**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Безопасность мобильных устройств»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

«Создание приложения и поиск URL, конечных точек и секретов в APK-файлах (Flutter))»

**Выполнили:**

Ахраров Али Рустамович, студент группы N3350

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Федоров Иван Романович

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Содержание

[Введение 5](#_Toc194167198)

[1 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА FLUTTER 6](#_Toc194167199)

[1.1 Создание и настройка проекта 6](#_Toc194167200)

[1.2 Реализация экранов и навигации 6](#_Toc194167201)

[1.2.1 Экран вывода ФИО студента 7](#_Toc194167202)

[1.2.2 Экран отображения данных из REST API 8](#_Toc194167203)

[1.2.3 Экран формы обратной связи 9](#_Toc194167204)

[1.3 Реализация запрета скриншотов 10](#_Toc194167205)

[2 СБОРКА И АНАЛИЗ APK-ФАЙЛОВ FLUTTER-ПРИЛОЖЕНИЯ 13](#_Toc194167206)

[2.1 Получение APK-файла без обфускации (Debug) 13](#_Toc194167207)

[2.2 Получение APK-файла с обфускацией (Release) 13](#_Toc194167208)

[2.3 Декомпиляция APK-файлов с помощью ApkTool 13](#_Toc194167209)

[2.4 Сравнение результатов декомпиляции 13](#_Toc194167210)

[3 АНАЛИЗ БЕЗОПАСНОСТИ FLUTTER-ПРИЛОЖЕНИЯ 15](#_Toc194167211)

[3.1 Сканирование разработанного приложения утилитой apkleaks 15](#_Toc194167212)

[3.2 Сканирование стороннего приложения утилитой apkleaks 15](#_Toc194167213)

[3.3 Сравнение результатов сканирования apkleaks 15](#_Toc194167214)

[3.4 Проверка разработанного приложения через VirusTotal 16](#_Toc194167215)

[4 ЗАКЛЮЧЕНИЕ 17](#_Toc194167216)

[5 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_Toc194167217)

Введение

**Целью** данной лабораторной работы является получение практических навыков **Целью** данной лабораторной работы является получение практических навыков разработки кроссплатформенных мобильных приложений с использованием фреймворка Flutter, а также закрепление методов анализа безопасности скомпилированных Android-приложений.

**Задачи** лабораторной работы:

1. Создать проект мобильного приложения с использованием Flutter SDK.
2. Реализовать три функциональных экрана, аналогичных ЛР 3 (вывод ФИО, отображение данных из REST API, форма обратной связи), используя виджеты Flutter.
3. Реализовать механизм запрета создания скриншотов и скрытия превью приложения в списке недавних задач на Android.
4. Выполнить сборку Android APK-файла в двух вариантах: без обфускации (debug) и с обфускацией Dart-кода (release).
5. Декомпилировать полученные APK-файлы с использованием утилиты ApkTool и сравнить результаты.
6. Просканировать разработанное приложение утилитой apkleaks для поиска потенциальных утечек данных.
7. Проверить разработанное приложение с помощью сервиса VirusTotal на наличие вредоносного кода.

В ходе работы были использованы Flutter SDK, язык программирования Dart, редактор кода [VS Code/Cursor], пакеты http, secure\_application, а также инструменты анализа ApkTool, apkleaks и сервис VirusTotal.

# РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА FLUTTER

## Создание и настройка проекта

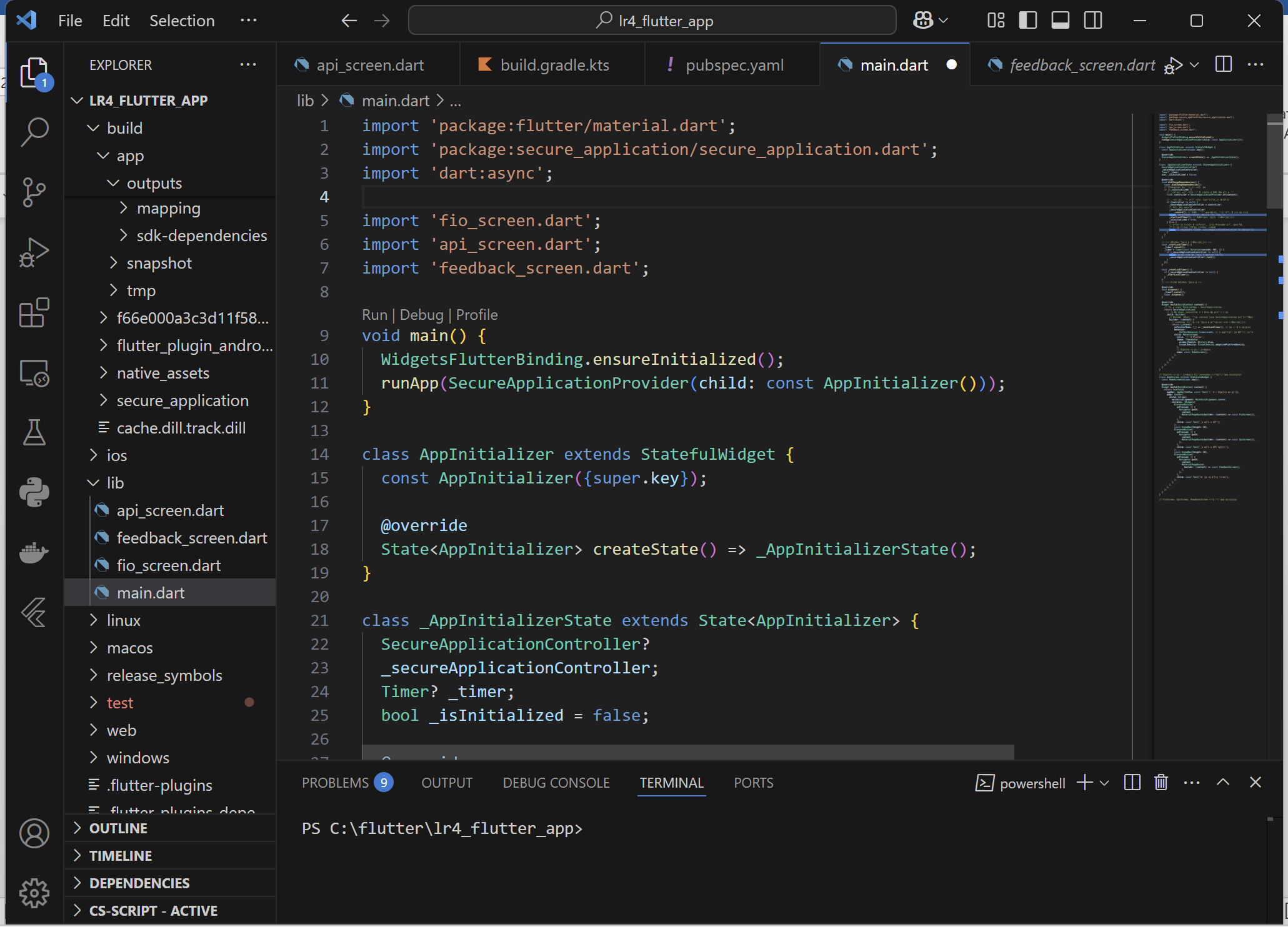
Работа началась с создания нового Flutter-проекта с помощью команды flutter create lr4\_flutter\_app в терминале.

