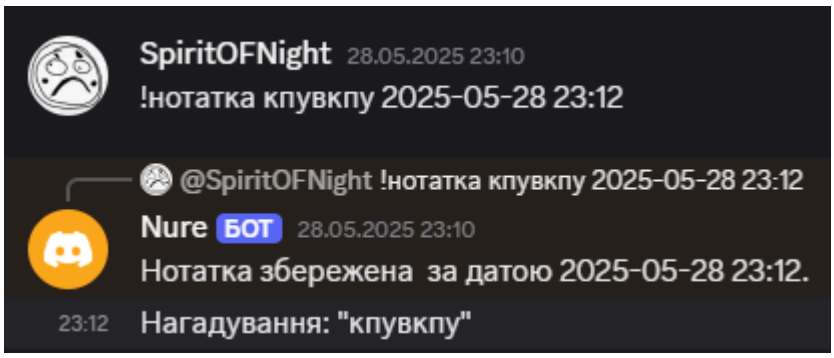


Рисунок 4.8 – Приклад повідомлення про нотатку

!нотатки - ця команда відображає список всіх збережених нотаток користувачем. Бот надсилає повідомлення з переліком усіх активних нотаток. Нотатки з датою позначені відповідною датою та спеціальним символом у дужках (рисунок 4.9).

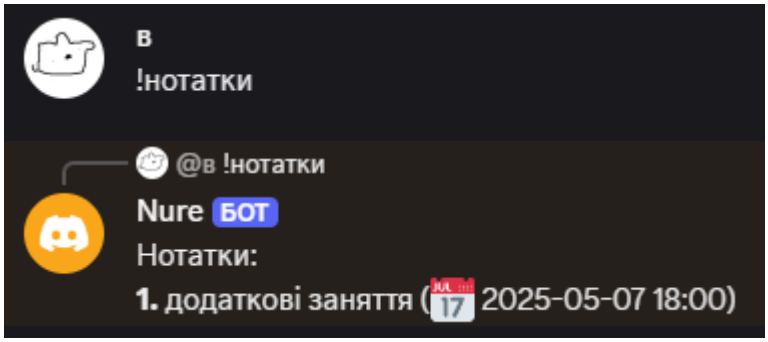


Рисунок 4.9 – Приклад виконання команди !нотатки

!видалити [номер] - для видалення конкретної нотатки з БД використовується команда !видалити. Після команди необхідно вказати номер нотатки, який відображається у списку при використанні команди !нотатки. Наприклад, !видалити 1 видалить першу нотатку зі списку (рисунок 4.10).

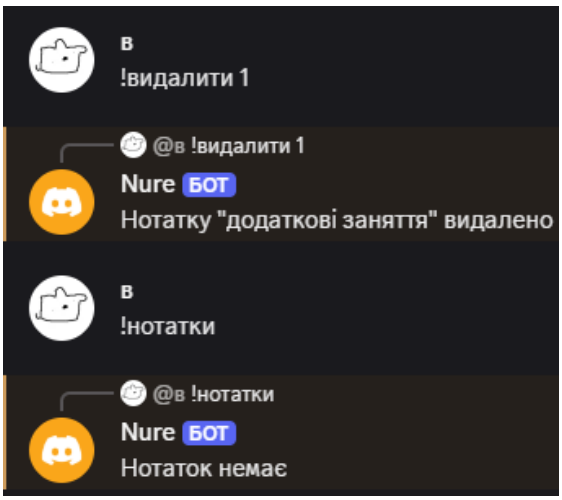


Рисунок 4.10 – Приклад виконання команди !видалити

!допомога - використання команди призводить до відображення короткого довідкового повідомлення, яке містить перелік усіх доступних команд бота та їх короткий опис.

Завдяки простому набору команд та інтерактивним елементам, як кнопки навігації, бот забезпечує користувачам доступ до розкладу занять та дозволяє зручно керувати особистими нотатками. Можливість персоналізації групи, отримання сповіщень та гнучке управління нагадуваннями надає перевагу над іншими, вже існуючими рішеннями. Він усуває необхідність постійного ручного відстеження розкладу та дозволяє тримати всі важливі нагадування під рукою в одному місці – безпосередньо у Discord.

