

Звіт

про виконання завдання з лабораторної роботи №1
з курсу “Розробка мобільних застосунків”
студентом Антонюком Олександром Андрійовичем
тема “Налаштування Android Studio. Створення першого проекту”
в 2024-2025 навчальному році

Завдання

1. Завантажте та встановіть Android Studio на комп'ютері.
(<https://developer.android.com/studio>)
2. Встановіть в Android Studio додаткові інструменти та плагіни (за бажанням), використовуючи SDK Manager. (Додайте у клас "скрін" з встановленими інструментами до завдання)
3. Створіть новий проект з підтримкою Kotlin та Empty Activity у якості базового шаблону.
4. Створіть та запустіть емулятор, використовуючи AVD Manager.
5. Замініть у файлі розмітки активності ConstraintLayout на LinearLayout та видаліть "лишні" атрибути у вкладених елементах.
6. Замініть текст у TextView на активності довільним власним та додайте його у рядкові ресурси.
7. Додайте на активність кнопку та довільно налаштуйте її властивості.
8. Додайте до проекту, використовуючи один із способів, обробник натиснення на кнопку, який би довільно змінював зміст текстового поля. Використайте при цьому можливості класу ViewModel (див приклади).
9. Зберіть та запустіть проект на емуляторі. (Додайте у клас "скрін" з емулятором та запусченим проектом).

10. Розмістіть проєкт у git-репозиторії та додайте відповідне посилання у якості виконаного завдання.

Розв'язок

Створимо React Native проєкт з використанням Ехро фреймворку та мови програмування TypeScript та виберемо шаблон Blank:

```
> npx create-expo-app@latest --template
✓ Choose a template: > Blank (TypeScript)
? What is your app named? > lab1
```

Рис. 1 – Створення нового проєкту

Створимо TSX-компонент MyButton, який приймає пропси appData (типу AppData, див. Рис. 3) і setAppData для зміни стану. Він використовує TouchableOpacity, оскільки на відміну від Button, він дозволяє як обробляти натискання, так і стилізувати компонент. Подія onPress змінює текст і колір за допомогою функції handleMyButtonClick().

```
1 type MyButtonProps = {
2   setAppData: Dispatch<SetStateAction<AppData>>,
3   appData: AppData,
4 }
5
6 const MyButton = (props: MyButtonProps) => {
7   const {appData, setAppData} = props;
8   const {text} = appData;
9
10  return (
11    <TouchableOpacity
12      style={buttonStyles.button}
13      onPress={() => handleMyButtonClick(appData, setAppData)}
14    >
15      <Text style={buttonStyles.buttonText}>{text}</Text>
16    </TouchableOpacity>
17  )
18 }
19
20 export default MyButton;
```

Рис. 2 – Створення MyButton компонента



```
1 export type AppData = {  
2   text: string,  
3   textColor: string,  
4 }
```

Рис. 3 – Тип AppData

Для обробки натискання на кнопку MyButton використовується функція handleMyButtonClick, яка приймає значення appData типу AppData, функцію setAppData та змінює сам текст та його колір:



```
1 export const handleMyButtonClick = (  
2   appData: AppData,  
3   setAppData: Dispatch<SetStateAction<AppData>>  
4 ) => {  
5   const {textColor} = appData;  
6  
7   setAppData({  
8     text: `Text is ${textColor}`,  
9     textColor: textColor  
10  })  
11 }
```

Рис. 4 – Функція handleMyButtonClick

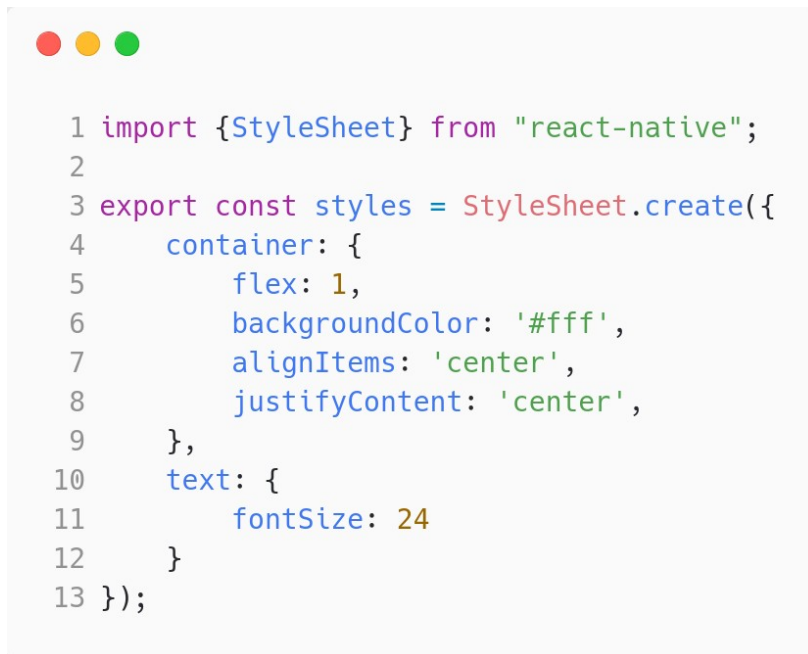
Створимо компонент App, який виконує роль головного екрану програми. Він використовує хук useState для збереження стану тексту, початкове значення

