

КУРСОВА РОБОТА

з дисципліни “Мультимедійні інтерфейси та
3D-візуалізація” на тему:

ПЕРСОНАЛЬНИЙ АСИСТЕНТ ДЛЯ ВЕДЕННЯ НОТАТОК ТА ПЛАНУВАННЯ ПОДІЙ. РОЗРОБЛЕННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ

Виконала студентка групи КП-41мп

Потапчук Анна Андріївна

Керівник роботи: д.т.н. Сулема Є.С.

До захисту допущено

(дата, підпис)

Захищено з оцінкою

(дата, підпис)

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

“ЗАТВЕРДЖЕНО”

Керівник роботи

_____ Є.С. Сулема

“ ____ ” _____ 2024 р.

ПЕРСОНАЛЬНИЙ АСИСТЕНТ ДЛЯ ВЕДЕННЯ НОТАТОК ТА
ПЛАНУВАННЯ ПОДІЙ

Технічне завдання

ПЗКС.045440-02-91

Виконавці:

Беліцький О. С.

Пецеля А. В.

Потапчук А. А.

Слободзян М. В.

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ	3
2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ	3
3. ПРИЗНАЧЕННЯ РОЗРОБКИ	3
4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ	4
5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ	7
6. ЕТАПИ ПРОЄКТУВАННЯ	8

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Персональний асистент для ведення нотаток та планування подій.

Галузь застосування: мультимедійні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на курсовий проєкт з дисципліни “Мультимедійні інтерфейси та 3D-візуалізація”.

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Персональний асистент для ведення нотаток та планування подій призначений для автоматизації процесів управління часом та інформацією користувача. Він забезпечує ефективну організацію щоденних завдань, подій та нотаток, інтегруючись з сучасними інструментами календаря та пропонуючи інтуїтивний інтерфейс з підтримкою голосових команд. Система спрямована на полегшення управління щоденними задачами для індивідуальних користувачів, які потребують швидкого доступу до планування подій, нагадувань, а також керування своїми нотатками. Голосове керування дозволяє користувачу взаємодіяти з календарем і нотатками без використання клавіатури чи миші, що робить застосунок особливо корисним для людей з обмеженими можливостями, а також для користувачів, які шукають більш зручний спосіб управління інформацією. Наявність мультимедійного помічника допоможе спростити та зробити процес взаємодії з системою більш інтуїтивним та доступним. Інтерактивний 3D-аватар забезпечить користувачу візуальні підказки під час голосових команд, надаючи відчуття реального спілкування з асистентом.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

4.1. Функціональні вимоги до програмного забезпечення

Інтерактивна система для ведення нотаток та планування подій повинна містити такі основні функції:

- 1) забезпечувати реєстрацію обов'язкового персонального облікового запису користувача для доступу до функціональності системи;
- 2) інформувати користувача про наявні команди при вході в систему;
- 3) забезпечувати можливість виконання наступних команд для управління нотатками:

а) Створення нотатки:

- команда: "Make a note";
- питання для уточнення: "How note should be named?", "Please dictate the note";
- результат: створена нотатка з вказаним ім'ям та вмістом.

б) Перегляд нотаток:

- команда: "What notes do I have?";
- питання для уточнення: "Do you want to hear more?"
- результат: анімований персонаж озвучує п'ять останніх назв нотаток та питає у користувача чи потрібно озвучити більше. У випадку ствердної відповіді анімований персонаж озвучує наступні п'ять імен нотаток.

с) Відтворення нотаток:

- команда: "Read the note";
- питання для уточнення: "What's the name of the note you'd like to hear";
- результат: анімований персонаж озвучує нотатку за вказаною назвою.

- 4) взаємодіяти з електронним календарем користувача для виконання наступних команд для управління подіями:

a) Створення події:

- команда: “Create an event”;
- питання для уточнення: “What is the name of the event?”, “When does it start?”, “When does it end?”;
- результат: створена подія з вказаною назвою, часом початку та кінця.

b) Розклад:

- команди: “What is planned for today?”, “What is planned for this week?”, “What is planned for the next week?”;
- результат: анімований персонаж озвучує користувачу список подій запланованих на сьогодні, цей тиждень або наступний тиждень.

c) Видалення події:

- команда: “Remove the event”;
- питання для уточнення: “What is the event name to remove?”;
- результат: вказана подія видалена або користувачу повідомляється про те, що такої події не існує.

- 5) надавати можливість вводу команд за допомогою голосу або вибору команди у відповідному меню;
- 6) надавати відповідь користувачу за допомогою відтворення згенерованого аудіо та дублювати відповідь у вигляді субтитрів;
- 7) відображати тривимірну анімовану модель під час використання застосунку;
- 8) надавати можливість перегляду записів про нотатки та події користувача із застосуванням інструментів пагінації, сортування та пошуку;
- 9) реагувати на неіснуючі голосові команди та незрозуміле мовлення користувача.

4.2. Нефункціональні вимоги до програмного забезпечення

Інтерактивна система для ведення нотаток та планування подій повинна забезпечувати такі нефункціональні можливості:

- 1) бути кросплатформною:
 - a) вебзастосунок повинен працювати на більшості сучасних браузерів (Chrome, Mozilla, Edge);
 - b) сервер повинен запускатися на різних ОС (Linux, Windows);
- 2) витримувати навантаження в 100 одночасних користувачів та 50 тисяч активних користувачів в місяць;
- 3) дані для автентифікації мають бути захищені, паролі не повинні зберігатися у відкритому вигляді;
- 4) мова інтерфейсу користувача – англійська.

4.3. Функціональні вимоги до моделі та анімації персонажу асистента

Функціональні вимоги моделі та анімації персонажа:

- 1) модель повинна бути багатополігональною, із накладеними матеріалами;
- 2) модель повинна бути виконана у вигляді футуристичного робота, наприклад, EVE із мультфільму “WALL-E”;
- 3) створений скелет моделі;
- 4) модель повинна виражати різні емоції та стан за допомогою анімації й зміни обличчя;
- 5) для кожної визначеної дії персонажа повинні бути реалізовані відповідні анімації:
 - "Привітання" - анімація, де модель махає рукою;
 - "Слухає" - модель робить невеликі рухи, на обличчі з'являється анімація прослуховування у вигляді голосової доріжки.
 - "Думає" - невеликі рухи вгору-вниз або з боку в бік, як ніби робот розмірковує, та вираз обличчя, що зображає завантаження.

- "Відповідає" - більш активні рухи тіла, а також щасливий вираз обличчя.
 - "Не зрозумів команду" - стримана анімація з жестом збентеження та виразом обличчя "помилка".
 - "Прощання" - анімація з прощальним жестом руки або нахилом корпусу.
- 6) персонаж не повинен бути повністю статичним, навіть коли не взаємодіє з користувачем. Необхідно створити просту анімацію, наприклад, невеликі випадкові погойдування або мимовільні рухи антен або інших дрібних деталей;
 - 7) вирази обличчя повинні точно відповідати стану помічника (радість, сум, збентеження);
 - 8) персонаж із анімаціями повинні бути експортовані у форматі glTF.

4.4. Нефункціональні вимоги до моделі та анімації персонажу асистента

Нефункціональні вимоги до моделі та анімації персонажа:

- 1) емоції мають бути чіткими та виразними, щоб користувачі могли легко їх розпізнати.
- 2) модель і анімації повинні бути плавними, без затримок. Потрібно уникати різких, незграбних рухів.
- 3) якщо помічник розмовляє або дає відповідь, анімації повинні бути синхронізовані з голосом.

5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проєкту повинна бути розроблена наступна документація:

- 1) 4 пояснювальні записки;
- 2) керівництво користувача.

6. ЕТАПИ ПРОЄКТУВАННЯ

Аналіз вимог до програмної системи	23.09.2024
Розроблення та узгодження технічного завдання	07.10.2024
Розроблення архітектури системи	21.10.2024
Розроблення основної логіки системи	01.11.2024
Розроблення компонент голосового інтерфейсу	12.11.2024
Розроблення моделей та анімацій	15.11.2024
Інтеграція компонент програмного продукту	23.11.2024
Тестування системи	03.12.2024
Оформлення технічної документації проєкту	09.12.2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет прикладної математики

“ЗАТВЕРДЖЕНО”

Керівник роботи

_____ Є.С.Сулема

“ ____ ” _____ 2024 р.