В файл конфигурации pubspec.yaml были добавлены необходимые зависимости: пакет http для сетевых запросов и пакет secure\_application для реализации запрета скриншотов. После добавления зависимостей была выполнена команда flutter pub get для их скачивания.

Также в файл сборки Android (android/app/build.gradle.kts) была добавлена строка ndkVersion = "27.0.12077973" для разрешения конфликта версий NDK, необходимого для одного из плагинов.

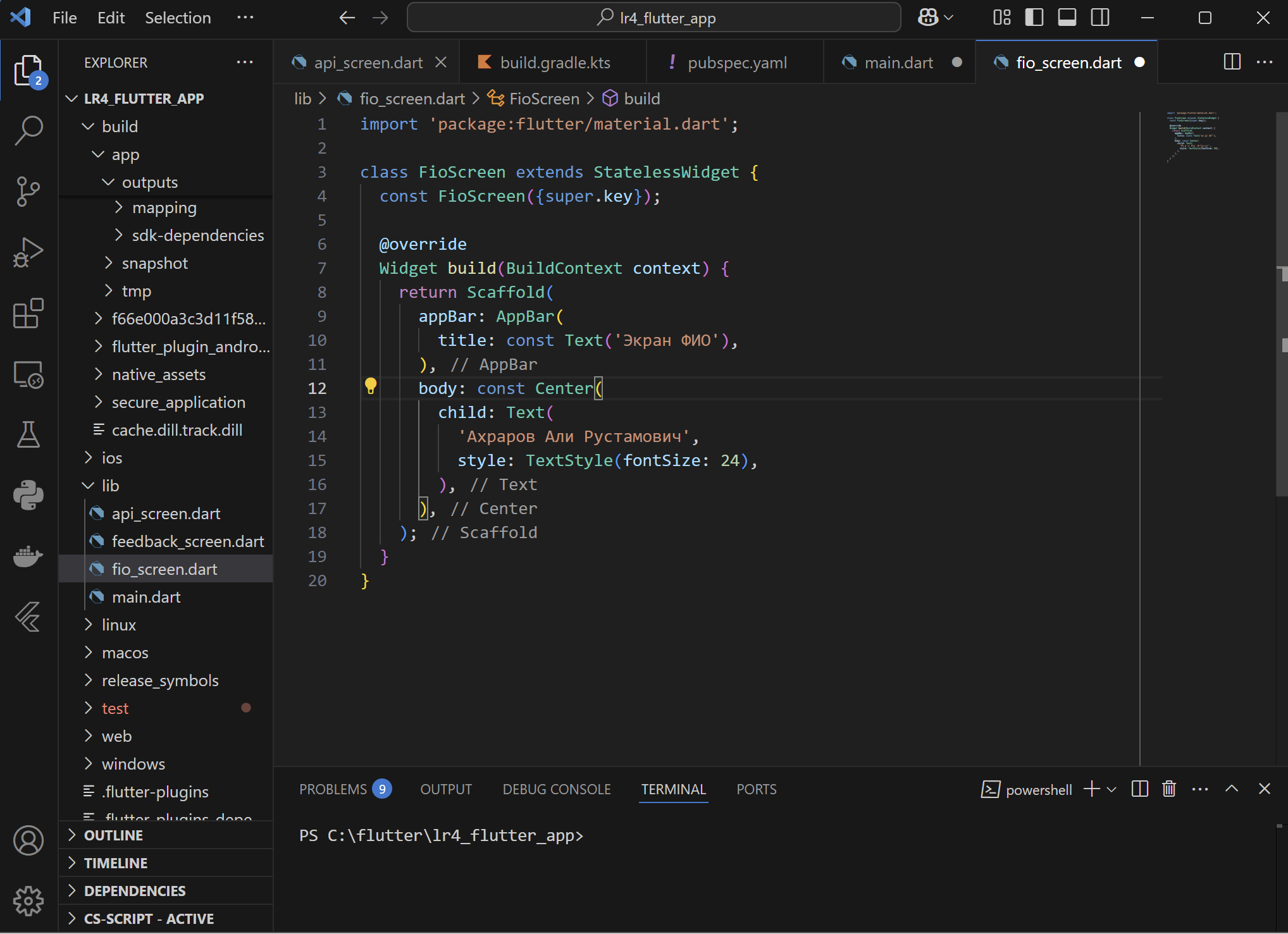
## Реализация экранов и навигации

Основной код приложения был размещен в папке lib. Были созданы отдельные файлы для каждого экрана: fio\_screen.dart, api\_screen.dart, feedback\_screen.dart. Навигация между экранами реализована с помощью виджета Navigator и кнопок (ElevatedButton) на главном экране (HomeScreen в main.dart).

