МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД

«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ

Реєстраційний №\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Поп Ільян Юрійович

студент 3-го курсу

денної форми навчання

групи ІПЗ

**Звіт**

ПРО ПРОХОДЖЕННЯ ТЕХНОЛОГІЧНОЇ (ВИРОБНИЧОЇ) ПРАКТИКИ

Тема: «Розробити бот для автоматизації рутинних завдань у системах

управління проєктами (Jira, Trello).»

**Спеціальність 121 Інженерія програмного забезпечення**

Рекомендована до захисту

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2025р.

Робота захищена

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2025р.

з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Підписи членів комісії:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівники практики від ВНЗ :

(керівники вказані в листі на пошті)

Ужгород – 2025

[**ВступРозділ 1 (опис бази практики)** 3](#_Toc202051736)

[**Розділ 2 (розкрити індивідуальне завдання)** 5](#_Toc202051737)

[**2.1. Основна ідея** 5](#_Toc202051738)

[**2.2. Що таке проект у розумінні програмування** 5](#_Toc202051739)

[**2.3. Що таке система керування проектом** 6](#_Toc202051740)

[**2.4. Проджект менеджер** 6](#_Toc202051741)

[**2.5. Що таке трелло** 6](#_Toc202051742)

[**2.6. Що таке джира** 7](#_Toc202051743)

[**2.7. React** 7](#_Toc202051744)

[**2.8. Trello api** 7](#_Toc202051745)

[**2.9. Wit api** 8](#_Toc202051746)

[**2.10. Axios** 9](#_Toc202051747)

[**2.11. Адаптивний інтерфейс** 9](#_Toc202051748)

[**2.12. Хостинг** 10](#_Toc202051749)

[**2.13. Гіт хаб** 11](#_Toc202051750)

[**2.14. Авторизація** 11](#_Toc202051751)

[**2.15. Підказки** 12](#_Toc202051752)

[**2.16. Вибір платформи trello** 13](#_Toc202051753)

[**2.17. Вибір платформи jira** 14](#_Toc202051754)

[**2.18. Перегляд дощок трелло** 14](#_Toc202051755)

[**2.19. Перегляд вмісту дошки трелло** 15](#_Toc202051756)

[**2.20. Створення списку трелло** 16](#_Toc202051757)

[**2.21. Створення картки трелло** 17](#_Toc202051758)

[**2.22. Видалення списку трелло** 17](#_Toc202051759)

[**2.23. Видалення картки трелло** 18](#_Toc202051760)

[**2.24. Отримання поточного розташування.** 19](#_Toc202051761)

[**Висновок** 20](#_Toc202051762)

**Вступ**

У сучасному світі діджиталізація відіграє важливу роль, проникаючи в усі сфери життя — від побуту до складних виробничих процесів. Це створює високий попит на програмне забезпечення, яке має бути не лише функціональним, а й зручним, адаптивним і ефективним. Ключовим етапом у створенні будь-якого програмного продукту є проектування — саме на цьому етапі визначаються структура майбутньої системи, її функціональні блоки та взаємодія між ними. У процесі проходження технологічної практики мною було розроблено програмний додаток, який дозволяє автоматизувати типові процеси в системі управління проектами Trello. Замість щоденного ручного входу в обліковий запис і повторного виконання однотипних дій, користувач може один раз авторизуватися, після чого взаємодія з платформою здійснюється за допомогою коротких текстових запитів. Такий підхід значно пришвидшує роботу, підвищує ефективність управління завданнями та мінімізує рутину для користувача.

**Розділ 1 (опис бази практики)**

**Розділ 2 (розкрити індивідуальне завдання)**

**2.1. Основна ідея**

Основною ідеєю мого проекту є автоматизація рутинних завдань проект-менеджера в таких платформах, як Trello. Метою є надання можливості швидко та зручно створювати, переглядати та видаляти списки й картки на дошці. Це дозволяє зменшити обсяг ручної роботи, пришвидшити виконання базових дій і зробити процес керування проектами ефективнішим і зручнішим для користувача.

**2.2. Що таке проект у розумінні програмування**

У сучасному світі діджиталізація відіграє важливу роль, проникаючи в усі сфери життя — від побуту до складних виробничих процесів. Це створює високий попит на програмне забезпечення, яке має бути не лише функціональним, а й зручним, адаптивним і ефективним. Ключовим етапом у створенні будь-якого програмного продукту є проектування — саме тут визначається структура майбутньої системи, її функціональні блоки та взаємодія між ними. Без якісного проектування неможливо забезпечити стабільну та масштабовану розробку, яка відповідатиме потребам користувачів. Як у будівництві необхідні план і контроль, так і в ІТ-проектах важливими є планування та управління. Вони допомагають координувати команду, розподіляти ресурси, дотримуватися строків і досягати цілей. Саме завдяки цим процесам розробка стає структурованою, передбачуваною й ефективною, що особливо важливо в умовах швидких змін у технологічному середовищі.

**2.3. Що таке система керування проектом**

Для спрощення процесу планування спеціалісти з програмного забезпечення дійшли до ідеї створити програмні рішення, які допомагають ефективно керувати розробкою. У процесі створення будь-якого програмного продукту виникає потреба не лише в написанні коду, а й у чіткій організації командної роботи, розподілі завдань, контролі термінів та моніторингу прогресу. Саме з цією метою було розроблено системи керування проєктами. Вони дозволяють створювати детальний план проєкту, розбивати його на окремі завдання або етапи, призначати відповідальних осіб, відстежувати статус кожного елемента роботи, а також виявляти затримки чи проблеми на ранніх етапах. Крім того, такі системи забезпечують зручну візуалізацію процесу, що полегшує розуміння загальної картини всіма учасниками команди. Завдяки цьому підвищується прозорість, узгодженість дій, гнучкість ухвалення рішень та якість кінцевого продукту. Поява таких інструментів, як Trello, Jira, Asana та інших, дала потужний поштовх діджиталізації процесів управління, зробивши їх ефективнішими та масштабованішими.

**2.4. Проджект менеджер**

Проджект-менеджер — це людина, яка відповідає за планування та контроль проєкту. До його обов’язків входить розподіл роботи між членами команди, контроль за виконанням завдань і якістю їх виконання, а також призначення відповідних завдань відповідним працівникам. Проджект-менеджер слідкує за тим, щоб усі етапи проєкту виконувалися вчасно і відповідали встановленим стандартам, забезпечуючи ефективну координацію між учасниками процесу та досягнення поставлених цілей.

