

Міністерство освіти і науки України  
Криворізький національний університет  
Факультет інформаційних технологій  
Кафедра автоматизації, комп'ютерних наук і технологій

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти – бакалавр

за освітньо-професійною програмою

«Комп'ютерні науки»

зі спеціальності

122 – Комп'ютерні науки

тема роботи:

***«Розробка мобільного додатку для ефективного керування справами»***

Виконав студент гр. КН-20 \_\_\_\_\_ Гречко Є. В.

Керівник \_\_\_\_\_ Рубан С. А.

Нормоконтроль \_\_\_\_\_ Маринич І. А.

Завідувач кафедри \_\_\_\_\_ Рубан С. А.

Кривий Ріг – 2024

# КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Бакалавр

Спеціальність: 122 – Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Зав. кафедрою: к.т.н. Рубан С.А.

« 27 » березня 2024 р.

## ЗАВДАННЯ

### на кваліфікаційну роботу бакалавра

студентові групи КН-20 Гречко Єлізаветі Вадимівні

**1. Тема кваліфікаційної роботи:** «Розробка мобільного додатку для ефективного керування справами»

затверджено наказом по університету № 235с від 27.03.2024 р.

**2. Термін здачі кваліфікаційної роботи:** 05.06.2024 р.

**3. Склад кваліфікаційної роботи:** Пояснювальна записка обсягом 53с., презентація у Microsoft PowerPoint (18 слайдів) в електронному та друкованому вигляді

**4. Консультанти кваліфікаційної роботи:**

Розділ 1-2

доц. Рубан С. А.

Нормоконтроль

доц. Маринич І. А.

## 5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>01.04.24</i>
2	<i>Розділ 1</i>	<i>05.04.24</i>
3	<i>Розділ 2</i>	<i>01.05.24</i>
4	<i>Висновки</i>	<i>25.05.24</i>
5	<i>Оформлення кваліфікаційної роботи</i>	<i>28.05.24</i>
6	<i>Підготовка презентації та графічного матеріалу</i>	<i>20.05.24</i>
7	<i>Підготовка доповіді до захисту</i>	<i>05.06.24</i>

6. Дата видачі завдання: 29.01.2024р.

Керівник \_\_\_\_\_ /Рубан С. А./

7. Запевнення: Я, Гречко Єлизавета Вадимівна, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Студент \_\_\_\_\_ /Гречко Є. В./

## АНОТАЦІЯ

Гречко Є. В. Розробка мобільного додатку для ефективного керування справами

Кваліфікаційна робота на здобуття ступеня вищої освіти – бакалавр, за спеціальністю 122 Комп'ютерні науки. Криворізький національний університет, Кривий Ріг, 2024.

Робота складається зі вступу, двох розділів, висновків, переліку використаної літератури з 41 позиції. Загальний обсяг роботи становить 53 сторінок, з яких основний зміст роботи викладено на 46 сторінках, включає 28 рисунків.

Було розглянуто концепцію менеджерів для ефективного керування справами. Описано механізми роботи подібних додатків, їх функціональні можливості та методи організації робочих процесів. Наведено обґрунтування вибору механізмів, які забезпечують зручність і ефективність користування додатком, оптимізуючи час і ресурси користувачів для досягнення максимального результату. Проведено порівняльний аналіз популярних менеджерів для керування справами, що дозволило визначити їх основні функції та характеристики. Це допомогло зрозуміти, які особливості варто реалізувати у власному додатку для досягнення максимальної ефективності та зручності у використанні. Для реалізації програмного забезпечення було обрано середу розробки Visual Studio Code та мову програмування React Native. Visual Studio Code забезпечує зручний процес розробки, а React Native дозволяє створювати кросплатформні мобільні додатки з єдиною кодовою базою, що економить час і ресурси. Використання цих інструментів сприяє швидкій та якісній розробці додатку для керування справами.

Додаток має повноцінний графічний інтерфейс завдяки можливостям мови React Native. Завдяки можливостям бібліотеки Mongoose реалізований бекенд для взаємодії з базою даних проєкту.

БАЗА ДАНИХ, REACT NATIVE, MONGOOSE, VISUAL STUDIO CODE,  
МЕНЕДЖЕР ДЛЯ ЕФЕКТИВНОГО КЕРУВАННЯ СПРАВАМИ

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>РОЗДІЛ 1 ВИВЧЕННЯ РИНКУ МОБІЛЬНИХ МЕНЕДЖЕРІВ ЗАДАЧ</b> .....	6
1.1 Аналіз проблеми .....	6
1.2 Визначення цілей та результатів проєкту .....	18
Висновки до розділу .....	23
<b>РОЗДІЛ 2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	24
2.1 Формулювання завдання для реалізації програмного рішення .....	24
2.2 Опис програмної реалізації системи.....	26
Висновки до розділу: .....	44
<b>ВИСНОВКИ</b> .....	46
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</b> .....	47

## ВСТУП

Ефективний тайм-менеджмент є невід'ємною частиною успішного життя. Планування часу має вирішальне значення для підвищення ефективності під час роботи та максимізації робочих зусиль. Завдяки грамотному розподіленню часу можна уникнути стресу та забезпечити контроль над завданнями і їх вчасним виконанням.

Чіткий шлях до цілей і систематичний контроль прогресу забезпечується саме плануванням, що сприяє успішному досягненню своєї мети. При зосередженні на важливих завданнях підвищується продуктивність за рахунок меншої кількості відволікаючих факторів.

Для ефективного використання робочих годин неабияк важлива оптимізація часу, завдяки чому він не буде витрачатися на неважливі завдання. Також завдяки ретельному плануванню можна досягнути балансу між роботою та відпочинком, що має велике значення для підтримки високої ефективності.

Тайм-менеджмент є необхідним інструментом для розвитку себе як особистості. Саме він дозволяє здобувати нові навички та відчувати систематичний прогрес у навчанні, сприяючи професійному та особистому розвитку. Ефективний розподіл часу має позитивний вплив на життя будь-якої людини, допомагає досягти балансу між професійним та особистим життям і цим підвищує якість життя. Тайм-менеджмент також сприяє зменшенню кількості стресу і уникненню перенавантаження, що має позитивний вплив на фізичне та психо-емоціональне здоров'я.

Ще одним позитивним наслідком раціонального планування часу є збереження власної енергії. Якщо уникати хаотичності у виконанні завдань і зосереджуватися лише на тому, що є важливим, витрати енергії зводяться до мінімуму. Також завдяки уникненню хаотичності забезпечується достатній час для відновлення та відпочинку, що є корисним для фізичного та психо-емоціонального здоров'я

Саме з цих причин одними із найважливіших навичок у житті кожної людини є навички планування свого часу. Адже вони допомагають з більшою ефективністю із підвищенням продуктивності, зменшенням стресу і досяганням поставлених цілей. Також навички тайм-менеджменту привносять баланс у життя людей.

Після обговорення цього питання з іншими людьми, було виявлено наступний ряд проблем:

- Відсутність звички планувати;
- Нестача мотивації для реалізації планів;
- Важко відстежити час, витрачений на роботу;
- Важко концентруватися на виконанні завдань;

Усі ці проблеми призводять до найбільшої – прокрастинації, з якою важко впоратися, якщо не позбутися її причин. Через це до планування свого часу необхідно підходити комплексно. Воно повинно бути цікавим, допомагати підтримувати мотивацію на високому рівні, результати роботи мають бути демонстраційними. Також перед менеджером задач стоїть важливе завдання: поступово навчити користувача концентруватися на виконанні завдань.

Саме тому було прийнято рішення розробити мобільний додаток для ефективного керування справами, який буде мотивувати людей та сприятиме виробленню звички до планування, дозволить людям ефективніше використовувати свій час і допоможе людям зосереджуватися на більш важливих і термінових завданнях.

# РОЗДІЛ 1

## ВИВЧЕННЯ РИНКУ МОБІЛЬНИХ МЕНЕДЖЕРІВ ЗАДАЧ

### 1.1 Аналіз проблеми

У сьогоднішній день, проблема ефективного планування та вибору ідеального мобільного додатка для керування справами полягає в тому, що на ринку представлено велику кількість додатків з різними функціями та можливостями. Це ускладнює вибір оптимального варіанту. Деякі додатки можуть бути надто насиченими функціями, тоді як інші можуть бути недостатньо функціональними. Важливо враховувати особисті потреби та вимоги до додатку, такі як інтерфейс, можливість синхронізації, нагадування та інші.

Найпопулярніші мобільні додатки для керування справами:

— Todoist – популярний додаток для організації завдань і списків справ, яким користуються більше 42 мільйонів людей і команд по всьому світу. Він сприяє зниженню ментального навантаження, підвищенню ефективності та формуванню корисних звичок [1, 2].

Todoist включає в себе такі функції як швидке додавання завдань одним дотиком, зокрема з деталями на кшталт «завтра о 16:00», які автоматично розпізнаються додатком. Кілька режимів відображення: календар, список та дошка. Фільтрація робочих та особистих завдань. Співпраця над проектами та досягнення душевного спокою. Синхронізація між iPhone, iPad, Apple Watch та комп'ютером. Інтеграція з календарем, голосовим асистентом, Outlook, Gmail, Slack та понад 60 іншими інструментами. Делегування завдань та співпраця над проектами будь-якого масштабу [2, 3].

Todoist вирізняється поєднанням простоти і потужних функцій, інтуїтивно зрозумілим інтерфейсом та можливістю організувати робочі та особисті завдання в одному місці (рис. 1.1). Додаток допомагає підтримувати ясність думок, фокусуватися на важливому та досягати поставлених цілей [1, 2, 3].



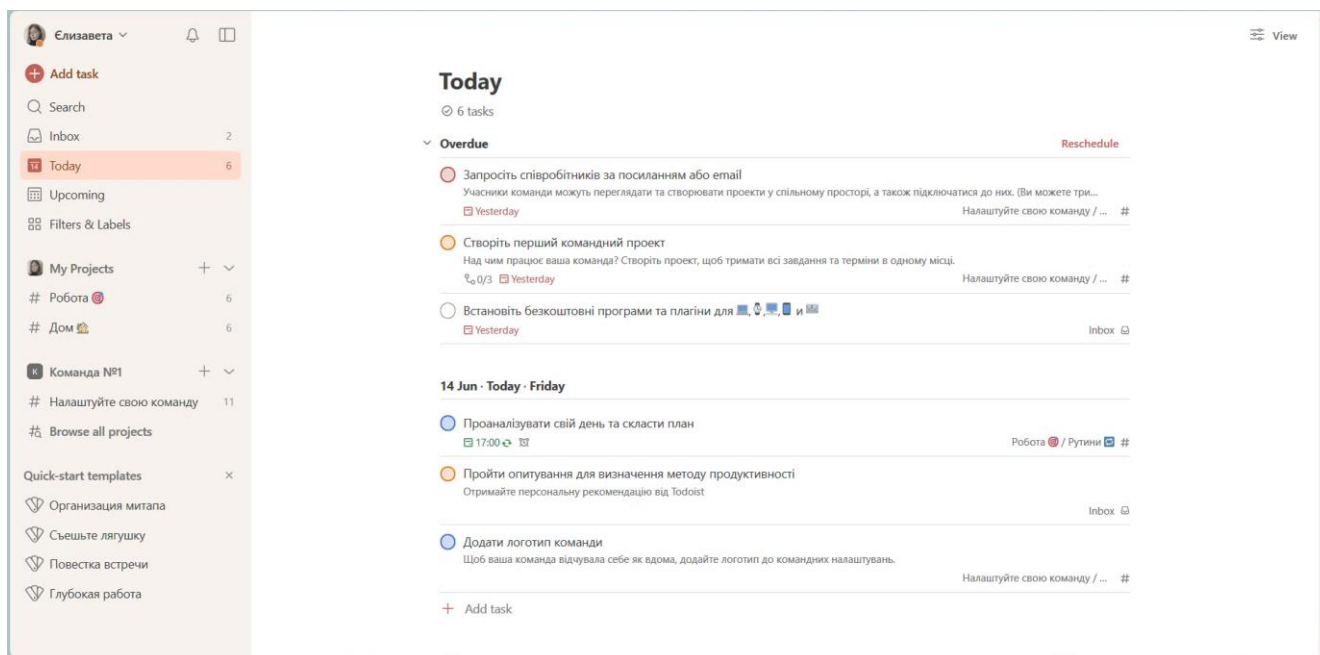


Рисунок 1.1 – Зображення головного екрану Todoist

— Trello – це візуальний інструмент для керування проектами, який використовує дошки, списки та картки для впорядкування та пріоритизації завдань (рис. 1.2). Дозволяє команді спільно працювати та слідкувати за прогресом [4].

Trello пропонує гнучку структуру завдань, яка дозволяє створювати і впорядковувати завдання для будь-яких проєктів та команд. Завдяки функції drag-and-drop, користувачі можуть легко переміщувати та оновлювати завдання, перетягуючи їх між списками та дошками. Інтуїтивно зрозумілий інтерфейс забезпечує швидке освоєння і ефективну роботу з додатком. Trello підтримує співпрацю та організацію, дозволяючи командам працювати разом над проєктами, планувати та керувати завданнями продуктивно та організовано. Незалежно від того, чи починає користувач новий проєкт, чи вдосконалює поточний, Trello адаптується до будь-яких потреб, спрощуючи і стандартизуючи робочий процес команди [5, 6, 7].

Перелічені функції роблять Trello потужним інструментом для керування проектами та завданнями, що значно сприяє підвищенню продуктивності та організації робочих процесів.

