

Звіт  
про виконання завдання з лабораторної роботи №2  
з курсу “Розробка мобільних застосунків”  
студентом Антонюком Олександром Андрійовичем  
тема “Знайомство із стандартними шаблонами в Android Studio”  
в 2024-2025 навчальному році

**Завдання**

1. У наданих інтернет-ресурсах ознайомтесь з типами активностей способом переходу між ними та способами створення обробників подій.
2. Запустіть Android Studio та створіть новий проект використовуючи у якості шаблону Navigation Drawer Activity (перша активність).
3. Ознайомтесь із структурою файлів розмітки та kotlin-файлів проекту.
4. Додайте до вже існуючих фрагментів першої активності новий фрагмент та відповідний йому пункт у навігаційне меню.
5. Додайте до проекту нову активність, використовуючи шаблон Tabbed Activity (друга активність).
6. Переконайтесь, що інформація про створену активність була додана до файлу з маніфестом та ознайомтесь із структурою файлів розмітки та kotlin-файлів нової активності.
7. Додайте до вже існуючих вкладок другої активності нову вкладку.
8. Додайте до навігаційного меню першої активності ще один пункт для переходу на другу активність, використовуючи засоби Navigation.
9. До створеної вкладки другої активності додайте кнопку по натисненні на яку буде здійснено перехід на першу активність, використовуючи Intent.

10. Створіть git-репозиторій для Вашого проекту, запишіть його на будь-який віддалений репозиторій та надайте на нього посилання у якості звіту про виконане завдання.

### Розв'язок

Створимо React Native проєкт з Expo фреймворком використовуючи Navigation (TypeScript) шаблон, що дає можливість створити програму з двома

```
> ls
> npx create-expo-app@latest --template
✓ Choose a template: > Navigation (TypeScript)
✓ What is your app named? ... project
✓ Downloaded and extracted project files.
> npm install
⋮
```

Рис. 1 – Створення проєкту

Запустимо створений проєкт:

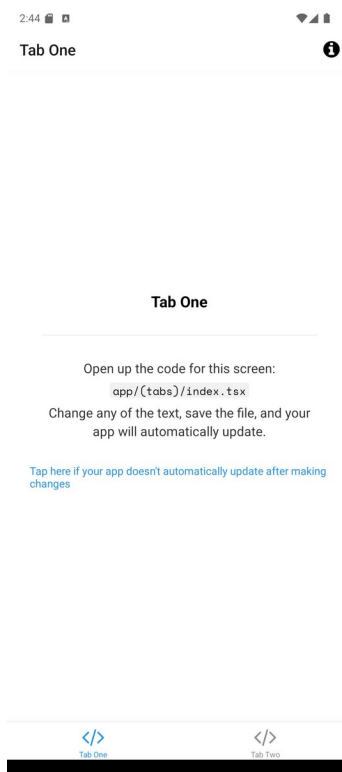


Рис. 2 – Перша вкладка створеного проєкту

Можна побачити, що було згенеровано сторінку, що містить заголовок “Tab One”, певний текст в середині та полосу з навігацією знизу для переходу на першу та другу вкладки.

Натиснемо на другу вкладку з написом “Tab Two”:

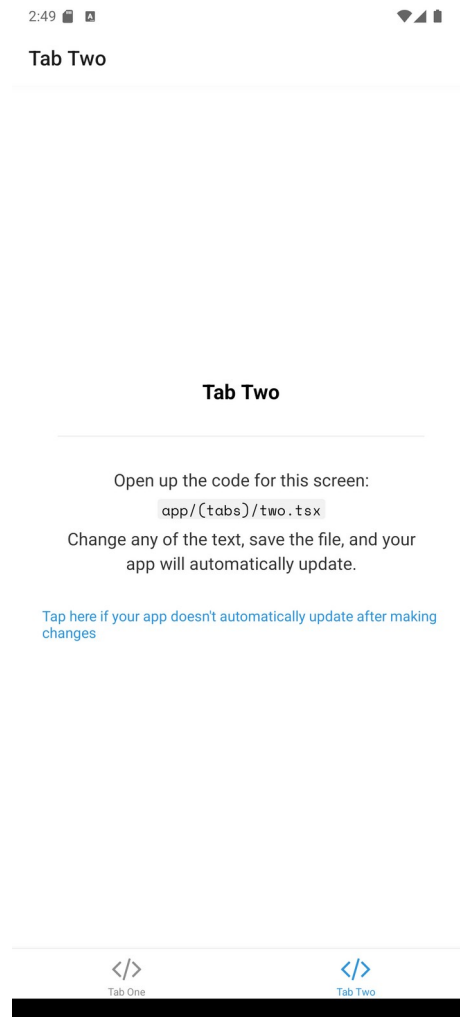


Рис. 3 – Друга вкладка створеного проєкту

Можна побачити, що відбулося перенаправлення з першої вкладки на другу, і контент цієї вкладки змінився. Іконка другої вкладки внизу екрана стала синьою, що вказує на її вибір та успішне перенаправлення.