**2.5. Що таке трелло**

Trello — це платформа, створена спеціально з метою управління проєктами. У ній користувач має можливість створювати дошки, на яких розміщуються списки, а вже на списках — завдання у вигляді карток. Проджект-менеджер може створювати як списки, так і картки, призначати виконавців для окремих завдань, переміщувати картки між списками відповідно до стадії виконання, а також видаляти непотрібні елементи у разі потреби. Завдяки своїй простоті, візуальності та гнучкості Trello є зручним інструментом для організації роботи як невеликих команд, так і великих проєктів.

**2.6. Що таке джира**

**Jira** — це платформа для керування проєктами, яка широко використовується в командах розробників програмного забезпечення. Вона дозволяє створювати проєкти та керувати різними типами завдань, такими як задачі, баги чи історії користувача. Проджект-менеджер може призначати завдання виконавцям, відстежувати їхній статус, змінювати етапи виконання або за потреби видаляти. Завдяки підтримці методологій Scrum і Kanban, розвиненим аналітичним інструментам і гнучким налаштуванням, Jira підходить як для невеликих команд, так і для складних багаторівневих проєктів.

**2.7. React**

**React** — це JavaScript-бібліотека, яка була використана для розробки цього проєкту. Завдяки її функціональним можливостям стало можливим швидко та зручно створити односторінковий фронт-енд додаток. Основу проєкту було згенеровано за допомогою інструмента **Create React App**, що автоматично налаштував необхідне середовище. Розробка здійснювалася в режимі розробника через команду start, яка дозволяла оперативно вносити зміни та бачити результат. Після завершення роботи застосунок було зібрано командою build, у результаті чого проєкт став готовим до розгортання.

**2.8. Trello api**

Для реалізації автоматизації в платформі Trello у проєкті було використано **Trello API**. Це програмний інтерфейс, який надає можливість інтегрувати користувацькі застосунки з серверами Trello та виконувати різні дії — створення, перегляд, редагування та видалення дощок, списків і карток. Доступ до API здійснюється шляхом авторизації: розробник надає свій унікальний ключ, а користувач — персональний токен. Такий підхід забезпечує надійний рівень безпеки: сторонні особи не можуть отримати доступ до чужих даних без відповідного токена. Крім того, ключ розробника дозволяє ідентифікувати джерело дій у разі зловживань, що унеможливлює використання додатку з шкідливою метою без відповідальності.

**2.9. Wit api**

Wit.ai було використано для розпізнавання наміру користувача у його запиті. Це дозволяє зробити взаємодію з додатком більш природною: користувачу не потрібно вводити статичні фрази типу "мої дошки" — він може писати по-різному, наприклад: "дай мені дошки", "покажи мої дошки" або навіть просто "дошки". Wit.ai аналізує текст, визначає, що саме має на увазі користувач, і повертає відповідну інтенцію програмі, яка вже виконує необхідну дію. Окрім визначення мети запиту, Wit.ai також здатний витягувати сутності з тексту. Наприклад, якщо користувач хоче переглянути всі списки певної дошки, він може написати "списки дошки Приклад", "списки Приклад" тощо — штучний інтелект розпізнає назву дошки як сутність і передасть її додатку для подальшої обробки. Для зв’язку програми з Wit.ai використовується відповідне API: текстовий запит надсилається на сервер Wit.ai, де обробляється, після чого у відповідь надходять структуровані дані — інтенція запиту та виявлені сутності.

