

УДК 004.75

DOI <https://doi.org/10.32782/tnv-tech.2023.2.9>

РОЗРОБКА ВЕБ-ДОДАТКІВ ТА СЕРВІСІВ НА ПЛАТФОРМІ NODE.JS

Кошова О. П. – кандидат педагогічних наук,
доцент кафедри комп'ютерних наук та інформаційних технологій
Полтавського університету економіки і торгівлі
ORCID ID: 0000-0003-0794-6774

Ольховська О. В. – кандидат фізико-математичних наук,
завідувач кафедри комп'ютерних наук та інформаційних технологій
Полтавського університету економіки і торгівлі
ORCID ID: 0000-0001-5366-5995

Тацій Д. С. – здобувач освіти спеціальності «Комп'ютерні науки»
Полтавського університету економіки і торгівлі
ORCID ID: 0009-0001-9146-4833

Олексійчук Ю. Ф. – кандидат фізико-математичних наук,
доцент кафедри комп'ютерних наук та інформаційних технологій
Полтавського університету економіки і торгівлі
ORCID ID: 0000-0002-0585-3307

Черненко О. О. – кандидат фізико-математичних наук,
доцент кафедри комп'ютерних наук та інформаційних технологій
Полтавського університету економіки і торгівлі
ORCID ID: 0000-0002-9084-0999

У роботі зроблено огляд систем дистанційного навчання, виділено їх позитивні та негативні сторони, також описано теорію та практичне застосування платформи Node.js. Сформульовано вимоги до функціональності для розроблюваного програмного забезпечення. Виконано опис проектних рішень, інструментів та підходів до розробки програмного забезпечення для системи дистанційного завдання у вигляді сервісу телеграм-боту для перегляду розкладу. Спроектовано та розроблено телеграм-бот для перегляду розкладу занять, як окремий сервіс.

Реалізація сервісу перегляду розкладу через телеграм-бот включає розробку як сервісу телеграм-боту, так і серверної логіки системи дистанційного навчання.

Після того, як користувач авторизувався в системі, в нього повинен з'являтися доступ до телеграм-бота, за допомогою якого він буде мати можливість переглядати розклад за власними групами.

Для того, щоб почати роботу з телеграм-ботом, користувач повинен перейти до налаштувань свого облікового запису в системі, де має бути кнопка для підключення до телеграм-боту, після натискання якої повинен здійснитися перехід до сторінки з телеграм-ботом для подальшої авторизації користувача в ньому. Після успішної авторизації телеграм-бот надає доступ до клавіатури головного меню, де знаходяться три кнопки: «Розклад на сьогодні», «Розклад на завтра» та «Розклад на тиждень». Після натискання кнопки «Розклад на сьогодні» телеграм-бот повинен відправити розклад на сьогоднішній день за всіма курсами користувача, відповідно до назви і у випадку натискання на кнопку «Розклад на завтра», також у обох випадках повинна бути клавіатура під повідомленням про розклад, що дозволить перемикатися між днями. В свою чергу після натискання кнопки «Розклад на тиждень», повинен бути відправлений розклад на поточний тиждень, і також з можливістю перемикатися до наступних або попередніх тижнів за допомогою стрілок вниз повідомлення. Якщо розкладу немає або його неможливо отримати, то телеграм-бот повинен повідомлення про відсутність розкладу.

Розроблене програмне забезпечення системи дистанційного навчання розміщено на хост сервісі, до нього мають доступ користувачі в будь-який час. Програмний продукт є результатом автоматизації навчального процесу у дистанційній формі. Саме тому він покриває основні потреби студентів та викладачів у навчальному процесі. Результати розробки впроваджено в початковий процес Полтавського університету економіки і торгівлі.

Ключові слова: бібліотека, веб-додаток, цикл подій, модуль.

Koshova O. P., Olkhovska O. V., Tatsii D. S., Oleksiichuk Yu. F., Chernenko O. O. Development of web-applications and services powered by Node.js

The work provides an overview of distance learning systems, highlights their positive and negative sides, and also describes the theory and practical application of the Node.js platform. Functionality requirements for the developed software are formulated. A description of design solutions, tools and approaches to the development of software for a remote task system in the form of a telegram service for viewing the schedule has been made. Designed and developed a Telegram bot for viewing the schedule of classes as a separate service.

The implementation of the schedule viewing service through the Telegram bot includes the development of both the Telegram bot service and the server logic of the distance learning system.

After the user has been authorized in the system, he should have access to the Telegram bot, with the help of which he will be able to view the schedule by his own groups.

In order to start working with the Telegram bot, the user must go to the settings of his account in the system, where there should be a button to connect to the Telegram bot, after clicking which a transition should be made to the page with the Telegram bot for further authorization of the user in him. After successful authorization, the Telegram bot provides access to the main menu keyboard, where there are three buttons: "Schedule for today", "Schedule for tomorrow" and "Schedule for a week". After pressing the "Schedule for the hour" button, the telegram bot should send the schedule for today for all courses of the user, according to the name, and in the case of clicking the button "Schedule for tomorrow", also in both cases there should be a keyboard under the message about the schedule, which will allow you to switch between days. In turn, after pressing the "Schedule for the week" button, the schedule for the current week should be sent, and it is also possible to switch to the next or previous weeks using the arrows at the bottom of the message. If there is no schedule or it cannot be obtained, the Telegram bot should report that there is no schedule.

The developed software of the distance learning system is placed on the host service, users can access it at any time. The developed system is the result of the automation of the educational process in a remote form. That is why it covers the basic needs of students and teachers in the educational process. The results of the development were implemented in the initial process of the Poltava University of Economics and Trade.