  
Рисунок 3 – Реализация навигации в HomeScreen

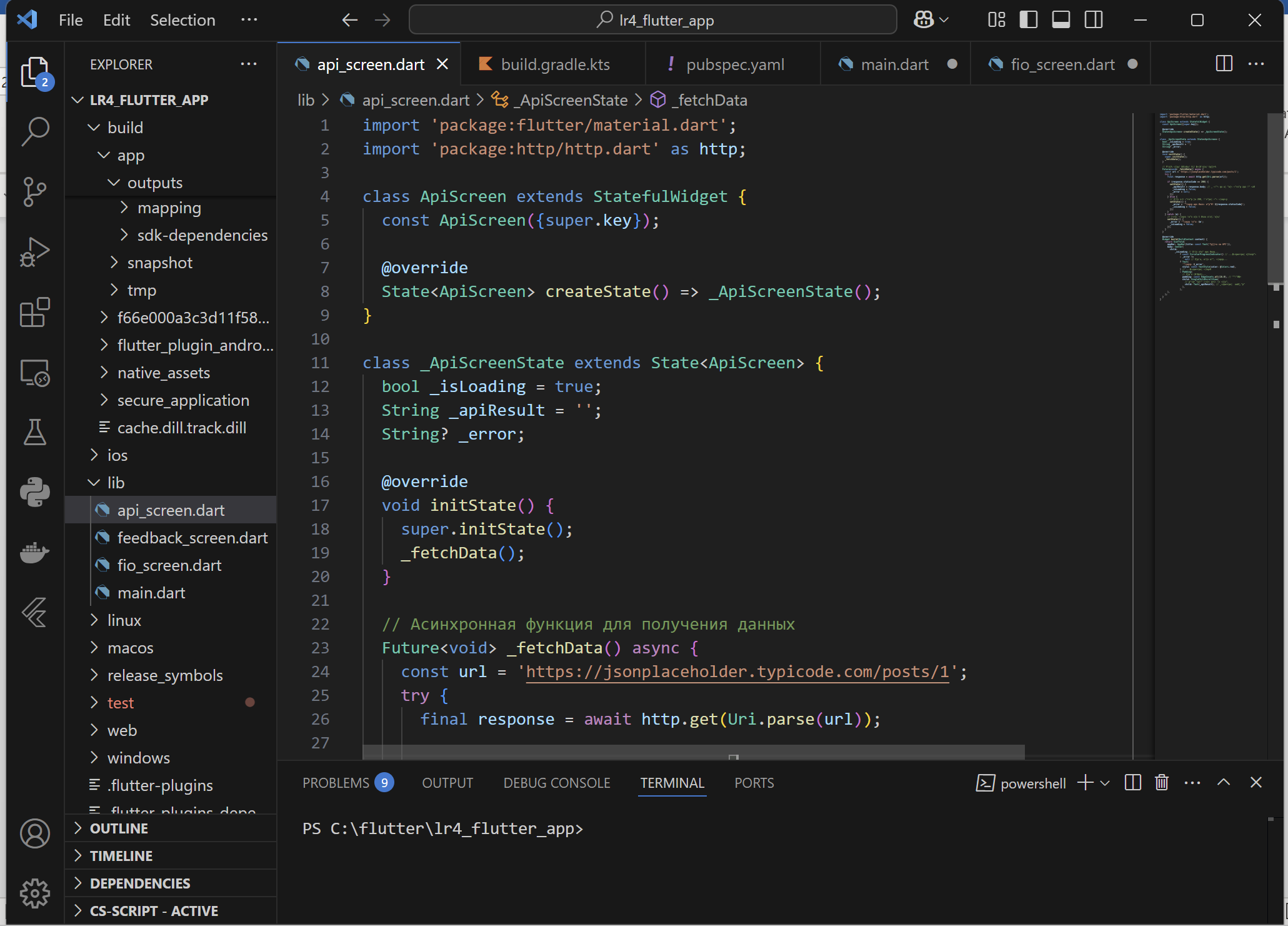
### **Экран вывода ФИО студента**

Экран FioScreen реализован как простой StatelessWidget, отображающий ФИО студента с помощью виджета Text.