якого - "Here's some text" з чорним кольором. Компонент містить масив кнопок, кожна з яких має свій текст та відповідний колір. Відображений текст змінює свій колір залежно від натискань на кнопки, які генеруються за допомогою методу `map`. Кожна кнопка передає свій текст, колір та функцію `setAppData` для оновлення стану через пропси до компонента `MyButton`.

```
1 const App = () => {
2   const [appData, setAppData] = useState<AppData>({
3     text: 'Here\'s some text',
4     textColor: 'black',
5   });
6
7   const buttons = [
8     {text: "Make text green", textColor: 'green'},
9     {text: "Make text yellow", textColor: 'yellow'},
10    {text: "Make text red", textColor: 'red'},
11  ] as AppData[];
12
13  return (
14    <View style={appStyles.container}>
15      <Text
16        style={[appStyles.text, {color: appData.textColor}]}
17      >{appData.text}</Text>
18      {buttons.map((button, index) => (
19        <MyButton
20          key={index}
21          setAppData={setAppData}
22          appData={button}
23        />
24      ))}
25      <StatusBar style="auto"/>
26    </View>
27  );
28 }
29
30 export default App;
```

Рис. 5 – Створення екрана головної сторінки

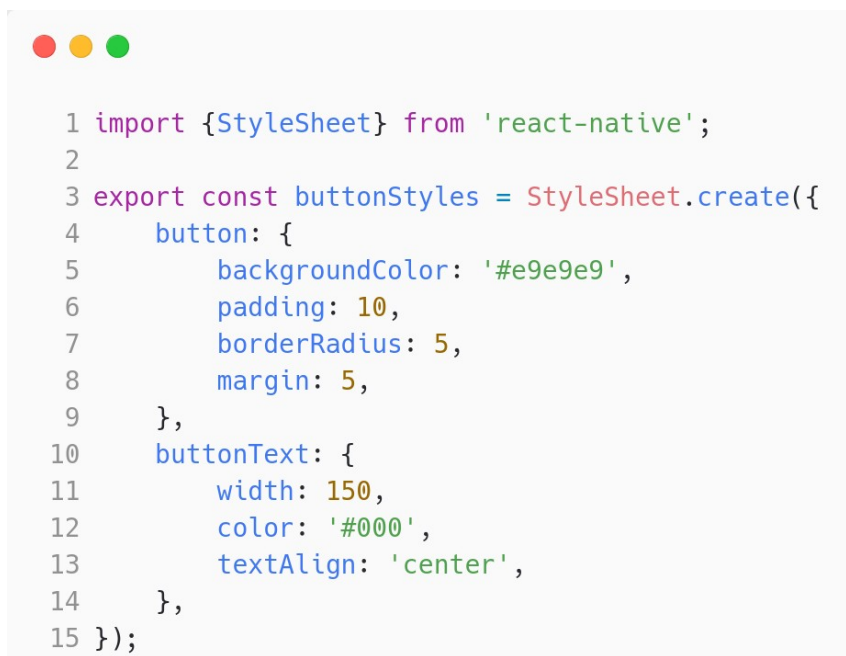
Для стилізації екрана головної сторінки використовується об'єкт стилів styles:



```
1 import {StyleSheet} from "react-native";
2
3 export const styles = StyleSheet.create({
4   container: {
5     flex: 1,
6     backgroundColor: '#fff',
7     alignItems: 'center',
8     justifyContent: 'center',
9   },
10  text: {
11    fontSize: 24
12  }
13 });
```