Key words: library, web-application, event loop, module.

Актуальність роботи. Досвід останніх років показав, що питання розвитку дистанційного навчання є досить актуальним [1–4], адже такий вид навчання допомагає у вирішенні проблеми, коли люди не можуть перебувати всі разом в одній аудиторії. Дистанційна форма навчання дозволяє приймати участь у освітньому процесі з будь-якої точки планети як студентам, так і викладачам, що є вагомим плюсом, бо надає змогу повноцінно навчатися або викладати людям, які живуть у важкодоступній місцевості, знаходяться за кордоном тощо. Також до переваг такої форми навчання можна віднести можливість підлаштувати навчання під свій індивідуальний графік, що дозволить навчатись у власному темпі та у зручний час, бо студент, навіть будучи відсутнім на занятті, має доступ до всіх матеріалів, які він може перечитати в будь-який момент, також може переглянути відео-запис лекції та зв'язатися з викладачем online, якщо в нього виникнуть питання. Отже, за допомогою дистанційного навчання студент одночасно може працювати та навчатися в декількох навчальних закладах.

Система дистанційного навчання не тільки забезпечує вище перелічені можливості, а й може надавати деякі додаткові: онлайн-тестування, архівація пройдених

курсів та оцінок студентів, чат з викладачем, онлайн-журнал оцінок, email-розсилка тощо. Наприклад, онлайн-тестування може зекономити час викладачеві на перевірку, адже система сама порахує кількість правильних відповідей та виставить оцінку, або її може виставити викладач за результатами тестування. У свою чергу онлайн-журнал може допомогти студенту дізнатися оцінки за здані роботи та оцінити свою успішність. Якщо студентам або викладачу потрібно переглянути інформацію за пройдені курси, то тут стане у пригоді архівація курсів, що дозволить переглянути матеріали, завдання та оцінки. Якщо ж у студента чи викладача виникають запитання, то онлайн-чат може допомогти у їх вирішенні. Також email-розсилка надає можливість студентам дізнаватися про нові завдання, оцінки, повідомлення тощо. Окрім цього, системи дистанційного навчання також можуть надати можливість взаємодії з ними через ботів в месенджерах, наприклад: телеграм-бот для швидкого та зручного перегляду розкладу занять, бо користувачу не потрібно буде для цього постійно заходити на сайт системи, що буде економити його час.

Огляд найбільш використовуваних на сьогодні онлайн-платформ, зокрема, Google Classroom [5], Moodle [6], Canvas [7], для навчання показав, що всі вони мають як переваги, так і недоліки.

Зважаючи на вказану вище актуальність, було вирішено розробити програмне забезпечення системи дистанційного навчання, а саме одну з її частин: функціональність перегляду розкладу занять за допомогою телеграм-боту.

Виклад основного матеріалу дослідження. Беручи до уваги постановку завдання та специфіку системи, що розробляється, а також в результаті аналізу існуючих інструментів та проектних рішень, було вирішено використати наступні засоби та інструменти розробки:

- платформа: Node.js;
- мова програмування: TypeScript;
- сервіс телеграм-боту: Telegraf, Axios;
- серверна частина: NestJs, Passport, TypeORM, Swagger;
- СУБД: PostgreSQL;
- інструмент адміністрування та розробки для PostgreSQL: pgAdmin;
- розподілене сховище даних, що зберігає інформацію в пам'яті: Redis;
- розподілена система контролю версій: Git;
- середовище розробки: Visual Studio Code;
- інструмент розробки дизайну: Figma;

Node.js – це середовище, із відкритим програмним кодом, для запуску додатків написаних на мові програмування JavaScript. Node.js кросплатформний, що показує його можливість працювати на різних операційних системах, таких як: Windows, Linux, Unix, macOS тощо [8].

Node.js був створений американським розробником Райаном Далом. Основною причиною, що підштовхнула Райана до створення Node.js, була відсутність у серверних платформ можливості ефективної обробки користувацьких запитів до системи вводу/виводу, оскільки це відбувалося шляхом виділення окремого потоку на кожний запит, що призводило до неефективного розподілу ресурсів, тому що код виконувався синхронно і виділений потік простоював не використовуючи всі виділені ресурси на повну. Node.js покликаний вирішити цю проблему за допомогою своєї неблокуючої моделі обробки запитів [9].

Неблокуюча модель обробки запитів означає можливість серверу безперервно обробляти користувацькі запити, не очікуючи виконання попередніх. Це стало

можливим завдяки одній з основних частин Node.js: бібліотеки `libuv`. Бібліотека `libuv`, яка написана на мові C і була створена для реалізації асинхронного, неблокуючого доступу до системи введення та виводу, за допомогою однієї з основних особливостей Node.js – Event loop (цикл подій), Event loop – це механізм неперервної обробки подій, які викликаються в процесі роботи серверу, що дає йому можливість працювати асинхронно. Event loop постійно очікує нову подію і, коли така трапляється, то він починає виконувати весь код, що стоїть за нею. У разі виклику блокуючої операції (операція вводу/виводу, що виконується бібліотекою `libuv` на мові C) передає її виконання у окремий потік, після чого продовжує очікувати нові події. Таким чином, Event loop обробляє всі запити в одному потоці і тільки за потреби передає у окремий потік, хоча кількість цих потоків обмежена, і цим забезпечує безперервну обробку користувачьких запитів до серверу. З огляду на все вище описане можна зробити висновок, що Node.js може обробляти одночасну велику кількість запитів на сервер, що робить його досить ефективним та невимогливим до ресурсів серверної машини. Єдиний мінус Node.js – складність обробки важких обчислень, адже, як було зазначено раніше, дана платформа працює в одному потоці, і якщо це не блокуюча операція, то вона виконується в цьому одному потоці, а інші операції чекають її виконання, що може уповільнити роботу серверу [10–12].