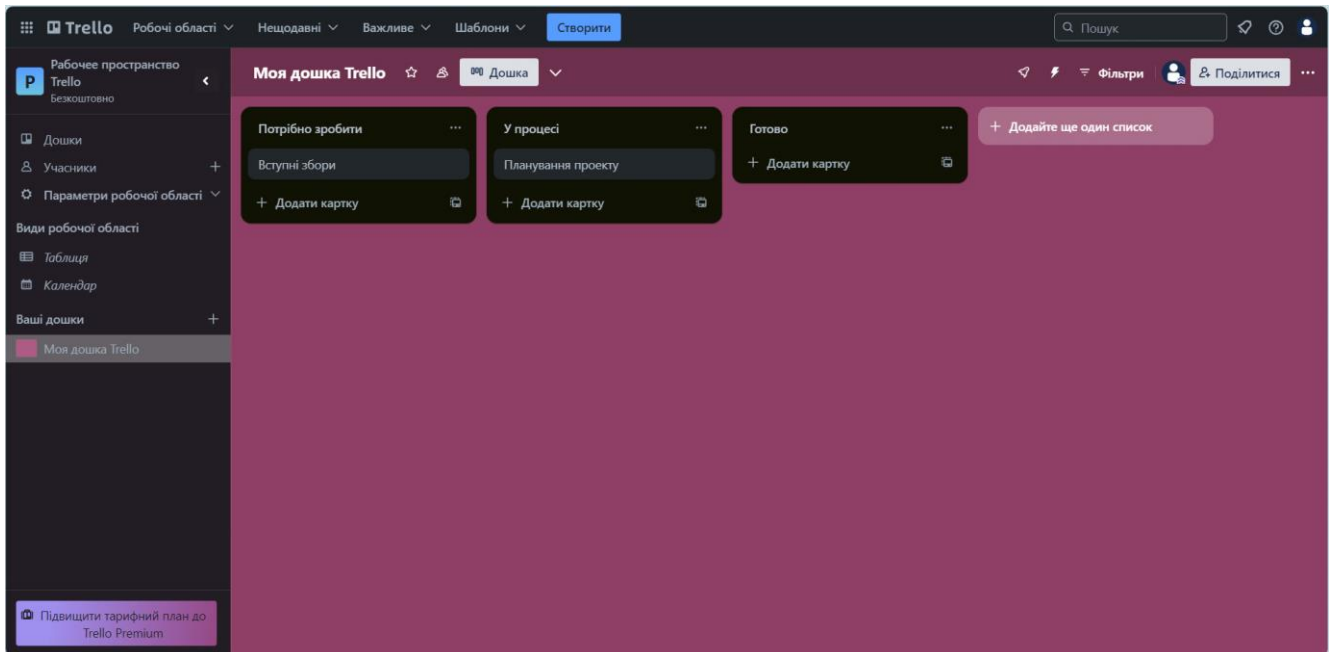


Рисунок 1.2 – Зображення інтерфейсу Trello

— Asana – це додаток для управління проектами та колективною роботою, який має багато функцій. Він включає в себе календарі, дошки, списки і діаграми Ганта для ефективного планування і контролю за виконанням завдань [4].

Додаток Asana пропонує ряд корисних можливостей для керування проектами. Він дозволяє структурувати завдання, встановлювати послідовності процесів і розподіляти плани на щоденні задачі, а також розбивати великі завдання на менші частини (рис. 1.3). Кожен проєкт у Asana може бути візуалізований у формі списків, дошок або графіків, що сприяє кращій організації робочих процесів і завдань. Планування та призначення завдань включає встановлення пріоритетів, термінів виконання і призначення відповідальних за завдання, що дозволяє ефективно керувати проектами і робочими завданнями. Asana також надає можливість використовувати шаблони для різних видів завдань і автоматизувати рутинні процеси що сприяє ефективному використанню часу і уникненню помилок. Моніторинг та аналітика здійснюється за допомогою додаткових розширень, завдяки яким можна відстежувати витрати часу на виконання окремих етапів проєктів і аналізувати продуктивність команди [8, 9, 10].

Asana представляє собою потужний високоефективний інструмент для управління проєктами та завданнями, який надає змогу командам і окремим користувачам систематично організовувати свою роботу, стратегічно планувати кожен етап та досягати запланованих цілей. Він дозволяє ефективно керувати ресурсами, підтримувати зв'язок у команді та залучати співробітників до спільної роботи над проєктами будь-якої складності.

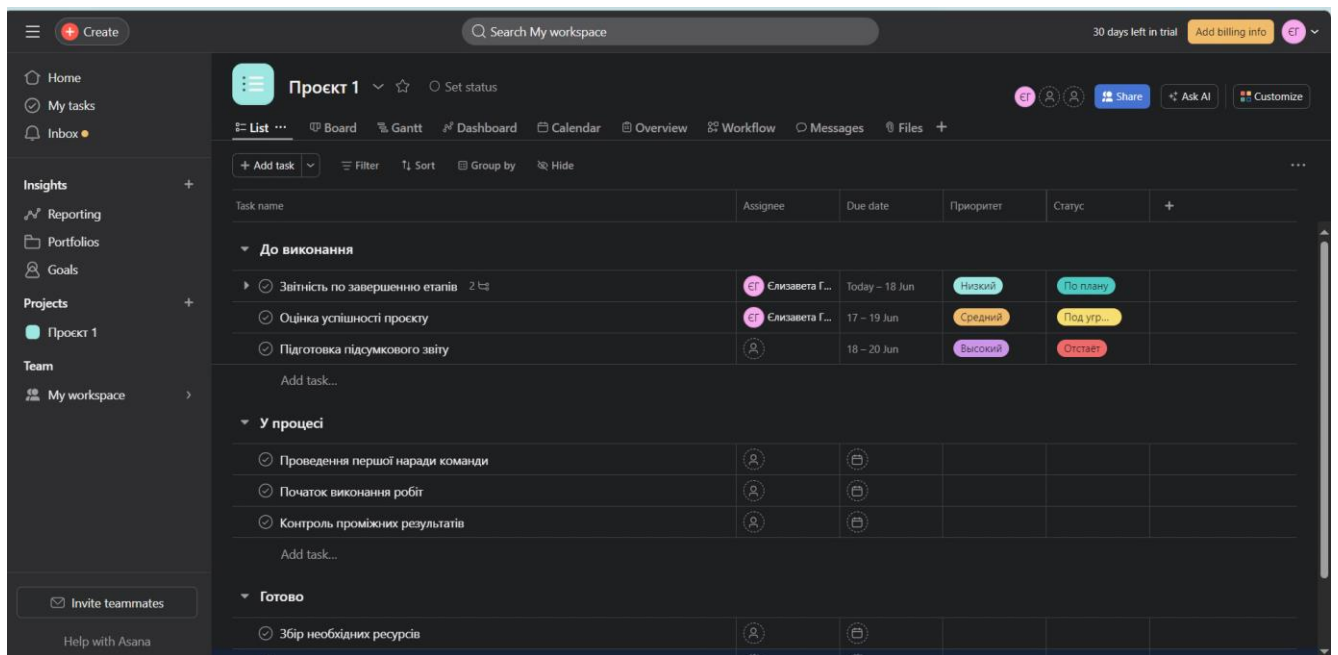


Рисунок 1.3 – Зображення інтерфейсу Asana

— Microsoft To-Do – це простий і легкий у використанні додаток для управління завданнями, який автоматично синхронізується з Microsoft 365. З його допомогою можна створювати списки завдань, встановлювати пріоритети і отримувати своєчасні нагадування (рис. 1.4) [4].

Microsoft To-Do включає в себе такі функції як особистий список справ – Microsoft To-Do надає можливість створювати особисті списки завдань та планів для кожного користувача. Також інтегрується з хмарними сервісами – програма базується на хмарних сервісах, що дозволяє синхронізувати дані між різними пристроями та зберігати їх у безпечному місці. Нагадування та терміни виконання – користувачі можуть встановлювати нагадування та терміни виконання для

кожного завдання, щоб ефективно планувати свій час та завдання. Спільні списки завдань – додаток також підтримує можливість спільної роботи над списками завдань, що дозволяє командам ефективно співпрацювати та досягати спільних цілей. Шаблони та автоматизація – програма надає можливість використовувати шаблони для різних видів завдань та автоматизувати деякі рутинні процеси, що сприяє підвищенню продуктивності користувачів [11, 12, 13].

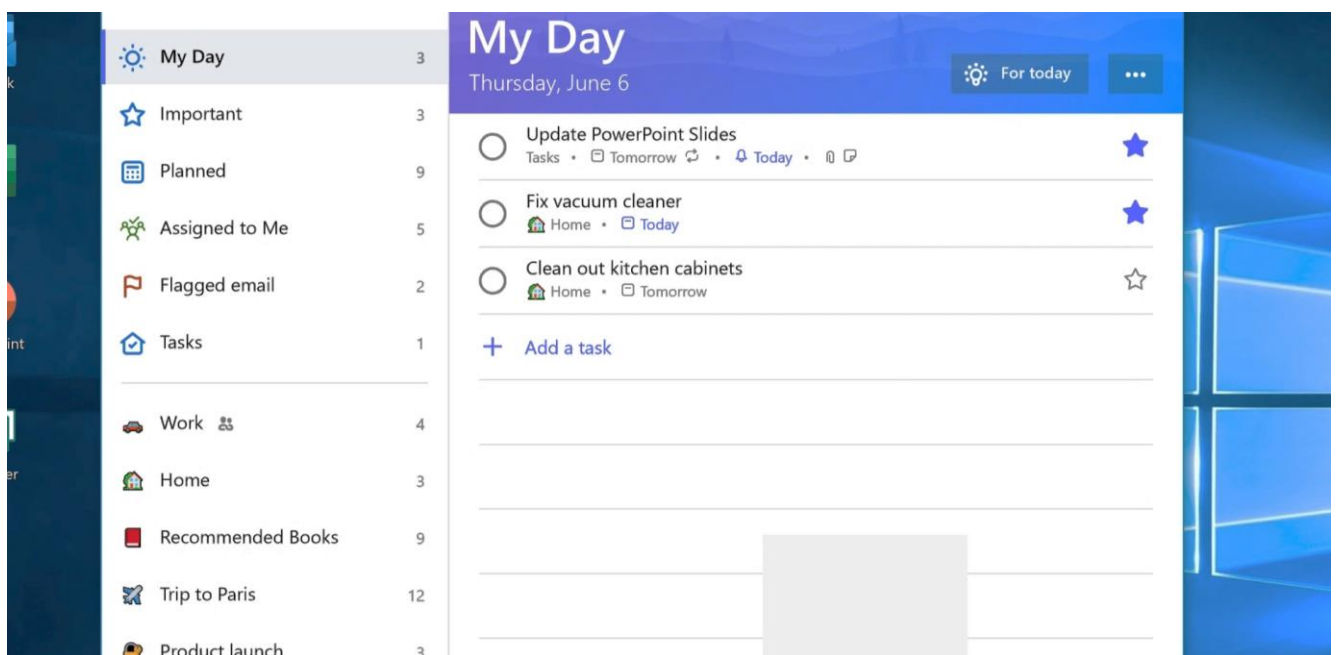


Рисунок 1.4 – Інтерфейс Microsoft To-Do

— Remember the Milk – це класичний інструмент для керування завданнями, який пропонує прості, але дуже корисні функції, включаючи можливість легко створювати нові завдання, встановлювати їм пріоритети для ефективнішої організації списку і отримувати нагадування, що допомагає вчасно згадувати про їх виконання (рис. 1.5) [4].

Remember the Milk пропонує широкий спектр функцій, спрямованих на полегшення організації завдань для користувачів. Додаток визволяє користувачів від необхідності пам'ятати про свої справи, автоматично запам'ятовуючи їх і надсилаючи нагадування вчасно. Нагадування в будь-якому місці – користувачі більше ніколи не забудуть про молоко виконати будь-яке інше завдання завдяки

системі нагадувань, яка оперативно і ефективно надсилає повідомлення. Виконання завдань разом – користувачі можуть спільно працювати над своїми списками завдань, а також делегувати їх іншим учасникам для швидшого і успішнішого виконання. Синхронізація на всіх пристроях – Remember The Milk забезпечує автоматичну синхронізацію даних на всіх пристроях користувачів, що дозволяє користувачам мати постійний доступ до своїх завдань та зручно ними керувати незалежно від місця і часу [14].

Цей додаток допомагає користувачам систематизувати свої завдання і успішно досягати їх вчасно та з високою ефективністю. Він забезпечує зручну можливість створення і організації завдань, а також їхнього планування з урахуванням пріоритетів.