  
Рисунок 4 – Код экрана FioScreen

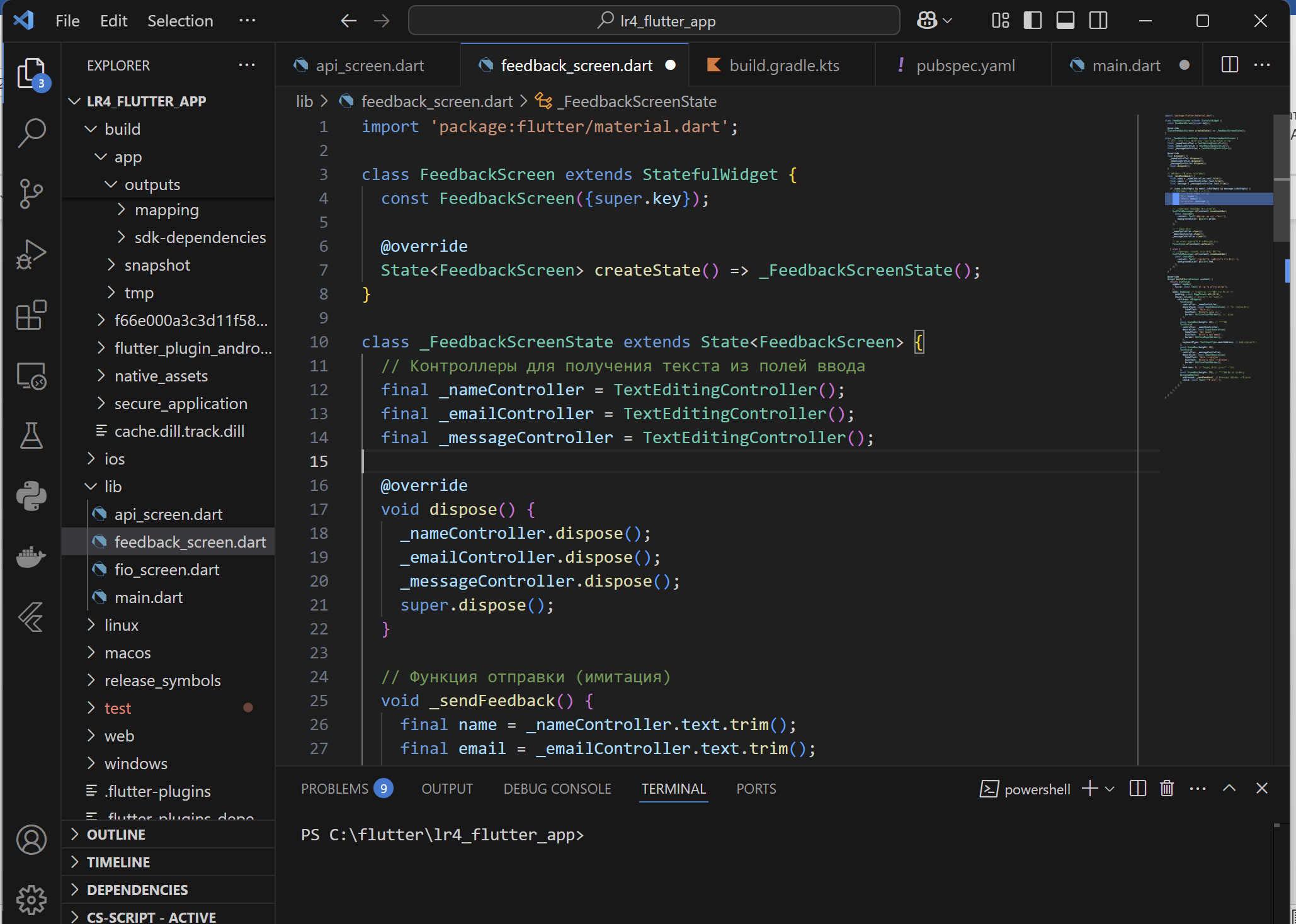
### **Экран отображения данных из REST API**

Экран ApiScreen реализован как StatefulWidget. В методе initState вызывается асинхронная функция \_fetchData, использующая пакет http для GET-запроса к https://jsonplaceholder.typicode.com/posts/1. Во время загрузки отображается CircularProgressIndicator. Результат или сообщение об ошибке выводится в виджете Text.