ПЕРСОНАЛЬНИЙ АСИСТЕНТ ДЛЯ ВЕДЕННЯ НОТАТОК ТА
ПЛАНУВАННЯ ПОДІЙ. РОЗРОБЛЕННЯ КЛІЄНТСЬКОЇ
ЧАСТИНИ

Пояснювальна записка

Виконавець:

_____ Потапчук А.А.

2024

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	3
ВСТУП.....	4
1. АНАЛІЗ ІНСТРУМЕНТІВ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ.....	5
1.1. TypeScript.....	5
1.2. Angular.....	5
1.3. Tailwind CSS.....	6
1.4. Three.js.....	6
2. РОЗРОБЛЕННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ.....	7
2.1. Аналіз вимог до розробленого програмного забезпечення.....	7
2.2. Опис вимог до клієнтської частини до персонального асистента для ведення нотаток та планування подій.....	7
2.3. Архітектура клієнтської частини.....	9
3. АНАЛІЗ РОЗРОБЛЕНИХ МОДУЛІВ.....	10
3.1. Особливості реалізації відображення необхідної анімації.....	10
3.2. Особливості реалізації голосового створення та пагінації.....	10
3.3. Особливості реалізації модуля запису та озвучування тексту.....	11
3.4. Тестування розроблених компонентів.....	11
3.5. Рекомендації щодо подальшого вдосконалення.....	12
ВИСНОВКИ.....	14
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	16

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

HTML — мова розмітки для створення структури веб-сторінок (заголовки, текст, зображення, посилання).

CSS — мова стилів для оформлення веб-сторінок (кольори, шрифти, розташування елементів).

3D — тривимірна графіка для відображення об'єктів із шириною, висотою і глибиною.

Рендерінг моделі — процес створення зображення 3D-моделі на основі її даних.

Фреймворк — набір готових інструментів і бібліотек для спрощення розробки програм або веб-додатків.

HTTP-запити — обмін даними між клієнтом і сервером у веб-протоколі (GET, POST тощо).

Роутизація — механізм маршрутизації запитів до відповідних сторінок чи компонентів у додатку.

WebGL — технологія для створення 2D і 3D графіки у веб-браузері без плагінів.

base64 — метод кодування даних у текстовий формат для передачі через мережу.

ВСТУП

В умовах стрімкого розвитку технологій та зростання попиту на мобільні рішення, важливим аспектом стає створення програмного забезпечення, яке відповідає вимогам зручності та інтуїтивності для кінцевого користувача. Одним із найбільш ефективних способів забезпечення високого рівня зручності є інтеграція голосових асистентів, які дозволяють швидко й ефективно виконувати різноманітні задачі за допомогою простих голосових команд. У сучасному світі, де швидкість і доступність інформації є ключовими факторами, голосові технології набирають популярності в різних сферах життя, від побутових завдань до професійної діяльності.

Важливим напрямком є розробка голосових асистентів для організації та планування діяльності, зокрема для створення нотаток та перегляду подій. Завдяки таким інтерфейсам, користувачі можуть за допомогою голосових команд легко створювати, змінювати та переглядати свої нотатки, планувати зустрічі, нагадування або переглядати майбутні події. Такий підхід дозволяє значно спростити процес управління часом і підвищити продуктивність.

Цей проєкт присвячений розробці голосового асистента для створення нотаток, планування та перегляду подій, який допоможе користувачам швидко і зручно організовувати свої справи за допомогою сучасних технологій розпізнавання мови та інтеграції з календарями та іншими сервісами. Основною метою є створення інтуїтивно зрозумілого інтерфейсу, що дозволить користувачам легко взаємодіяти з програмним забезпеченням та ефективно виконувати завдання, пов'язані з організацією часу та нотаток.

1. АНАЛІЗ ІНСТРУМЕНТІВ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ

1.1. TypeScript

TypeScript — це мова програмування, що є надмножиною JavaScript. Вона додає типізацію та інші потужні можливості, які дозволяють розробникам зменшити кількість помилок у коді, покращити читабельність і забезпечити більшу передбачуваність роботи програм. TypeScript компілюється в чистий JavaScript, що дозволяє його використовувати у всіх браузерах.

TypeScript дозволяє виявляти помилки на етапі компіляції завдяки статичній типізації. Це особливо важливо для великих проєктів, де контроль за типами даних значно знижує кількість помилок у коді.

1.2. Angular

Angular — це популярний фреймворк для розробки односторінкових веб-додатків. Розроблений компанією Google, Angular використовує TypeScript як основну мову програмування і пропонує безліч функціональних можливостей для створення масштабованих та підтримуваних веб-додатків.

Angular дозволяє будувати додатки на основі компонентів, що полегшує повторне використання коду, тестування та підтримку додатків.

Інтеграція з іншими бібліотеками: Angular має вбудовані механізми для інтеграції з різними бібліотеками та інструментами, такими як роутизація, форми, HTTP-запити.

Angular дозволяє створювати як малі, так і великі додатки завдяки модульності й інтегрованим інструментам для побудови складних рішень.

1.3. Tailwind CSS

Tailwind CSS — це утилітний CSS-фреймворк, який дозволяє створювати адаптивні і стильні інтерфейси користувача без написання великих стилізованих CSS-класів. Він пропонує велику кількість готових утилітних класів для роботи з кольорами, відступами, шрифтами, розмірами і так далі.

Завдяки готовим класам, розробники можуть швидко налаштувати стиль інтерфейсу без необхідності писати великий CSS. Це значно пришвидшує процес розробки.

1.4. Three.js

Three.js — це JavaScript-бібліотека, яка дозволяє створювати 3D-графіку для веб-додатків. Вона використовує WebGL для рендерингу графіки в браузері та забезпечує простий інтерфейс для створення складних 3D-сцен, анімацій і взаємодій.

Three.js приховує складність роботи з WebGL, надаючи розробникам простий API для створення 3D-сцен і рендерингу.

Оскільки Three.js працює в браузері за допомогою WebGL, розроблені 3D-сцени можна використовувати на будь-якому пристрої без необхідності додаткових плагінів.

Бібліотека підтримує складні 3D-анімації та інтерактивність, що дозволяє створювати як прості, так і висококласні 3D-інтерфейси.

2. РОЗРОБЛЕННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ

2.1. Аналіз вимог до розробленого програмного забезпечення

Графічний інтерфейс користувача (GUI) є важливою складовою будь-якого програмного забезпечення, оскільки він визначає взаємодію користувача з додатком. Для розробки ефективного та зручного інтерфейсу необхідно враховувати низку вимог, що стосуються його функціональності, зовнішнього вигляду, зручності використання та інтерактивності. Аналіз вимог до графічного інтерфейсу допомагає створити продукт, що буде не тільки інтуїтивно зрозумілим, але й ефективним у роботі. Нижче наведено ключові вимоги, які мають бути враховані при розробці графічного інтерфейсу програмного забезпечення.

2.2. Опис вимог до клієнтської частини до персонального асистента для ведення нотаток та планування подій

Одним із ключових етапів розробки програмного забезпечення є аналіз вимог. Він охоплює процес збору вимог, їх упорядкування, виявлення взаємозв'язків, а також подальше документування.

Отже, маємо такі вимоги стосовно клієнтської частини:

Інтерфейс повинен бути простим у використанні та зрозумілим для користувачів різного рівня підготовки. Головна мета — зменшити час на навчання та адаптацію до системи, а також знизити кількість помилок, які можуть виникнути через незрозумілий дизайн.

- Простота та очевидність елементів: Користувач повинен з першого погляду розуміти, які елементи інтерфейсу є клікабельними, які функції вони виконують.

- Логічна структура: Меню, кнопки та інші елементи повинні бути

організовані таким чином, щоб користувач не витрачав часу на пошук потрібної функції.