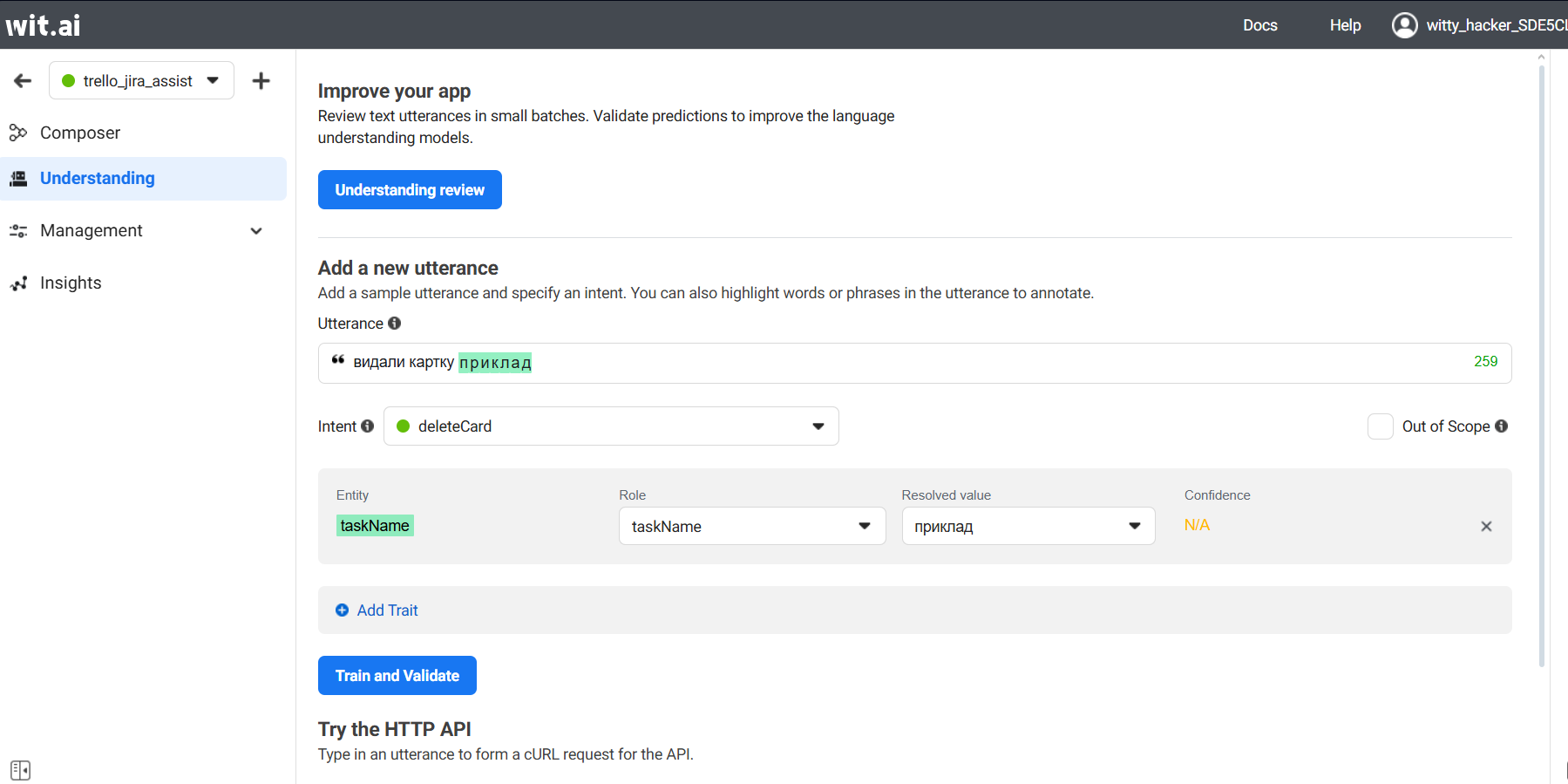


Рисунок 2.1. Приклад навчання штучного інтелекту

**2.10. Адаптивний інтерфейс**

Було створено адаптивний інтерфейс для забезпечення доступності додатку з будь-якого пристрою. Користувач має змогу комфортно взаємодіяти з додатком як із комп’ютера, так і з планшета або телефону. Для реалізації адаптивності були використані відносні одиниці виміру, зокрема vh висота вікна перегляду та vw ширина вікна перегляду, що дозволяє елементам динамічно підлаштовуватися під розміри екрану. Крім того, застосовано медіа-запити для визначення фактичної ширини та висоти пристрою користувача. Це дає змогу виявити, чи використовується вертикальна чи горизонтальна орієнтація екрана, і відповідно адаптувати розмітку. Завдяки цьому інтерфейс автоматично підлаштовується під розміри й тип пристрою, забезпечуючи зручність користування в будь-якому форматі.

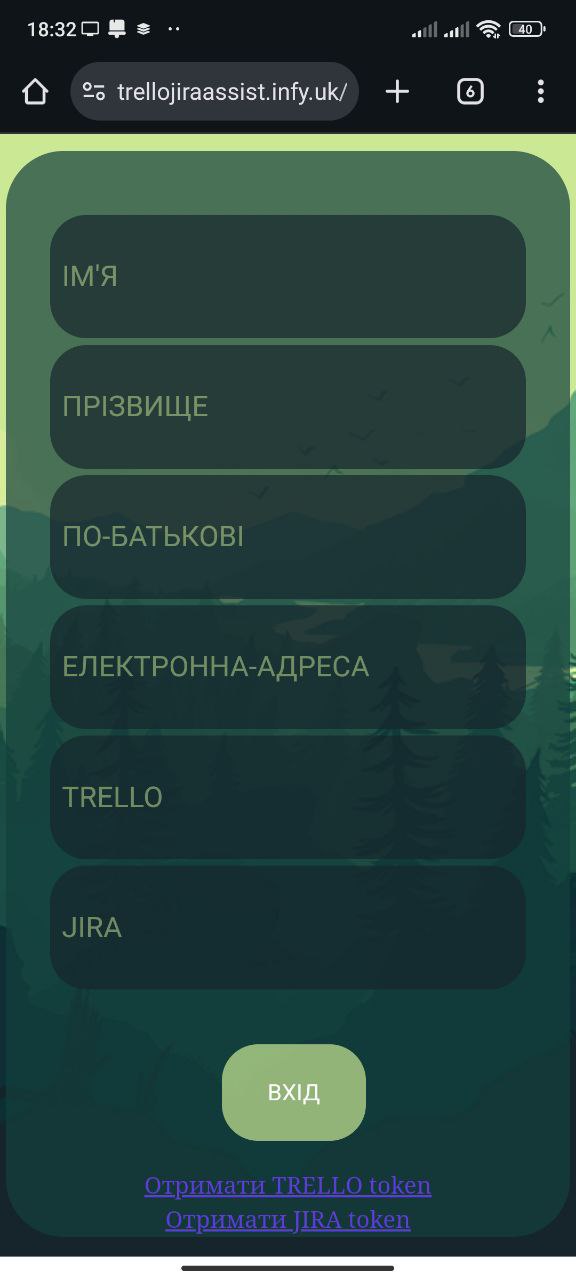


Рисунок 2.2. Вигляд додатку з екрану смартфону

**2.11. Хостинг**

Для хостингу було обрано сервіс InfinityFree, на якому був задеплоєний уже зібраний проєкт. Цей сервіс надає можливість безкоштовно розміщувати вебдодатки, а також створити доменне ім’я з використанням одного з доступних субдоменів. У моєму випадку було обрано субдомен infy.uk. Після завершення розгортання додаток став доступним за адресою: <https://trellojiraassist.infy.uk>. Деплой здійснювався за допомогою FTP-з’єднання. У якості програмного забезпечення для цього було використано FileZilla. На початку процесу InfinityFree надав облікові дані, необхідні для підключення до сервера по FTP. Після введення цих даних з’явилась можливість підключитися до віддаленої директорії, куди було завантажено вміст папки build — тобто фінальну версію зібраного проєкту, готову до публічного доступу.

**2.12. Гіт хаб**

Як систему контролю версій у проєкті було використано **Git**. Усі версії та зміни коду зберігалися в репозиторії, розміщеному на моєму персональному GitHub-акаунті. Це дозволило ефективно відстежувати хід розробки, повертатися до попередніх станів коду та забезпечити зручну структуру проєкту. Репозиторій доступний за посиланням: <https://github.com/Ilian-Pop/trello-jira_assist>.

**2.13. Авторизація**

Для авторизації необхідно ввести такі дані: прізвище, ім’я та по батькові, електронну адресу, а також токени для Jira та Trello, які можна отримати за посиланням, розміщеним унизу панелі авторизації. Усі введені дані проходять валідацію. Прізвище, ім’я та по батькові мають починатися з великої літери, можуть містити всі літери українського алфавіту та апостроф, але не можуть закінчуватися на апостроф. Електронна адреса обов’язково повинна завершуватися на «@gmail.com». При першому введенні поля виглядають звичайно, без підсвітки, однак у разі спроби відправити невалідні дані відповідні поля автоматично підкреслюються. Підкреслення зберігається до моменту, поки користувач не введе коректні дані — після цього поле повертається до нормального вигляду.

