

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

ЛАБОРАТОРНАЯ РАБОТА 2
Создание базового проекта

По курсу

Разработка мобильных приложений

Выполнил **Килеев С.И.**

Студент Гр 211-321

Проверил **ФИО**

Захаров И.А.,

Речинский В.А.

Натур В.В.

Москва, 2024

Цель работы: Создать основной макет мобильного приложения с использованием ключевых компонентов выбранной мобильной платформы или с помощью компонентов React Native.

Задачи:

1. Выбор мобильной платформы и настройка среды разработки
Выбрать мобильную платформу и настроить среду разработки для создания проекта.
2. Создание основного макета приложения
Написать базовый код, включающий в себя основные компоненты (например, экраны, кнопки, текстовые поля) для формирования основы мобильного приложения.
3. Комментирование кода и его листинг
Добавить комментарии к коду, объясняющие цель каждого компонента или модуля, а также основные шаги и решения, принятые при создании базового проекта.

Отчет по выполнению:

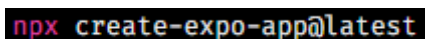
A terminal window with a dark background. The command 'npx create-expo-app@latest' is entered in a light-colored font. The 'npx' part is in a lighter shade, and 'create-expo-app@latest' is in a slightly darker shade.

Рисунок 1 - Создание react-native приложения используя expo

Далее зададим название нашему проекту. В моем случае dev-mobile-plot-vuz.

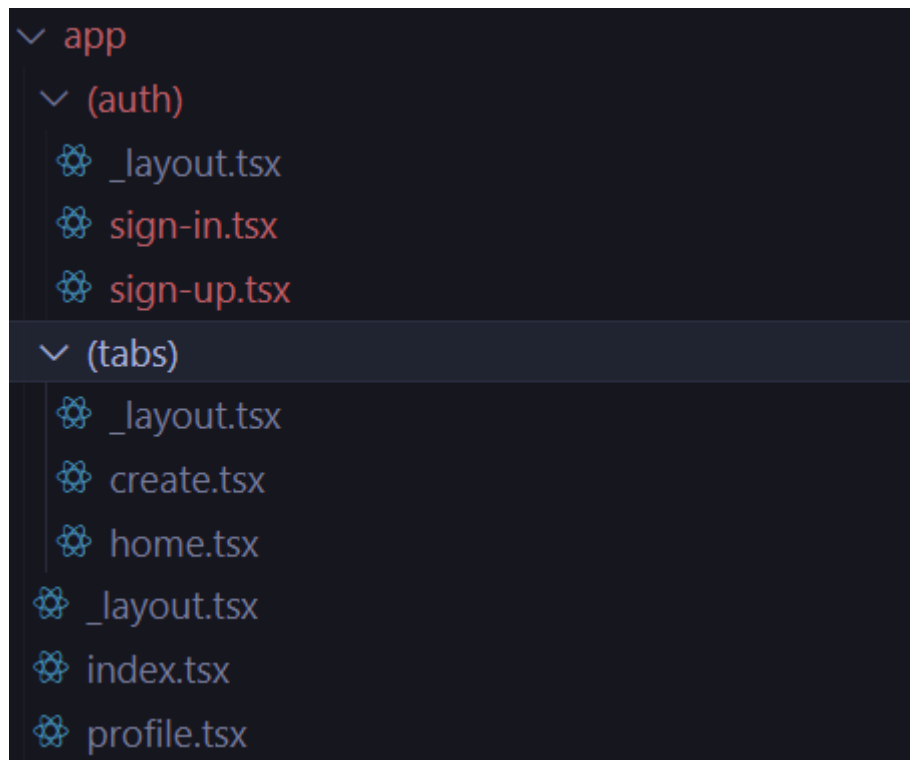


Рисунок 2 - Создадим базовую структуру навигации в проекте

```

1  import { StyleSheet, Text, View } from 'react-native'
2  import React from 'react'
3  import { Slot, Stack } from 'expo-router'
4  import GlobalProvider from '@context/GlobalProvider'
5
6
7  import '../global.css'
8
9  const RootLayout = () => {
10    return (
11      <GlobalProvider>
12        <Stack>
13          <Stack.Screen name='index' options={{headerShown: false}}></Stack.Screen>
14          <Stack.Screen name='(auth)' options={{headerShown: false}}></Stack.Screen>
15          <Stack.Screen name='(tabs)' options={{headerShown: false}}></Stack.Screen>
16        </Stack>
17      </GlobalProvider>
18    )
19  }
20
21  export default RootLayout
22
```