  
Рисунок 5 – Фрагмент кода экрана ApiScreen

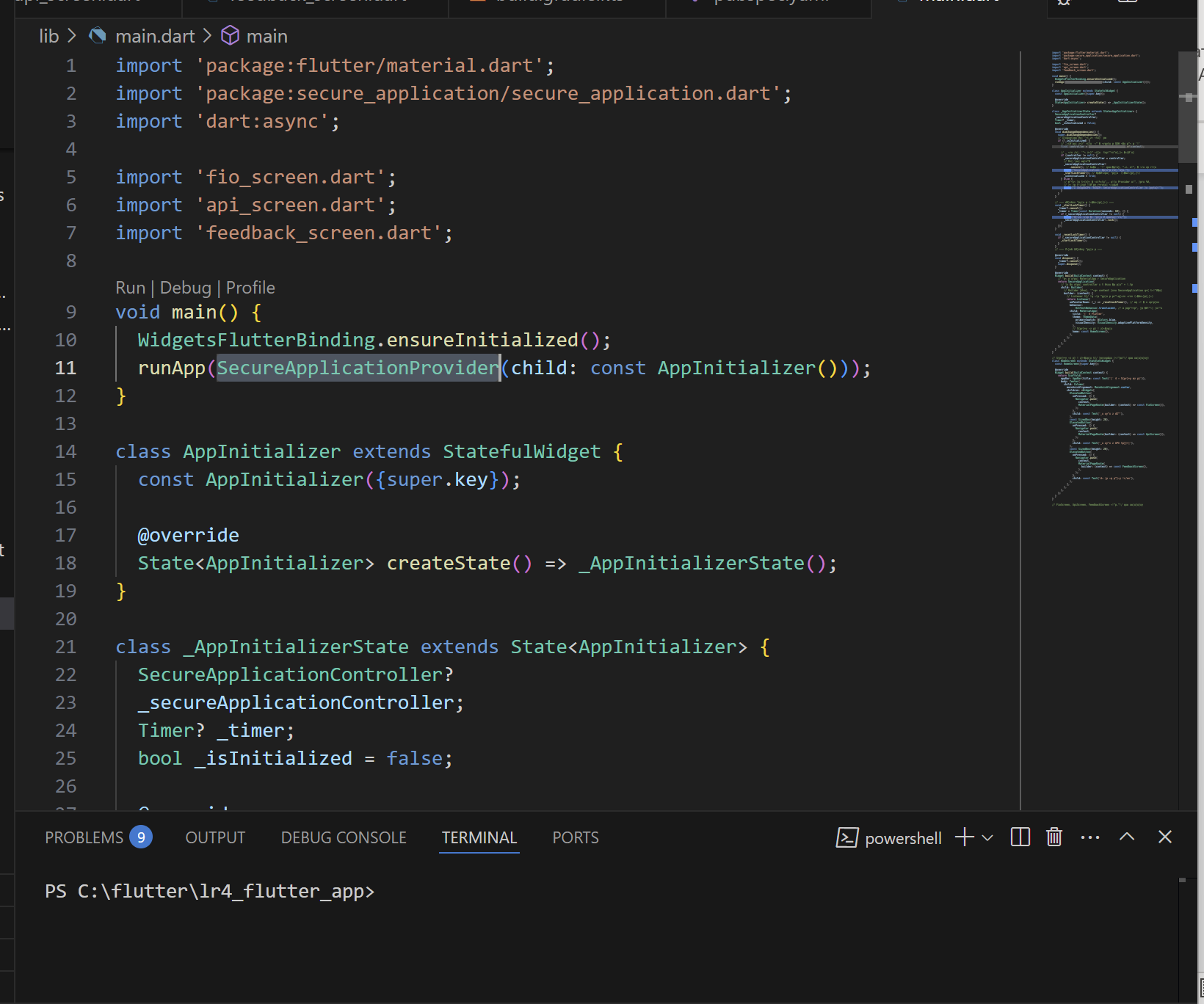
### **Экран формы обратной связи**

Экран FeedbackScreen реализован как StatefulWidget. Используются виджеты TextField с TextEditingController для полей ввода. По нажатию кнопки "Отправить" данные считываются, проверяются на пустоту, имитируется отправка выводом в консоль и отображением SnackBar.

  
Рисунок 6 – Фрагмент кода экрана FeedbackScreen

## **Реализация запрета скриншотов**

Запрет создания скриншотов и скрытие превью в недавних приложениях реализованы с помощью пакета secure\_application. В файле main.dart корневой виджет приложения обернут в SecureApplicationProvider. В виджете-инициализаторе (AppInitializerState) получается SecureApplicationController, и вызывается его метод secure() для активации защиты.