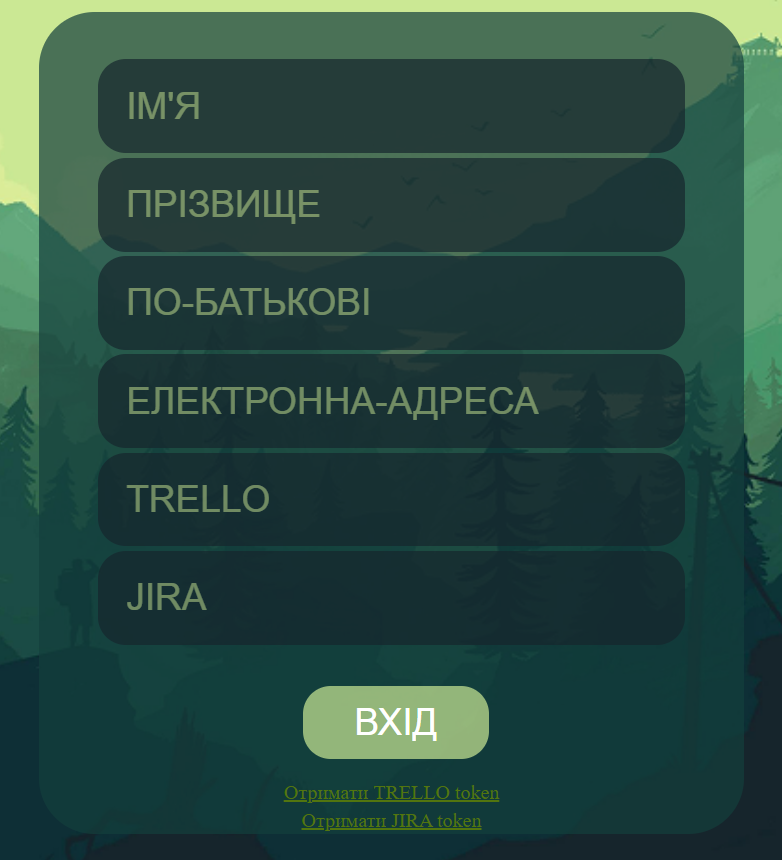


Рисунок 2.3. Сторінка авторизації

**2.14. Підказки**

Дане програмне забезпечення має розширений функціонал, тому для нових користувачів, які лише починають ознайомлення з додатком, була реалізована команда help, яку можна викликати, звернувшись із проханням про допомогу — наприклад, написавши "допоможи", "мені потрібна допомога" тощо. У відповідь на запит система виводить підказки щодо використання та приклади запитів, що допомагає швидше освоїти функціонал і полегшує перші кроки у взаємодії з інтерфейсом.

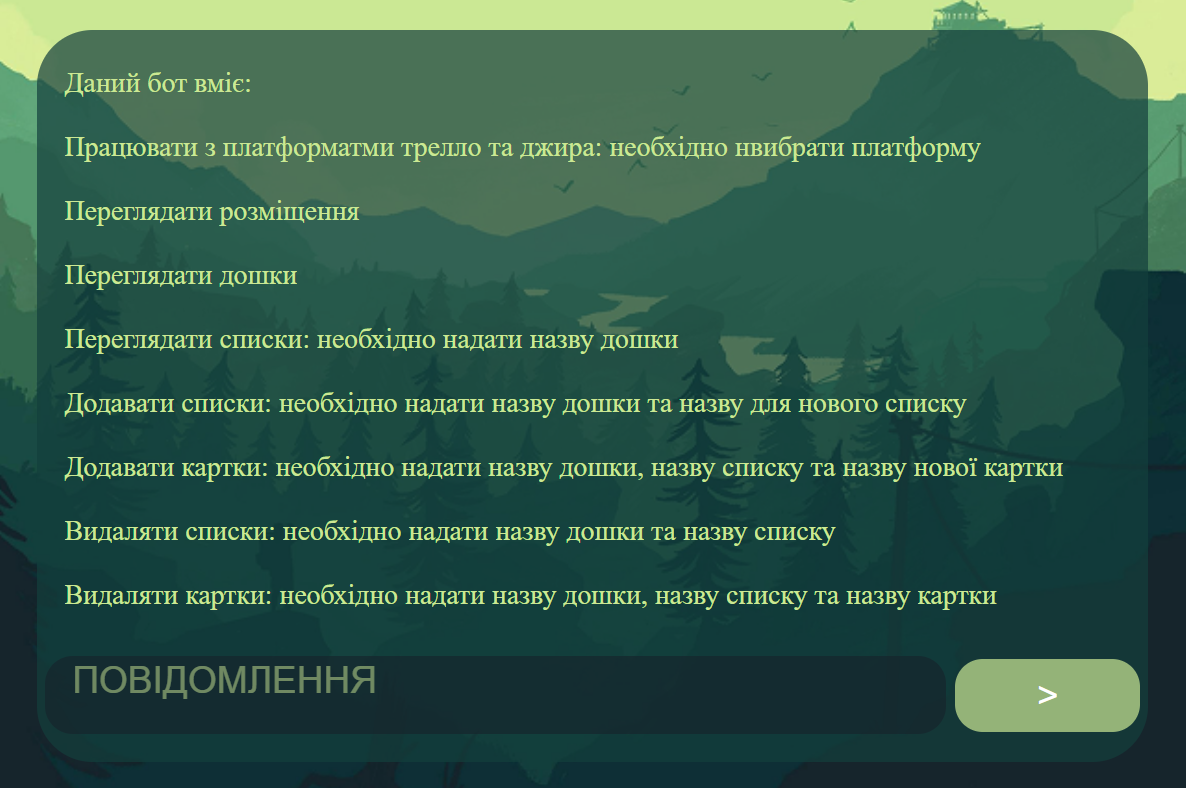


Рисунок 2.4. Допомога новим користувачам

**2.15. Вибір платформи trello**

Для вибору платформи, тобто трелло або джира, використовується штучний інтелект Wit.ai. Це означає, що користувачу не обов’язково вводити статичну фразу на кшталт "Trello", щоб обрати платформу для взаємодії. Він може сформулювати запит у довільній формі — наприклад: "використовувати платформу Trello", "використовувати Trello", або навіть просто написати "Trello". У всіх цих випадках програма завдяки Wit.ai розпізнає намір користувача та правильно обере відповідну платформу для подальшої роботи.

