Приватний заклад вищої освіти

Одеський технологічний університет «ШАГ»

Кафедра інформаційних технологій та фундаментальної підготовки

Випускна кваліфікаційна робота

«ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ РАНЖОВАНОГО УПРАВЛІННЯ БІЗНЕС-ЗАДАЧАМИ (за прикладом сервісу Notion»

на здобуття бакалаврського рівня вищої освіти

зі спеціальності «122 Комп'ютерні науки»

Розробники:

№ з/п	ПІБ	Роль
1	Червона А.О	Розробка UI/UX - дизайну
2	Боярчук Б.М	Розробка (backend)
3	Котліков А.О	Розробка (frontend)
4	Величко В.М	Розробка UI/UX - дизайну

Ментори:

№ з/п	ПІБ, посада	Напрямок
1	Черкезян Крістіне Арутюнівна	Розробка ПЗ
2	Судавная Т.О.	Графічний дизайн

Автор звіту

_____ Котліков Альберт Олександрович

Анотація до проєкту

Причиною проекту стало прагнення забезпечити користувачів інструментом, який допоможе упорядкувати їхні ідеї, плани та інформацію. У сучасному світі, де кожен день насичений різноманітними завданнями, зустрічами та дедлайнами, мати організовану систему для збереження та управління інформацією стає критично важливим. Веб-застосунок для створення та управління сторінками з нотатками та планами стає незамінним помічником для багатьох користувачів – від студентів і професіоналів до творчих особистостей та підприємців.

Чому людям потрібні додатки для нотаток і планування?

Усі ми стикаємося з необхідністю зберігати та систематизувати інформацію. Традиційні методи, такі як паперові блокноти або стікери, хоча й мають свою користь, але не завжди здатні задовольнити вимоги сучасного темпу життя. Ось кілька ключових причин, чому цифрові інструменти для нотаток і планування стали настільки популярними:

1. Управління великим обсягом інформації:

 Щодня ми маємо справу з величезною кількістю інформації: від робочих завдань до особистих нагадувань. Веб-застосунок для нотаток дозволяє ефективно організувати цю інформацію, забезпечуючи доступ до неї в будь-який час і з будь-якого пристрою.

2. Підвищення продуктивності:

- Планування завдань і проєктів допомагає чітко бачити свої пріоритети, уникати перевантаження та ефективно керувати своїм часом. Завдяки цифровим інструментам можна створювати гнучкі та адаптивні плани, які легко коригуються залежно від обставин.
- 3. Покращення співпраці:

 У багатьох проектах потрібна спільна робота. Додатки для нотаток і планування дозволяють легко ділитися інформацією з колегами, організовувати спільні зустрічі та координувати дії всієї команди. Це значно спрощує комунікацію і допомагає досягати спільних цілей.

4. Творче самовираження:

 Веб-застосунки для нотаток часто пропонують безліч інструментів для візуалізації ідей, упорядкування інформації через простий і зрозумілий користувацький інтерфейс. Це ідеально підходить для творчих особистостей, які шукають зручний спосіб організувати свої думки та проекти.

5. Зручність і доступність:

 Використання цифрових нотаток забезпечує доступ до інформації з будь-якого місця. Завдяки хмарним технологіям, усі ваші дані завжди під рукою, незалежно від того, чи ви перебуваєте в офісі, вдома або в дорозі.

Як наш проект задовольняє ці потреби?

Розробляючи наш веб-застосунок, ми прагнули створити інструмент, який би не тільки відповідав вимогам користувачів, але й перевершував їхні очікування. Основна ідея нашого проекту полягає в наданні універсального і гнучкого рішення для організації інформації, яке можна адаптувати до будь-яких потреб.

1. Різноманітність типів сторінок:

 Ми запропонували п'ять основних типів сторінок: порожні сторінки для простих нотаток, дошки для візуального управління завданнями, списки для організації завдань, таблиці для структурованих даних та галереї для зберігання зображень. Це дозволяє користувачам вибрати оптимальний формат для кожного типу інформації.