Інтерфейс має бути структурованим таким чином, щоб користувач міг безперешкодно переміщатися між різними екранами і функціями програми. Це забезпечує ефективну роботу з програмним забезпеченням і скорочує час на виконання завдань.

- **Висвітлений список доступних команд:** Відображення списку доступних команд для ознайомленості.

- **Голосова навігація між командами:** Додаток автоматично відображає необхідну інформацію на панелі відображення.

Інтерфейс повинен надавати користувачам чіткий зворотний зв'язок при кожній їх взаємодії з додатком. Кожен крок або дія повинні бути супроводжені відповідним відгуком від системи.

- **Візуальний зворотний зв'язок:** При взаємодії з користувачем має відображатись необхідна анімація та підсвічуватись мікрофон при запису мовлення.

- **Оповіщення та повідомлення:** Система повинна інформувати користувача про важливі події або помилки (наприклад, «Нотатка успішно збережена» чи «Невірна команда»).

Графічний інтерфейс повинен бути спрямований на полегшення виконання основних завдань користувачем. Для цього необхідно оптимізувати робочі процеси, зменшуючи кількість кроків, що необхідні для досягнення результату.

- **Мінімізація кількості кліків:** Основні функції мають бути доступні за мінімальну кількість кліків, що значно спрощує роботу з програмним забезпеченням.

- **Шаблони та заготовки:** Для створення нотаток і подій можуть бути використані форми, які автоматично заповнюються під час мовлення користувача та мають можливість редагування.

2.3. Архітектура клієнтської частини

Загалом розроблену систему можна охарактеризувати як клієнт-серверну архітектуру. Вона складається з серверної частини, яка обробляє основні операції з даними, та клієнтської частини, що забезпечує графічну та голосову взаємодію з користувачем. Однак, наразі зацентруємо увагу на архітектурі клієнтської частини.

Клієнтська частина має вигляд односторінкового сайту та містить в собі три основні модулі: модуль відображення команд, модуль анімації та відображення текстової відповіді та модуль голосового вводу. При голосовому вводі команди модуль відображення команд виводить на екран необхідну компоненту в залежності від озвученої команди. Всього клієнтська частина налічує 6 компонентів відображення команд, а саме: створення нотатки, відображення нотатків, читання нотатки, створення події, відображення запланованих подій на сьогодні/цей тиждень/наступний тиждень, видалення події.

Окрім цього, клієнтська частина містить в собі блок реєстрації та авторизації. Реєстрація складається з трьох етапів: етап введення персональних даних для входу, етап верифікації електронної адреси, етап інтеграції з Google calendar.

3. АНАЛІЗ РОЗРОБЛЕНИХ МОДУЛІВ

3.1. Особливості реалізації відображення необхідної анімації

Як було зазначено вище, для відображення 3D моделі у веб-додатку було обрано бібліотеку Three.js.

Для початку, було налаштовано сцену, а саме виставлено камеру, виставлено навколишнє та пряме освітлення, ініціалізовано рендеринг.

Після цього, було створено завантажено моделі з анімаціями. Усього наша система містить в собі 7 видів анімацій, а саме: вітання, відповідь, прослуховування, обмірковування, очікування, помилка та прощання.

Клієнтська частина містить у собі сервіс, в якому знаходиться observer, у який в потрібний момент додається необхідна анімація, а модуль анімації в свою чергу реагує на зміни у observer, запускаючи необхідну анімацію. Наприклад, під час натискання на мікрофон додається анімація прослуховування, після чого модуль анімації автоматично видаляє зі сцени попередній об'єкт з анімацією та додає на сцену новий об'єкт та відіграє задану анімацію.

3.2. Особливості реалізації голосового створення та пагінації

Компоненти створення містять в собі форми для редагування інформації стосовно нотатки та події. При виконанні команди створення сервер надсилає питання, на які користувач має відповісти, як от назва нотатки/події, текст нотатки, дата початку/закінчення події. При отриманні відповіді на запитання система автоматично заповнює форми розпізнаним текстом, після чого користувачу надається можливість редагування даної форми. Після завершення користувач натискає кнопку відправлення і нотатка / подія успішно створюється.

Компонента відображення списку нотаток містить в собі пагінацію. На початку дана компонента відображає перші 5 нотаток. Після чого озвучується питання чи має бажання користувач отримати більше нотаток. В разі ствердної відповіді клієнтська частина відправляє на сервер номер наступної сторінки і отримує наступні 5 нотаток. Даний цикл триває до моменту, поки користувач не озвучить не ствердну відповідь.

3.3. Особливості реалізації модуля запису та озвучування тексту

Запис та зупинка запису реагують на включення або виключення кнопки запису.

Під час запису на початку запитуються права на доступ до мікрофона користувача. Якщо доступ до мікрофона отримано, створюється новий об'єкт `MediaRecorder`, який починає запис звуку з потоку. Після чого дані під час запуску поступово зберігаються у масив. Після виключення кнопки запису зібрані аудіофрагменти об'єднуються в один об'єкт, який представляє записане аудіо у форматі `audio/wav`.

При озвучуванні тексту із серверу надходить рядок, що містить аудіо в форматі `base64`. Тож, коли клієнтська частина отримує даний рядок, вона декодує рядок, що містить аудіо в `base64`, у звичайний бінарний рядок. Це необхідно для перетворення даних з `base64` у бінарний формат, який можна обробляти як аудіо. Після чого кожен символ перетворюється в байт та записується у спеціальний масив для того, щоб створити об'єкт, що містить в собі аудіофайл у форматі `MPEG`, який відіграється через створену URL через HTML-елемент `audio`.

3.4. Тестування розроблених компонентів

Тестування системи відбулося шляхом мануального тестування.

Для тестування було написано та протестовані такі варіанти:

Гарні сценарії:

- створення нотатки
- відображення нотаток
- пагінація нотаток
- читання нотатки
- створення події
- відображення події на сьогодні/на тиждень/на наступний тиждень
- видалення події

Перевірка на обробку поганих сценаріїв:

- озвучення команди, якої не існує
- створення нотатки без назви або опису
- читання неіснуючої нотатки
- створення події без назви/дати
- створення події з некоректною датою
- видалення неіснуючої події

3.5. Рекомендації щодо подальшого вдосконалення

У процесі роботи над персональним асистентом для ведення нотаток і планування подій із використанням голосового модуля було визначено низку напрямів і функцій, які можуть покращити розроблене програмне забезпечення, зокрема:

- Розширення списку голосових команд для управління нотатками та подіями, щоб забезпечити більшу гнучкість у взаємодії.
- Інтеграція з іншими видами календарів (наприклад, Outlook, Apple Calendar), для більш зручної синхронізації своїх подій.

- Додавання мобільної версії додатку для забезпечення доступу до асистента з будь-якого пристрою.
- Оптимізація алгоритмів обробки команд, задля зменшення затримки під час виконання запитів.

ВИСНОВКИ

Метою цієї курсової роботи було розроблення персонального асистента для ведення нотаток та планування подій, який має забезпечити користувачів зручними інструментами для організації та управління їхньою діяльністю. В рамках цієї роботи було розроблено клієнтську частину програми, яка включає функціональність створення, відображення та читання нотаток, а також можливість планування подій з їх подальшим відображенням на сьогодні, на тиждень чи наступний тиждень.

Аналіз засобів розробки таких систем показав доцільність використання сучасних веб-технологій, таких як Angular, TypeScript та популярних фреймворків для побудови зручних інтерфейсів. Важливими компонентами стали інтеграція з базою даних для збереження нотаток і подій, а також реалізація пагінації для ефективного відображення великої кількості даних.

Розроблена система має такі основні функціональні можливості:

- створення нотатки
- відображення нотаток
- пагінація нотаток
- читання нотатки
- створення події
- відображення події на сьогодні/на тиждень/на наступний тиждень
- видалення події

Робота виконана у повному обсязі, з урахуванням всіх вимог технічного завдання. Тестування продукту було проведено відповідно до затвердженої методики та програми тестування, що забезпечило надійність та ефективність розробленої системи.