Рис. 6 – Стилiзацiя екрана головної сторiнки

Для стилізації компонента MyButton використовується об'єкт стилів buttonStyles:



```
1 import {StyleSheet} from 'react-native';
2
3 export const buttonStyles = StyleSheet.create({
4   button: {
5     backgroundColor: '#e9e9e9',
6     padding: 10,
7     borderRadius: 5,
8     margin: 5,
9   },
10  buttonText: {
11    width: 150,
12    color: '#000',
13    textAlign: 'center',
14  },
15 });
```

Рис. 7 – Стилiзацiя компонента MyButton

Запустимо програму з виконанням емулятора:

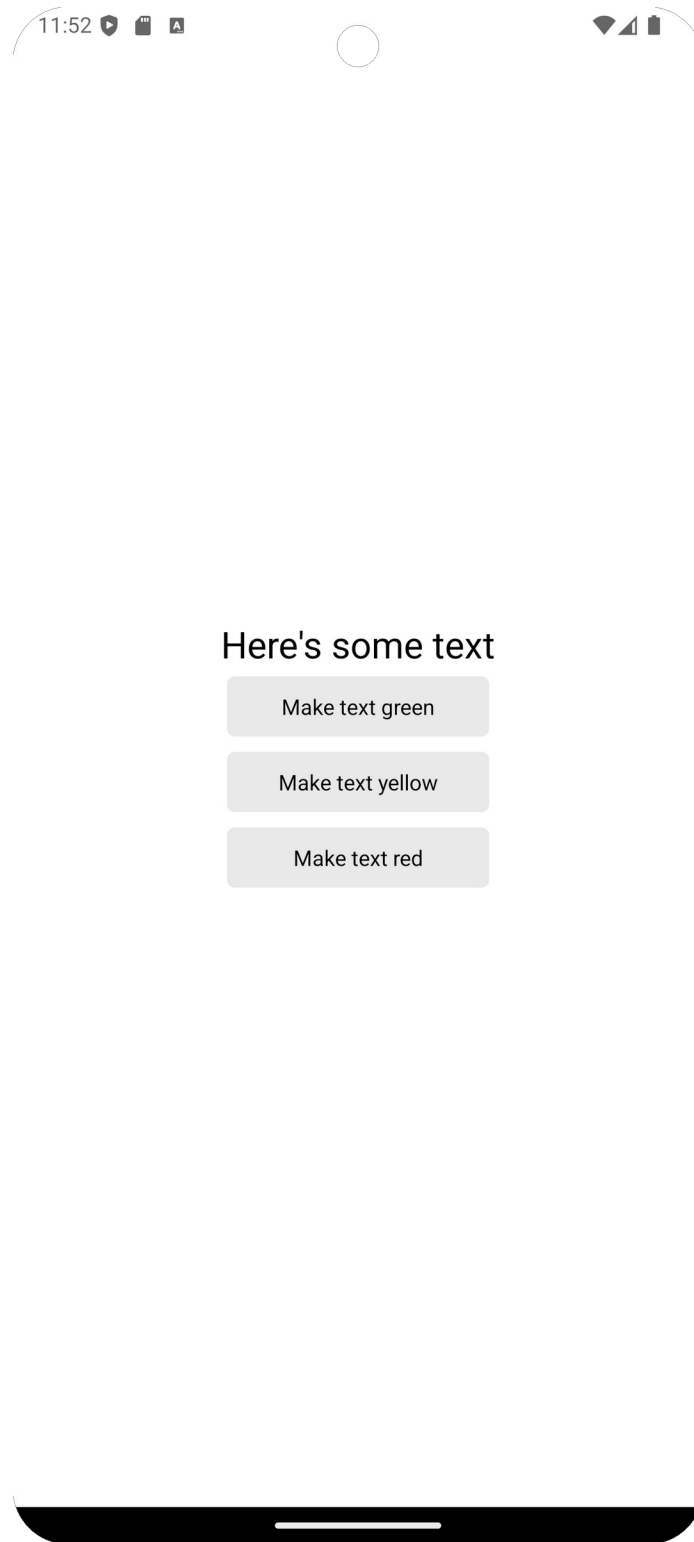


Рис. 8 – Зображення програми під час першого запуску

Можна почати, що на екрані зв'явився текст чорного колюору та три кнопки.