2. Інтуїтивно зрозумілий інтерфейс:

 Завдяки інтеграції з сучасними UI/UX-інструментами, такими як Figma, і використанню Angular, ми створили зручний і приємний у використанні інтерфейс. Він дозволяє легко створювати, редагувати та організовувати сторінки.

3. Гнучкість і кастомізація:

 Користувачі можуть налаштовувати сторінки за своїм смаком, змінювати зовнішній вигляд елементів і адаптувати інтерфейс під свої потреби. Це забезпечує індивідуальний підхід і підвищує зручність використання.

4. Ефективне збереження та доступ до даних:

 Інформація зберігається автоматично при виході зі сторінки, що дозволяє користувачам зосередитися на своїх завданнях без необхідності постійно думати про збереження даних.

Наша мета – створити веб-застосунок, який допоможе користувачам ефективно керувати своїм часом, завданнями та ідеями. У світі, де кожна хвилина має значення, ми прагнемо надати інструмент, який допоможе зосередитися на важливому і досягати успіху. Наш веб-застосунок для нотаток і планування – це відповідь на виклики сучасного життя, який забезпечує зручність, гнучкість і продуктивність у повсякденній діяльності.

Загальний опис.

Проект представляє собою веб-застосунок, що дозволяє користувачам створювати, редагувати, видаляти та ділитися сторінками з нотатками. Життєвий цикл проекту: Авторизація та автентифікація, створення нотаток, редагування нотаток, видалення нотаток, додавання нотатки у список "Favorites". Мета проекту – створити зручний та надійний інструмент для управління особистими нотатками, що сприяє підвищенню продуктивності користувачів та полегшує обмін інформацією.

Вимоги до функціоналу. Обов'язкова частина

Процес реєстрації:

- 1. Реєстрація:
 - Для того, щоб зареєструватися, користувач повинен ввести свою електронну адресу в відповідне поле вводу на сторінці реєстрації.
 - Після введення електронної адреси, протягом однієї хвилини на зазначену адресу буде надіслано код авторизації.
 - Користувач повинен ввести отриманий код авторизації в поле вводу, яке з'явиться нижче після підтвердження введеної електронної адреси.

- Після успішного введення коду авторизації, користувач отримає токен доступу, що дозволить йому повноцінно користуватися застосунком.
- 2. Авторизація:
 - Після реєстрації користувач може входити в систему, використовуючи свою електронну адресу та отриманий токен.
 - Токен доступу зберігається на пристрої користувача, що дозволяє автоматично входити в систему без необхідності повторного введення коду авторизації.
 - В разі втрати токену або необхідності його оновлення, користувач може повторно пройти процедуру отримання коду авторизації.

Цей процес забезпечує високий рівень безпеки, дозволяючи уникнути несанкціонованого доступу до облікових записів користувачів, а також спрощує процес входу та реєстрації.

Створення сторінок. Процес створення сторінок у нашому веб-застосунку включає наступні кроки:

- 1. Вибір типу сторінки:
 - Користувач повинен вибрати тип сторінки, яку він хоче створити.
 - Типи сторінок включають: Empty Page (Порожня сторінка), Board (Дошка), List (Список), Table (Таблиця) та Gallery (Галерея).
- 2. Введення імені сторінки:
 - Після вибору типу сторінки, користувач повинен ввести її ім'я у відповідне поле.
- 3. Методи створення сторінок:
 - HomeComponent: Після реєстрації або видалення сторінки, користувач потрапляє на HomeComponent, де він може створити нову сторінку.
 - Пункт меню "New Page": Користувач також може створити сторінку через пункт меню "New Page". При цьому відкривається модальне вікно, в якому потрібно вибрати тип сторінки та ввести її ім'я.