Використання персонального асистента для ведення нотаток та планування подій дозволить значно спростити процес організації діяльності, підвищити ефективність управління часом користувачів, а також забезпечить зручність і доступність для різних категорій користувачів. Це може стати корисним інструментом для людей, які активно працюють з інформацією та потребують надійного та інтуїтивно зрозумілого помічника для планування та організації свого часу.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. TypeScript Programming. [Електронний ресурс]. — Режим доступу:
<https://code.visualstudio.com/docs/languages/typescript>
2. An introduction to Three.js. [Електронний ресурс]. — Режим доступу:
<https://humaan.com/blog/web-3d-graphics-using-three-js/>
3. Getting Started with Angular Material [Електронний ресурс]. — Режим доступу: <https://material.angular.io/guide/getting-started>
4. Tailwind [Електронний ресурс]. — Режим доступу:
<https://tailwindui.com/documentation>
5. Record Audio in JS and upload as wav or mp3 file to your backend [Електронний ресурс]. — Режим доступу:
<https://franzeus.medium.com/record-audio-in-js-and-upload-as-wav-or-mp3-file-to-your-backend-1a2f35dea7e8>

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет прикладної математики

“ЗАТВЕРДЖЕНО”

Керівник роботи

_____ Є.С.Сулема

“ ____ ” _____ 2024 р.

ПЕРСОНАЛЬНИЙ АСИСТЕНТ ДЛЯ ВЕДЕННЯ НОТАТОК ТА
ПЛАНУВАННЯ ПОДІЙ
Керівництво користувача
ПЗКС.045440-05-34

Виконавці:

_____ Потапчук А.А.

_____ Беліцький О.С.

_____ Слободзян М.В.

_____ Пецеля А.В.

ЗМІСТ

1. Опис структури системи.....	3
2. Опис реєстрації та авторизації.....	3
3. Опис взаємодії з нотатками.....	5
4. Опис взаємодії з подіями.....	8

1. Опис структури системи

Система складається з наступних сторінок:

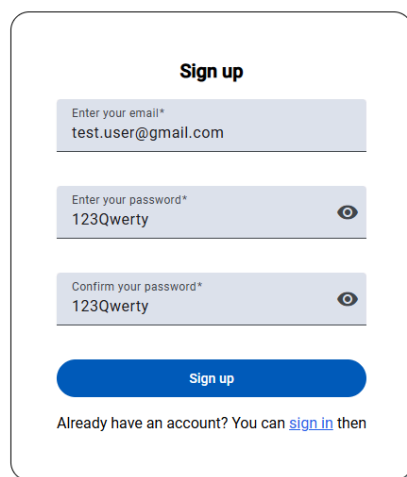
- створення нотатки
- відображення нотаток
- читання нотатки
- створення події
- відображення події на сьогодні/на тиждень/на наступний тиждень
- видалення події

Кожна сторінка містить зверху хедер, де відображені найменування поточної сторінки з навігацією на головну сторінку.

2. Опис реєстрації та авторизації

Для користування програмою необхідно зареєструватись у системі. Реєстрація складається з трьох етапів: введення інформації для входу в систему, підтвердження електронної адреси та інтеграції з Google calendar.

Для проходження першого етапу реєстрації необхідно ввести електронну адресу, пароль та підтвердження паролю.



Sign up

Enter your email*
test.user@gmail.com

Enter your password*
123Qwerty

Confirm your password*
123Qwerty

Sign up

Already have an account? You can [sign in](#) then

Рисунок 1 – Форма реєстрації

Після введення усіх даних та натискання на кнопку “Sign up” користувачу на вказану поштову адресу має надійти лист з кодом верифікації. Даний код необхідно ввести у поле, зображене на рис. 2.

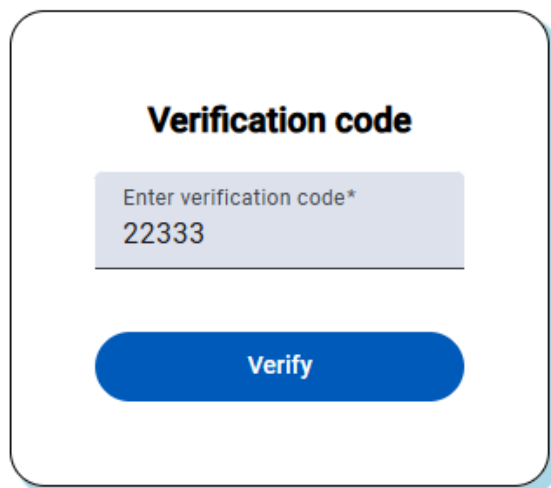
A screenshot of a verification code form. It has a light blue rounded rectangular border. At the top, the text "Verification code" is centered in bold black font. Below it is a light blue input field with the placeholder text "Enter verification code*" and the value "22333" entered. At the bottom, there is a blue rounded button with the text "Verify" in white.

Рисунок 2 – Форма коду підтвердження

Після успішної верифікації, користувачу необхідно додати Google Calendar ID. Ідентифікатор календаря можна знайти у налаштуваннях календаря. Даний ідентифікатор необхідно ввести у поле, що зображено на рис. 3. В разі успішного додавання користувача буде зареєстровано у системі.

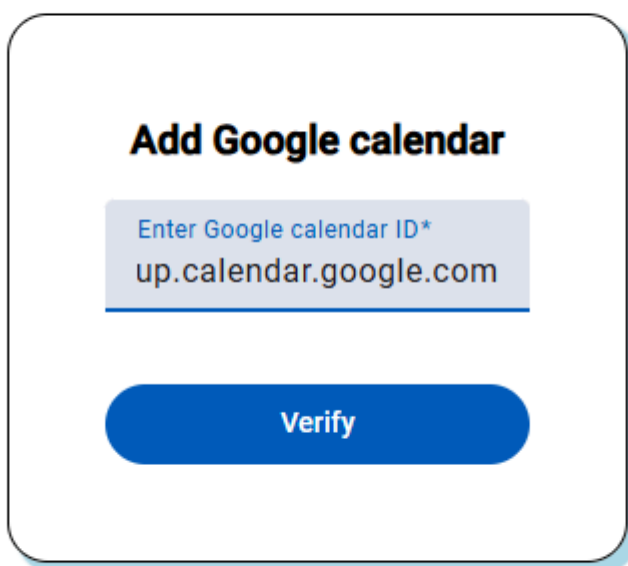
A screenshot of a form to add a Google calendar. It has a light blue rounded rectangular border. At the top, the text "Add Google calendar" is centered in bold black font. Below it is a light blue input field with the placeholder text "Enter Google calendar ID*" and the value "up.calendar.google.com" entered. At the bottom, there is a blue rounded button with the text "Verify" in white.

Рисунок 3 – Форма вводу Google Calendar ID

Після того, як користувача було успішно авторизовано у системі, його переносить на головний екран системи, де його очікує персональний асистент. Також на головній сторінці, що зображена на рис. 4, для ознайомлення присутні основні команди, які допомагають здійснювати керування помічником.