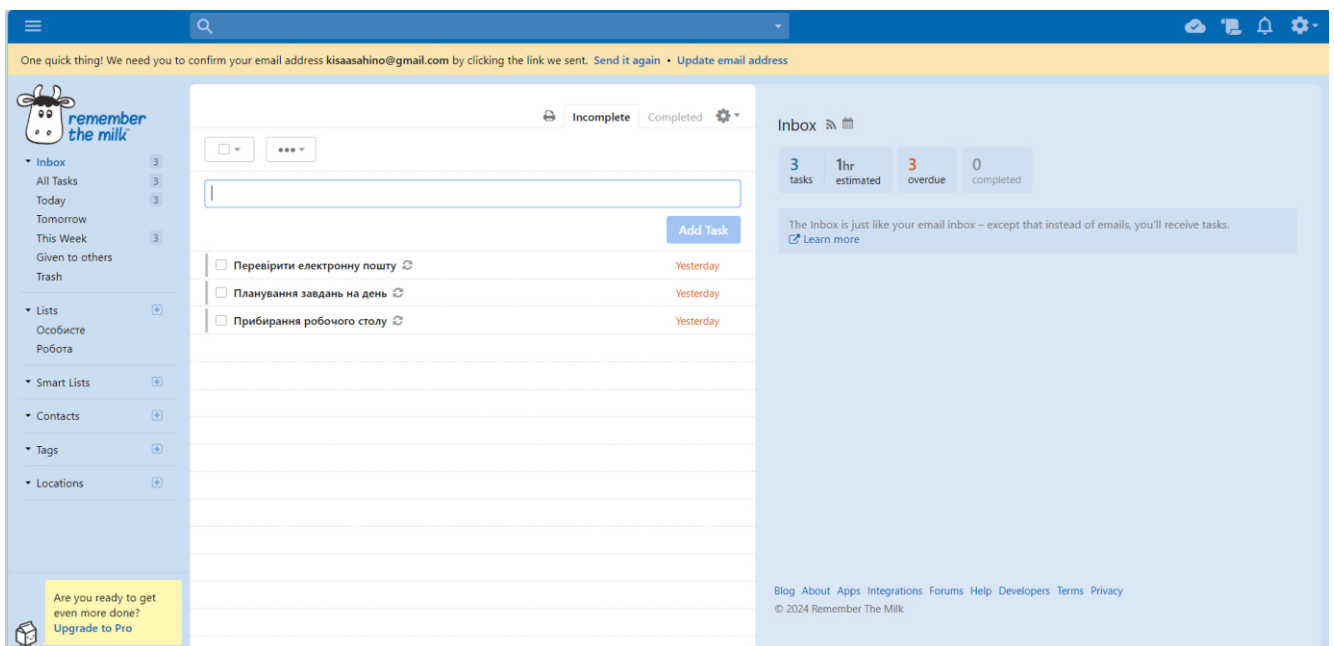


Рисунок 1.5 – Зображення інтерфейсу Remember the Milk

— Jira – це платформа для комунікації між працівниками, що спрямована на ефективне відстеження проєктів. У програмі користувачі можуть вказувати тематику проєкту, його категорію, встановлювати пріоритети, додавати складові частини та контент (рис. 1.6). Для більш детальнішого опису завдань доступні додаткові поля або можливість залишати коментарі [4].

Jira включає в себе такі функції як планування та управління проектами – Jira дозволяє створювати проекти, розбивати їх на завдання та підзавдання, встановлювати терміни виконання та призначати відповідальних. Відстеження помилок та проблем – Jira надає можливість реєструвати, відстежувати та вирішувати помилки, проблеми та запити користувачів. Agile-методології – Jira підтримує Agile-методології, такі як Scrum та Kanban, дозволяючи командам планувати спринти, вести беклог та відображати роботу на дошках Kanban. Звітність та аналітика – Jira пропонує широкий спектр звітів та аналітики, що дозволяє відстежувати прогрес проекту, швидкість команди та ефективність процесів. Інтеграції та розширення – Jira може бути інтегрована з іншими інструментами, такими як Confluence, Bitbucket, Slack та багатьма іншими, а також має великий вибір додатків для розширення функціоналу [15].

Jira дозволяє командам здійснювати ефективне планування, відстежувати та керувати проектною роботою, що сприяє підвищенню продуктивності та забезпечує прозорість в робочих процесах. Вона допомагає командам краще координуватися і досягати своїх цілей шляхом створення структурованих проектних управлінських систем.

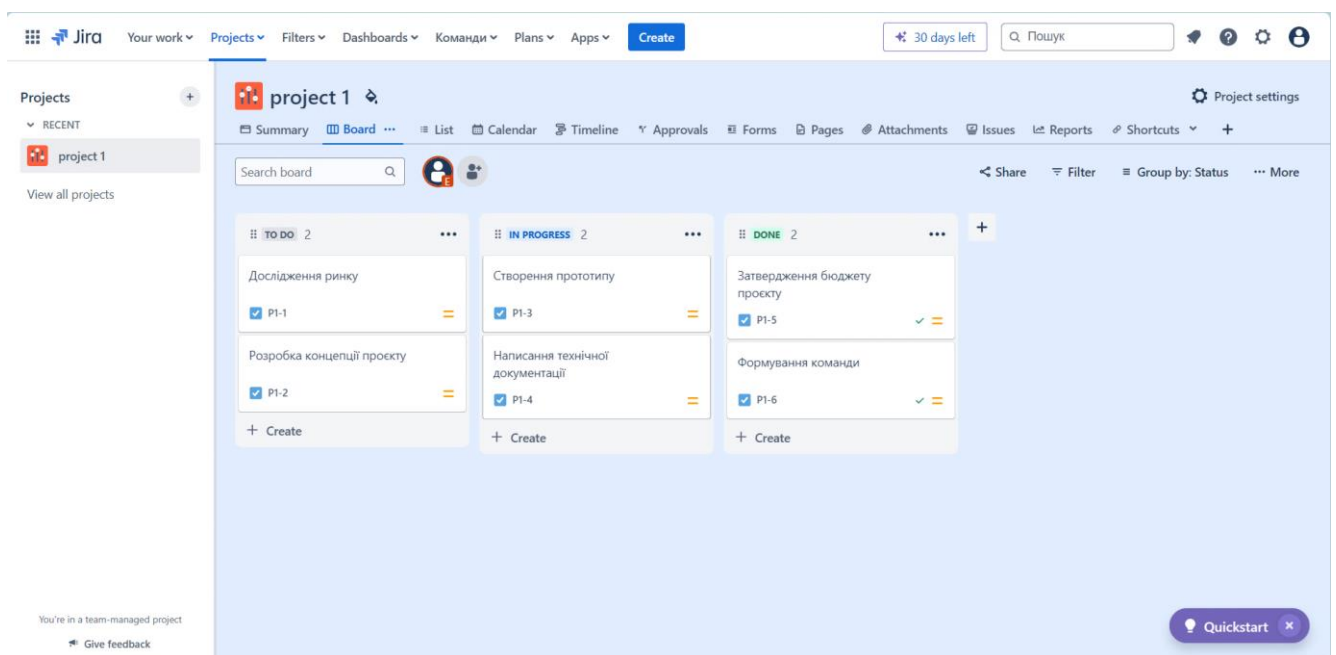


Рисунок 1.6 – Зображення інтерфейсу Jira

— Hour blocks: Day planner — це інструмент для планування дня, який допомагає користувачам сконцентруватися на виконанні конкретних завдань протягом певних годин. Цифровий календар дозволяє ефективно розподіляти час і зосереджуватися на пріоритетних задачах, забезпечуючи більшу продуктивність у повсякденних заняттях [4].

Hour blocks: Day planner включає в себе такі функції як розподіл дня на годинні блоки – додаток дозволяє розділити день на годинні інтервали для ефективного планування завдань (рис. 1.7). Простота використання – для додавання завдання потрібно лише натиснути на годинний блок та ввести опис. Додаток автоматично генерує іконку для завдання. Навчання на основі попередньої активності – з часом додаток починає пропонувати блоки на основі попередніх дій користувача, роблячи планування ще простішим. Інтеграція з календарем – додаток автоматично імпортує події з календаря iOS. Приємний дизайн – інтерфейс розроблений для максимального задоволення користувача, з дружнім тоном мови та спокійною кольоровою схемою [16].

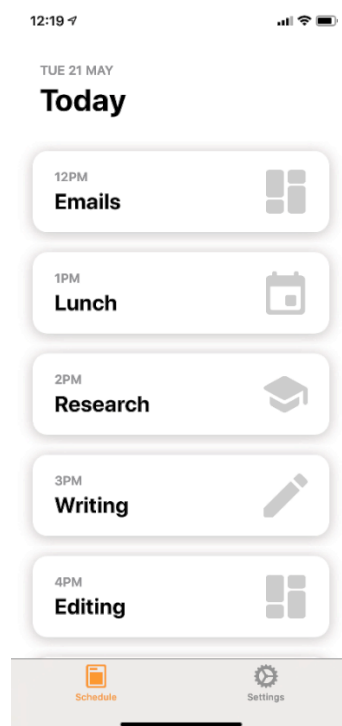


Рисунок 1.7 – Зображення головного екрану Hour blocks: Day planner

— Wrike – це потужна корпоративна платформа управління роботою, яка не лише допомагає компаніям досягати успіху, але й значно покращує співпрацю між співробітниками. Вона забезпечує комплексний підхід до управління проектами, де організації можуть ефективно планувати, відстежувати і керувати завданнями на різних етапах виконання. Гнучкість Wrike полягає в можливості адаптуватися до різних типів проектів і бізнес-потреб, що дозволяє забезпечувати оптимальне використання ресурсів і здійснювати необхідні корекції під час реалізації завдань. Видимість забезпечується завдяки зрозумілому інтерфейсу і звітності, яка дозволяє стежити за прогресом проектів і вчасно виявляти можливі проблеми. Контроль над процесами допомагає забезпечити високу ефективність робочих груп і досягнення стратегічних цілей компанії [17, 18].

Wrike пропонує різноманітність інструментів для ефективного управління роботою, включаючи різноманітність виглядів подачі інформації. Зокрема, він підтримує використання Gantt-діаграм для візуалізації термінів і залученості ресурсів, календарі для планування завдань за датами, таблиці для структурування даних та звіти для аналізу продуктивності (рис. 1.8). Що стосується автоматизації, Wrike має потужні та інтуїтивно зрозумілі засоби для автоматизації рутинних процесів, що дозволяє значно заощаджувати час. Він дозволяє перетворювати повторювані робочі процеси на автоматичні за кілька простих клацань, спрощуючи роботу і підвищуючи ефективність управління проектами. Мобільні додатки Wrike забезпечують повноцінний функціонал, що дозволяє користувачам мати доступ до всіх можливостей платформи навіть у режимі офлайн, працюючи з ними без підключення до Інтернету. Це дає можливість отримувати оновлення та взаємодіяти з проектами у реальному часі з будь-якого місця і в будь-який час, забезпечуючи безперебійну роботу навіть при відсутності інтернет-з'єднання. Ці додатки включають усі потужні інструменти, які дозволяють користувачам отримувати оновлення в реальному часі з будь-якого місця. Реальні часові панелі та аналітика – це інструменти, які допомагають командам досягати результатів. Вони дозволяють глибше розуміти плани, відстежувати прогрес, аналізувати дані



та отримувати повну картину швидко. Також доступна можливість створення звітів з використанням різних метрик, розрахунків та KPI, що дозволяє візуалізувати ключові результати оперативно. Для спрощення процесу перевірки та схвалення використовуються інструменти, які дозволяють порівнювати різні активи та отримувати оперативний зворотний зв'язок у реальному часі на документи або відео. Це значно полегшує комунікацію та співпрацю між учасниками процесу, а також створює базу для подальшої автоматизації схвалення, що дозволяє ефективно використовувати час і ресурси команди [19].

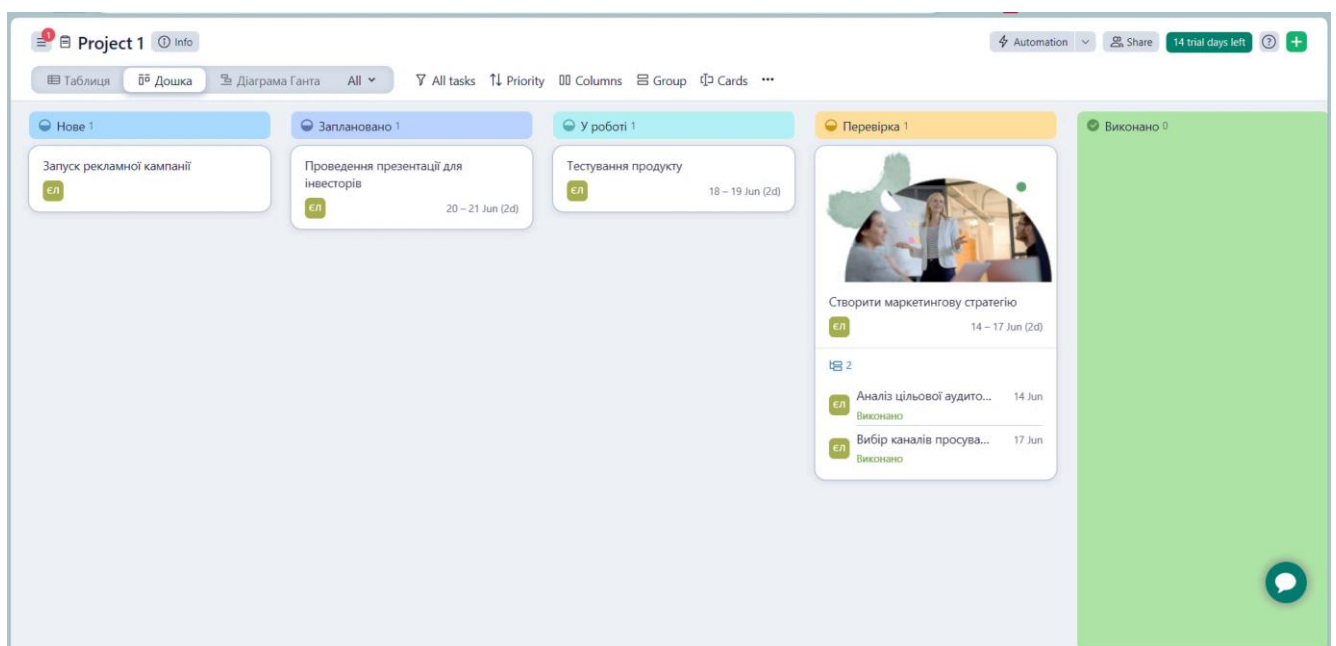


Рисунок 1.8 – Зображення головного екрану Wrike

— Notion – це багатофункціональна платформа для організації завдань і управління проектами, яка об'єднує в собі функції блокнота, бази даних, планувальника та багато іншого. Він підтримує колективну роботу та можливість коментування, дозволяючи командам разом редагувати документи та обмінюватися ідеями. Це сприяє покращенню комунікації та ефективній співпраці [18].