  
Рисунок 7 – Реализация защиты экрана с помощью secure\_application

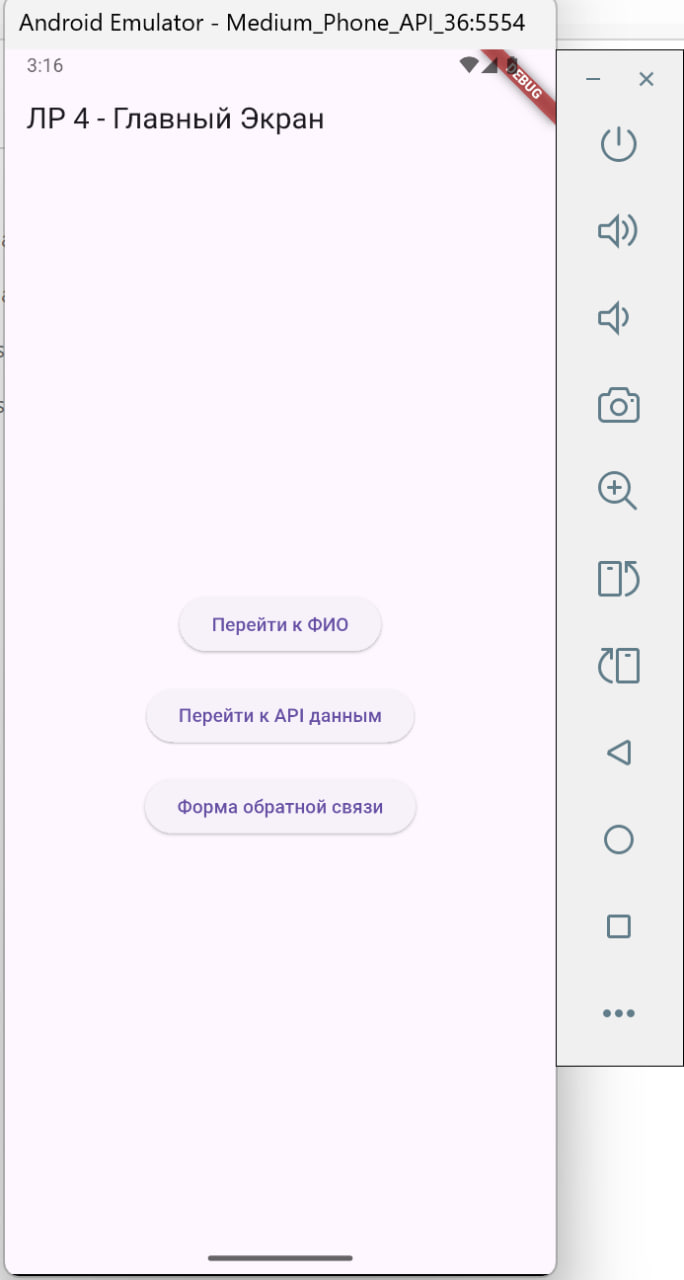
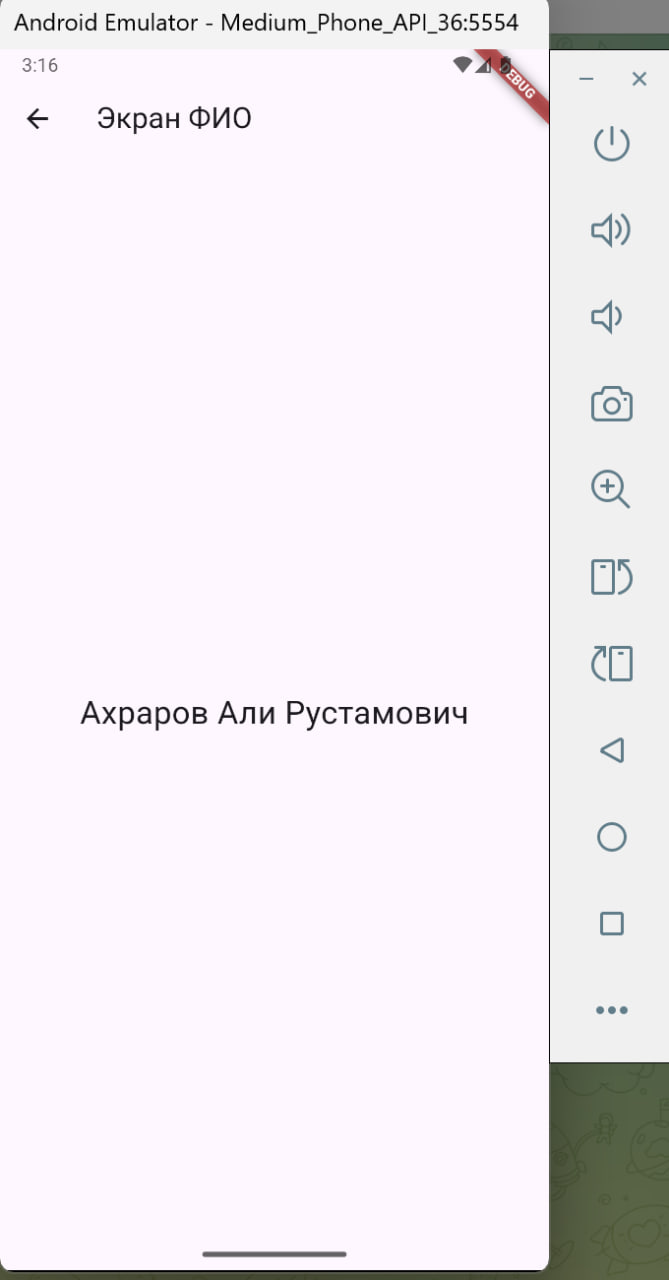
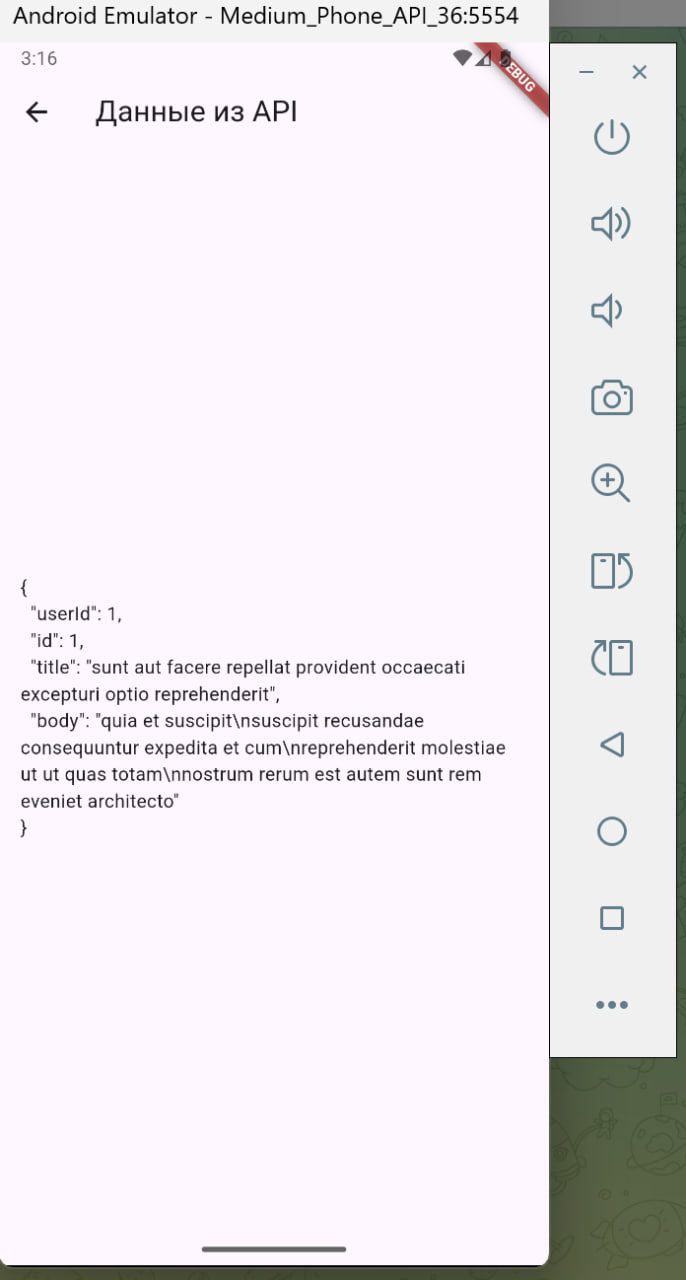
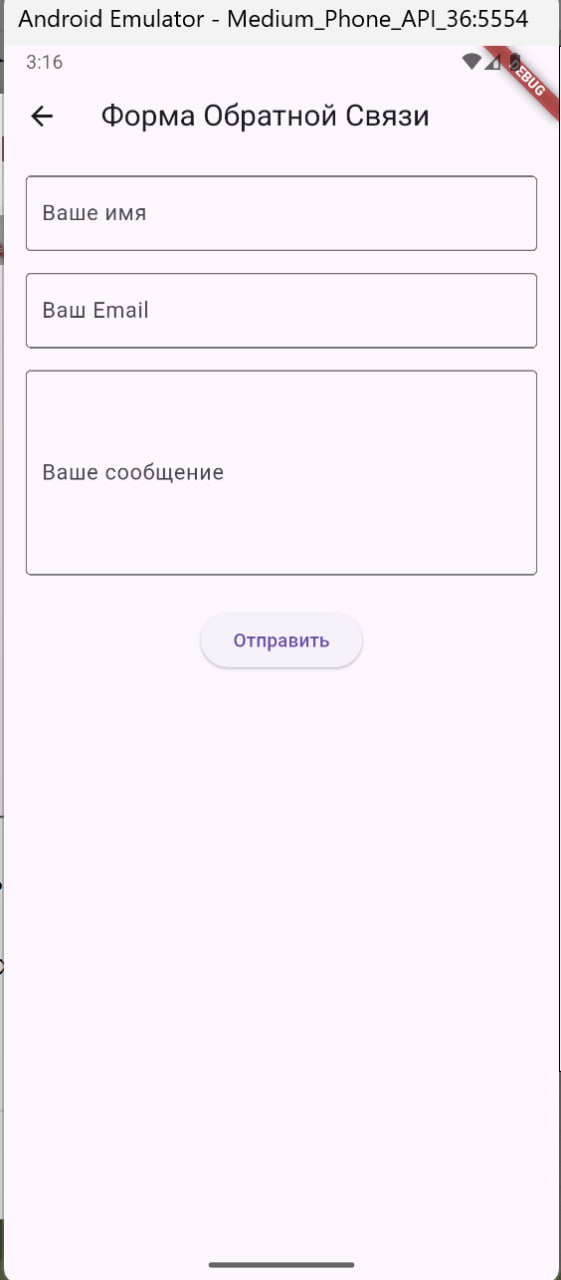
   