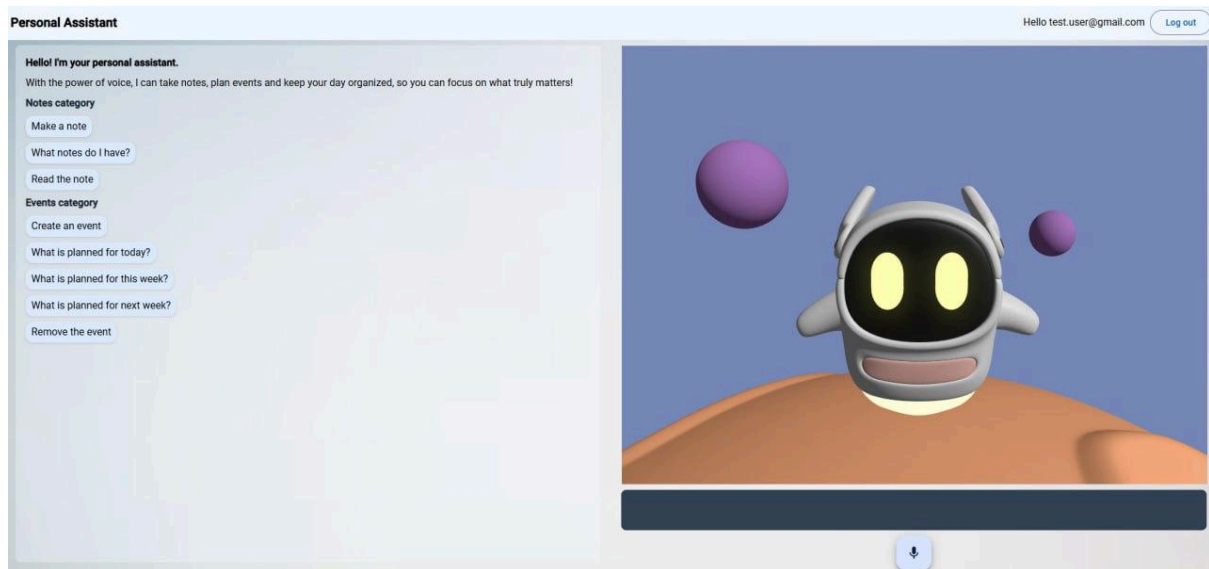


Рисунок 4 – Головна сторінка системи

3. Опис взаємодії з нотатками

3.1. Опис створення нотатки

На головній сторінці користувачу необхідно увімкнути мікрофон та сказати команду “Make a note”. Після розпізнавання команди буде виведено форму для створення нотатки. Спочатку необхідно проговорити назву нотатки. Після розпізнавання назви нотатки система автоматично заповнить поле “Name”. Після цього користувач матиме можливість озвучити текст нотатки. Аналогічно до назви, текст, сказаний користувачем, буде розпізнано та записано в поле “Description”.

Після заповнення форми користувач матиме змогу відредагувати нотатку за необхідності. Далі необхідно натиснути на кнопку “Submit” для збереження нотатки. Дану сторінку можна побачити на рис. 5.

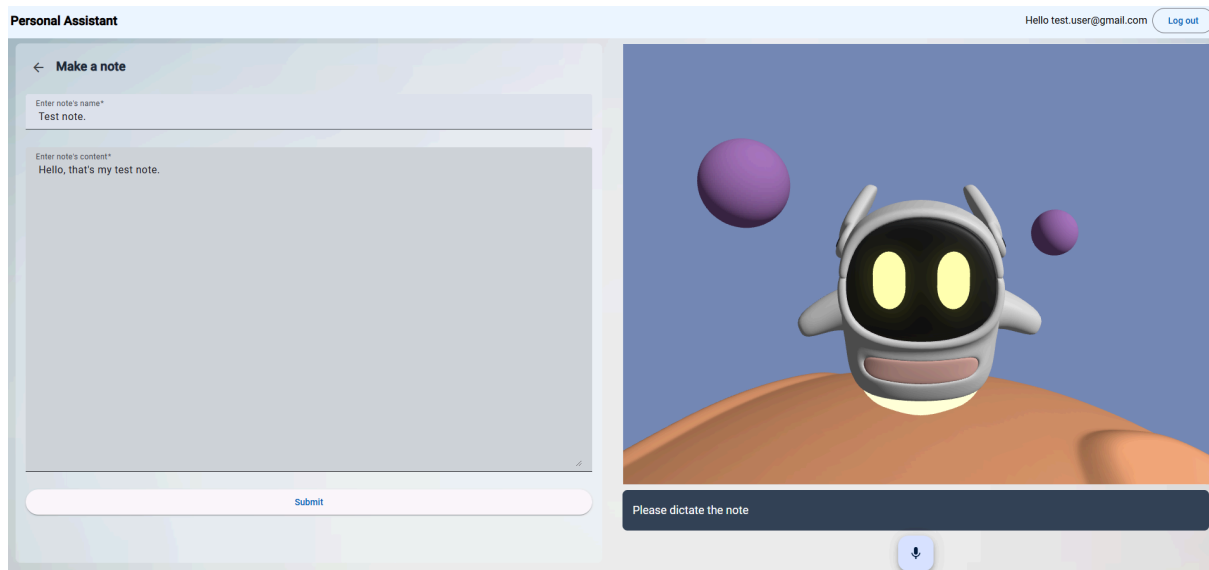


Рисунок 5 – Форма створення нотатки

У разі успішного збереження нотатки асистент сповістить користувача про успішне збереження нотатки (рис. 6).

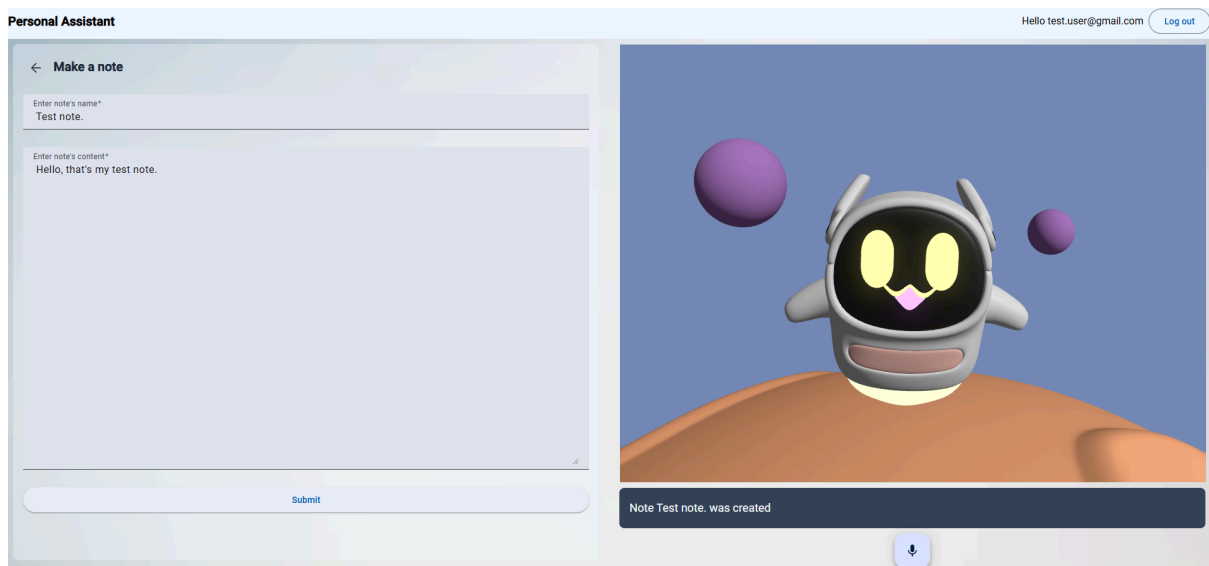


Рисунок 6 – Сповіщення аватара про успішне створення нотатки

3.2. Опис відображення нотаток

На головній сторінці користувачу необхідно увімкнути мікрофон та сказати “What notes do I have?”. Після цього користувача переведе в окреме меню, де він зможе переглянути свої нотатки (рис. 7). Якщо нотаток багато, то користувач може попросити показати більше нотаток.