Notion включає в себе такі функції як створення нотаток – Notion надає можливість створювати різноманітні типи нотаток, від простого тексту до складних

структур з таблицями, зображеннями та формулами. Управління базами даних – платформа дозволяє створювати бази даних для зберігання та організації інформації, наприклад, списки завдань, контакти чи ідеї. Планування проєктів – Notion пропонує інструменти для планування проєктів, включаючи канбан-дошки, календарі та нагадувачі, що допомагає ефективно відстежувати прогрес. Співпраця та обмін знаннями – платформа спрощує командну роботу, забезпечуючи ефективну співпрацю над проєктами, обмін ідеями та відстеження прогресу. Адаптивність та гнучкість – Notion дозволяє користувачам адаптувати робочий простір під свої потреби, використовуючи готові шаблони або створюючи власні (рис. 1.9). Мобільність – Notion доступний як веб-версія, так і у вигляді мобільних додатків для iOS та Android, що забезпечує зручний доступ до робочого простору з будь-якого пристрою [20, 21, 22].

Завдяки своїй гнучкості та багатофункціональності, Notion дозволяє користувачам створювати індивідуальні робочі простори та організувати інформацію у зручний для них спосіб, що робить його популярним серед різних команд і окремих користувачів [18].

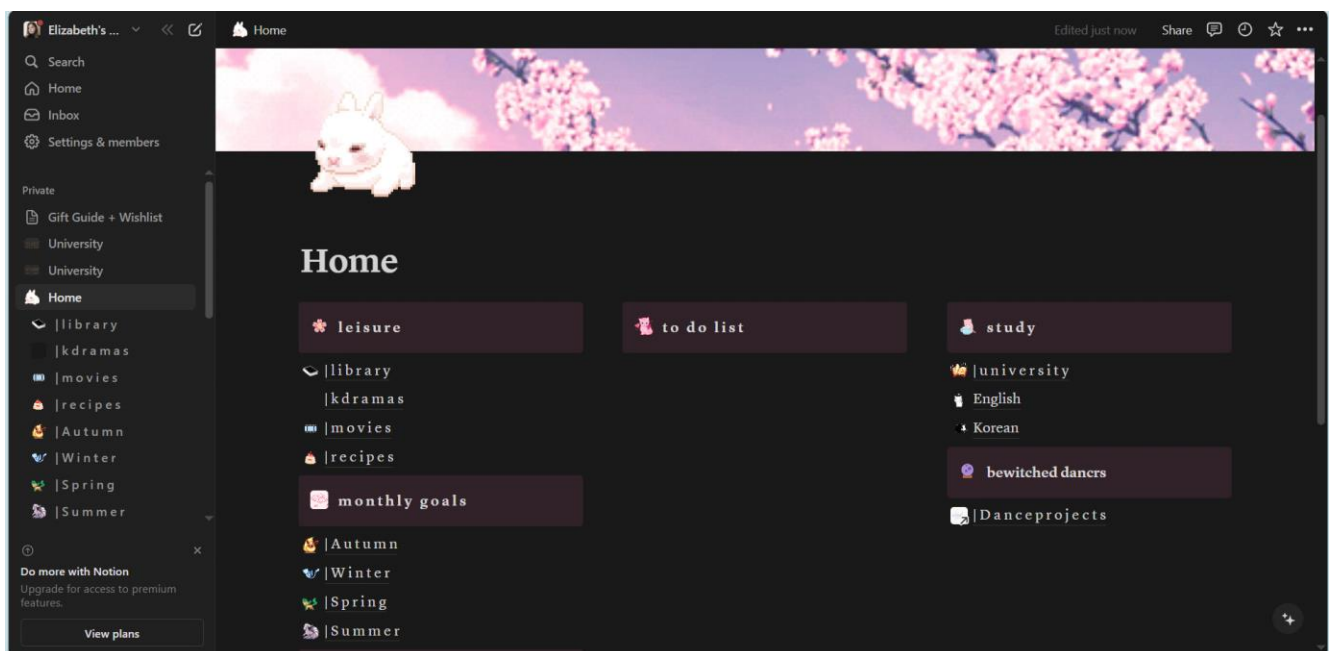


Рисунок 1.9 – Зображення головного екрану Notion

— ClickUp – це інтуїтивно зрозуміла та потужна американська платформа для управління проектами і завданнями. З її допомогою команди можуть ефективно планувати, відстежувати та координувати свою роботу, що підвищує продуктивність і покращує результати проектів. Сервіс пропонує безліч інструментів для оптимізації робочих процесів та поліпшення командної співпраці [18].

ClickUp надає користувачам широкий спектр можливостей для управління завданнями та проектами. Користувач може легко створювати завдання, встановлювати терміни виконання, призначати відповідальних і відстежувати прогрес проектів. Платформа пропонує більше 15 динамічних переглядів, включаючи списки, дошки, календарі та діаграми Ганта, що дозволяє командам налаштовувати робочі процеси відповідно до їхніх потреб і зручності. Це забезпечує гнучкість і можливість ефективного управління проектами на будь-якому етапі їх виконання. ClickUp полегшує командну взаємодію через розширені інструменти співпраці та комунікації, такі як вбудовані чати, коментарі та можливість згадувань. Платформа інтегрується з широким спектром популярних додатків, включаючи Google Drive, Zoom, Slack і інші, що дозволяє об'єднувати всі робочі інструменти в одному місці для зручності і ефективності робочих процесів. Такий підхід забезпечує плавну інтеграцію і сприяє синергії між різними аспектами проектної діяльності. ClickUp пропонує рішення для автоматизації, що сприяє ефективному використанню часу завдяки шаблонам, тригерам та діям. Платформа дозволяє відстежувати час, витрачений на завдання, та надає аналітичні звіти щодо продуктивності для оптимізації робочих процесів. ClickUp забезпечує надійність функцій безпеки, що є критичним аспектом для будь-якої організації, незалежно від її розміру. Інтеграція високих стандартів безпеки дозволяє користувачам впевнено працювати з конфіденційною інформацією та даними проектів, забезпечуючи захист від потенційних загроз і витоків даних. Можливість масштабування ClickUp робить його ідеальним вибором для компаній будь-якого розміру, незалежно від їх поточних потреб та майбутніх вимог. Платформа легко

адаптується до зростаючих вимог бізнесу і може ефективно підтримувати збільшення обсягів роботи, число користувачів та складність проектів [20, 23, 24].

ClickUp — це цінний інструмент, призначений для команд, що мають на меті підвищення ефективності співпраці, продуктивності та управління проектами. Він забезпечує широкий спектр функціональних можливостей, дозволяючи налаштовувати робочі процеси під конкретні потреби команди. ClickUp дозволяє організаціям зосередитися на своїх основних завданнях, не турбуючись про технічні аспекти управління проектами та безпеки даних. Завдяки своєму інтуїтивно зрозумілому інтерфейсу, ClickUp стає ідеальним рішенням для тих, хто шукає ефективний інструмент для управління завданнями та проектами [20].

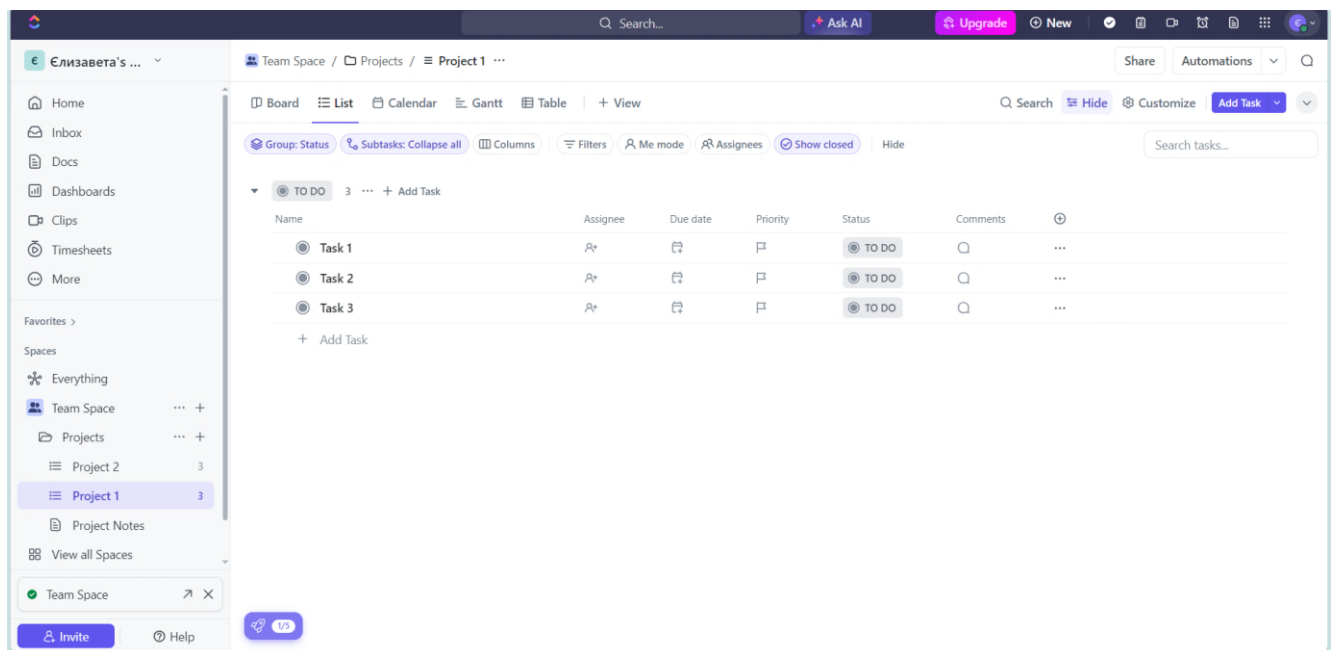


Рисунок 1.10 – Інтерфейс ClickUp

## 1.2 Визначення цілей та результатів проекту

Даний додаток має на меті не лише створити інструмент для ведення розкладу, але й надати користувачам стимули та мотивацію для покращення управління своїм часом. Завдяки використанню наочного прогресу, менеджер задач може перетворити рутинне планування на захопливий процес, який не лише

полегшить управління часом, але й надихне людей на досягнення поставлених цілей.

Метою даного проєкту є створення мобільного додатку для ефективного керування справами, який відповідає основним вимогам ефективного управління часом і надає широкий спектр можливостей для задоволення потреб різних користувачів. Фокусом розробки є не просто надання інструменту для управління розкладом, а створення комплексного додатку, який допомагає вирішувати найрізноманітніші завдання і проблеми.

Виходячи з цього, можна виділити наступні основні цілі:

- Мотивація користувача до планування;
- Підвищення продуктивності користувача;
- Створення оптимального робочого середовища;
- Розробка системи розстановки пріоритетів

Основні функції, що забезпечать виконання ключових цілей проєкту:

Система розстановки пріоритетів у менеджері задач діє як ефективний орієнтир для користувачів, допомагаючи їм зосереджуватися на найбільш важливих і термінових завданнях. Завдяки такій системі, користувачі можуть чітко бачити, які завдання потребують негайної уваги, а які можуть почекати, що сприяє ефективному плануванню та розподілу часу. Це, в свою чергу, сприяє кращому розумінню і впровадженню стратегії управління часом, що є ключовим фактором для досягнення більш важливих результатів у професійному та особистому житті. Розстановка пріоритетів допомагає уникнути перенавантаження, дозволяючи користувачам працювати продуктивніше та зосереджуватися на досягненні своїх цілей з максимальною ефективністю.

Відстеження власного прогресу також є надзвичайно важливою функцією, яка сприяє збереженню мотивації та ентузіазму. Якщо людина не бачить чітких ознак свого руху до мети, вона може легко втратити віру в свої сили і зневіритися. Це часто призводить до того, що людина залишає справу, в якій могла б досягти значного успіху. Завдяки регулярному відстеженню прогресу можна відчути реальний рух, що надає додаткового стимулу і впевненості у собі і своїх

можливостях, підтримуючи постійне бажання продовжувати працювати заради мети і досягати поставлених цілей.

Розбиття великих задач на більш малі підзавдання також є дуже важливою функцією у додатку для ефективного керування часом. Ця можливість сприяє більш глибокому розумінню завдання, допомагаючи користувачам розділяти комплексні проекти на більш керовані та досяжні етапи. Завдяки цьому, кожен крок стає зрозумілішим і легше реалізовуваним. Виконуючи завдання покроково, користувач отримує змогу чітко усвідомити, що саме потрібно зробити в кожний конкретний момент. Це дозволяє фокусувати увагу на конкретній частині завдання, що значно підвищує продуктивність та ефективність роботи. Крім того, успішне завершення маленьких підзадач дає відчуття досягнення, яке мотивує продовжувати рухатися вперед, наближаючись до загальної мети.

Необхідними цілями для даного проекту будуть наступні:

— Персоналізація та кастомізація:

Перша ціль фокусується на важливості індивідуалізації додатку для планування для кожного користувача. Можливість кастомізації і персоналізації дозволяє створити інструмент, який відповідає унікальним потребам та вподобанням кожного користувача, забезпечуючи високий рівень зручності використання.

— Створення звички планування:

Друга ціль спрямована на створення не тільки інструменту, але і партнера, який допоможе користувачам розвинути звичку планування. Інтерфейс та функції повинні заохочувати до регулярного планування і допомагати сформувати цю корисну звичку.

— Покращення продуктивності:

Третя ціль полягає в тому, щоб менеджер задач не лише був інструментом для фіксації завдань, а й активною підтримкою для досягнення більших результатів та підвищення ефективності використання часу.

— Збереження позитивного досвіду використання:

Остання ціль визначає, що користувачі повинні відчувати задоволення та позитивні емоції від використання менеджера задач. Це означає створення інтерфейсу, який буде зручним у використанні та сприятиме позитивному досвіду від взаємодії з продуктом.

Ці цілі взаємодіють між собою, утворюючи єдиний напрямок для реалізації проєкту. Їхнє досягнення сприятиме створенню інструменту, який відповідає вимогам та очікуванням різних користувачів.