Оскільки бібліотека `libuv` написана на мові програмування C, то щоб мова JavaScript могла використовувати інтерфейс цієї бібліотеки, їй потрібен засіб, що буде виконувати роль «мосту». В Node.js цим мостом є рушій JavaScript під назвою V8. V8 було створено компанією Google для запуску та ефективного виконання коду, написаного на мові програмування JavaScript, в браузері. Розробники Node.js вдалося дістати цей рушій прямо з браузера і зробити можливим ефективний запуск коду JavaScript поза межами браузера з можливістю комунікувати з операційною системою серверу за допомогою вище описаної бібліотеки `libuv`.

З огляду на те, що Node.js використовує мову програмування JavaScript, це дає можливість написання клієнтських та серверних додатків на одній мові, що є безумовним плюсом, адже розробник може розробляти повноцінні веб-додатки без вивчення нових мов.

Також невід'ємною частиною Node.js є модулі [12], адже будь-який проект складається з окремих, пов'язаних між собою, модулів. Модуль – це окремий файл з певним набором класів, функцій та змінних, які можуть бути використані в будь-якій іншій частині проекту. Модуль може бути представлений у вигляді одного окремого самостійного файлу, або файлу, що використовує інші модулі. В Node.js є три типи модулів: основні, локальні та сторонні. Основні – це модулі, які вбудовані в Node.js і надають базовий набір інструментів для розробки додатків, наприклад: можливість проводити операції з файлами, розгорнути веб-сервер тощо. Локальні – це модулі, які створюються самим розробником в середині проекту. Сторонні – це модулі, що створюються для реалізації певної функціональності, яку можна буде використовувати в інших проектах як бібліотеку чи фреймворк.

Серед основних модулів Node.js є декілька найважливіших [10]:

- 1) `http` – містить набір інструментів для створення та розгортання веб-серверу;
- 2) `util` – містить функції, які можуть допомогти при розробці додатків, або інших модулів;
- 3) `fs` – містить набір інструментів для операцій пов'язаних з файлами;
- 4) `querystring` – містить функції для роботи з URL-параметрами;
- 5) `stream` – містить набір інструментів для роботи з потоками даних.

Для розповсюдження сторонніх модулів для Node.js використовується менеджер пакетів NPM (Node Package Manager), що дозволяє будь-кому опублікувати власний модуль в публічний доступ, і відповідно завантажити його будь-кому у власний проект. NPM є популярним і має величезну спільноту, тому містить велику кількість різних пакетів бібліотек та фреймворків, що дає можливість з великою долею вірогідності знайти потрібне рішення під конкретну задачу.

За допомогою Node.js можна створювати різні додатки або сервіси, наприклад: месенджери, соціальні мережі, односторінкові веб-додатки, стрімінгові сервіси, платіжні сервіси тощо.

На даний момент серед найвідоміших компаній, що використовують Node.js можна побачити: NASA, Trello, Netflix, PayPal, LinkedIn, Walmart, Uber, Twitter, Yahoo, eBay та GoDaddy.

Беручи до уваги все вище описане, було вирішено використати Node.js в рамках розробки системи дистанційного навчання, а саме завдання по розробці функціональності розкладу у вигляді телеграм-боту, як окремого сервісу для забезпечення повноти роботи системи, що буде складатися з серверної частини для керування даними розкладу та самого телеграм-боту для його перегляду.

Для реалізації поставленого завдання використано сторонній модуль (фреймворк) для Node.js під назвою NestJS, що дозволяє значно спростити реалізацію серверних додатків, шляхом розділення всією функціональності на окремі модулі, які містять: контролери для обробки запитів, сутності, що описують поля для бази даних; і сервіси, що витягують, оброблюють та передають дані між контролерами та базою даних у вигляді сутностей.

Наступним кроком була реалізація доповнення розробленого модуля для роботи з курсами у вигляді: нових обробників контролера з курсів, що приймають запити на створення, отримання, редагування та видалення даних про сутності розкладу; сутність розкладу, яка описує наступні поля: дата, час початку, час закінчення; і сервісу, що оброблює та передає дані про розклад між контролером та базою даних.

Далі було додано функціональність авторизації користувача системи в телеграм-боті, шляхом реалізації наступних частин: сервіс та модуль кешування даних, для зберігання тимчасового ключа, який використовується для авторизації в телеграм-боті; перехоплювачі користувачських запитів отримання розкладу для перевірки на те, чи справді ці запити відправляються з телеграм-боту авторизованим в ньому користувачем; сервіс для роботи із сутністю розкладу та відповідний обробник для авторизації в самому телеграм-боті.

Насамкінець, розроблено сервіс для отримання розкладу в телеграм-боті та функціональність його виводу користувачу в телеграм-чаті.

В результаті було реалізовано повноцінний телеграм-бот для перегляду розкладу та серверну частину для його комунікації з системою дистанційного навчання.

Процес роботи з телеграм-ботом можна описати наступним чином:

1) Користувач заходить в систему дистанційного навчання, переходить до сторінки налаштування власного облікового запису та натискає на кнопку «Підключити Telegram-розклад» (рис. 1).

2) Після натискання кнопки на сервері створюється телеграм-токен і користувача перенаправляє до сторінки з телеграм-ботом, де в посиланні буде вказано цей токен (рис. 2).