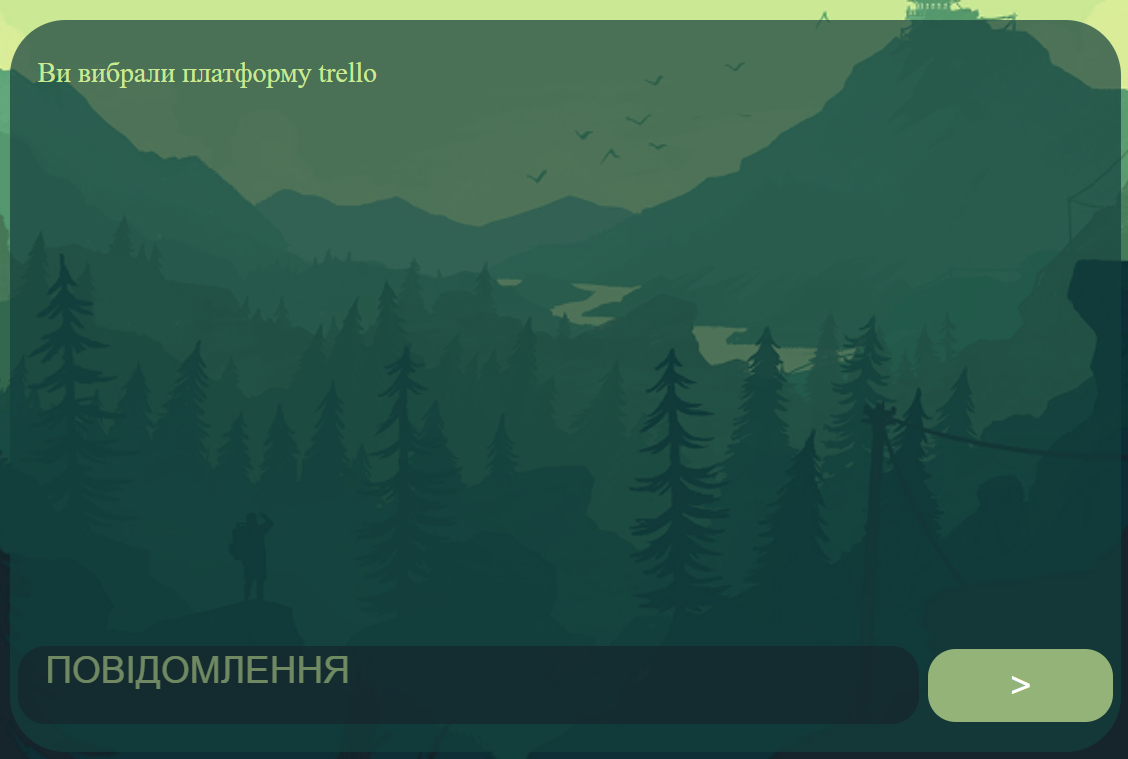


Рисунок 2.5. Вибір платформи трелло

**2.16. Перегляд дощок трелло**

При введенні тексту, пов’язаного з отриманням дощок, штучний інтелект Wit.ai аналізує запит користувача та повертає завдання типу getBoard. Мій додаток розпізнає це завдання, після чого автоматично виконує запит до API Trello для отримання списку всіх дощок, пов’язаних з токеном, який був збережений під час авторизації. Отримані дані використовуються для подальшої взаємодії, зокрема — для виведення назв дощок, вибору потрібної, а також для виконання наступних дій, таких як перегляд вмісту або створення списків та карток. Такий підхід дозволяє зробити роботу з ботом інтуїтивно зрозумілою та максимально автоматизованою.

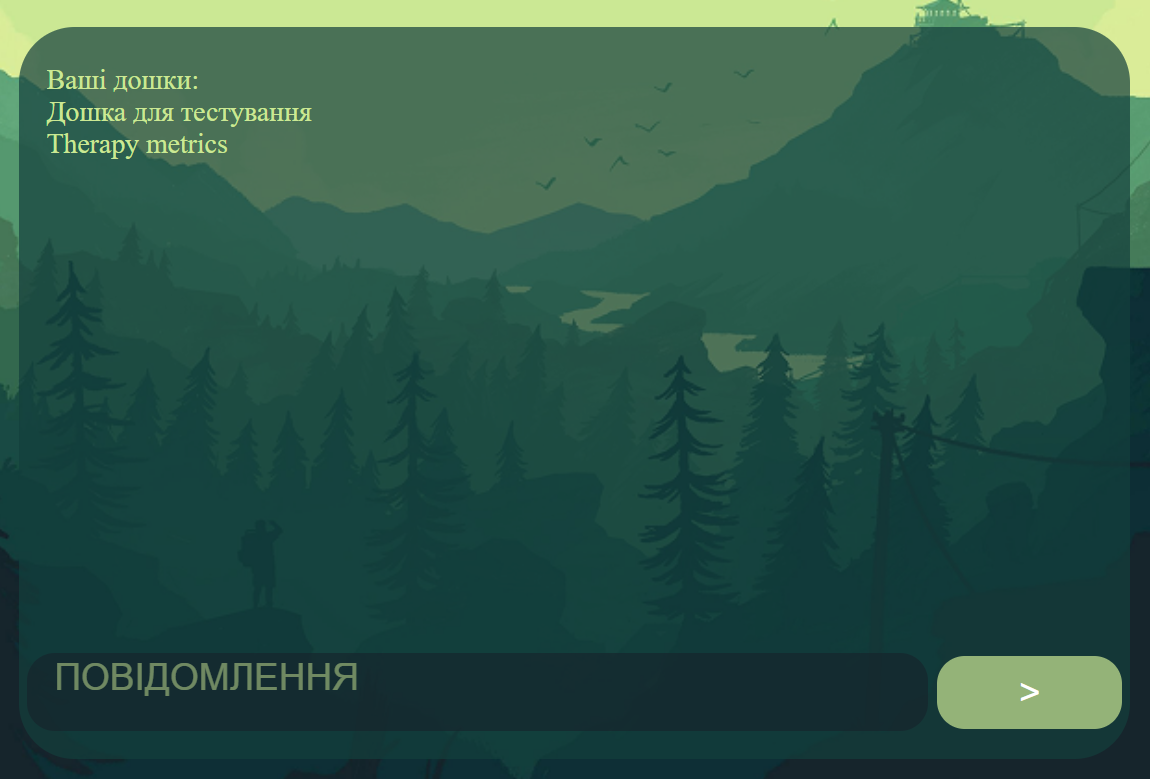
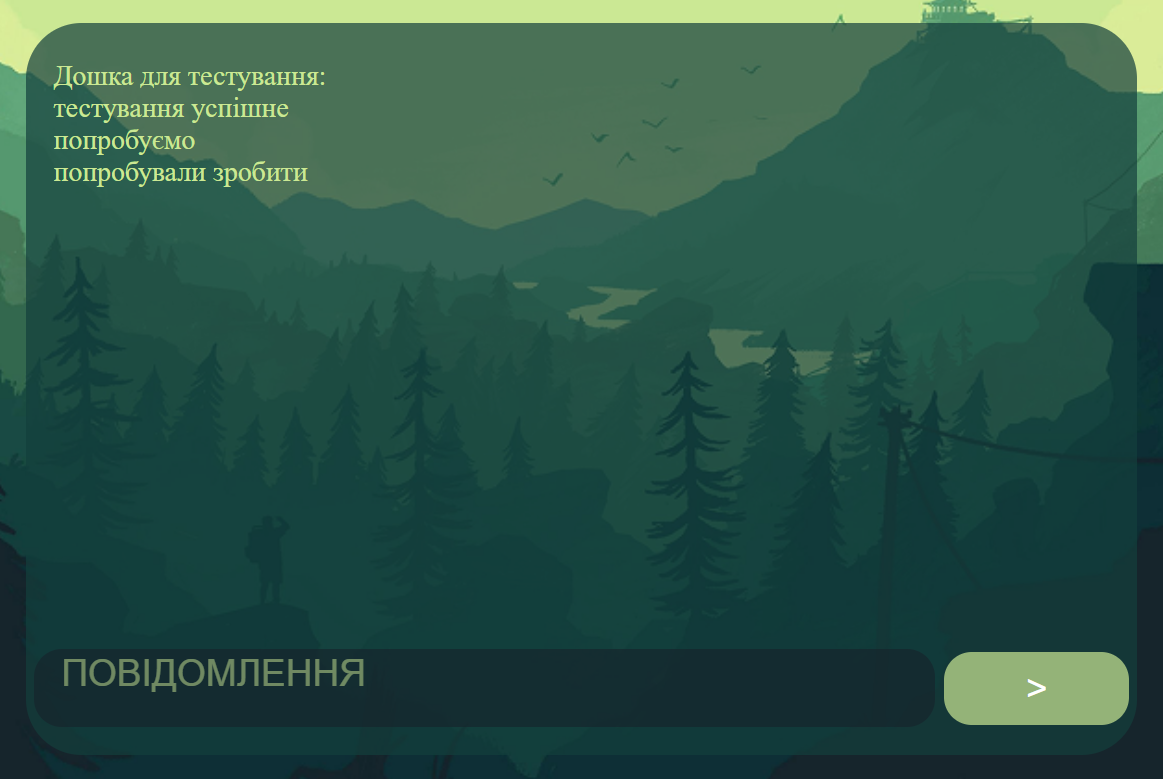