Типи сторінок та їх функціональні можливості:

- 1. Empty Page (Порожня сторінка):
 - Створення тексту з заголовком.
 - Можливість форматування тексту (жирний, курсивний, підкреслений).
- 2. Board (Дошка):
 - Створення списків з картками за зразком Trello.
 - Додавання списків через кнопку "Add list".
 - Додавання карток до списків через кнопку "Add card".
 - Редагування та перетягування карток між списками.
 - Видалення, копіювання списків або додавання карток через меню, що відкривається при натисканні на три крапки в правому верхньому куті списку.
- 3. List (Список):
 - Створення та редагування списків.
 - Додавання елементів списку через кнопку "New".
 - Перетягування елементів списку.
- 4. Table (Таблиця):
 - Створення таблиці.
 - Додавання рядків через кнопку "Add row".
 - Додавання стовпців через кнопку "Add column".
- 5. Gallery (Галерея):
 - Створення галереї з фотографіями.
 - Додавання зображень через кнопку "New", що відкриває вікно вибору файлів на комп'ютері.
 - Зміна імені картки з зображенням.
 - Зміна зображення при натисканні на картинку.

Редагування сторінок:

Після створення сторінки, її ім'я з'являється в лівому меню. Редагування сторінок у нашому веб-застосунку включає наступні функціональні можливості:

- 1. Збереження інформації:
 - Коли користувач виходить зі сторінки, вся інформація автоматично зберігається на сервері.
 - При повторному заході на сторінку, дані завантажуються з сервера, забезпечуючи безперервність роботи користувача.
- 2. Видалення сторінки:

- Через ліве меню: Користувач може видалити сторінку, натиснувши на три крапки біля її імені в лівому меню та вибравши пункт "Delete item".
- Через хедер: Користувач також може видалити сторінку, натиснувши на три крапки, розташовані справа в хедері сторінки, та вибравши пункт "Delete item".
- Після видалення сторінка переміщується до списку "Deleted items".
- 3. Відновлення сторінки:
 - Для відновлення сторінки з списку "Deleted items" користувач повинен натиснути на три крапки біля її імені та вибрати пункт "Recover item".
 - Відновлена сторінка знову з'явиться в лівому меню та буде доступна для подальшого редагування.
- 4. Додавання сторінки до списку обраного (Favorites):
 - Щоб додати сторінку до списку "Favorites", потрібно натиснути на три крапки біля її імені в лівому меню та вибрати пункт "Fav item".

Ці функції забезпечують користувачам можливість ефективно керувати своїми сторінками, зберігаючи та відновлюючи важливу інформацію у будь-який час.

Шаблони сторінок:

У нашому веб-застосунку користувачі мають можливість створювати сторінки на основі готових шаблонів. Це значно спрощує процес налаштування сторінок і дозволяє швидко приступити до роботи з ними.

Доступ до шаблонів

Для створення сторінки на основі шаблону користувачу необхідно:

- 1. Натиснути на пункт меню "Templates".
- 2. Вибрати потрібний шаблон зі списку доступних.

Шаблони для типів сторінок

Готові шаблони доступні для всіх типів сторінок, за винятком **Empty Page** (Порожня сторінка). Це дозволяє швидко налаштовувати більш складні

сторінки з попередньо визначеним вмістом та структурою. Нижче наведені доступні шаблони для кожного типу сторінки:

- 1. Board (Дошка):
 - Personal Planning: Шаблон для особистого планування, що включає списки для щоденних завдань, тижневих цілей та довгострокових планів. Це зручно для організації особистих справ і задач.

2. List (Список):

- **Books to Read**: Шаблон для створення списку книг, які ви плануєте прочитати.
- Тravel Kit: Шаблон для організації списку речей для подорожей.
 Включає категорії, такі як "Одяг", "Електроніка", "Документи" та "Інше", для зручного пакування.

3. Table (Таблиця):

• Weekly Routine: Шаблон таблиці для планування тижневого розпорядку. Містить колонки для днів тижня та рядки для розкладу завдань або подій на кожен день. Це корисно для управління своїм часом та задачами протягом тижня.