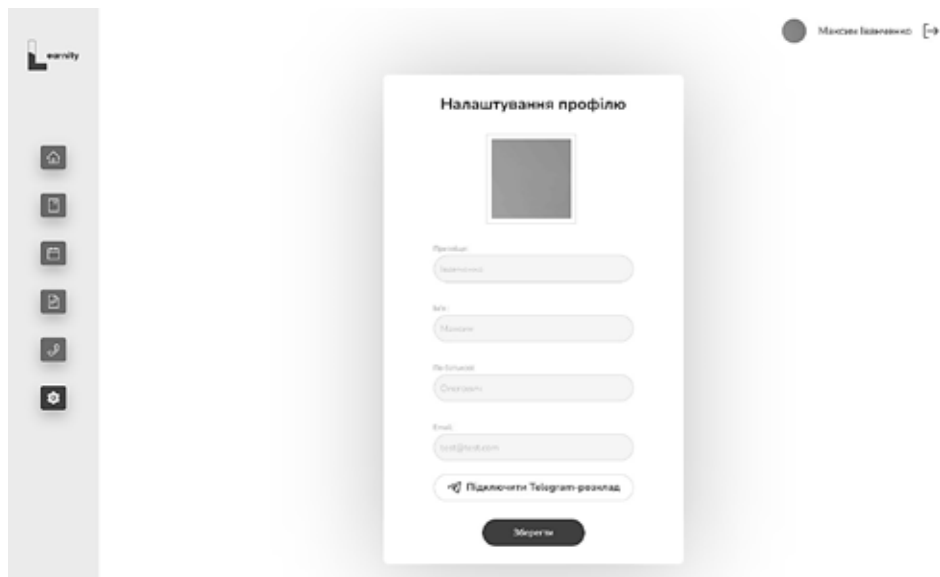


Рис. 1. Сторінка налаштування облікового запису

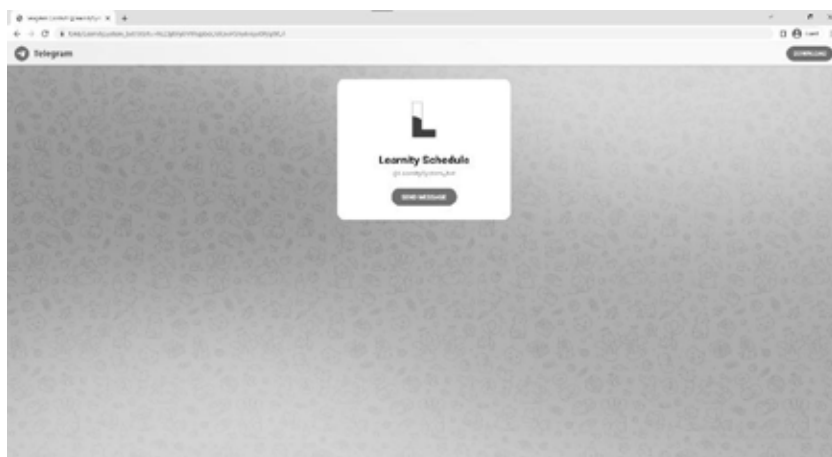


Рис. 2. Сторінка телеграм-боту з токеном у посиланні

3) Далі користувач переходить до чату з телеграм-ботом, де після початку листування буде здійснено спробу авторизації за допомогою серверу системи дистанційного навчання і, у випадку успіху, бот відправить повідомлення з привітанням та головним меню (рис. 3). У випадку помилки під час авторизації буде відправлено відповідне повідомлення з помилкою (рис. 4). Після успішної авторизації сервер зберігає телеграм-ідентифікатор користувача в базі даних і в подальшому звіряє його з тим, який приходить від телеграм-боту, щоб впевнитися, що бот використовується авторизованим користувачем. Також у користувача з'явиться можливість швидкого переходу до чату з телеграм-ботом з головної сторінки системи за допомогою кнопки «Telegram-розклад» (рис. 5).

