

ЛАБОРАТОРНА РОБОТА № 2

Тема:

Побудова вкладеної навігації та оптимізація відображення великих списків у React Native із використанням компонентів **FlatList** та **SectionList**.

Мета:

- ознайомлення з принципами навігації у мобільних застосунках React Native;
 - вивчення архітектури вкладеної навігації;
 - набуття практичних навичок використання Drawer Navigator та Stack Navigator;
 - засвоєння методів передачі параметрів між екранами;
 - опанування ефективного відображення великих наборів даних;
 - вивчення механізму віртуалізації списків;
 - практичне застосування компонентів **FlatList** та **SectionList**;
 - формування навичок оптимізації продуктивності мобільних застосунків.
-

1. Короткі теоретичні відомості

1.1 Навігація у React Native

У мобільних застосунках навігація забезпечує переходи між екранами. Бібліотека **React Navigation** реалізує різні типи навігаторів.

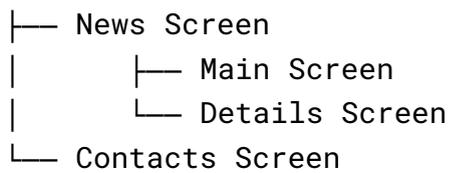
Основні типи:

- **Stack Navigator** — послідовна навігація між екранами;
- **Drawer Navigator** — бокове меню застосунку;
- **Tab Navigator** — нижня (або верхня) панель вкладок.

У складних застосунках використовується вкладена навігація, коли один навігатор містить інший.

Типова структура:

Drawer Navigator



1.2 Віртуалізація списків

При відображенні великої кількості елементів використання звичайного прокручуваного контейнера призводить до значного навантаження на пам'ять.

Компонент **FlatList** використовує механізм віртуалізації:

- відображаються лише видимі елементи;
- невидимі елементи видаляються;
- компоненти перевикористовуються під час прокрутки.

1.3 Компонент FlatList

FlatList застосовується для відображення однорідних списків даних.

Основні можливості:

- динамічне підвантаження даних;
- оновлення списку жестом (Pull-to-Refresh);
- оптимізований рендеринг.

Параметри оптимізації:

- `initialNumToRender`
- `maxToRenderPerBatch`
- `windowSize`

1.4 Компонент SectionList

SectionList використовується для відображення групованих даних.

Кожна секція містить:

- заголовок;
- набір елементів.

2. Порядок виконання роботи

2.1 Налаштування проєкту

1. Створити новий проєкт Expo(Blank):

```
npx create-expo-app@latest --template
```

2. Встановити бібліотеки навігації:

<https://reactnavigation.org/docs/stack-navigator>

<https://reactnavigation.org/docs/drawer-navigator>

3. Запустити застосунок та перевірити його працездатність.
-

2.2 Модель даних

Згенерувати тестові дані. Об'єкт повинен містити щонайменше:

- id
 - title
 - description
 - image
-

2.3 Реалізація списку новин (FlatList)

Створити головний екран застосунку зі списком новин.

Обов'язковий функціонал

1. Pull-to-Refresh

- використати `refreshing`;
- реалізувати `onRefresh`;
- імітувати мережевий запит через `setTimeout`.

2. Infinite Scroll

Реалізувати підвантаження даних:

- `onEndReached`;
- `onEndReachedThreshold`.

3. Візуальні компоненти списку

Обов'язково використати:

- `ListHeaderComponent`;

- ListFooterComponent;
- ItemSeparatorComponent.

4. Оптимізація

Задати параметри:

```
initialNumToRender  
maxToRenderPerBatch  
windowSize
```

2.4 Побудова навігації

Створити структуру:

```
Drawer Navigator  
└─ Stack Navigator  
    └─ MainScreen  
    └─ DetailsScreen
```

Реалізувати:

- перехід між екранами (<https://reactnavigation.org/docs/navigating>);
 - передачу параметрів;
 - динамічний заголовок екрану деталей;
 - усунення подвійного header-а.
-

2.5 Екран контактів (SectionList)

Створити екран **ContactsScreen**.

Реалізувати **SectionList** з:

- sections;
 - renderItem;
 - renderSectionHeader;
 - keyExtractor;
 - ItemSeparatorComponent.
-

2.6 Кастомізація Drawer Menu

Створити власний компонент меню (drawerContent), який містить:

- Аватар;
 - ПІБ;
 - Група;
 - Пункт меню - Новини;
 - Пункт меню - Контакти;
-

3. Контрольні запитання

1. Чим відрізняється FlatList від ScrollView?
 2. Що таке віртуалізація списків?
 3. Як здійснюється передача параметрів між екранами?
 4. Що таке вкладена навігація?
 5. У яких випадках застосовується SectionList?
-

4. Формат задачі

Репозиторій: MobileLabsRN2026/lab2. Надати доступ до репозиторію та відправити лист на пошту kipz_nsi@ztu.edu.ua

Додати файл README.md, що містить:

- інструкцію запуску;
- опис реалізованого функціоналу;
- скріншоти роботи застосунку;
- висновки(див. п. контрольні запитання).