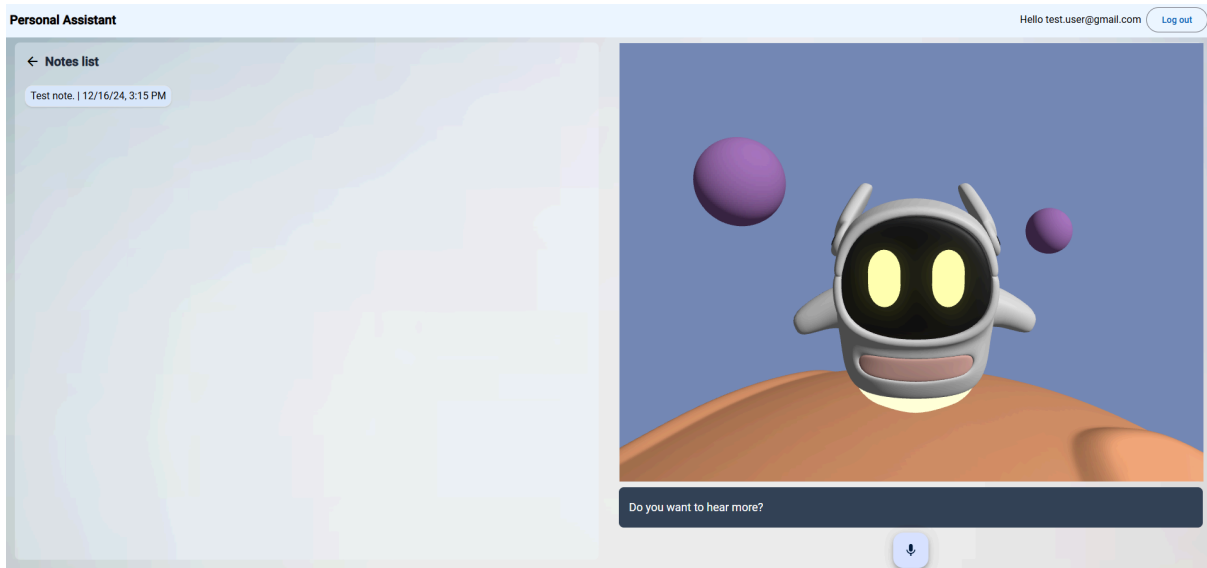


Рисунок 7 – Меню перегляду нотаток

3.3. Опис читання нотатки

На головній сторінці користувачу необхідно увімкнути мікрофон та сказати “Read the note”. Після цього користувача переведе в окреме меню, де асистент запитає користувача про ім’я нотатки, яку він хоче почути (рис. 8).

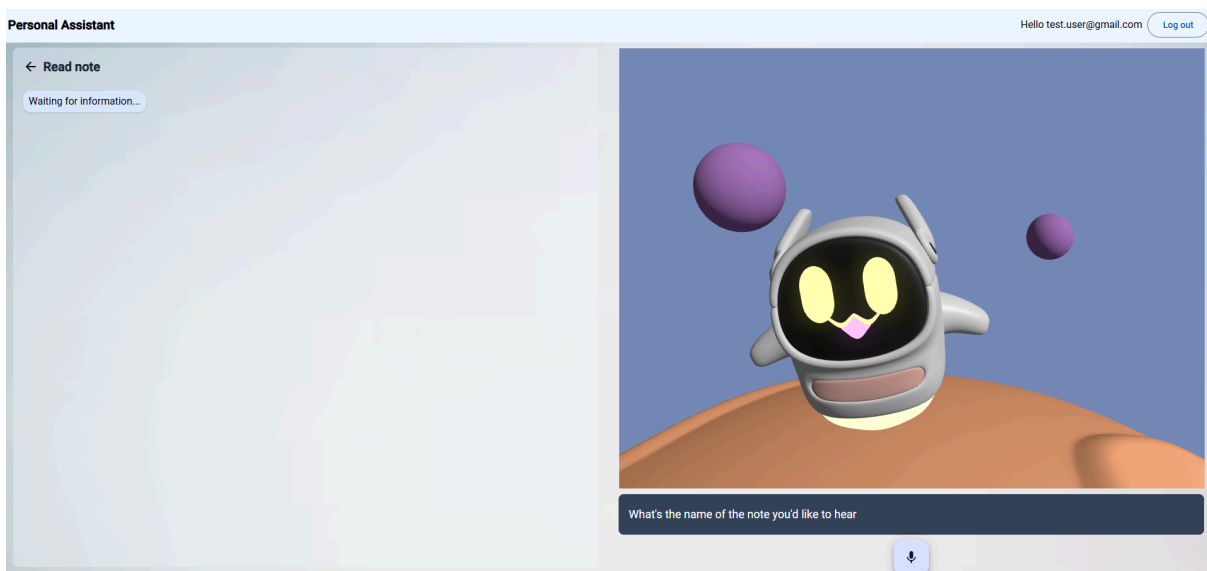


Рисунок 8 – Меню очікування назви нотатки

Після введення назви нотатки з’явиться вікно (рис. 9), у якому буде текст нотатки та асистент почне її озвучувати.

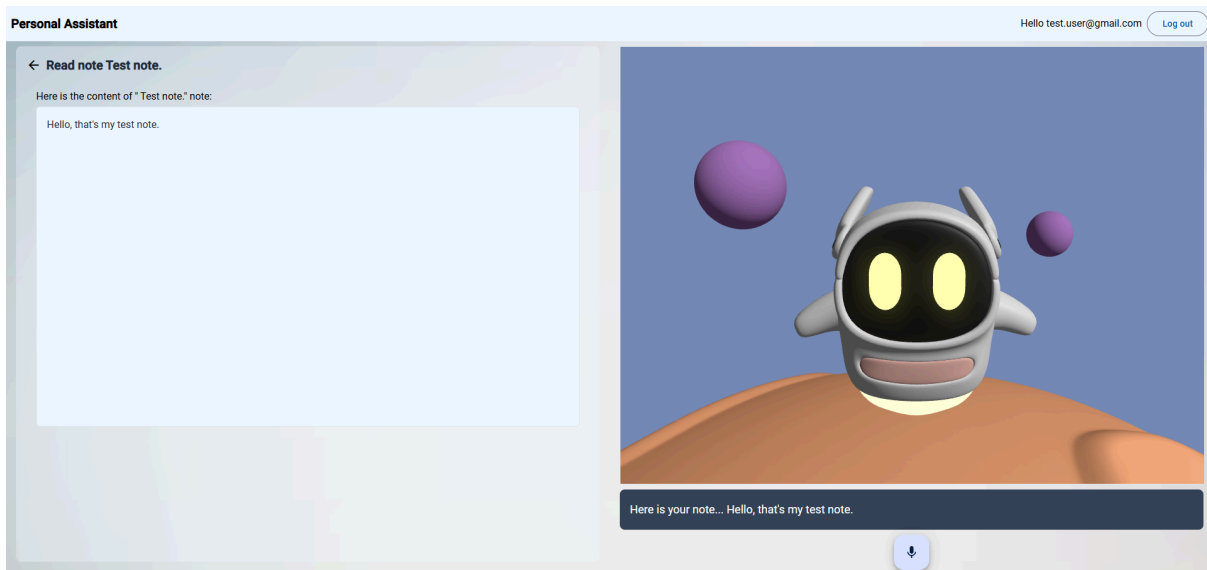


Рисунок 9 – Вікно з текстом нотатки

4. Опис взаємодії з подіями

4.1. Опис створення події

На головній сторінці користувачу необхідно увімкнути мікрофон та сказати “Create an event”. Після цього користувача переведе в окреме меню (рис. 10), де асистент запитає користувача про ім’я події, час початку та час кінця цієї події. Після заповнення відповідних полів необхідно натиснути кнопку “Submit” (рис. 11).