Бажані цілі проєкту:

— Широкий функціонал та гнучкість:

Першою бажаною ціллю є забезпечення широкого спектру функціоналу, що охоплює різні аспекти управління часом. Гнучкість та налаштовуваність є ключовими елементами, які дозволяють пристосувати менеджер задач до унікальних потреб та стилю життя кожного користувача.

— Ефективність та вплив на продуктивність:

Другою бажаною ціллю є забезпечення відчуття реального впливу на продуктивність користувачів. Кожен інструмент і функція повинні сприяти досягненню цілей користувачів та максимальній ефективності використання часу.

— Позитивний вплив на психологічний комфорт:

Важливо не лише надавати користувачам інструменти для управління часом, але й позитивно впливати на їхній психологічний стан. Зручний дизайн повинен створювати приємний та позитивний досвід використання додатку, аби люди не відчували дискомфорту під час планування та загалом мали позитивний досвід в управлінні своїм часом, а також поверталися у додаток для керування справами.

— Постійна мотивація та стимуляція:

Четвертою бажаною ціллю є забезпечення системи постійної мотивації та стимулювання, яка підтримує користувачів у регулярному плануванні та допомагає виробити цю корисну звичку.

— Висока залученість та задоволеність користувачів:

Остаточна мета – створити не тільки багатофункціональний додаток для керування справами, але і менеджер задач, що утримує високий рівень

задоволеності та залученості користувачів. Активна та лояльна спільнота користувачів є ключовим показником успіху проєкту.

Додаткові вигоди у даному проєкті можуть включати:

— Управління змінами:

Управління змінами в додатках для управління задачами дозволяє адаптувати плани та процеси до змін у середовищі та потребах клієнтів. Це означає, що користувачі можуть легко модифікувати свої завдання, враховуючи нові вимоги або умови, що змінюються з часом. Такі додатки надають інструменти для швидкої зміни пріоритетів, перегляду та редагування завдань, а також для оновлення інформації про ресурси та виконання завдань у відповідь на зміни. Це сприяє підвищенню адаптивності та ефективності управління проєктами, що є критичним для досягнення успіху в сучасних умовах швидко змінного середовища [25].

— Управління ресурсами:

Управління ресурсами у менеджері задач дозволяє користувачам планувати та організовувати ефективне використання ресурсів для проєктів. Вони можуть розподіляти ресурси відповідно до потреб проєктів, забезпечуючи оптимальне використання робочих сил, матеріалів та інших активів. Такий підхід сприяє зниженню витрат, підвищенню продуктивності та забезпеченню успішного виконання завдань в установлені строки [26].

— Управління продуктивністю:

Управління продуктивністю через додатки для керування справами створює умови для активного відстеження та аналізу продуктивності. Це дозволяє користувачам ефективніше управляти своїми завданнями, знаходити оптимальні рішення та досягати успішних результатів у своїх проєктах. Такі інструменти стимулюють постійне покращення продуктивності та сприяють досягненню поставлених цілей [23, 24].

— Оптимізація використання часу:

Додатки для керування справами створюють умови для планування та організації часу, що дозволяє користувачам ефективно розпоряджатися своїм часом



та досягати успіху у своїх проєктах. Вони сприяють налагодженню систематичного контролю над часом і сприяють досягненню високих результатів у роботі[25, 26].

— Аналіз продуктивності:

Функції аналізу продуктивності в додатках для управління задачами не лише відображають історію виконання завдань, але й надають корисні інсайти щодо оптимізації планування та використання часу. Вони аналізують, як ефективно користувач виконує завдання, підказують, як можна покращити продуктивність та розподіл часу [25, 26].

Ці додаткові вигоди роблять проєкт більш комплексним та привабливим для різних типів користувачів, що сприяє не тільки управлінню часом, але й різнобічному розвитку та покращенню життя.

## Висновки до розділу

У першому розділі кваліфікаційної роботи було проведено детальний аналіз існуючих додатків для керування справами. Досліджено функції і можливості популярних програмних продуктів, зокрема: Todoist, Trello, Asana, Microsoft To-Do, Remember the Milk, Jira, Hour blocks: Day planner, Wrike, Notion та ClickUp. На основі проведеного аналізу було виявлено, що головним недоліком цих додатків є потреба в занадто довгому налаштуванні робочого простору перед початком планування задач.

Цей аналіз дозволив ідентифікувати основні, необхідні та додаткові цілі, які були визначені як ключові для створення мобільного додатку для ефективного управління справами. Також було визначено головні функції, які є необхідними та обов'язковими при реалізації додатку типу менеджера завдань.

Також було визначено яких результатів необхідно досягти для того, аби вважати мобільний додаток для ефективного управління справами успішно реалізованим у роботі. На основі цього розроблено вимоги для реалізації додатку для керування справами

## РОЗДІЛ 2

### РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 2.1 Формулювання завдання для реалізації програмного рішення

Проект було розроблено з використанням мови програмування React Native, що дозволяє забезпечити доступ до додатка на різних мобільних пристроях.

Користувач може створити свій особистий обліковий запис, до якого буде мати доступ лише він. Це досягається шляхом введення імені користувача, електронної пошти та пароля.

Після простої реєстрації у застосунку, користувач переходить на головний екран, де будуть відображені додані ним завдання (рис. 2.1). До додавання завдань, користувач може бачити напис, що завдань немає і заклик до додавання нового завдання. У верхній частині екрану розташована навігація по категоріям завдань: всі, робота, персональні. Також справа від цих вкладок можна побачити кнопку додавання задачі до списку. Після натискання на цю кнопку з'явиться спливаюче вікно, де можна ввести назву задачі, або обрати її із шаблонів і обрати категорію у якій ця задача буде обрана. Якщо користувач знаходиться на вкладці «всі», відображаються вся завдання, якщо переходить до однієї з категорій «робота», або «персональне», відображаються завдання, що були внесені до категорії відповідно.

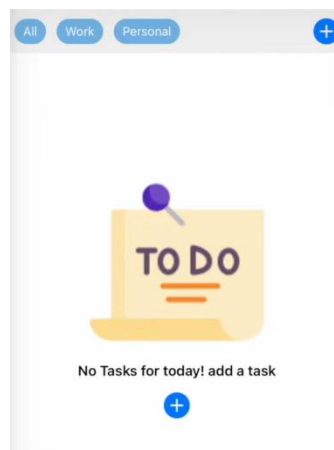


Рисунок 2.1 – Головний екран до додавання на нього завдань користувача

Задачі:

Кожна задача може ділитися на більш дрібні підзавдання, в налаштуваннях завдання, які викликаються простим натисканням на задачу, можна додати декілька підзавдань за бажанням користувача.

Для задачі можна встановити пріоритет. Це робиться за допомогою натискання на прапорець, що розташований у правому куті задачі (рис. 2.2), і тоді вона буде відображатися вгорі списку.

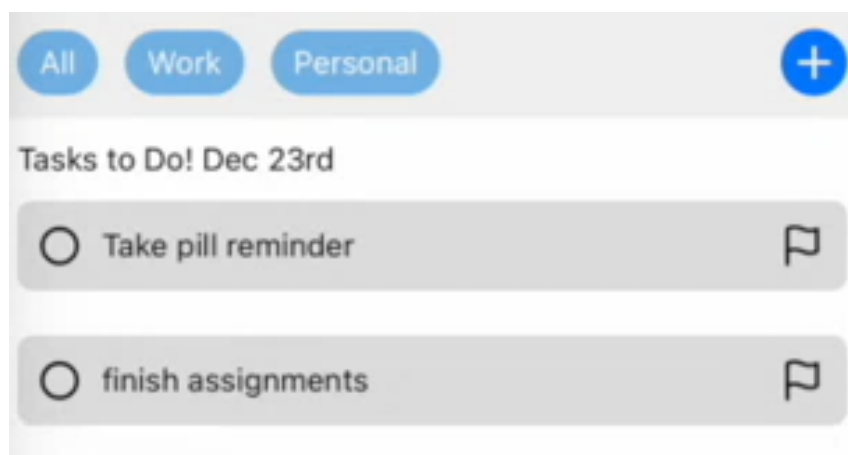


Рисунок 2.2 – Зображення інтерфейсу задач із можливістю встановлення пріоритету прапорцем

Також існує можливість встановлення кінцевого терміну виконання завдання. Це також можна встановити за допомогою налаштувань у вже доданій задачі. Це дозволяє користувачам точно планувати та контролювати часові рамки виконання завдань, що важливо для ефективного управління своїми справами і вчасного виконання планів.

У календарі користувач може переглянути історію виконаних завдань за кожен конкретний день, а також переглянути завдання, заплановані на поточний день. Це дозволяє користувачам зручно відстежувати свою продуктивність, оцінювати виконану роботу і планувати активності на майбутні дні.

У профілі користувача відображено його аватар, кількість завдань, які він вже виконав сьогодні, та кількість завдань, які залишилися до виконання. Крім того,

відображена статистика, яка враховує виконані завдання і ті, які ще потрібно виконати, що дозволяє користувачам оцінювати свою продуктивність та планувати подальші дії. Поряд із аватаром відображається інформація про те, скільки днів користувач вже використовує додаток для планування своїх справ. Це дозволяє користувачам відстежувати тривалість їхнього досвіду використання додатка та оцінювати свою активність у керуванні власними справами.

Місія даного проєкту – перетворити планування та управління часом у захопливий та надихаючий процес, що сприяє особистому та професійному розвитку користувачів. Проєкт ставить перед собою завдання створити корисний та ефективний менеджер для керування власними справами, який надає дієві інструменти для досягнення цілей, підтримує позитивні звички, сприяє психологічному комфорту та підвищує якість життя користувачів. Мета проєкту полягає в тому, щоб не лише змінити підхід до планування часу, але й надати інструмент для гармонійного балансу між досягненням цілей та збереженням якості життя.

## 2.2 Опис програмної реалізації системи

Середовищем розробки було обрано Visual Studio Code, розроблений Microsoft, що є потужним текстовим редактором, доступним для операційних систем Windows, Linux і macOS. Цей редактор коду позиціонується як «легкий», проте він містить всі необхідні функції для кросплатформної розробки хмарних та веб-додатків. Visual Studio Code пропонує багатий набір інструментів, таких як вбудований налагоджувач, інтеграція з Git для версіонування, підсвічування синтаксису, IntelliSense для інтелектуального автозавершення коду, та інструменти для рефакторингу, що допомагають покращити якість коду [27].

Окрім основних функцій, Visual Studio Code виділяється своїми великими можливостями для налаштування та персоналізації. Користувачі можуть створювати власні теми оформлення, налаштовувати комбінації клавіш під свої потреби, та використовувати конфігураційні файли для адаптації редактора під

специфічні проекти. Цей редактор коду є безкоштовним і розробляється як програмне забезпечення з відкритим вихідним кодом, що дозволяє спільноті розробників вносити свої зміни, доповнення та покращення. Проте готові збірки програми поширюються під пропрієтарною ліцензією, забезпечуючи певний рівень контролю, стабільності та офіційної підтримки з боку Microsoft. Завдяки цьому користувачі отримують не лише гнучкий та функціональний інструмент для розробки, але й впевненість у його надійності та постійних оновленнях. [27].

Для того, щоб розпочати роботу над кодом програми, перш за все необхідно належним чином підготувати середовище розробки для роботи із React Native. Цей процес включає кілька важливих етапів, які забезпечать оптимальні умови для написання, тестування та налагодження коду.

Спочатку необхідно встановити Node.js, який можна завантажити з офіційного сайту. Він включає в себе необхідний для встановлення пакетів React Native менеджер пакетів – npm.

Далі встановлюємо Expo CLI. Це набір команд, які забезпечують інтерфейс для взаємодії розробника з різними інструментами Expo. Цей інструментарій дозволяє виконувати широкий спектр завдань, таких як запуск локального сервера для розробки, генерація статичних файлів, а також запуск додатків на різних платформах. За допомогою команди `npm install -g expo-cli` встановлюємо безпосередньо сам Expo CLI [28, 29].

Наступним кроком буде створення нового проекту за допомогою команди `nx create-expo-app`. Ця команда використовується для автоматичного створення та налаштування нового проекту Expo, базованого на React Native. Цей інструментарій значно спрощує процес створення нових проектів, дозволяючи розпочати роботу з мінімальними зусиллями. Команда `nx create-expo-app` виконує низку важливих дій, спрямованих на полегшення процесу створення нового проекту. Вона автоматично встановлює необхідні залежності, пропонує вибір шаблонів для різних типів проектів, налаштовує параметри для оптимальної роботи, створює нові файли та директорії, налаштовує мета-інформацію для проекту, а також встановлює всі необхідні інструменти та залежності. Завдяки цій

команді, розробник може зосередитися на написанні коду, а не на рутинних налаштуваннях, що дозволяє швидко розпочати роботу над проєктом [30].

Наступним кроком у розробці даного проєкту є інтеграція Expo Router, що забезпечить зручну та ефективну маршрутизацію. Для цього додаємо необхідні залежності з допомогою команди `npx expo install expo-router react-native-safe-area-context react-native-screens expo-linking expo-constants expo-status-bar` яка вводиться у термінал Visual Studio Code [31].

Після встановлення залежностей, необхідно внести зміни у файл конфігурації проєкту. Файл `package.json` є важливим для налаштування параметрів додатку. У цьому файлі потрібно додати або змінити поле `main`, щоб воно вказувало на `expo-router/entry`. Це важливий крок, оскільки він вказує Expo використовувати роутер для маршрутизації у додатку. Тому у `package.json` додаємо наступну частину коду: «`main`»: «`expo-router/entry`» [31].

Ще одним важливим етапом у налаштуванні проєкту є додавання функціональності діплінкінгу. Це дозволяє відкривати додаток за допомогою спеціальних URL-посилань, що дуже зручно для інтеграції з іншими додатками та веб-сайтами, а також для покращення користувацького досвіду. Діплінкінг полегшує перехід до конкретних розділів або екранів додатку, що може бути особливо корисним для великих проєктів з численними функціями. Для налаштування діплінкінгу потрібно внести зміни у файл конфігурації додатку, а саме у «`app.json`». У цьому файлі потрібно додати поле «`scheme`»: «`your-app-scheme`», яке визначає схему URL для додатку, де «`your-app-scheme`» у даному випадку замінюється на «`todo-list`» [31].