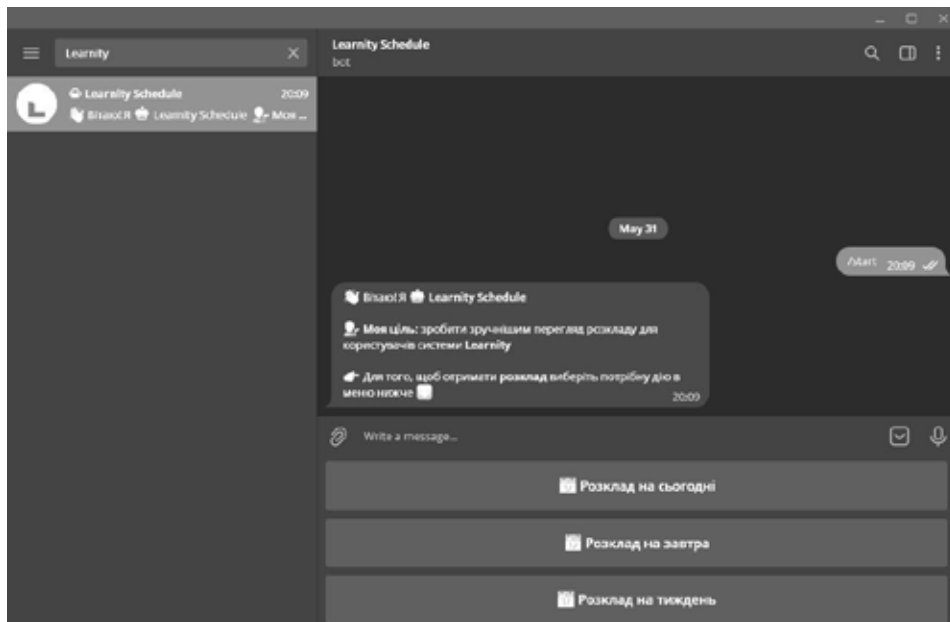


Рис. 3. Повідомлення з привітанням

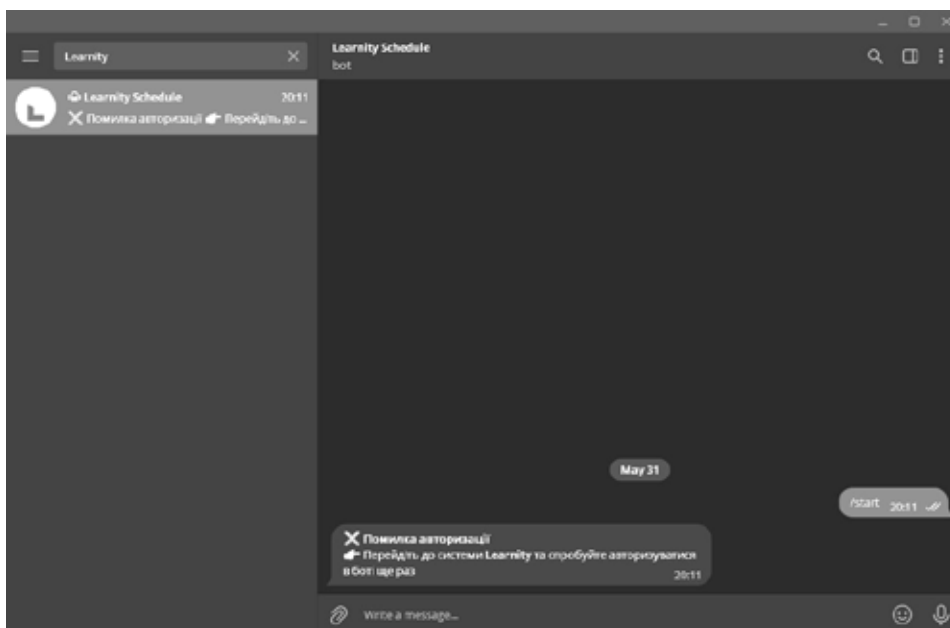


Рис. 4. Повідомлення про помилку авторизації

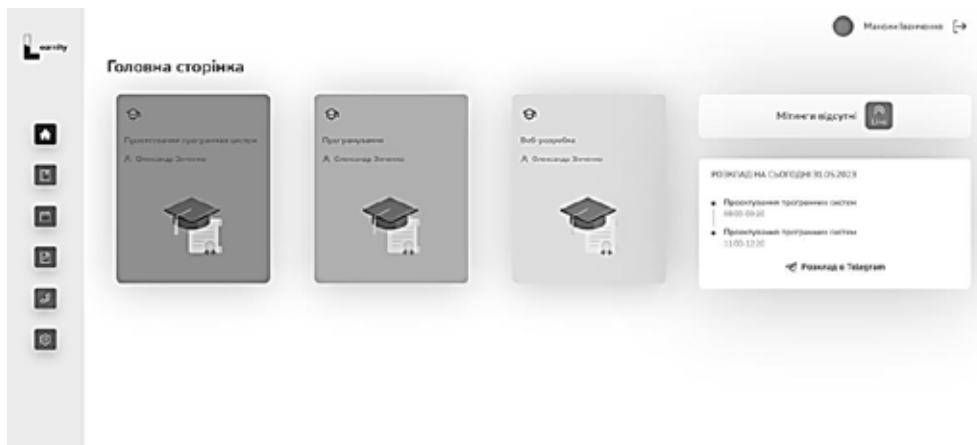


Рис. 5. Головна сторінка системи дистанційного навчання

4) Тепер користувач має змогу повноцінно використовувати телеграм-бота для перегляду свого розкладу занять системи дистанційного навчання, для цього потрібно натискати відповідні кнопки у головному меню, наприклад: «Розклад на сьогодні», після чого з'явиться відповідне повідомлення з розкладом на сьогодні (рис. 6). Для інших кнопок будуть відправлятися відповідні повідомлення подібні до повідомлення з розкладом на сьогодні (рис. 7).