Рисунок 2.6. Вивід дощок користувача

**2.17. Перегляд вмісту дошки трелло**

При введенні тексту, пов’язаного з отриманням списків, система спочатку перевіряє, чи є збережені дані про будь-яку дошку. Якщо така інформація вже введена, здійснюється запит до API Trello для отримання ідентифікатора дошки та списків, що їй належать. У разі відсутності таких даних повідомлення користувача додатково аналізується за допомогою штучного інтелекту на наявність сутності з назвою дошки. Якщо її вдається визначити, вона використовується для запиту та зберігається для подальшої роботи. Це забезпечує зручну та безперервну взаємодію з додатком.

  
Рисунок 2.7. Отримання вмісту дошки

**2.18. Створення списку трелло**

Для створення списку необхідно або мати вже введені дані про дошку, яка буде використана для цієї операції, або ввести нові. Крім того, обов’язковою умовою є введення назви для нового списку. Процес створення починається з отримання ідентифікатора вибраної дошки. Після цього за допомогою запиту до серверів Trello у вказаній дошці створюється новий список з попередньо введеною назвою. У разі успішного виконання операції користувачу виводиться повідомлення про успішне створення списку. Якщо ж створення не вдалося, система повідомляє про невиконання завдання.

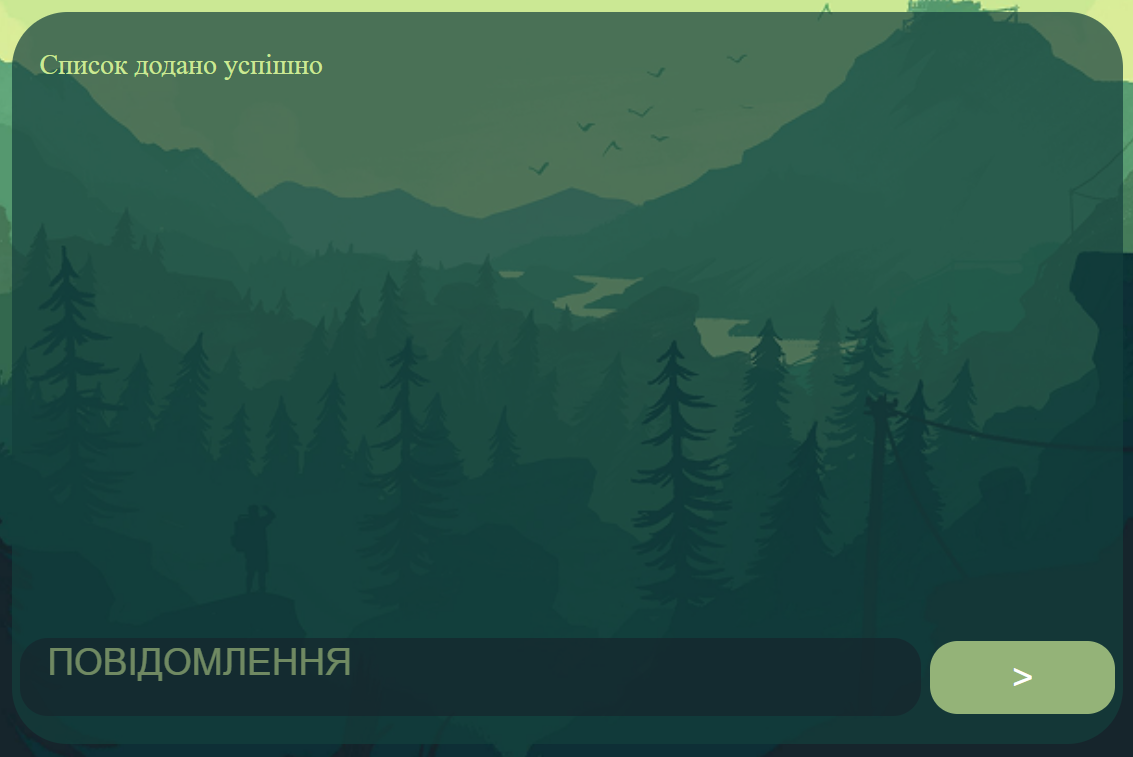


Рисунок 2.8. Успішне створення списку

**2.19. Створення картки трелло**

Для створення картки необхідно або мати вже введені дані про дошку та список, які будуть використані для цієї операції, або ввести нові. Крім того, обов’язковою умовою є введення назви для нової картки. Процес починається з отримання ідентифікаційного ключа дошки. Далі відбувається перевірка наявності вказаного списку в межах цієї дошки та отримання ідентифікатора самого списку. Лише після того, як усі дані успішно перевірено та підтверджено, здійснюється спроба створення картки. У разі успішного виконання операції користувач отримує повідомлення про успішне створення картки. Якщо ж виникає помилка — як і у випадку зі створенням списку — користувачу виводиться повідомлення про неуспішність операції, з можливістю повторного введення або уточнення даних.