4. Gallery (Галерея):

 Moodboard: Шаблон для створення візуальних колажів або настрій-дошок. Дозволяє зберігати та організовувати зображення, що відображають ваші ідеї, проекти або просто натхнення. Підходить для дизайнерів, художників та творчих людей.

Використання шаблонів

При виборі шаблону відповідна сторінка створюється з попередньо заповненим вмістом. Це дозволяє користувачам миттєво почати роботу, редагуючи лише необхідні деталі, без потреби створювати структуру з нуля.

Шаблони розроблені так, щоб бути максимально зручними і універсальними, забезпечуючи швидкий старт та ефективне використання функцій веб-застосунку.

Пошук сторінок:

У веб-застосунку реалізовано зручну функцію пошуку, яка дозволяє швидко знаходити створені сторінки. Це особливо корисно при роботі з великою кількістю сторінок різних типів.

Використання функції пошуку

Щоб знайти потрібну сторінку, потрібно:

- 1. Відкрити меню "Search".
- 2. Ввести назву сторінки в пошукове поле.

Фільтрація результатів пошуку

Для точнішого пошуку можна скористатися додатковими фільтрами:

1. Фільтрація за типом сторінки:

- Вибрати тип сторінки зі списку фільтрів, що дозволяє обмежити результати до конкретного типу. Наприклад:
 - Board (Дошка): Відображаються тільки сторінки типу дошки.
 - List (Список): Показуються тільки сторінки у форматі списків.
 - **Table (Таблиця)**: Відображаються сторінки з таблицями.
 - Gallery (Галерея): Відображаються лише галереї.
- Ця фільтрація зручна для швидкого знаходження потрібного типу сторінок серед багатьох.

2. Фільтрація за алфавітом:

 Результати пошуку можна впорядкувати за алфавітом. Це корисно, якщо відома лише частина назви сторінки або якщо потрібно переглянути всі сторінки в алфавітному порядку.

Приклад використання фільтрів

Розглянемо приклади фільтрації для більш точного пошуку:

- Пошук сторінки типу "Board":
 - Після натискання на "Search", у пошуковому полі вводиться "Project Planning".
 - Далі обирається фільтр "Board" зі списку типів сторінок.
 - Результати будуть показані тільки для сторінок типу "Board", що відповідають введеному тексту.

• Пошук сторінки за алфавітом:

- У пошуковому полі вводиться частина назви, наприклад, "Travel".
- Після цього можна впорядкувати результати за алфавітом, щоб швидко знайти сторінки, що починаються на "Travel", такі як "Travel Plans" або "Travel Kit".

Функція пошуку дозволяє швидко знаходити необхідні сторінки, забезпечуючи ефективне управління та легкий доступ до інформації у веб-застосунку.

Зміна теми (обоїв):

Веб-застосунок пропонує користувачам можливість змінювати тему або обої своєї сторінки, щоб зробити роботу з ним ще більш приємною та персоналізованою. Ця функція дозволяє налаштовувати зовнішній вигляд сторінок відповідно до особистих уподобань або настрою користувача.

Як змінити обої сторінки?

- Натиснути кнопку "Change Cover": Для зміни обоїв необхідно натиснути на кнопку "Change Cover", яка знаходиться в правому верхньому кутку сторінки. Ця кнопка легко доступна, що забезпечує швидкий і зручний доступ до функції зміни обоїв.
- Вибірати тему з випадаючого меню: Після натискання на кнопку відкриється випадаюче меню з доступними темами. У цьому меню можна переглянути різноманітні варіанти обоїв, які пропонує застосунок.
- Застосування обраної теми: Після вибору бажаної теми із випадаючого списку, вона автоматично застосовується до поточної сторінки. Це миттєво змінює зовнішній вигляд сторінки, забезпечуючи новий фон і естетичне оформлення.