Рисунок 3 - В rootlayout создадим стеки

```

import { StyleSheet, Text, View } from 'react-native'
import { StatusBar } from 'expo-status-bar'
import React from 'react'
import { Link, Redirect } from 'expo-router'
import { useGlobalContext } from '@context/GlobalProvider'

export default function App() {
  const {isLoading, isLoggedIn} = useGlobalContext();

  if(!isLoading && isLoggedIn){
    return <Redirect href="/home"/>
  }

  return (
    <View className='flex-1 items-center justify-center bg-gray-100'>
      <Text className='text-3xl'>Test!</Text>
      <StatusBar style="auto"></StatusBar>
      <Link href="/sign-in" style={{ color: 'blue' }}>Go To sign-in</Link>
    </View>
  )
}

```

Рисунок 4 - Создадим главную страницу в которую пользователь попадает при заходе в приложение

```

return (
  <SafeAreaView className="bg-gray-50 h-full">
    <ScrollView>
      <View className='w-full justify-center px-4 my-6 min-h-[88vh]'>
        <Icons name='home' size={45} />

        <Text className='text-2xl text-black mt-10 font-semibold'>Log in</Text>

        <FormField
          title="Username"
          value={form.username}
          handleChangeText={(e: string) => setForm({ ...form, username: e })}
          otherStyles="mt-7"
          placeholder={undefined}/>

        <FormField
          title="Password"
          value={form.password}
          handleChangeText={(e: string) => setForm({ ...form, password: e })}
          otherStyles="mt-7" placeholder={undefined}/>

        <CustomButton
          title="Sign In"
          handlePress={submit}
          containerStyles="mt-7"
          isLoading={isSubmitting} textStyles={undefined}/>

        <View className='justify-center pt-5 flex-row gap-2'>
          <Text className='text-lg text-black font-normal'>Don't have an account?

          </Text>
          <Link href="/sign-up" className='text-lg font-semibold color-orange-500'>Sign Up</Link>
        </View>

      </View>
    </ScrollView>
  </SafeAreaView>
)
}

```

Рисунок 5 - Создадим экран sign-in для авторизации пользователя

```

return (
  <SafeAreaView className="bg-gray-50 h-full">
    <ScrollView>
      <View className='w-full justify-center px-4 my-6 min-h-[88vh]'\>
        <Ionicons name='home' size={45} />

        <Text className='text-2xl text-black mt-10 font-semibold'\>Sign Up</Text>

        <FormField
          title="Username"
          value={form.username}
          handleChangeText={(e: string) => setForm({ ...form, username: e })}
          otherStyles="mt-7" placeholder={undefined}/>

        <FormField
          title="Email"
          value={form.email}
          handleChangeText={(e: string) => setForm({ ...form, email: e })}
          otherStyles="mt-7"
          keyboardType="email-address" placeholder={undefined}/>

        <FormField
          title="Password"
          value={form.password}
          handleChangeText={(e: string) => setForm({ ...form, password: e })}
          otherStyles="mt-7" placeholder={undefined}/>

        <CustomButton
          title="Sign Up"
          handlePress={submit}
          containerStyles="mt-7"
          isLoading={isSubmitting} textStyles={undefined}/>

        <View className='justify-center pt-5 flex-row gap-2'\>
          <Text className='text-lg text-black font-normal'\>Do you have an account?
            </Text>
          <Link href="/sign-in" className='text-lg font-semibold color-orange-500'\>Sign In</Link>
        </View>

      </View>
    </ScrollView>
  </SafeAreaView>
)

```

Рисунок 6 - Создадим экран sign-up для регистрации пользователя

Ссылка на репозиторий: <https://github.com/Ctfy21/labs-RMP>