Натиснемо на першу кнопку:

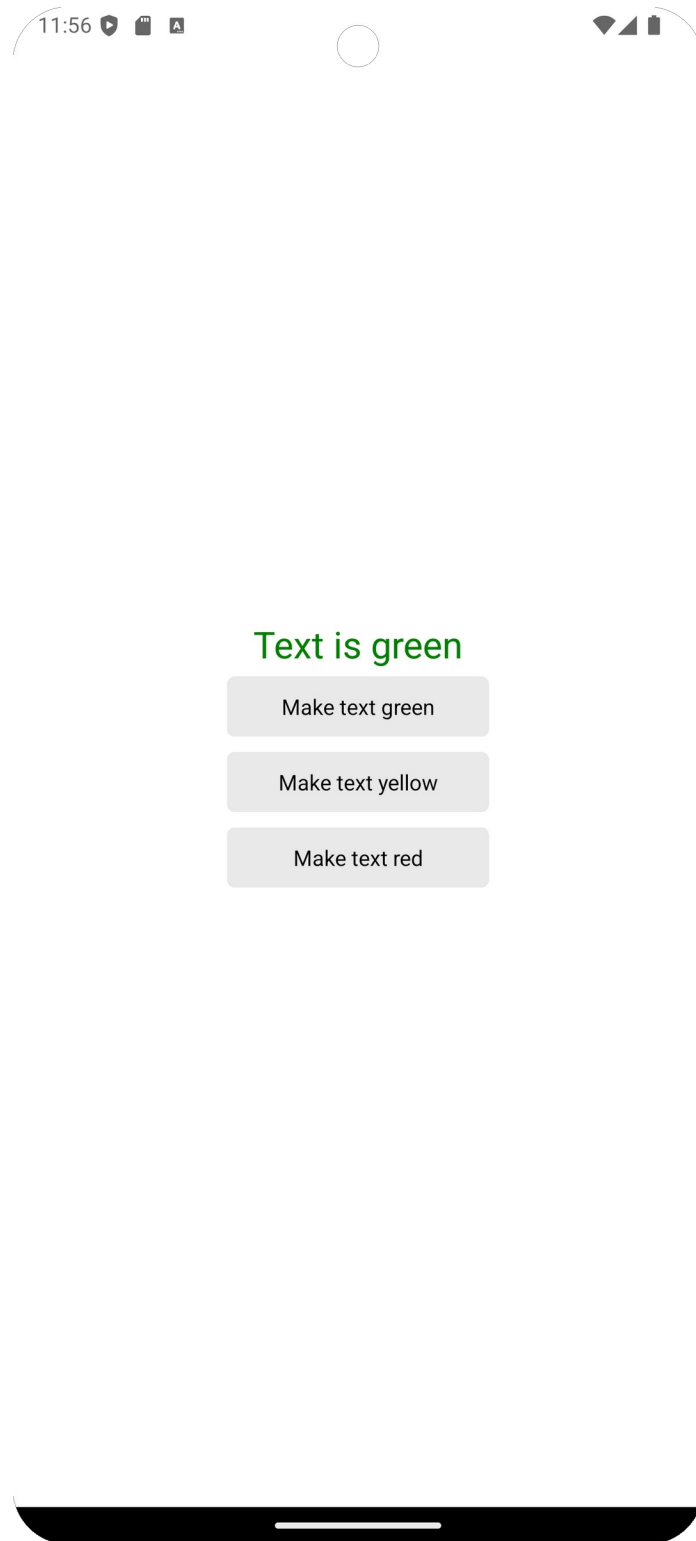


Рис. 9 – Зображення програми після натискання першої кнопки

В результаті натискання, програма змінила текст на “Text is green” та змінила колір компонента Text на зелений.

Натиснемо на другу кнопку:



Рис. 10 – Зображення програми після натискання другої кнопки

В результаті натискання, програма змінила текст на “Text is yellow” та змінила колір компонента Text на жовтий.

Натиснемо на третю кнопку:



Рис. 11 – Зображення програми після натискання третьої кнопки

В результаті натискання, програма змінила текст на “Text is red” та змінила колір компонента Text на червоний.

Висновок: у результаті лабораторної роботи було створено проєкт з використанням React Native та фреймворку Expo, в якому реалізовано три кнопки та текст, що змінюється при натисканні на ці кнопки. Для відображення тексту використано стандартний компонент Text, а для кнопок — власний компонент MyButton. Цей компонент побудований на основі TouchableOpacity, який, на відміну від звичайного Button, дозволяє кастомізувати стилі та обробляти події натискання.