Архітектура проекту

Проект реалізований з використанням клієнт-серверної архітектури. Фронтенд розроблений на основі Angular (TS, HTML), що забезпечує динамічний та інтерактивний користувацький інтерфейс. Бекенд написаний на мові програмування C#, що забезпечує надійну та масштабовану серверну частину. Клієнт-серверна архітектура дозволяє ефективно розділити обробку даних між клієнтською та серверною частинами, забезпечуючи швидкий відгук користувацького інтерфейсу та надійне збереження даних на сервері.

Команда проекту

Над проектом працювала команда з чотирьох осіб. Програмуванням займалися два розробники: один (Котліков Альберт) спеціалізувався на фронтенді, використовуючи Angular, а інший (Боярчук Богдан) працював над бекендом на мові С#. Дизайном сайту займалися два дизайнери (Червона Анна та Величко Владислав), які розробили зручний та привабливий користувацький інтерфейс. Завдяки їх спільним зусиллям, проект був реалізований на високому рівні якості.

ФРОНТЕНД

Для розробки фронтенду я використовував Angular 15 версії, що дозволило створити динамічний і зручний користувацький інтерфейс. Важливим аспектом роботи було те, що деякі частини інтерфейсу, такі як хедер, випадаючі списки та модальні вікна, були реалізовані з використанням Bootstrap. Однак, оскільки у нас був свій унікальний стиль для всіх елементів сайту, я взяв з Bootstrap лише функціональну частину, а верстку довелося переробити відповідно до наших вимог.

Цей підхід забезпечив відповідність готового інтерфейсу затвердженим макетам та підтримку високої якості користувацького досвіду, дотримуючись при цьому нашого унікального стилю дизайну.

Моя задача як фронтенд розробника полягала в тому, щоб реалізувати інтерфейс сайту в точності відповідно до макетів, наданих дизайнерами через Figma. Всі технічні завдання та макети були надані дизайнерами саме в цьому інструменті.

Для розробки фронтенду я використовував Angular, що дозволило створити динамічний і зручний користувацький інтерфейс. Важливим аспектом роботи було те, що деякі частини інтерфейсу, такі як хедер, випадаючі списки та модальні вікна, були реалізовані з використанням Bootstrap. Однак, оскільки у нас був свій унікальний стиль для всіх елементів сайту, я взяв з Bootstrap лише функціональну частину, а верстку довелося переробити відповідно до наших вимог.

В Angular вся структура додатку ділиться на компоненти. Компоненти – це відокремлені, самостійні частини інтерфейсу, які можна використовувати повторно. У нашому проекті є два типи компонентів:

- 1. Компоненти, які присутні на екрані завжди:
 - Ці компоненти присутні на екрані незалежно від того, яка посилання вказана в адресному рядку браузера.
 - До статичних компонентів належать меню, модальні вікна та хедер.
- 2. Компоненти, які показуються через Router Outlet:
 - Router Outlet це директива Angular, яка служить як контейнер для компонентів, що повинні завантажуватися в залежності від поточного маршруту. Коли користувач переходить за різними URL, Angular визначає, який компонент потрібно завантажити, і відображає його у Router Outlet.
 - Через Router Outlet я показую сторінку реєстрації, лендинг (головну сторінку) та всі сторінки, які створює користувач.

Цей підхід дозволяє ефективно управляти відображенням різних частин інтерфейсу, забезпечуючи чітку структуру та зручність навігації для користувача.

Дані сторінок користувача

На фронтенді сторінки користувача зберігаються для того, щоб відобразити інформацію на екрані та передати іі на бекенд за допомогою затипу. Для цього у компоненті кожного типу сторінки э відповідні поля.