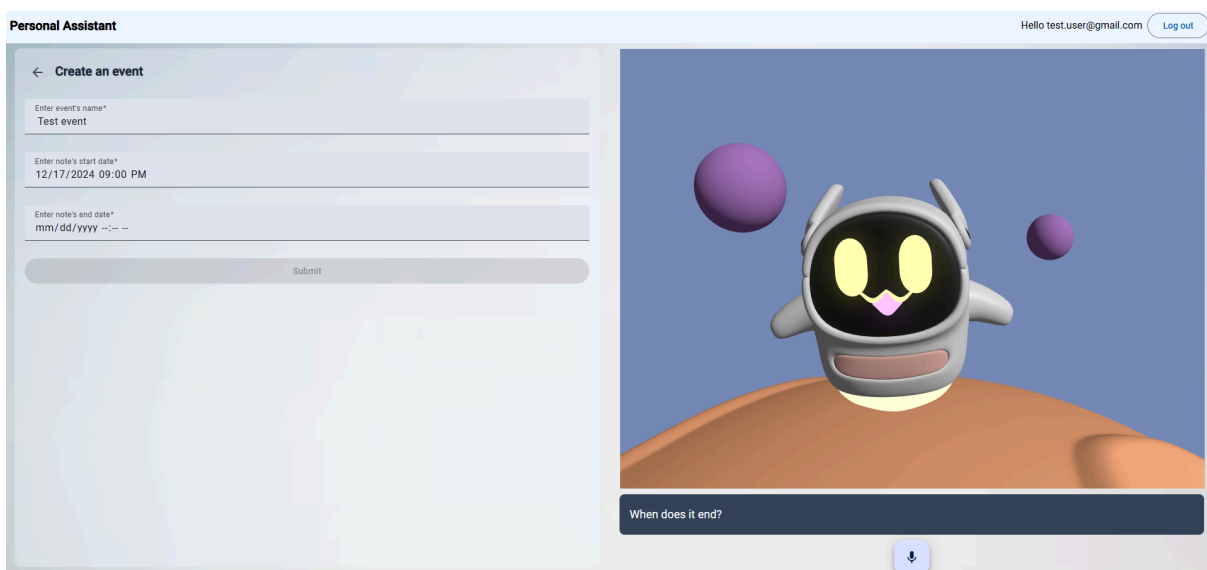


Рисунок 10 – Вікно створення події

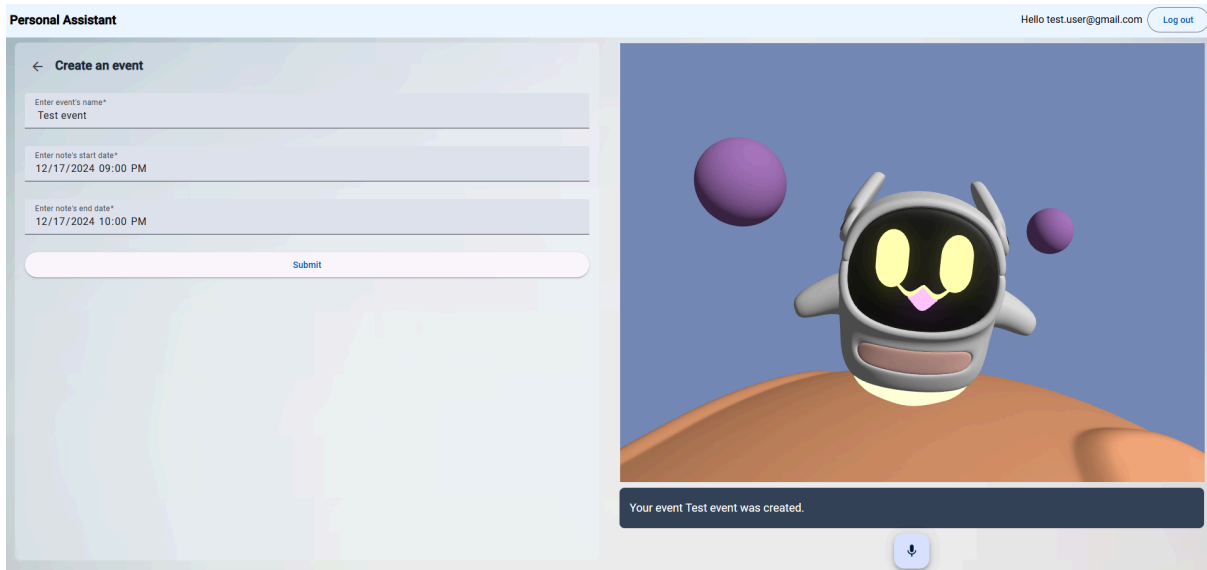


Рисунок 11 – Активна кнопка “Submit”

4.2. Відображення події

На головній сторінці користувачу необхідно увімкнути мікрофон та сказати “What is planned for *time*”. Замість “*time*” необхідно сказати часовий проміжок, який цікавить користувача, наприклад, “today”, “this week” або “next week”. Після цього користувача переведе в окреме меню (рис. 12-13), де користувач зможе переглянути події заплановані на той час, який вказав.

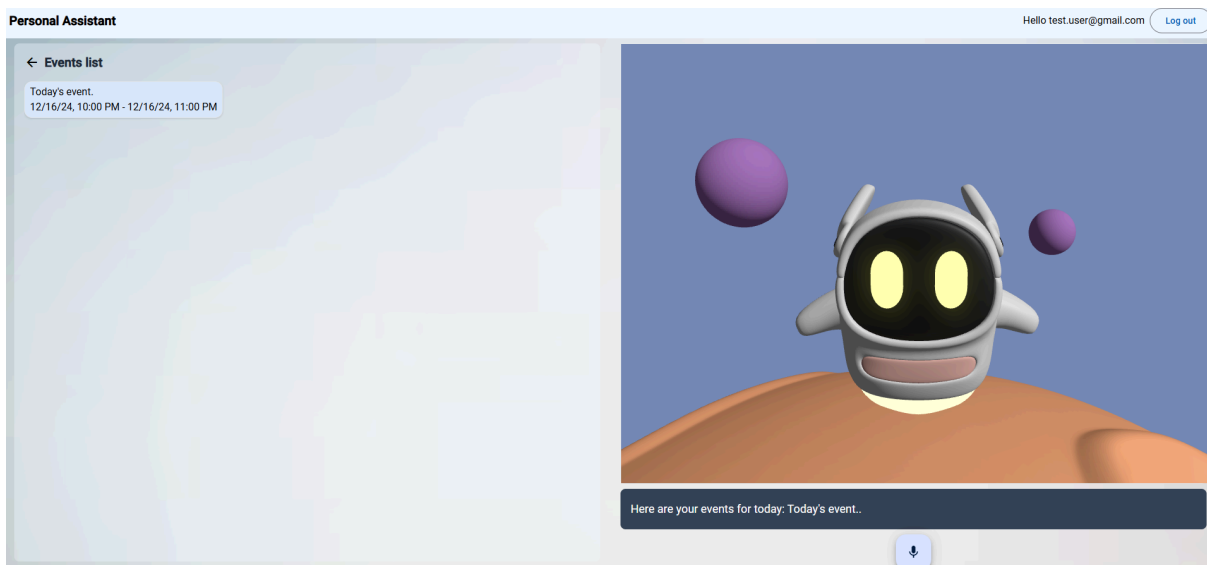


Рисунок 12 – Події, якщо в команді вказати “today”

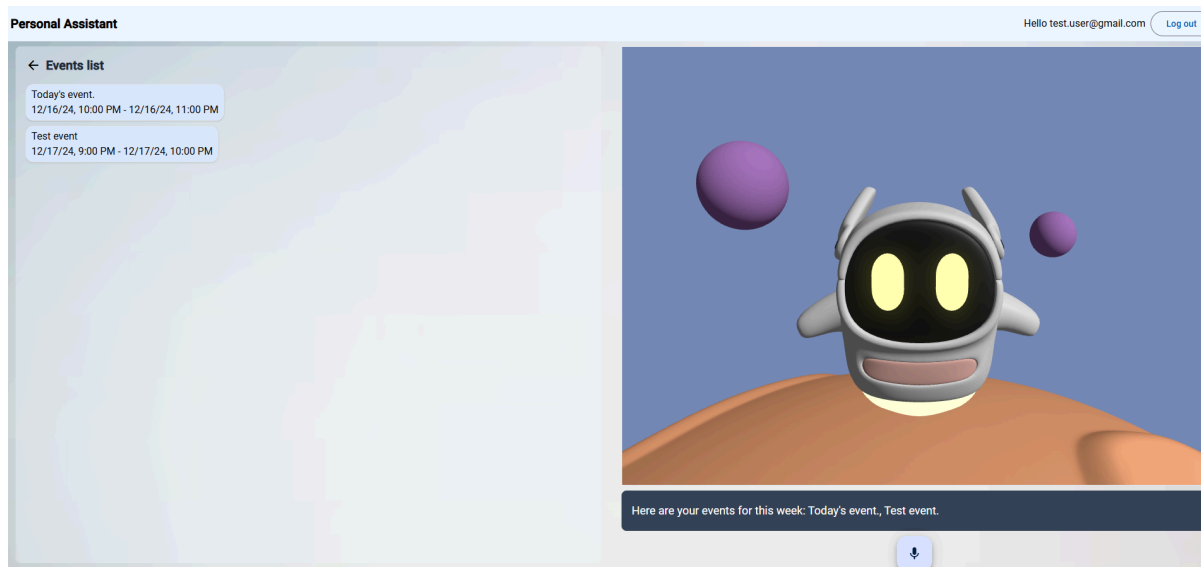


Рисунок 13 – Події, якщо в команді вказати “this week”

Всі події відмічені у власному Google Calendar користувача (рис. 14).