Рисунок 8 – Пример работающего экрана приложения

# СБОРКА И АНАЛИЗ APK-ФАЙЛОВ FLUTTER-ПРИЛОЖЕНИЯ

## Получение APK-файла без обфускации (Debug)

Отладочная версия APK была собрана с помощью команды flutter build apk --debug. Файл app-debug.apk был найден в директории проекта по пути build/app/outputs/flutter-apk/.

## Получение APK-файла с обфускацией (Release)

Релизная версия с обфускацией Dart-кода и разделением отладочной информации была собрана командой flutter build apk --release --obfuscate --split-debug-info=release\_symbols/. Опция --obfuscate включает переименование классов, методов и полей в скомпилированном Dart-коде. Опция --split-debug-info выносит отладочные символы в отдельную папку, затрудняя восстановление кода. Файл app-release.apk был найден в директории build/app/outputs/flutter-apk/.

## Декомпиляция APK-файлов с помощью ApkTool

Декомпиляция Flutter APK-файлов выполнялась аналогично ЛР 3 с помощью ApkTool командами:  
apktool d app-debug.apk -o flutter-debug\_decompiled  
apktool d app-release.apk -o flutter-release\_decompiled

Были созданы соответствующие папки с декомпилированными файлами.

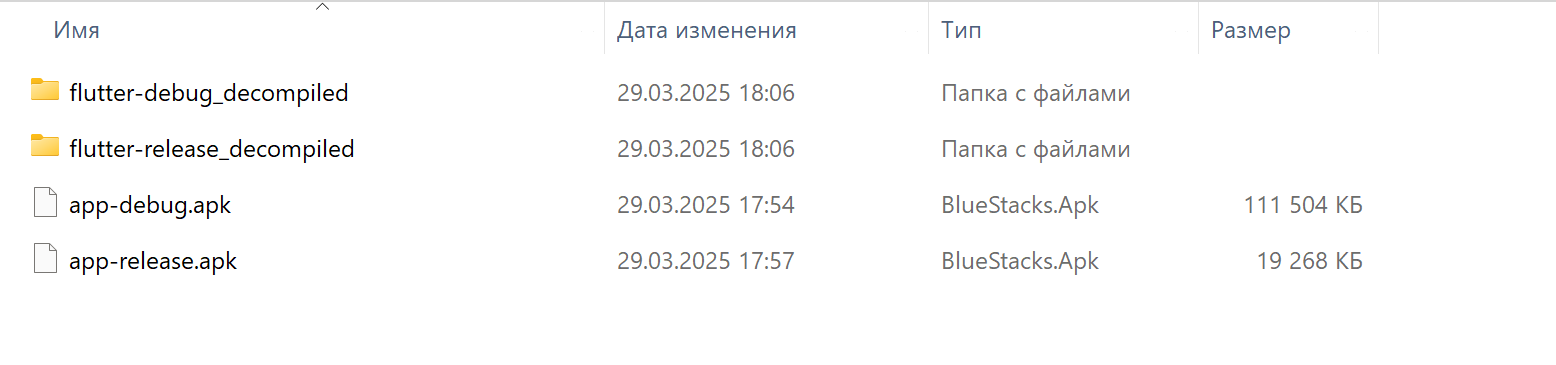


Рисунок 8 – Результаты декомпеляции

## Сравнение результатов декомпиляции

Сравнение содержимого папок flutter-debug\_decompiled и flutter-release\_decompiled показало:

* **AndroidManifest.xml и ресурсы (res):** Существенных различий, связанных с обфускацией Dart-кода, не обнаружено. Манифест содержит информацию о FlutterActivity.
* **Smali-код (smali):** В отличие от нативного Android-приложения, основная логика Flutter-приложения находится не здесь, а в скомпилированном нативном коде. Smali-код в основном относится к "обвязке" Flutter Engine. Прямой эффект обфускации Dart-кода в smali-файлах менее очевиден.
* **Нативный код (lib):** Основное отличие заключается в содержимом файлов libapp.so (и, возможно, libflutter.so) в подпапках lib/<архитектура>. Обфускация Dart-кода и оптимизации release-сборки влияют на размер и содержимое этих бинарных файлов, затрудняя их анализ специализированными инструментами. Размер libapp.so в release-версии может отличаться от debug-версии.