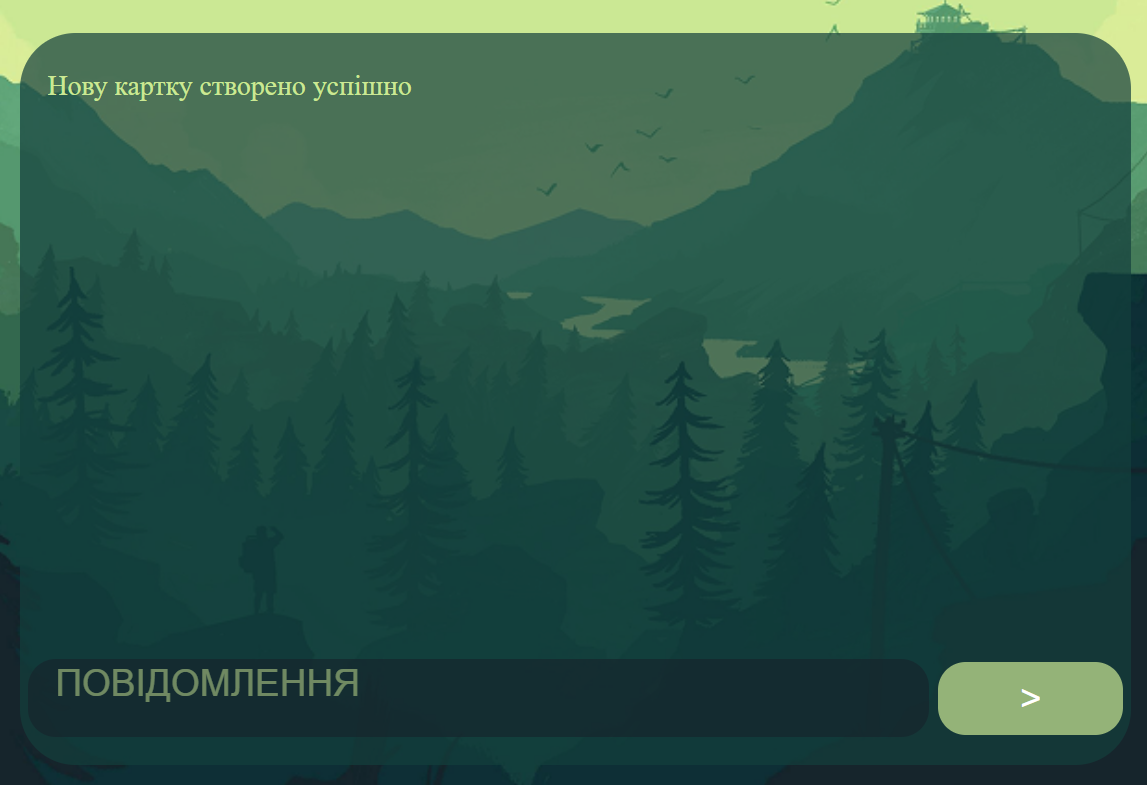


Рисунок 2.9. Успішне створення картки

**2.20. Видалення списку трелло**

Для видалення списку необхідно або мати вже введені дані про дошку та список, який потрібно видалити, або ввести нові. Процес видалення починається з того, що програма отримує ідентифікаційний код вибраної дошки. Після цього, на основі цього коду, здійснюється пошук відповідного списку та його ідентифікатора. Далі, за допомогою цього ідентифікатора, надсилається запит на сервер для видалення списку. У разі успішного виконання операції користувачу виводиться повідомлення про успішне видалення. Якщо ж операція не виконується з якихось причин, система повідомляє про неуспішність дії, що дозволяє користувачу вжити необхідних заходів для виправлення ситуації.