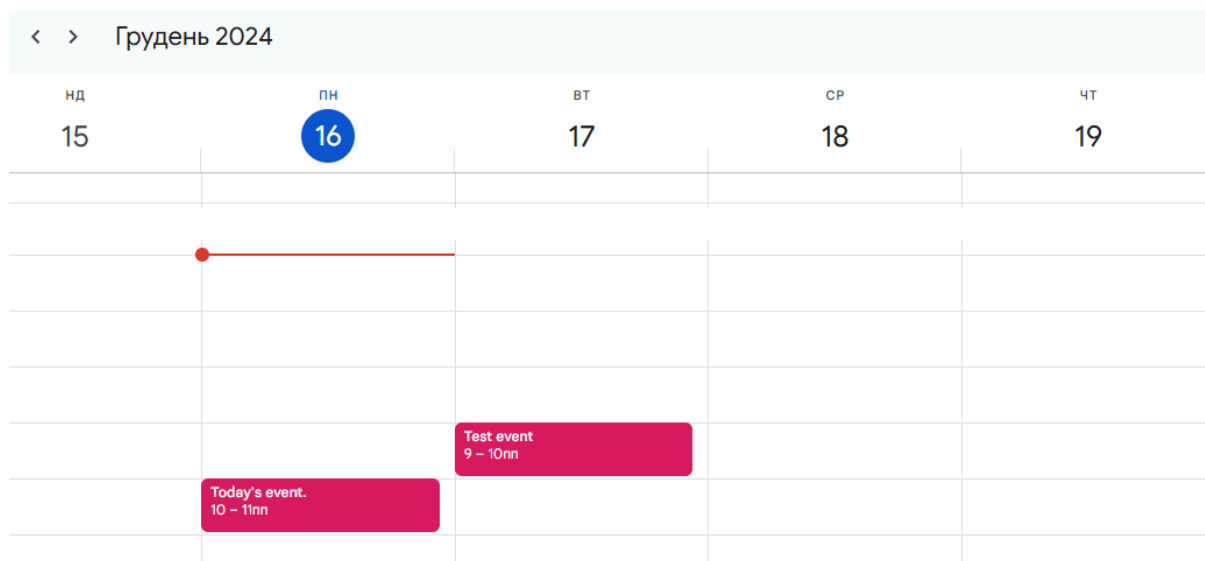


Рисунок 14 – Відображення подій в Google Calendar

4.3. Видалення події

На головній сторінці користувачу необхідно увімкнути мікрофон та сказати “Remove the event”. Після цього користувача переведе в окреме меню (рис. 15), де асистент запитає користувача про ім'я події, яку необхідно видалити. Після цього подію буде видалено.

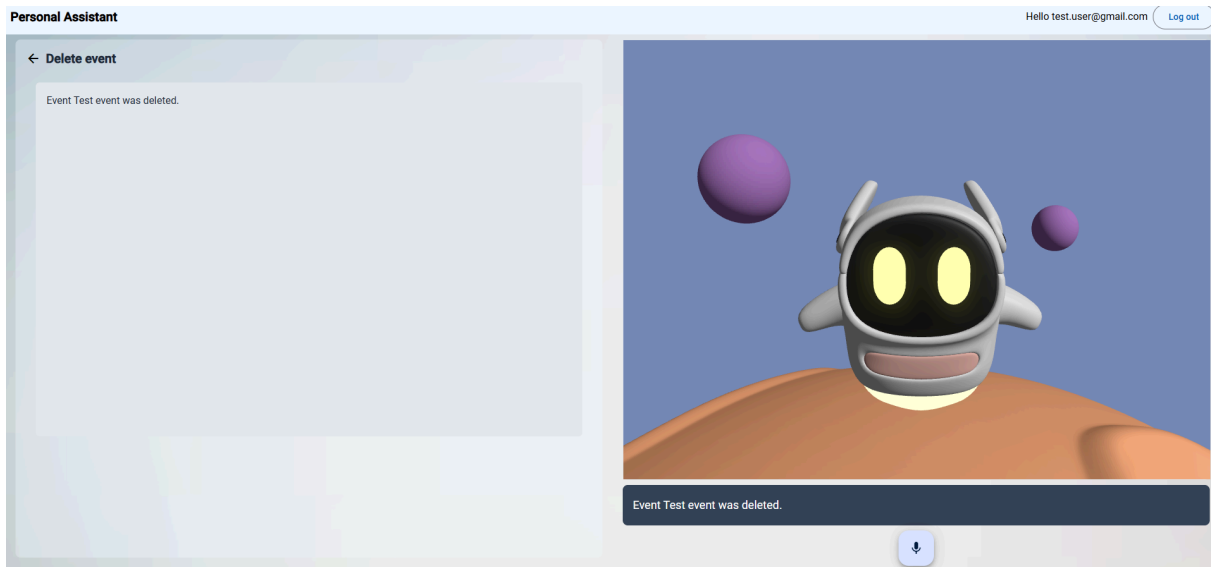


Рисунок 15 – Відображення видалення події

Видалення події також відбувається і в Google Calendar користувача (рис. 15).

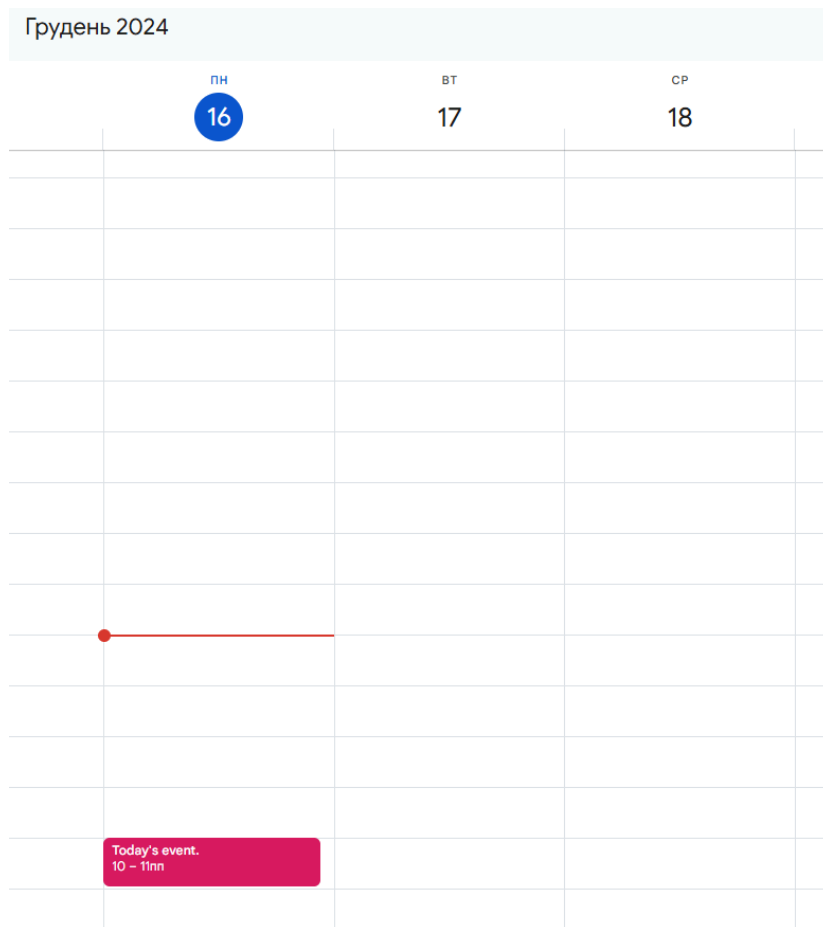


Рисунок 15 – Відображення подій в Google Calendar