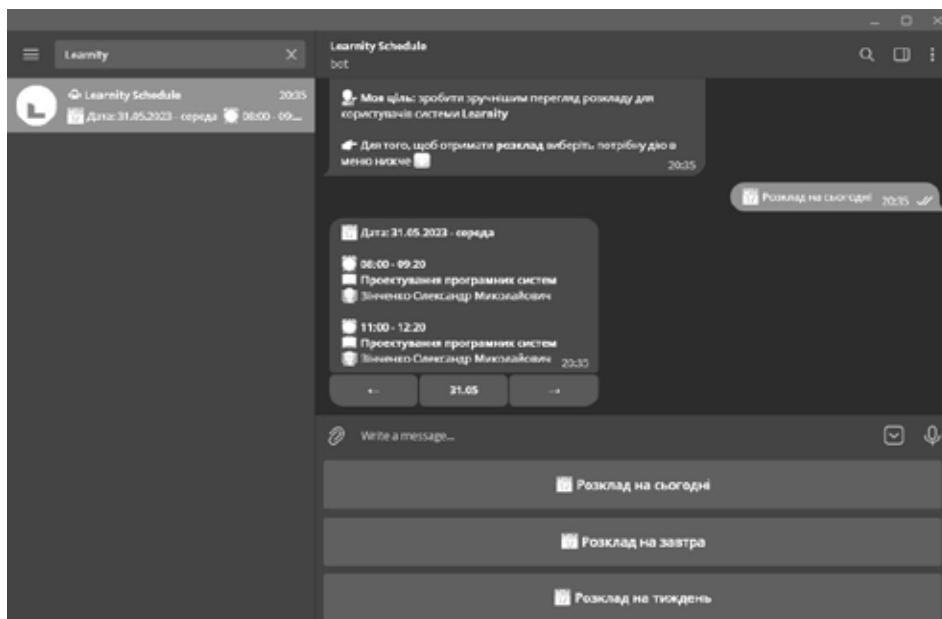


Рис. 6. Повідомлення з розкладом на сьогодні

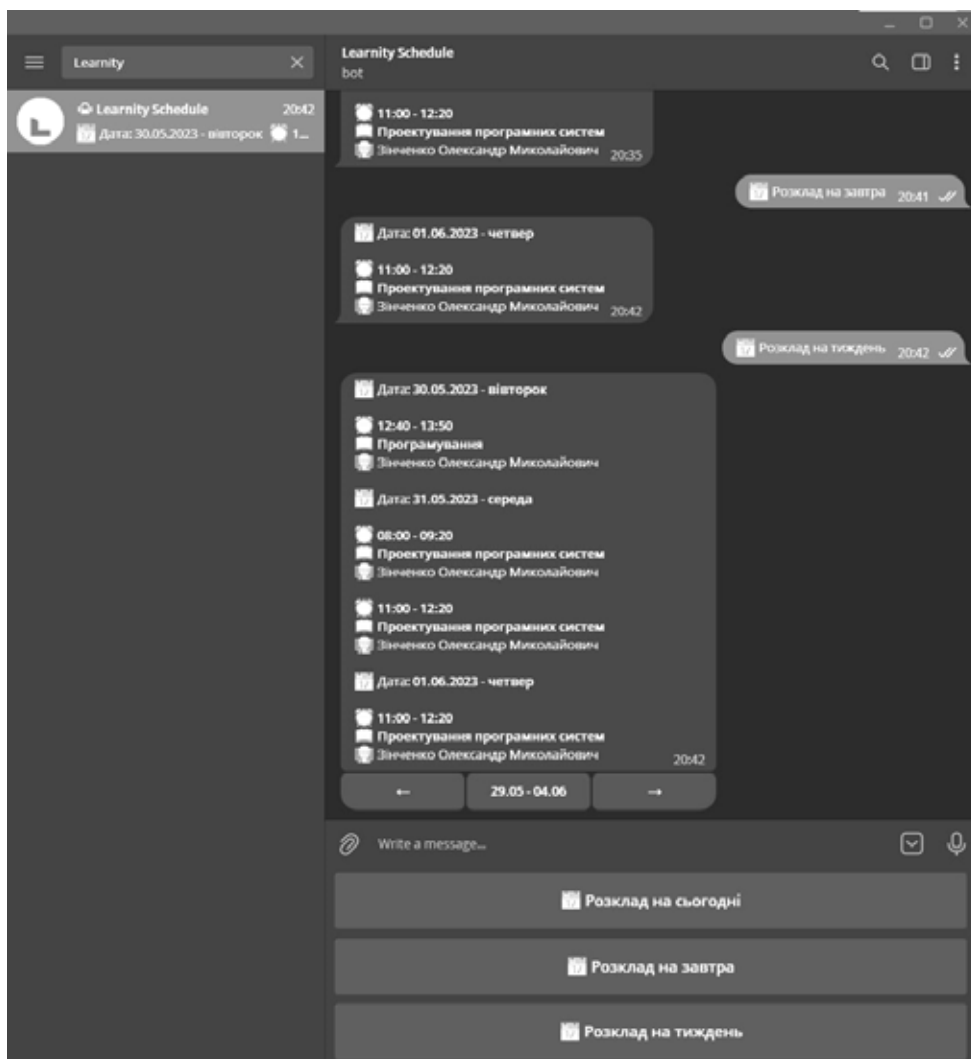


Рис. 7. Повідомлення про розклад на завтра і на тиждень

5) Також в телеграм-боті присутня можливість гортати розклад, натискаючи на відповідні стрілки під останнім повідомленням. При натисканні такої стрілки в залежності від вибраного виду розкладу (тиждень для «Розклад на тиждень» та день для «Розклад на сьогодні» і «Розклад на завтра»), користувачу буде відправлено розклад на наступний або попередній день чи тиждень, а кнопки зі стрілками буде прибрано з минулого повідомлення (рис. 8). Якщо розкладу не буде знайдено на потрібну дату, то буде виведено відповідне повідомлення (рис. 9).

Отже, враховуючи теоретичні відомості та практичне застосування платформи Node.js у розробці різних веб-застосунків та сервісів, можна зробити висновок, що ця платформа є досить гнучкою та дозволяє розробляти програмне забезпечення під різні цілі: від сайтів до мобільних додатків, причому не потребуючи великих витрат на серверне обладнання для ефективної роботи.