Щоб налаштувати проєкт для використання Expo Router, важливим кроком є додавання відповідного плагіну у конфігураційний файл `Babel`. `Babel` – це надзвичайно корисна утиліта, яка перетворює сучасний JavaScript-код у формат, сумісний з різними браузерами та середовищами виконання. Це означає, що розробники можуть безпечно використовувати новітні функції та можливості JavaScript, не турбуючись про те, чи буде цей код коректно працювати у всіх браузерах. `Babel` забезпечує легку інтеграцію та сумісність, дозволяючи

зосередитися на створенні інноваційних рішень, не витрачаючи час на вирішення проблем сумісності. У конфігураційний файл Babel, який зазвичай називається «babel.config.js» додається плагін «plugins: ['expo-router/babel']», щоб забезпечити коректну роботу маршрутизації у додатку [31, 32].

Тепер наступним кроком є створення маршрутизатора у дереві файлів проєкту. Для цього на глобальному рівні створюється нова папка із назвою «app». У цій папці необхідно створити файл під назвою «index.js» (рис. 2.3). Цей файл по суті стане кореневим шляхом до каталогу додатку, забезпечуючи організовану структуру для навігації та маршрутизації у проєкті.



Рисунок 2.3 – Структура маршрутизатору проєкту

Створюється новий термінал, що надає можливість для взаємодії з операційною системою через командний рядок. Потім створюється нова папка із назвою «ari». Коли ця папка створена можна перейти в неї викликавши у терміналі команду «cd ari». Це змінює поточну робочу директорію терміналу на «ari», що означає, що будь-які подальші команди та операції будуть виконуватися у цій конкретній папці. Далі, вводиться команда «npm init», щоб ініціалізувати створення файлу package.json. Цей файл використовується для опису проєкту Node.js, включаючи ім'я проєкту, версію, авторів, залежності та інші важливі метадані. Ініціалізація npm init дозволяє створити цей файл та налаштувати його відповідно до проєкту. У налаштуваннях у даному випадку змінюється тільки description на backend. У результаті у папці «ari» було створено файл package.json (рис. 2.4)

```
api > {} package.json > license
1  {
2    "name": "api",
3    "version": "1.0.0",
4    "description": "backend",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC"
11 }
12
```

Рисунок 2.4 – Програмний код створеного package.json

Також необхідно встановити деякі залежності у папку api. Для цього у термінал вводиться команда `yarn add body-parser cors express mongoose nodemon jsonwebtoken`. `Body-parser` – це програмне забезпечення середнього рівня, яке використовується для обробки даних у форматах JSON, рядків та URL-кодівання в запитах Node.js. Він забезпечує можливість отримання доступу до вмісту запиту в зазначених форматах для подальшої обробки і використання. `Cors` – це модуль, який надає можливість серверу Node.js керувати запитом CORS (Cross-Origin Resource Sharing). Цей механізм безпеки дозволяє контролювати взаємодію веб-сайтів з різних доменів, щоб забезпечити безпеку та обмежити доступ до ресурсів. `Express` – це високопопулярний фреймворк для створення веб-додатків на платформі Node.js. Він забезпечує простий і ефективний механізм для створення API, веб-серверів, обробки маршрутів і керування запитом та відповідями. `Mongoose` – це бібліотека для Node.js і MongoDB, яка надає об'єктно-документний (ODM) інтерфейс для взаємодії з базою даних MongoDB. Вона дозволяє визначати структури документів (схеми), які використовуються для зберігання даних в MongoDB. За допомогою Mongoose можна легко виконувати операції CRUD (створення, читання, оновлення, видалення) з об'єктами даних, що забезпечує



зручну і ефективну роботу з базою даних. це інструмент, який допомагає в автоматичному перезавантаженні сервера Node.js при зміні файлів у вашому проєкті. Він особливо зручний під час розробки, оскільки не потрібно вручну перезапускати сервер після внесення змін у код. Jsonwebtoken – це бібліотека для генерації і верифікації JSON Web Tokens (JWT) в середовищі Node.js. JWT є стандартом, призначеним для створення безпечних та компактних токенів, які використовуються для автентифікації та обміну даними між різними сторонами, такими як сервер і клієнт, або між двома серверами [33, 34].

Після виконання даної команди, яка встановлює необхідні залежності, вміст файлу package.json був автоматично оновлений(рис. 2.5). Це означає, що до цього файлу було додано інформацію про шість нових залежностей разом з відповідними їх версіями.

```
api > {} package.json > ...
1  {
2    "name": "api",
3    "version": "1.0.0",
4    "description": "backend",
5    "main": "index.js",
6    > Debug
7    "scripts": {
8      "test": "echo \"Error: no test specified\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "body-parser": "^1.20.2",
14     "cors": "^2.8.5",
15     "express": "^4.18.2",
16     "jsonwebtoken": "^9.0.2",
17     "mongoose": "^8.0.3",
18     "nodemon": "^3.0.2"
19   }
20 }
```

Рисунок 2.5 – Оновлений програмний код package.json

Тепер у скрипти цього коду необхідно додати скрипт start, який буде відповідати за запуск основного додатку. Це дуже важливо для зручності розробки та тестування, оскільки скрипт start забезпечує можливість швидкого і легкого запуску додатку. Це значно спрощує процес розробки, оскільки дає можливість запускати додаток однією командою замість ручного виконання кожного кроку. Тепер код package.json має наступний вигляд, зображений на рисунку 2.6, де також відображені залежності з відповідними версіями.

```
api > {} package.json > {} scripts > start
1  {
2    "name": "api",
3    "version": "1.0.0",
4    "description": "backend",
5    "main": "index.js",
6    "scripts": {
7      "start": "nodemon index.js",
8      "test": "echo \"Error: no test specified\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "body-parser": "^1.20.2",
14     "cors": "^2.8.5",
15     "express": "^4.18.2",
16     "jsonwebtoken": "^9.0.2",
17     "mongoose": "^8.0.3",
18     "nodemon": "^3.0.2"
19   }
20 }
21
```

Рисунок 2.6 – Програмний код зміненого package.json

Надалі для роботи над цим проєктом необхідно підключити базу даних. Зважаючи на потреби проєкту, було обрано сервіс MongoDB. Це високоефективна система управління даними типу NoSQL, що спеціалізується на зберіганні та управлінні великими обсягами даних у документах. Розроблена компанією MongoDB Inc., вона славиться своєю високою швидкістю, здатністю легко масштабуватися і гнучкістю у структурі даних, що дозволяє зручно працювати з

різноманітними типами інформації. Вона також відрізняється високою здатністю обробляти значні обсяги інформації. Продуктивність MongoDB дозволяє ефективно виконувати операції з даними. Інтеграція та сумісність з іншими системами, підтримка геопросторових даних і можливість реплікації даних роблять MongoDB відмінним інструментом для сучасних додатків та сервісів, де потрібна ефективна робота з даними [35].

Для користування цим сервісом необхідно пройти просту реєстрацію. Після чого необхідно створити новий проєкт, який у даному випадку називається «todo-list». У даному проєкті використовується безкоштовний тарифний план сервісу MongoDB, адже немає необхідності у збереженні великої кількості даних. Далі налаштовується ім'я користувача і пароль. У цьому випадку для імені користувача було використано ім'я Yelyzaveta. Надалі вказується ір адреса для якої буде дозволено доступ до бази даних.

Далі у тій самій папці, де створено файл package.json, необхідно створити файл з назвою index.js, де розміститься основний код програми. Цей файл index.js є вхідною точкою додатку, де буде описано логіку роботи, обробку запитів та відповідей сервера.

Після оголошення змінних та встановлення необхідних налаштувань, додається звертання на створену за допомогою сервісу базу даних. Для цього на сервісі необхідно обрати бажаний спосіб з'єднання, у цьому випадку це «Connecting with MongoDB for VS Code» та скопіювати рядок, наданий сервісом (рис. 2.7).

### 3. Connect to your MongoDB deployment.

Paste your connection string into the Command Palette.

```
mongodb+srv://Yelyzaveta:<password>@cluster0.qxkblz2.mongodb.net/
```

Replace **<password>** with the password for the **Yelyzaveta** user. Ensure any options are URL encoded. [↗](#)

Рисунок 2.7 – Вікно підключення у MongoDB

Замість використання <password> вставляється реальний пароль, який було встановлено при створенні проєкту у MongoDB. Після успішного підключення у файлі index.js додаються обробки обіцянок для управління подальшими діями з базою даних та код для запуску сервера. Програмна реалізація встановлення з'єднання зі створеною базою даних зображена на рисунку 2.8. Частина з then() виконується, якщо підключення до MongoDB успішне і друкує повідомлення про те, що підключення до MongoDB відбулося у консоль. Частина з catch() виконується, якщо під час спроби підключення виникає помилка і друкує повідомлення про це.

```
mongoose
  .connect("mongodb+srv://Yelyzaveta:Yelyzaveta@cluster0.zjqdesc.mongodb.net/")
  .then(() => {
    console.log("Connected to MongoDB");
  })
  .catch((error) => {
    console.log("Error connectin to mongoDb", error);
  });

app.listen(port, () => {
  console.log("Server is running on port 3000");
});
```

Рисунок 2.8 – Частина коду, що встановлює зв'язок з кластером MongoDB

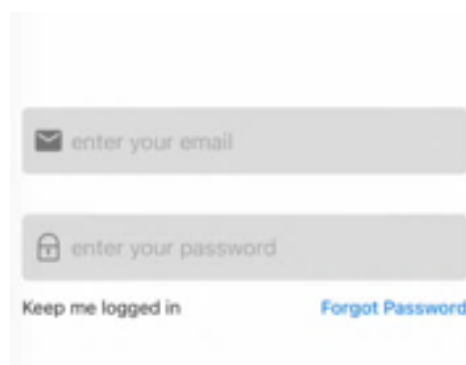
На цьому етапі в папці app створюється нова папка з назвою tabs, де будуть розміщені три основні екрани: домашня сторінка, профіль та календар. Кожен з цих екранних компонентів буде відповідати за окрему функціональність додатку. Додатково, в тій же папці app створюється ще одна папка, в якій буде зберігатися весь код, пов'язаний з екраном автентифікації та реєстрації користувачів. Ця структура допоможе організувати код таким чином, щоб він був легко зрозумілим та підтримуваним. У файлі index.js першим екраном вказується екран автентифікації, що забезпечує початковий вхід користувачів до системи. Це означає, що нові користувачі спочатку будуть перенаправлені на сторінку для введення своїх облікових даних перед тим, як отримають доступ до інших функцій

додатку. Таким чином, забезпечується безпека та індивідуальний доступ до облікових записів кожного користувача (рис. 2.9).

```
todo-list > app > JS index.js > ...
1  import { StyleSheet, Text, View } from "react-native";
2  import React from "react";
3  import { Redirect } from "expo-router";
4
5  const index = () => {
6    | return <Redirect href="/(authenticate)/login" />;
7  };
8
9  export default index;
10
11 const styles = StyleSheet.create({});
12
13 // -> "/"
14 |
```

Рисунок 2.9 – Код файлу index.js у папці app

На самому екрані автентифікації необхідно ввести свою електронну пошту та пароль у спеціально призначені для цього поля введення даних (рис. 2.10, 2.11). Вказані облікові дані використовуються для перевірки особистості користувача і надання доступу до відповідних ресурсів. Цей процес є важливим етапом для забезпечення безпеки та захисту користувацької інформації, гарантує, що тільки авторизовані користувачі можуть отримати доступ до додатку. Після введення електронної пошти та пароля необхідно натиснути кнопку для підтвердження, щоб завершити процес автентифікації та отримати доступ до облікового запису.



The image shows a login form with two input fields: "enter your email" (with an envelope icon) and "enter your password" (with a lock icon). Below the fields are two options: "Keep me logged in" (with a checkbox) and "Forgot Password" (a blue link).

Рисунок 2.10 – Поля для введення паролю та електронної пошти

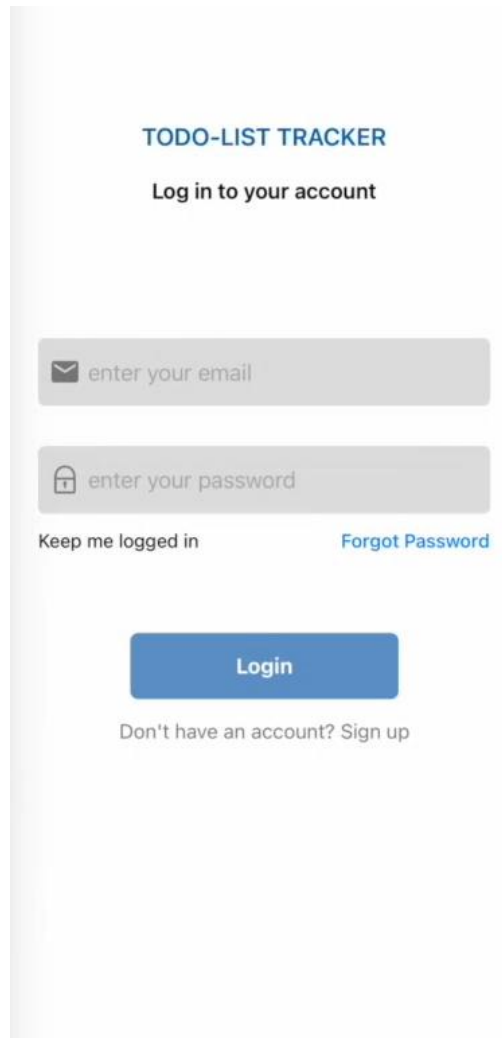


Рисунок 2.11 – Інтерфейс екрану автентифікації у додатку

Також на екрані автентифікації передбачена кнопка для тих користувачів, які ще не зареєстровані в системі. При натисканні на цю кнопку, користувачу відкриється екран реєстрації у додатку, де буде потрібно ввести своє ім'я, електронну пошту та пароль. Цей процес реєстрації допоможе створити новий обліковий запис для користувача, дозволяючи йому повністю скористатися всіма функціями додатку. Крім того, якщо користувач випадково потрапив на екран реєстрації, він може легко повернутися до екрану автентифікації за допомогою спеціальної кнопки «Вже маєте аккаунт?» (рис. 2.12). Ця функціональність забезпечує гнучкість і зручність для користувачів, дозволяючи їм без зусиль перемикатися між процесами реєстрації та автентифікації.