ВИСНОВКИ

У кваліфікаційній роботі було успішно розроблено та реалізовано Discord-бота. Бот надає зручний спосіб користувачам швидко отримувати актуальний розклад для своєї групи, керувати сповіщеннями про початок занять та створювати персональні нагадування. Його функціональність охоплює такі команди, як !група - для вибору групи, !розклад - для перегляду розкладу з можливістю навігації по тижнях та оновленням даних, !повідомлення - для керування сповіщеннями, !нотатка - для створення нових нотаток з можливістю встановлення нагадувань, !нотатки - для перегляду списку всіх нотаток, !видалити - для їх видалення, а також !допомога - для швидкого доступу до інформації про команди.

Аналізуючи існуючі рішення було виявлено, що більша частина проектів або не пропонувала повного спектра необхідних функцій, або не підтримувала усі операційні системи. Бот має усунути усі недоліки, будучи універсальним для будь-якої операційної системи та маючи можливість нагадувати про заняття.

Основною мовою програмування виступив JavaScript та програмне середовище Node.js. Взаємодія з Discord API здійснювалась за допомогою бібліотеки discord.js, вона надавала необхідні інструменти для роботи з командами, взаємодіями та подіями у Discord. Для зберігання даних користувачів, їхніх груп, нотаток та самого розкладу була інтегрована та вбудована СУБД SQLite. Комбінація усіх технологій дозволила створити стабільного, розширюваного та простого у підтримці Discord-бота.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Discord. Вікіпедія. Discord. URL: https://uk.wikipedia.org/wiki/Discord (дата звернення: 17.05.2025).

2. App store. eUniversity. URL: https://apps.apple.com/ua/app/euniversity/id6739423337 (дата звернення: 17.05.2025).

3. Google play. eUniversity. URL: https://play.google.com/store/apps/details?id=com.nupp.euniversity (дата звернення: 17.05.2025).

4. Your Schedule In Your Phone: University Launches Telegram Chat Bot For Students. Національний університет "Полтавська Політехніка Імені Юрія Кондратюка". 31.08.2020. URL: https://nupp.edu.ua/en/news/your-schedule-in-your-phone-university-launches-telegram-chat-bot-for-students.html (дата звернення: 17.05.2025)..

5. Nodejs. about. URL: https://nodejs.org/en/about (дата звернення: 18.05.2025).

6. Npm Docs. About npm. URL: https://docs.npmjs.com/about-npm (дата звернення: 18.05.2025).

7. MDN Web Docs. JavaScript. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript (дата звернення: 18.05.2025).

8. Techtarget. Node.js (Node). URL: https://www.techtarget.com/whatis/definition/Nodejs (дата звернення: 18.05.2025).

9. Bun. What is Bun?. URL: https://bun.sh/docs (дата звернення: 18.05.2025).

10. Вікіпедія. Раян Дал. URL: https://uk.wikipedia.org/wiki/%D0%A0%D0%B0%D1%8F%D0%BD_%D0%94%D0%B0%D0%BB (дата звернення: 18.05.2025).

11. Deno Land Inc. Deno. URL: https://deno.com/ (дата звернення: 18.05.2025).

12. Discord.js. URL: https://discord.js.org/docs/packages/discord.js/main (дата звернення: 18.05.2025).

13. Discord.py. URL: https://discordpy.readthedocs.io/en/stable/ (дата звернення: 18.05.2025)

14. SQLite. URL: https://www.sqlite.org/index.html (дата звернення: 18.05.2025)

15. Github. node-sqlite3. URL: https://github.com/TryGhost/node-sqlite3 (дата звернення: 18.05.2025).

16. PostgreSQL. URL: https://www.postgresql.org/ (дата звернення: 18.05.2025)

17. MySQL. URL: https://www.mysql.com/ (дата звернення: 18.05.2025)