Тепер створимо третю вкладку. Для цього необхідно створити новий файл three.tsx та стилі в tabThreeStyles.ts (див. Рис. 5):




```

1  const TabThreeScreen = () => {
2    const router = useRouter();
3
4    return (
5      <View style={tabThreeStyles.container}>
6        <Text style={tabThreeStyles.text}>Tab Three</Text>
7        <Button
8          onPress={() => router.push("/(tabs)/")}
9          title={'Go to Tab One'}
10        />
11      </View>
12    );
13  }
14
15  export default TabThreeScreen;

```

Рис. 4 – Третя вкладка

Ця вкладка містить компонент View, що включає текст і кнопку для переходу на першу вкладку, реалізоване за допомогою бібліотеки expo-router.



```

1  import { StyleSheet } from 'react-native';
2
3  export const tabThreeStyles = StyleSheet.create({
4    container: {
5      flex: 1,
6      alignItems: 'center',
7      justifyContent: 'center',
8      flexDirection: 'column',
9    },
10   link: {
11     fontSize: 14,
12     fontWeight: 'medium',
13     marginBottom: 16,
14   },
15   text: {
16     fontSize: 20,
17     fontWeight: 'bold',
18     marginBottom: 16,
19   }
20 });

```

Рис. 5 – Стилзація третьої вкладки

Щоб відобразити третю вкладку в нижній частині екрана, необхідно додати перехід до неї у файлі макету `_layout.tsx`. Для цього слід додати ще один компонент `Tabs.Screen`, вказати ім'я файлу, задати заголовок сторінки та використати згенеровану іконку:

```
1 function TabBarIcon(props: {
2   name: React.ComponentProps<typeof FontAwesome>['name'];
3   color: string;
4 }) {
5   return <FontAwesome size={28} style={{ marginBottom: -3 }} {...props} />;
6 }
7
8 export default function TabLayout() {
9   const colorScheme = useColorScheme();
10
11   return (
12     <Tabs
13       screenOptions={{
14         tabBarActiveTintColor: Colors[colorScheme ?? 'light'].tint,
15         headerShown: useClientOnlyValue(false, true),
16       }}>
17       <Tabs.Screen
18         name="index"
19         options={{
20           title: 'Tab One',
21           tabBarIcon: ({ color }) => <TabBarIcon name="code" color={color} />
22         }}
23       />
24       <Tabs.Screen
25         name="two"
26         options={{
27           title: 'Tab Two',
28           tabBarIcon: ({ color }) => <TabBarIcon name="code" color={color} />,
29         }}
30       />
31       <Tabs.Screen
32         name="three"
33         options={{
34           title: 'Tab Three',
35           tabBarIcon: ({ color }) => <TabBarIcon name="code" color={color} />,
36         }}
37       />
38     </Tabs>
39   );
40 }
```

Рис. 6 – Відредагований макет

В результаті чого, можна побачити третю вкладку на екрані з заголовком та кнопкою, що переходить на першу вкладку.

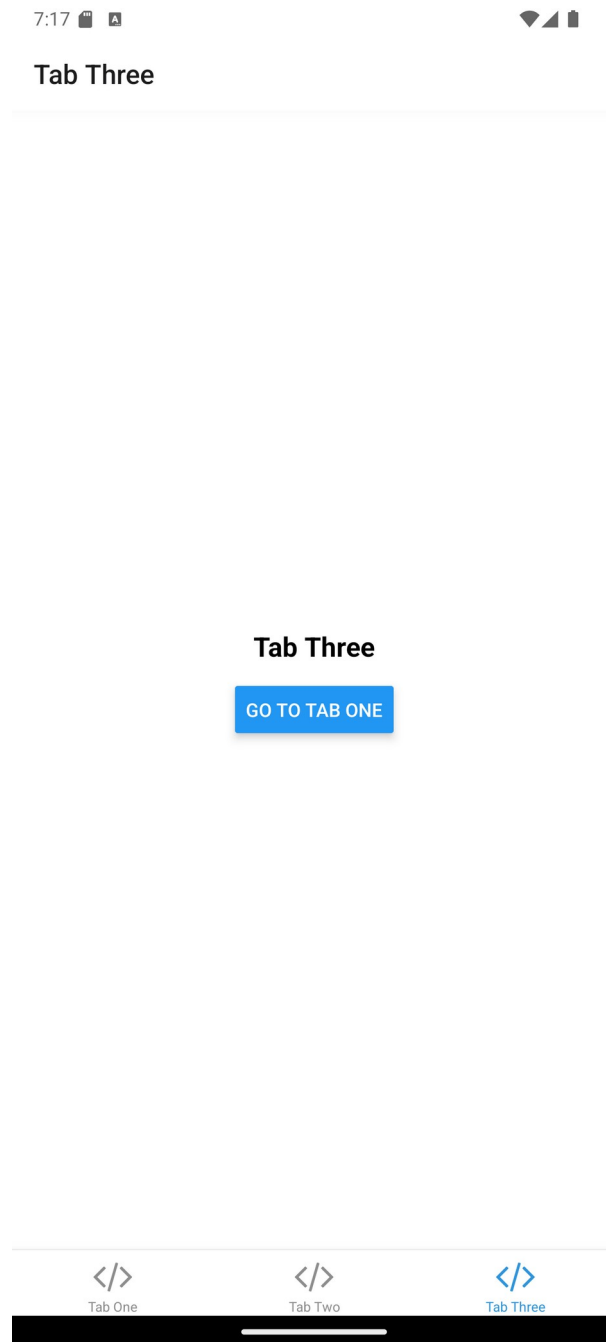


Рис. 7 – Третя вкладка

**Висновок:** у ході виконання роботи було ознайомлено зі стандартними шаблонами React Native та фреймворком Expo. Створено проєкт із трьома

вкладками, для навігації між якими використовується навігаційна панель. На третій вкладці додано кнопку, яка дозволяє перейти на першу вкладку.