The image shows a registration form for an application titled "TODO-LIST TRACKER". The form is centered on a light gray background. At the top, the title "TODO-LIST TRACKER" is displayed in blue, followed by the text "register to your account" in black. Below this, there are three input fields, each with a gray background and a light gray border. The first field has a person icon and the text "enter your name". The second field has an envelope icon and the text "enter your email". The third field has a lock icon and the text "enter your password". Below these fields is a blue button with the text "Register" in white. At the bottom, there is a link that says "Already have an account? sign Up" in a smaller, gray font.

Рисунок 2.12 – Інтерфейс екрану реєстрації у додатку

Також варто відзначити, що при успішній реєстрації нового облікового запису користувача з'являється відповідне повідомлення про успішну реєстрацію. Це повідомлення виконує важливу роль, оскільки інформує користувача про те, що всі введені ним дані були коректно прийняті і оброблені системою. Повідомлення про успішну реєстрацію з'являється у вигляді спливаючого вікна на екрані, що дає користувачу зрозуміти, що процес створення облікового запису завершений (рис. 2.13). Такий підхід додає додатковий рівень зручності та впевненості для користувача, оскільки він отримує візуальне підтвердження успішного виконання своєї дії.

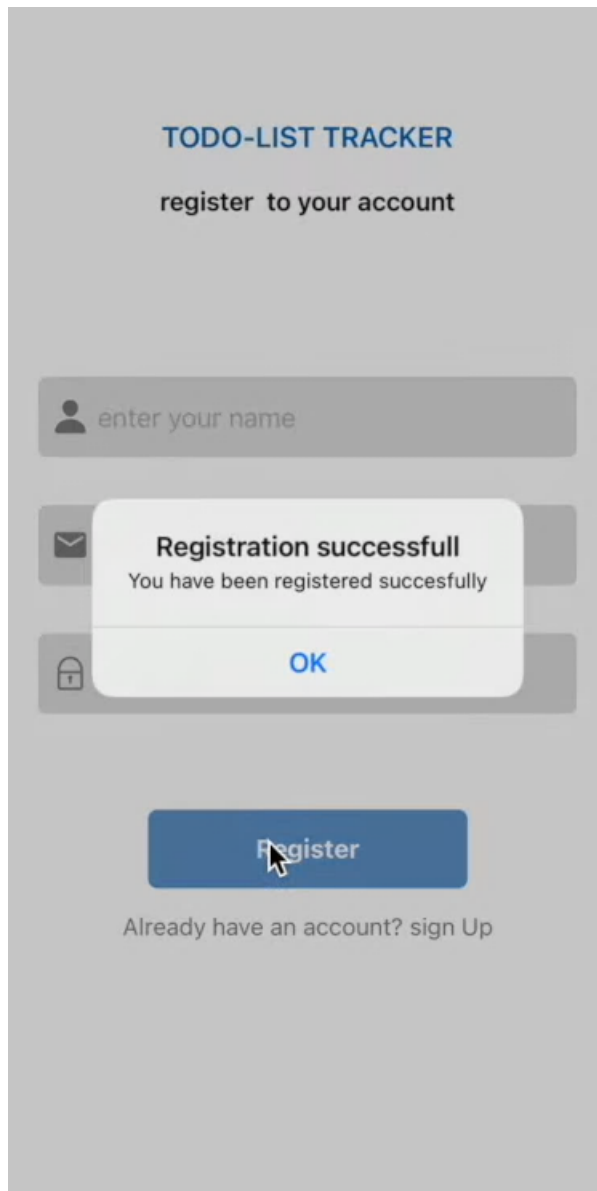


Рисунок 2.13 – Зображення спливаючого вінка про успішну реєстрацію

Для функціоналу додатку, який буде використовуватися для ефективного керування справами та задачами, найважливішим аспектом є можливість додавання нових задач і формування зручного списку з них. Ця функція є ключовою, оскільки вона дозволяє користувачам організувати свої завдання у структурований та доступний спосіб.

У додатку реалізована функція, яка надає користувачу декілька корисних опцій. По-перше, користувач може обирати категорію задач з трьох заздалегідь запропонованих варіантів, що значно спрощує процес організації справ. По-друге, додаток пропонує вже існуючі шаблони задач на вибір, що робить процес



планування ще більш оперативним та ефективним. І, по-третє, користувач має можливість створювати власні задачі, якщо стандартні варіанти не відповідають його потребам (рис. 2.14). Це було зроблено з метою пришвидшити процес планування, а також забезпечити користувачеві позитивний досвід користування додатком. Завдяки такому підходу користувачі можуть швидко і легко додавати нові задачі, не витрачаючи багато часу на їх категоризацію.

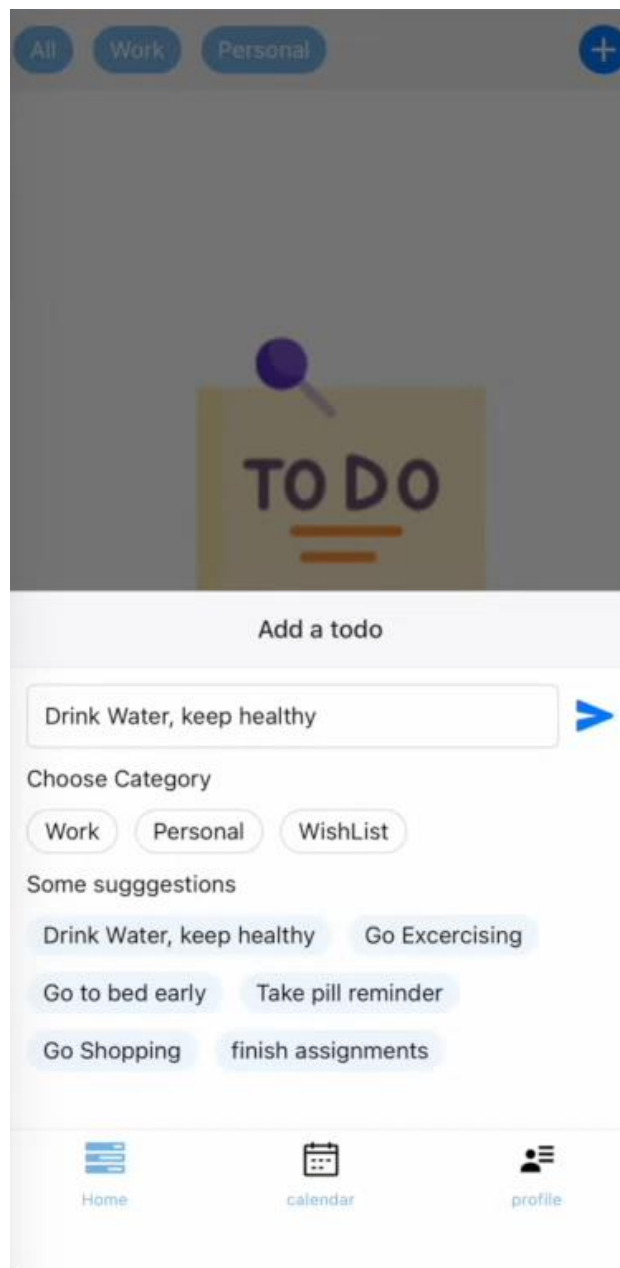


Рисунок 2.14 – Вікно додавання задачі до списку

Для наочного відображення прогресу та більш зручної організації списку завдань, екран з задачами в додатку розділено на дві окремі секції: задачі, що потрібно виконати, та вже виконані завдання (рис. 2.15). Це рішення дозволяє користувачам легко орієнтуватися у своєму робочому навантаженні та бачити, які завдання ще потребують уваги, а які вже успішно завершені. Такий підхід не лише підвищує ефективність управління справами, але й створює відчуття досягнення, адже користувач може візуально відстежувати свій прогрес. Завдяки цьому розподілу, кожен користувач отримує чітку картину свого робочого процесу, що сприяє кращій організації часу та підвищує мотивацію до виконання поставлених задач.

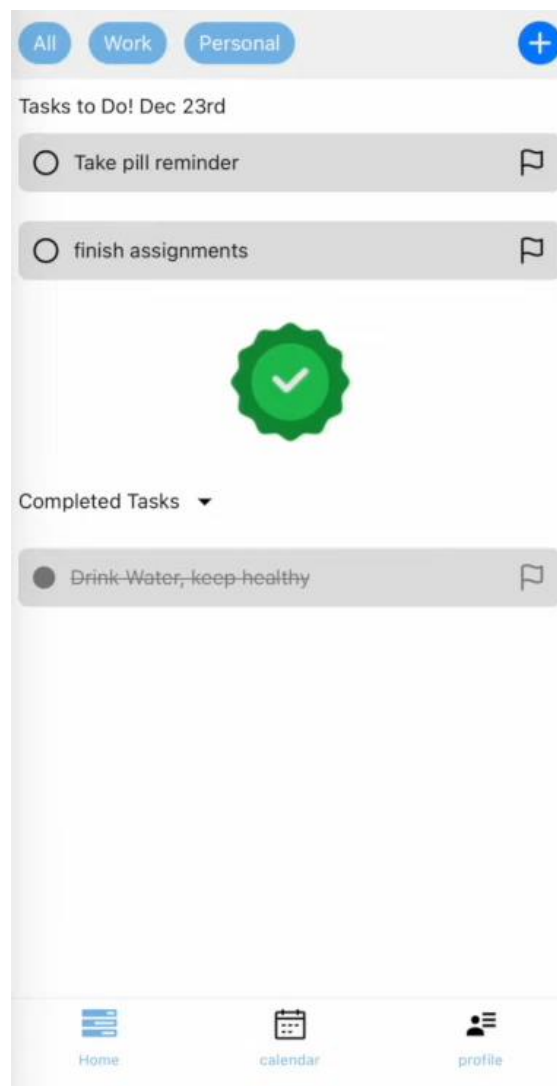


Рисунок 2.15 – Домашня сторінка зі списком задач

Для ще більш зручного управління справами доступно редагування вже доданих задач (рис. 2.16). Користувач може змінювати категорію задачі, її назву, додавати підзавдання, встановлювати кінцеву дату виконання, налаштовувати час нагадувань, задавати періодичність повторення завдання і додавати нотатки до задачі. Це забезпечує більш гнучке та персоналізоване планування справ, що відповідає індивідуальним потребам користувача.

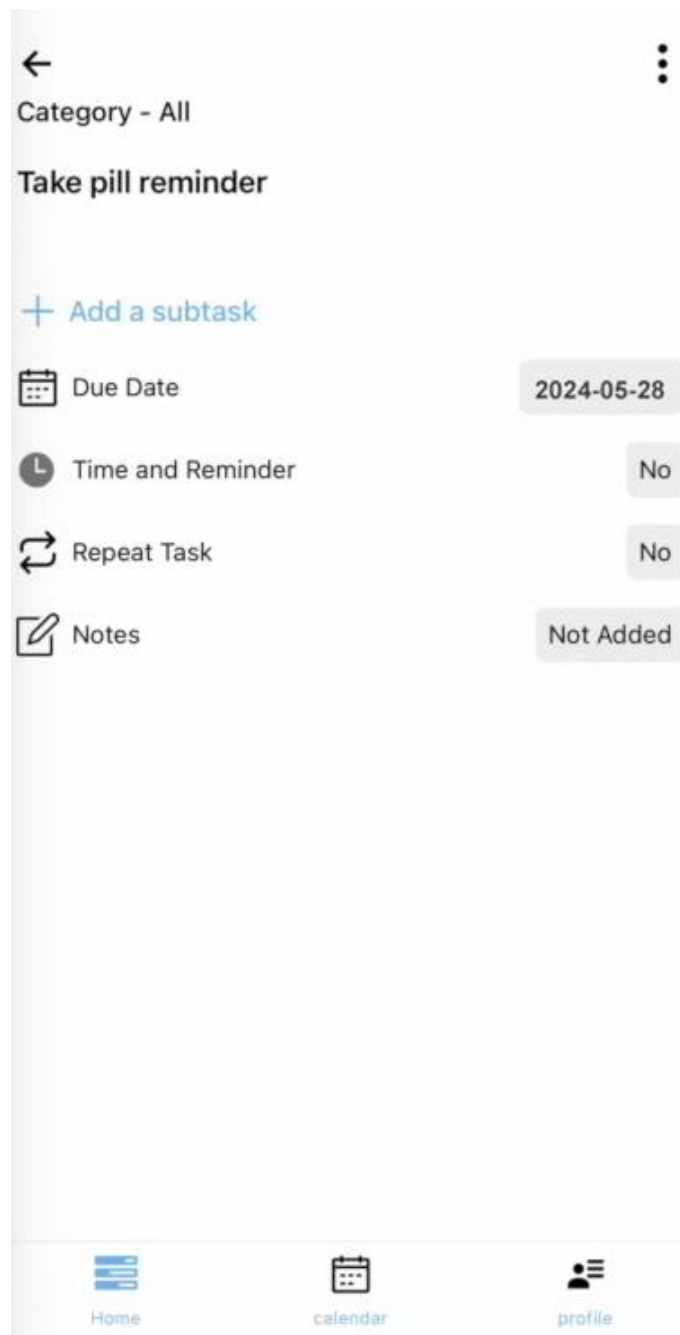


Рисунок 2.16 – Вікно редагування задачі

Для реалізації екрану із календарем необхідно встановити бібліотеку `react-native-calendars`. Команда `yarn add react-native-calendars` використовується для додавання бібліотеки `react-native-calendars` до проєкту React Native. Бібліотека `react-native-calendars` надає різні компоненти для роботи з календарями у мобільних додатках, створених за допомогою React Native. Вона містить різноманітні компоненти календарів, такі як календар на місяць, тиждень або день, а також календар із можливістю вибору діапазону дат. Це дозволяє легко інтегрувати календарі у додаток для різних потреб. Бібліотека пропонує багато варіантів налаштування зовнішнього вигляду та поведінки календарів, що дозволяє змінювати кольори, шрифти, стилі та інші аспекти, щоб календарі відповідали дизайну додатку [36].