Поля Empty page:

```
id: string = "";;
headerInput: string = "";;
mainText: string = "";
```

Поля List:

Поля Board:

```
interface Card {
  id: string;
  name: string;
  description?: string;
  datetime?: Date;
  files?: string[];
  properties?: Property[];
```

```
id: string;
name: string;
cards: Card[];
}
id: string = '';
lists: List[] = [];
headerInput: string = '';
newListName: string = '';
newListVisible: boolean = true;
selectedFiles: File[] = [];
currentLink: string = "";
```

interface List {

Поля Table:

```
tableData: (string | null)[][] = [];
tableDefault: (string | null)[][] = [
 ['Name', 'Number', 'Text'],
 [null, null, null],
```

```
[null, null, null]
];
id: string = '';
headerInput: string = 'Untitled';
currentLink: string = "";
```

Поля Gallery:

```
interface Card {
 description: string;
 base64Image: string; // Base64 string for the image
export interface GalleryDTO {
 [key: string]: any;
 noteId: string;
```

```
currentLink: string;
content: Card[];
}
gallery: Card[] = [];
headerInput: string = "";
id: string = "";
selectedFiles: File[] = []; // Список файлов, которые нужно отправить
currentLink: string = "";
```

Зберігання інформації сторінок користувача

Оскільки ми (фронтенд і бекенд розробники) не мали достатньо ресурсів для того, щоб зберегти кожну зміну сторінки в реальному часі, ми прийняли рішення зберігати інформацію про сторінку лише при виході з неї. Це означає, що дані сторінки зберігаються на сервері тільки тоді, коли користувач завершує роботу з нею.

Хоча цей підхід може не бути таким динамічним, як збереження кожної зміни в реальному часі, він дозволяє економити ресурси сервера і зменшує кількість запитів до бази даних. Такий компроміс дозволяє нам забезпечити зручну роботу зі сторінками, при цьому не перевантажуючи сервер. У розробці на Angular іноді можна стикнутися з особливістю, що деструктор компонента викликається лише при закритті сторінки, на якій він знаходиться. Це може призвести до труднощів у збереженні та оновленні даних при переміщенні між сторінками одного типу.

У вашому випадку маршрути для створення нових сторінок користувачами виглядають наступним чином:

path "createnewpage/emptypage/:id" component CreatenewpageComponent
path "createnewpage/board/:id" component CreatenewboardComponent

```
path "createnewpage/list/:id" component CreatenewlistComponent
path "createnewpage/gallery/:id" component CreatenewgalleryComponent
path "createnewpage/table/:id" component CreatenewtableComponent
```

Angular не викликає деструктор при переході з, наприклад, createnewpage/emptypage/1 на createnewpage/emptypage/2. Тому для вирішення цієї проблеми додатково було використано підписку на подію зміни адреси в рядку браузера.

Також, для закриття сторінки використовується декоратор @HostListener('window:beforeunload', ['\$event']), який реагує на подію закриття активної вкладки браузера.

Таким чином, у веб-застосунку POST запит, що містить інформацію про сторінку, відправляється на бекенд у наступних випадках:

- при закритті вкладки браузера
- при оновленні вкладки браузера
- при переході назад або вперед
- при створенні нової сторінки

Ці прийоми дозволяють забезпечити коректне збереження та оновлення даних при переміщенні між сторінками одного типу та закритті вкладок.

Особливості Empty page

У нашому веб-застосунку тип сторінки **Empty Page** є унікальним серед інших типів сторінок, оскільки для редагування тексту на ній використовується елемент contenteditable div. Це рішення було обрано через його здатність забезпечувати просте та інтуїтивне редагування тексту без необхідності використання складних текстових редакторів або додаткових бібліотек.

Contenteditable div є HTML-елементом, який дозволяє користувачам безпосередньо редагувати вміст сторінки. Його основна перевага полягає в тому, що він підтримує базові функції форматування тексту, такі як жирний, курсивний, підкреслений і закреслений текст. Завдяки цьому користувачі можуть легко застосовувати ці стилі, просто вибравши відповідні опції в інтерфейсі застосунку. Оскільки contenteditable div дозволяє редагувати вміст без необхідності використання складних інтерфейсів або додаткових бібліотек, це рішення стало оптимальним для типу сторінки Empty Page. Це дозволяє користувачам зосередитися на створенні і форматуванні тексту, не відволікаючись на складні функції. Крім того, використання стандартних HTML-тегів для форматування забезпечує сумісність з різними браузерами та платформами, що важливо для забезпечення безперебійної роботи застосунку.