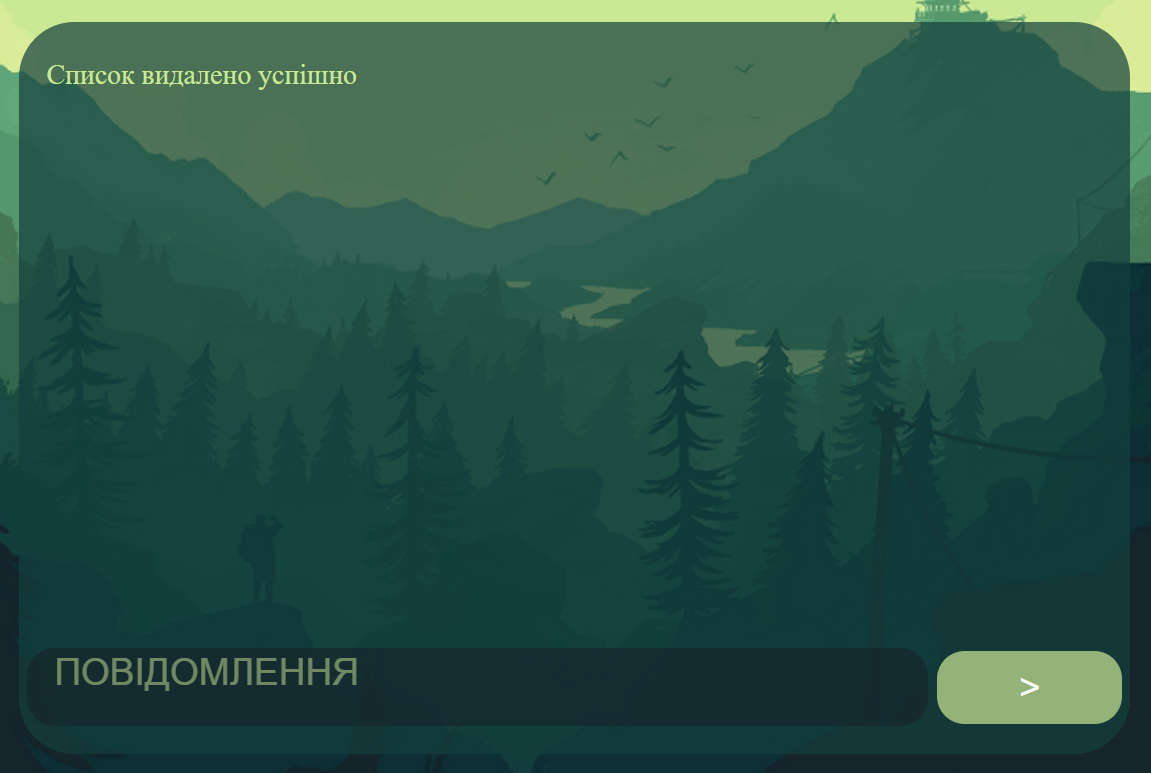
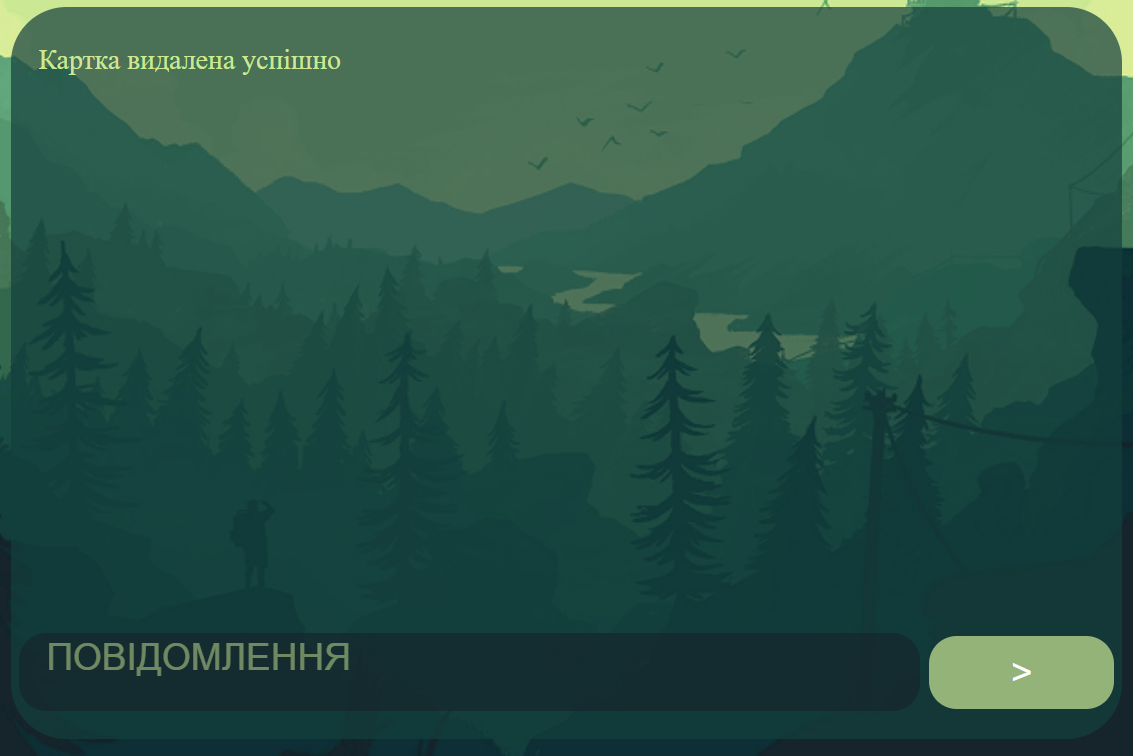


Рисунок 2.10. Успішне видалення списку

**2.21. Видалення картки трелло**

Для видалення картки необхідно або вже мати введені дані про дошку, список та саму картку, яку потрібно видалити, або ввести нові. Процес видалення починається з отримання ідентифікатора вибраної дошки. Після цього, на основі цього ідентифікатора, виконується пошук відповідного списку, що є її нащадком. Далі починається пошук усіх карток, які належать цьому списку. Серед них за назвою визначається потрібна картка, після чого відбувається пошук її ідентифікатора. Коли ідентифікатор картки отримано, здійснюється запит до сервера на її видалення. У разі успішного виконання операції користувач отримує повідомлення про успішне видалення. Якщо ж видалення не вдалося, користувач отримує повідомлення про невиконання операції.

  
Рисунок 2.11. Успішне видалення картки

**2.22. Отримання поточного розташування**

Для того щоб користувачу було зручніше орієнтуватися в додатку та розуміти поточне розташування в структурі, було реалізовано спеціальну команду position. Її можна викликати, сформулювавши запит у довільній формі, наприклад: "дай мені моє розташування", "дай розташування", або просто "розташування". Після виконання цієї команди користувач отримує відповідь, яка містить інформацію про вибрану платформу, дошку, список і картку. Якщо якийсь із цих елементів не був раніше обраний або заданий, він просто не буде відображатися у відповіді. Таким чином, користувач завжди може дізнатися, на якому етапі роботи з додатком він перебуває, навіть у випадку неповного контексту.

**2.23. Вихід**

У випадку, коли користувач хоче вийти зі свого облікового запису, йому достатньо або висловити бажання вийти, або просто попрощатися з ботом.

**Висновок**

У ході проходження практики мною було реалізовано вебдодаток, який автоматизує рутинні завдання в системі Trello. Користувач може переглядати, створювати та видаляти дошки, списки та картки, використовуючи прості текстові запити. У деяких випадках для виконання дії достатньо одного слова, що значно спрощує роботу з платформою. Було створено адаптивний інтерфейс, який забезпечує зручне користування як з комп’ютера, так і з мобільних пристроїв. Для розпізнавання команд у вільній формі використано штучний інтелект Wit.ai, що дозволяє уникнути статичних фраз і зробити взаємодію більш природною. Основою проєкту є бібліотека React, яка забезпечила ефективну розробку односторінкового додатку. Реалізовано валідацію даних при авторизації: ім’я, прізвище, по батькові та електронна пошта перевіряються на коректність, а токени — на наявність. У разі введення некоректного токена додаток виводить відповідне повідомлення та не виконує запит. Результатом стала зручна система, яка поєднує автоматизацію, інтелектуальне розпізнавання запитів і доступність з будь-якого пристрою.

**Список джерел та використаної літератури**

1. «Trello api doc» – [Електронний ресурс] – [https://developer.atlassian.com/cloud/trello/rest/api-group-actions/#api-group-actions](https://developer.atlassian.com/cloud/trello/rest/api-group-actions/%23api-group-actions%20)
2. «Wit ai doc» – [Електронний ресурс] – <https://wit.ai/docs>
3. «React doc» – [Електронний ресурс] – <https://react.dev>