За допомогою `react-native-calendars` можна легко маркувати певні дати, наприклад, виділити важливі події, нагадування або дедлайни. Це робить календарі не тільки зручними для перегляду, але й інформативними. Важливою перевагою є підтримка різних платформ: `react-native-calendars` підтримує як iOS, так і Android, що дозволяє створювати кросплатформні додатки з однаковим функціоналом календарів. Бібліотека також підтримує обробку подій, таких як вибір дати, прокручування та натискання на елементи календаря, що дозволяє легко інтегрувати календарі з іншими частинами додатку та створювати інтерактивні інтерфейси [36].

Отже, у цьому додатку реалізовано календар, який відображає поточну дату і підсвічує обрану користувачем дату. Користувач може переглядати інформацію про виконані завдання за обраною датою безпосередньо на екрані календаря. Це дозволяє користувачам ефективно відстежувати свою продуктивність і підтримувати свою вмотивованість на високому рівні (рис. 2.17).

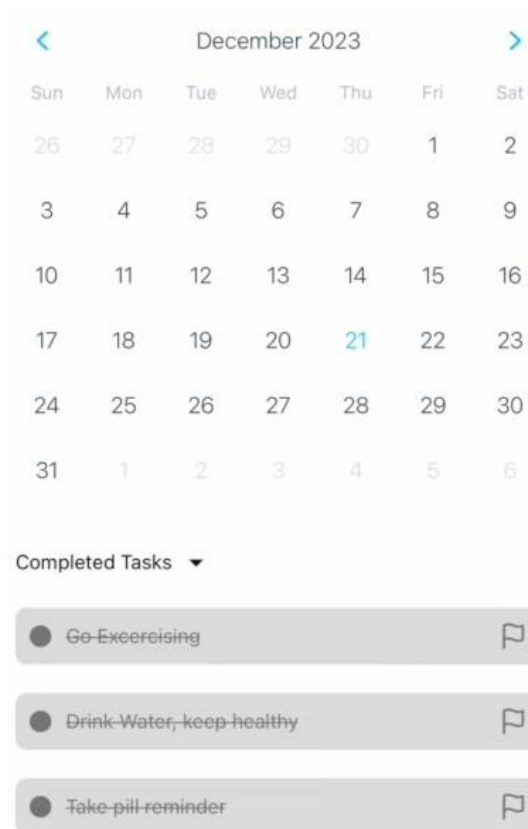


Рисунок 2.17 –Сторінка календаря зі списком виконаних за обраний день задач

На сторінці профілю користувача (рис. 2.18), крім основної інформації, такої як аватар і кількість виконаних і запланованих задач, також відображається інформація про те, скільки днів користувач вже використовує додаток для планування свого дня. Крім цього, статистика на сторінці показує співвідношення між виконаними і запланованими до виконання завданнями. Це дозволяє користувачеві оцінити свою продуктивність і ступінь досягнення цілей, а також мотивує до виконання запланованих завдань. Це дозволяє користувачеві мотивуватися до виконання запланованих справ. Крім того, на цій сторінці доступний перегляд задач, запланованих на наступний тиждень, що дозволяє тримати під контролем свою продуктивність і плани.



Рисунок 2.18 –Сторінка профілю користувача

Висновки до розділу:

Для проєкту реалізації мобільного додатку для ефективного керування справами було уважно обрано та обґрунтовано технології розробки, що найкраще відповідають вимогам та цілям проєкту. Обрані технології мають забезпечити оптимальну продуктивність, надійність та зручність у використанні, враховуючи вимоги до кросплатформенності та можливості інтеграції з різними сервісами та API.

Було ретельно сформульовано завдання для реалізації програмного рішення, враховуючи дослідження подібних додатків, що були проведені у першому розділі роботи. Ці дослідження стали основою для визначення функціональних і нефункціональних вимог до мобільного додатку для ефективного керування справами.

В рамках розділу було розроблено проєкт програмного забезпечення мобільного додатку для ефективного керування справами. Також він включає опис його роботи та програмну реалізацію, що відповідають вимогам, сформульованим на попередніх етапах розробки.

## **ВИСНОВКИ**

У першому розділі кваліфікаційної роботи детально досліджено існуючі додатки для ефективного керування справами. Проведений аналіз включав оцінку функцій та можливостей кожного додатка, та виявлення основних переваг і недоліків цих додатків загалом. На основі отриманих даних були сформульовані чіткі вимоги та цілі, які визначили подальший напрямок розробки програмного забезпечення для керування справами.

У другому розділі кваліфікаційної роботи було чітко сформульовано завдання для розробки програмного забезпечення, детально розглянуто і продемонстровано дизайн різних елементів мобільного додатку для ефективного керування справами. На основі визначених вимог було успішно розроблено додаток, який повністю відповідає поставленим завданням. Його інтерфейс є простим та інтуїтивно зрозумілим, що робить його використання легким і зручним для користувачів.



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Todoist | A To-Do List to Organize Your Work & Life. Todoist. URL: <https://todoist.com/?locale=en> (дата звернення: 03.05.2024).
2. Inc. Todoist: To-Do List & Planner. App Store. URL: <https://apps.apple.com/us/app/todoist-to-do-list-planner/id572688855> (дата звертання: 03.05.2024).
3. Inc D. Todoist: to-do list & planner – Додатки в Google Play. Android Apps on Google Play. URL: <https://play.google.com/store/apps/details?hl=uk&id=com.todoist> (дата звернення: 03.05.2024).
4. 10 додатків для оптимізації робочого процесу - Bazilik Media. Bazilik Media. URL: <https://bazilik.media/10-dodatktiv-dlia-optymizatsii-robochoho-protsesu/> (дата звернення: 03.05.2024).
5. Що таке Trello та як ним користуватись >>> Блог хостера HOSTiQ.ua. Блог хостера HOSTiQ.ua. URL: <https://hostiq.ua/blog/ukr/what-is-trello-2/> (дата звернення: 03.05.2024).
6. Керуйте проектами своєї команди звідусіль | Trello. Manage Your Team's Projects From Anywhere | Trello. URL: <https://trello.com/uk> (дата звернення: 03.05.2024).
7. Trello 101: як користуватися дошками й картками Trello | Trello. Manage Your Team's Projects From Anywhere | Trello. URL: <https://trello.com/uk/guide/trello-101> (дата звернення: 03.05.2024).
8. Підбірка найкращих інструментів для віддаленої роботи. eSputnik. URL: <https://esputnik.com/uk/blog/korysni-instrumenty-dlya-komandnoi-roboty> (дата звернення: 03.05.2024).
9. Почніть використовувати Asana для особистих проектів. Cloudfresh. URL: <https://cloudfresh.com/ua/cloud-blog/yak-efektyvno-vykorystovuvaty-asana-u-riznyh-sferah/> (дата звернення: 03.05.2024).

10. Asana Product. Partnerway. URL: <https://www.partnerway.com/services/asana-ua> (дата звернення: 03.05.2024).
11. Початок роботи з Microsoft To-Do - Підтримка від Microsoft. Microsoft Support. URL: <https://support.microsoft.com/uk-ua/office/початок-роботи-з-microsoft-to-do-762cbbf9-7fc1-48e5-b619-005622da89d0> (дата звернення: 03.05.2024).
12. Настроювання Microsoft To Do - Підтримка від Microsoft. Microsoft Support. URL: <https://support.microsoft.com/uk-ua/office/настроювання-microsoft-to-do-490c1a8c-2333-4952-8125-841afadb9620> (дата звернення: 03.05.2024).
13. To Do - допомога та навчання. Microsoft Support. URL: <https://support.microsoft.com/uk-ua/todo> (дата звернення: 03.05.2024).
14. Remember The Milk: Online to-do list and task management. Remember The Milk: Online to-do list and task management. URL: <https://www.rememberthemilk.com/?hl=en-GB> (дата звернення: 04.05.2024).
15. Jira Software - Features | Atlassian. Collaboration software for software, IT and business teams. URL: <https://www.atlassian.com/software/jira/features> (дата звернення: 04.05.2024).
16. WrestleWorks. Hour Blocks: Day Planner. App Store. URL: <https://apps.apple.com/us/app/hour-blocks-day-planner/id1456275153> (дата звернення: 04.05.2024).
17. ТОП CRM система URL: <https://ratelist.top/r/12#> (дата звертання: 04.05.2024)
18. Сервіси для керування проектами: ефективні інструменти для командної роботи. Інтернет маркетинг статті від Владислави Рикової: SEO, SMM, PPC. URL: <https://vlada-rykova.com/ua/top-10-sistem-upravleniya-proektami-dlya-ukrainy/#Wrike> (дата звернення: 04.05.2024).
19. Versatile & Robust Project Management Software | Wrike. Versatile & Robust Project Management Software | Wrike. URL: <https://www.wrike.com/> (дата звернення: 04.05.2024).

20. 10 Best Work Management Software & Tools (June 2024). Unite.AI. URL: <https://www.unite.ai/best-work-management-software-tools/> (дата звернення: 04.05.2024).
21. Шиманська В. РОБОТА З NOTION: ЦЕ ВАРТО ЗНАТИ. ANT MEDIA. URL: <https://www.theantmedia.com/post/roboata-z-notion-ce-var-to-znati> (дата звернення: 04.05.2024).
22. Учасники проєктів Вікімедіа. Notion (програмне забезпечення) – Вікіпедія. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Notion\\_\(програмне\\_забезпечення\)](https://uk.wikipedia.org/wiki/Notion_(програмне_забезпечення)) (дата звернення: 05.05.2024).
23. Як користуватися сервісом ClickUp: 7 актуальних способів. Apix-Drive. URL: <https://apix-drive.com/ua/blog/useful/7-sposobiv-vikoristannja-clickup> (дата звернення: 15.05.2024).
24. ClickUp – це більше, ніж система управління проєктами та краща альтернатива Bitrix24. URL: <https://www.seotm.com/ua/blog/clickup-upravlinnia-proektami-ta-alternativa-bitrix24.html> (дата звернення: 16.05.2024).
25. Учасники проєктів Вікімедіа. Управління проєктами – Вікіпедія. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Управління\\_проєктами](https://uk.wikipedia.org/wiki/Управління_проєктами) (дата звернення: 16.05.2024).
26. Менеджмент: секрети та стратегії. Блог | Центр Персонального Розвитку. URL: <https://ua.crp-wroclaw.com/blog/менеджмент-секрети-та-стратегії/> (дата звернення: 16.05.2024).
27. Учасники проєктів Вікімедіа. Visual Studio Code – Вікіпедія. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://uk.wikipedia.org/wiki/Visual_Studio_Code) (дата звернення: 16.05.2024).
28. Expo CLI. Expo Documentation. URL: <https://docs.expo.dev/more/expo-cli/> (дата звернення: 20.05.2024).
29. The New Expo CLI URL: <https://blog.expo.dev/the-new-expo-cli-f4250d8e3421> (дата звернення: 20.05.2024).
30. create-expo-app. Expo Documentation. URL: <https://docs.expo.dev/more/create-expo/> (дата звернення: 20.05.2024).

31. Install Expo Router. Expo Documentation.  
URL: <https://docs.expo.dev/router/installation/#manual-installation> (дата звернення: 23.05.2024).
32. Що таке Babel в JavaScript?. ItProger. URL: <https://itproger.com/ua/spravka/javascript/babel> (дата звернення: 23.05.2024).
33. Raji A. Understanding body-parser, cookie-parser, morgan, nodemon, pm2, serve-favicon, cors, dotenv, fs-extra, moment in Express.js. DEV Community.  
URL: <https://dev.to/yemiklein/understanding-body-parser-cookie-parser-morgan-nodemon-pm2-serve-favicon-cors-dotenv-fs-extra-moment-in-expressjs-2fpm> (дата звернення: 23.05.2024).
34. Barros S. Authentication: how to create a NodeJS application using JWT. LinkedIn: Log In or Sign Up.  
URL: <https://www.linkedin.com/pulse/authentication-how-create-nodejs-application-using-jwt-sam-barros> (дата звернення: 23.05.2024).
35. Учасники проектів Вікімедіа. MongoDB – Вікіпедія. Вікіпедія.  
URL: <https://uk.wikipedia.org/wiki/MongoDB> (дата звернення: 23.05.2024).
36. Yarn. Yarn. URL: <https://classic.yarnpkg.com/en/package/react-native-calendars> (дата звернення: 23.05.2024).
37. Моркун Н. В., Маринич І. А. Методичні вказівки до виконання кваліфікаційної роботи бакалавру для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”. Кривий Ріг : Видавничий центр КНУ, 2019. 50 с.
38. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).
39. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).
40. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація).

41. ДСТУ 3651.0-97 Метрологія. Одиниці фізичних величин. Основні одиниці фізичних величин Міжнародної системи одиниць. Основні положення, назви та позначення Київ, Держстандарт України, 1998. 27 с. (Інформація та документація).