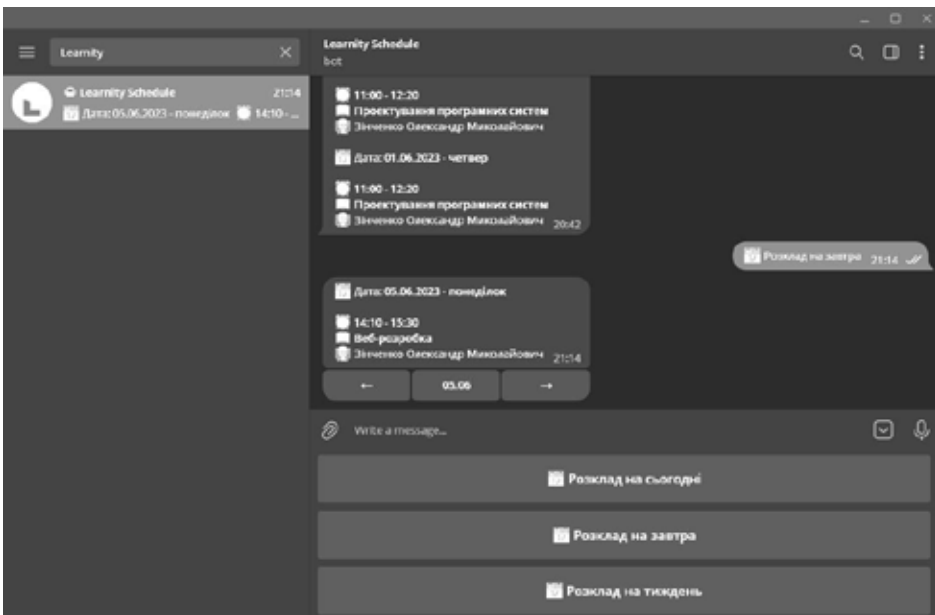


Рис. 8. Повідомлення про розклад на декілька днів вперед

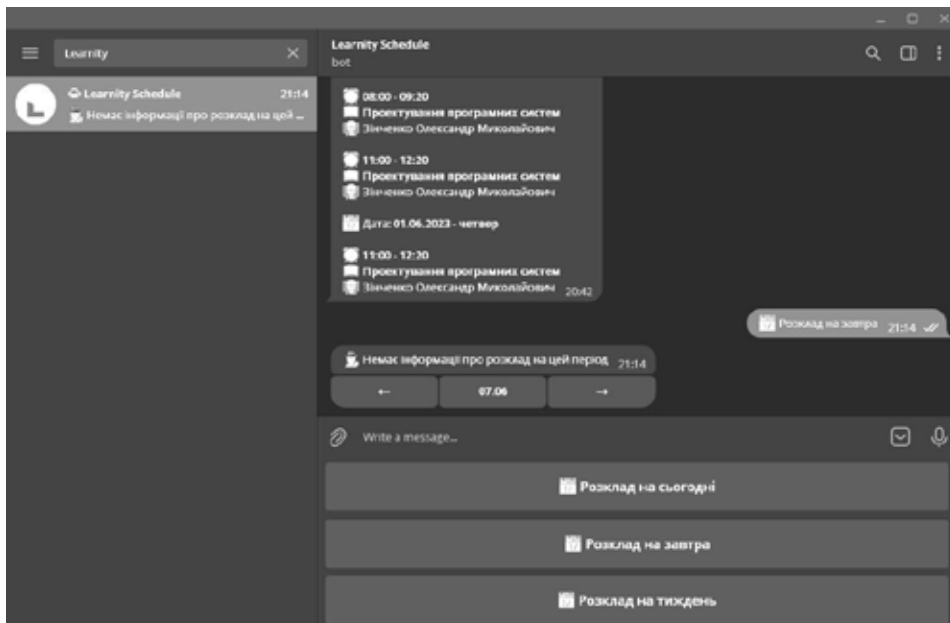


Рис. 9. Повідомлення про відсутність розкладу

Висновки. В наш час питання організації дистанційного навчання є досить актуальним, бо воно допомагає людині безперешкодно здобувати знання, вирішуючи проблеми: часу, доступності, місцезнаходження та комунікації. З огляду на цю актуальність і було вирішено провести детальний аналіз цієї області та скласти