Таким чином, для Empty Page було прийнято рішення використовувати contenteditable div, оскільки це забезпечує простий і ефективний спосіб peдarування тексту з можливістю застосування основних стилів форматування. Інші типи сторінок в нашому застосунку мають свої специфічні функціональні можливості, які не вимагають використання contenteditable div, що робить Empty Page особливим і зручним інструментом для створення текстових нотаток з гнучким форматуванням.

Особливості Board i List

Сторінки List та Board у нашому веб-застосунку мають особливі функціональні можливості, які відрізняють їх від інших типів сторінок. Ці можливості реалізовані завдяки використанню бібліотеки CDK Drag and Drop для Angular. Ця бібліотека забезпечує просту і гнучку реалізацію функцій перетягування елементів, що дозволяє створювати інтерактивні списки та дошки з можливістю динамічного переміщення елементів.

Використання CDK Drag and Drop

Бібліотека **CDK Drag and Drop** є частиною Angular Component Dev Kit (CDK) і надає готові рішення для реалізації функціоналу перетягування. Вона підтримує широкі можливості для налаштування та інтеграції в Angular-застосунки, дозволяючи легко додавати можливості перетягування до компонентів.

Реалізація на сторінці Board

На сторінці **Board** CDK Drag and Drop використовується для організації та управління списками і картками за принципом канбан-дошки, подібно до Trello. Користувач може створювати кілька списків, а також додавати до них картки. Завдяки можливостям бібліотеки, ці картки можна легко перетягувати між різними списками, що забезпечує зручне управління завданнями та проектами. Користувачеві достатньо натиснути на іконку олівця, утримуючи ліву кнопку миші, і перетягнути її до потрібного списку.

Реалізація на сторінці List

Сторінка **List** пропонує користувачам інструмент для створення і управління списками завдань або елементів. CDK Drag and Drop забезпечує можливість перетягування окремих елементів списку всередині одного списку або між різними списками. Це дозволяє користувачам легко реорганізовувати елементи списків відповідно до їхніх пріоритетів або змінюваних потреб.

Для переміщення елемента списку достатньо натиснути на нього, утримуючи ліву кнопку миші, і перетягнути його до потрібного місця. Ця функція дозволяє підтримувати порядок у списках і ефективно керувати завданнями.

Переваги використання CDK Drag and Drop

Використання CDK Drag and Drop на сторінках Board і List надає кілька ключових переваг:

- Інтерактивність: Користувачі можуть динамічно змінювати порядок елементів, що забезпечує гнучкість і зручність в управлінні контентом.
- Простота інтеграції: CDK Drag and Drop легко інтегрується в Angular-застосунки і підтримує різноманітні конфігурації та налаштування.
- Поліпшення користувацького досвіду: Зручні та інтуїтивні функції перетягування роблять роботу зі сторінками більш ефективною і приємною.

Завдяки цим можливостям, сторінки **List** та **Board** є потужними інструментами для організації та управління інформацією, що дозволяють користувачам легко адаптувати і реорганізовувати контент відповідно до їхніх потреб.

Перелік використаних технологій

Nº	Назва технологіі
1	Visual Studio Code Розробник: Microsoft
2	Angular (TypeScript, HTML, CSS) Розробник: Google

3	Cdk Drag Drop Розробник: Google
4	Radmin VPN Розробник: Famatech Corp.
5	Postman Розробник: Ankit Sobti, Abhinav Asthana, and Abhijit Kane
6	Bootstrap Розробник: <u>small team of developers</u> on GitHub