список технологій та інструментів, потрібних для розробки програмного забезпечення системи дистанційного навчання, саме її частини у вигляді функціональності перегляду розкладу за допомогою телеграм-бота. Результатом проведеної роботи є розробка програмного забезпечення системи дистанційного навчання, а саме її частини у вигляді телеграм-бота для Полтавського університету економіки та торгівлі з урахуванням поставлених вимог. Проведено тестування усіх модулів та доступних можливостей програмного продукту. В перспективі планується працювати над його удосконаленням.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Гайдай Г. Ю., Грешнов А. Ю. Комп'ютерна система для проведення лабораторних досліджень у дистанційному форматі (розробка серверної частини веб-додатку). *Таврійський науковий вісник. Серія: Технічні науки*, 2023, (6), 10–17. <https://doi.org/10.32851/tnv-tech.2022.6.2>
2. Слабінога М. О., Чабан С. В. Розробка веб-додатків в контексті оптимізації їх швидкодії. *Таврійський науковий вісник. Серія: Технічні науки*, 2022, (3), 63–69. <https://doi.org/10.32851/tnv-tech.2022.3.7>
3. Круглик В. С., Астаф'єв В. Ю. Особливості реалізації семантичної нейронної мережі створення генератора навчальних кросвордів. *Вісник Кременчуцького національного університету імені Михайла Остроградського*. Кременчук : КрНУ, 2021. Випуск 2(127), С. 81–88.
4. Almahamid, S. M., Tweiqat, A. F., Almanaseer, M. S. University website quality characteristics and success: lecturers' perspective. *International Journal of Business Information Systems*, 2016, 22(1), P. 41–61.
5. About Classroom – Classroom Help. URL: <https://support.google.com/edu/classroom/answer/6020279> – Назва з екрану.
6. Moodle App Overview. URL: <https://moodledev.io/general/app/overview> – Назва з екрану.
7. What is Canvas? URL: <https://community.canvaslms.com/t5/Canvas-Basics-Guide/What-is-Canvas/ta-p/45> – Назва з екрану.
8. Node.js. URL: <https://nodejs.org/en> (дата звернення 15.04.2023).
9. Taha Sufiyan. What is Node.js: A Comprehensive Guide. URL: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs> (дата звернення 15.04.2023).
10. Bonjar Basumatary, Nishant Agnihotri. Benefits and Challenges of Using NodeJS. *International Journal of Innovative Research in Computer Science & Technology*. URL: <https://www.ijirest.org/DOC/13-benefits-and-challenges-of-using-nodejs.pdf> (дата звернення 16.04.2023).
11. Nadiia Shevchuk. Is Node.js Still Relevant for Startup Development in 2023? URL: <https://www.altamira.ai/blog/node-js-for-startup-development-in-2023> (дата звернення: 16.04.2023).
12. Олександр Ветошкін. Як працює Event Loop в NodeJS: внутрішня будова, фази та приклади. URL: <https://dou.ua/forums/topic/31698> (дата звернення 16.04.2023).
13. Коротун О., Вакалюк Т., Зубрицький В., Гордієнко І. Теоретичні аспекти розробки системи управління навчанням. *Таврійський науковий вісник. Серія: Технічні науки*, 2022, (1), 36–46. <https://doi.org/10.32851/tnv-tech.2022.1.4>

REFERENCES:

1. Haidai, H. Yu., Hrieshnov, A. Yu. (2023) Kompiuterna systema dlia provedennia laboratornykh doslidzhen u dystantsiinomu formati (rozrobka servernoi chastyny veb-dodatku) [Computer system for conducting laboratory research in a remote format (development of the server part of the web application)]. *Tavriiskyi naukovyi visnyk*.

- Seriia: Tekhnichni nauky [Taurian Scientific Bulletin. Series: Technical sciences]*, (6), 10–17. <https://doi.org/10.32851/tnv-tech.2022.6.2> [in Ukrainian].
2. Slabinoha, M. O., Chaban, S. V. (2022) Rozrobka veb-dodatkov v konteksti optymizatsii yikh shvydkodii [Development of web applications in the context of optimizing their speed]. *Tavriiskyi naukovyi visnyk. Seriia: Tekhnichni nauky [Taurian Scientific Bulletin. Series: Technical sciences]*, (3), 63–69. <https://doi.org/10.32851/tnv-tech.2022.3.7> [in Ukrainian].
3. Kruhlyk, V. S., Astafiev, V. Yu. (2021) Osoblyvosti realizatsii semantychnoi neironnoi merezhi stvorennia heneratora navchalnykh krosvordiv [Peculiarities of implementing a semantic neural network for creating an educational crossword generator]. *Visnyk Kremenchut'skoho natsionalnoho universytetu imeni Mykhaila Ostrohrad'skoho [Visnyk Kremenchug National University named after Mykhailo Ostrogradskyi]*. Kremenchuk: KrNU. Vypusk 2 (127). P. 81–88 [in Ukrainian].
4. Almahamid S. M., Tweiqat A. F., Almanaseer M. S. (2016) University website quality characteristics and success: lecturers' perspective. *International Journal of Business Information Systems*, 22(1), P. 41–61 [in English].
5. About Classroom – Classroom Help. URL: <https://support.google.com/edu/classroom/answer/6020279> [in English].
6. Moodle App Overview. URL: <https://moodledev.io/general/app/overview> [in English].
7. What is Canvas? [Електронний ресурс]. – Режим доступу: URL: <https://community.canvaslms.com/t5/Canvas-Basics-Guide/What-is-Canvas/ta-p/45> [in English].
8. Node.js. URL: <https://nodejs.org/en> (date of request: 15.04.2023) [in English].
9. Taha, Sufiyan. What is Node.js: A Comprehensive Guide. URL: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs> (date of request: 15.04.2023) [in English].
10. Bonjar Basumatary, Nishant Agnihotri. Benefits and Challenges of Using NodeJS. *International Journal of Innovative Research in Computer Science & Technology*. URL: <https://www.ijircst.org/DOC/13-benefits-and-challenges-of-using-nodejs.pdf> (date of request: 16.04.2023) [in English].
11. Nadiia Shevchuk. Is Node.js Still Relevant for Startup Development in 2023? URL: <https://www.altamira.ai/blog/node-js-for-startup-development-in-2023> (date of request: 16.04.2023) [in English].
12. Oleksandr Vetoshkin. Yak pratsiuie Event Loop v NodeJS: vnutrishnia budova, fazy ta pryklady [How Event Loop works in NodeJS: internal structure, phases and examples]. URL: <https://dou.ua/forums/topic/31698> (date of request: 16.04.2023) [in Ukrainian].
13. Korotun, O., Vakaliuk, T., Zubrytskyi, V., Hordiienko, I. (2022) Teoretychni aspekty rozrobky systemy upravlinnia navchanniam [Theoretical aspects of learning management system development]. *Tavriiskyi naukovyi visnyk. Seriia: Tekhnichni nauky [Taurian Scientific Bulletin. Series: Technical sciences]*, (1), 36–46. <https://doi.org/10.32851/tnv-tech.2022.1.4> [in Ukrainian].