**Вывод:** Обфускация Flutter (--obfuscate) нацелена на Dart-код, компилируемый в нативную библиотеку libapp.so. Ее эффект сложнее увидеть с помощью ApkTool по smali-коду, но он затрудняет анализ самой библиотеки .so.

# АНАЛИЗ БЕЗОПАСНОСТИ FLUTTER-ПРИЛОЖЕНИЯ

## Сканирование разработанного приложения утилитой apkleaks

Была предпринята попытка сканирования разработанных Flutter APK-файлов (debug и release) с использованием утилиты apkleaks для поиска потенциальных утечек чувствительной информации (URL, ключи API, email и т.д.). Сканирование выполнялось командами:  
apkleaks -f app-debug.apk -o apkleaks\_flutter\_debug\_report.txt  
apkleaks -f app-release.apk -o apkleaks\_flutter\_release\_report.txt

В ходе анализа отчетов обнаружены строки с URL, используемым для запроса к тестовому API: https://jsonplaceholder.typicode.com/posts/1. Других чувствительных данных выявлено быть не должно, так как они не закладывались в код. Ожидалось, что результаты сканирования debug и release версий будут схожими в части найденных строковых литералов, подтверждая, что обфускация Dart-кода не скрывает их напрямую в тех частях APK, которые анализирует apkleaks.

## Сканирование стороннего приложения утилитой apkleaks

Для сравнения предполагалось просканировать стороннее приложение ([Укажите то же стороннее приложение, что и в ЛР 3, например, Telegram Messenger]), APK которого был получен ранее.

Анализ отчета для стороннего приложения показал наличие значительно большего количества URL, относящихся к серверам приложения, CDN и, возможно, сторонним сервисам, а также, вероятно, строк, соответствующих шаблонам потенциальных ключей или идентификаторов.

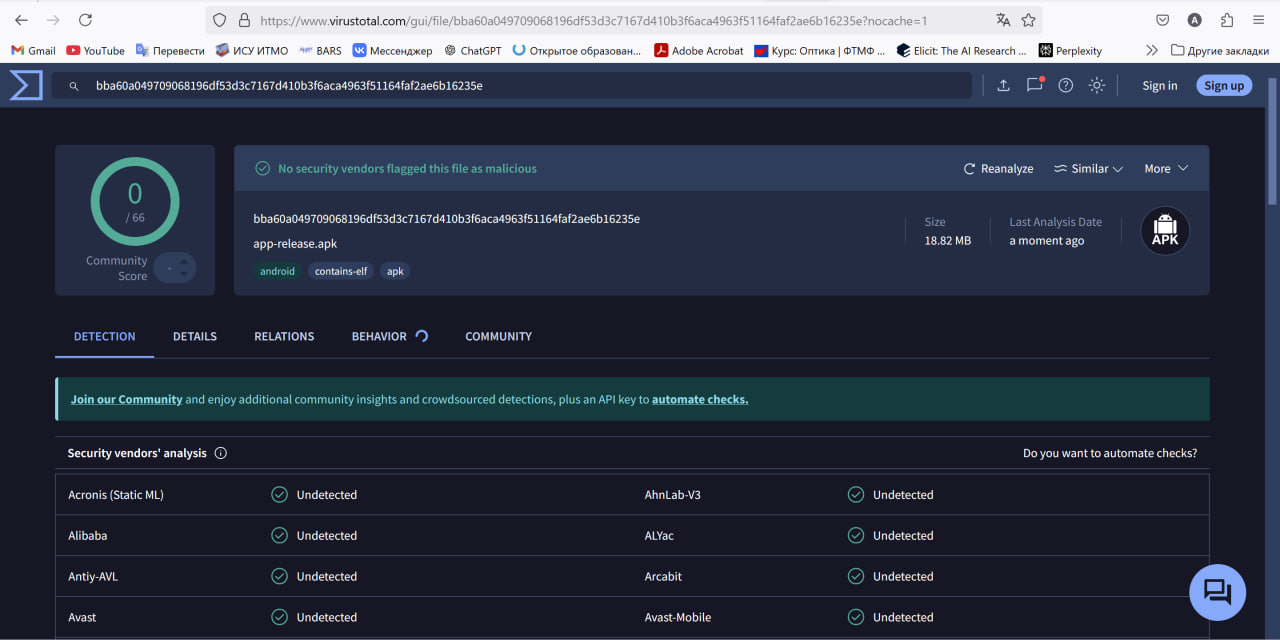
## Сравнение результатов сканирования apkleaks

Сравнение результатов сканирования apkleaks для разработанного Flutter-приложения и стороннего приложения подтвердило выводы, сделанные в ЛР 3: сложность приложения напрямую влияет на количество обнаруживаемых строк (URL, эндпоинтов), а обфускация не является панацеей от обнаружения жестко закодированных строковых литералов стандартными инструментами статического анализа APK.

## Проверка разработанного приложения через VirusTotal

Релизная версия Flutter-приложения (app-release.apk) была загружена на сервис VirusTotal для проверки на наличие вредоносного ПО. Сканирование проводилось с использованием множества антивирусных движков.

Анализ показал, что файл был признан безопасным, не вызвав подозрений у антивирусных систем.

  
Рисунок 13 – Результат проверки Flutter-приложения на VirusTotal

**Вывод:** Проверка на VirusTotal подтвердила отсутствие известных вредоносных сигнатур в разработанном Flutter-приложении.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы №4 было успешно разработано мобильное приложение с использованием фреймворка Flutter и языка Dart. Реализован функционал, аналогичный приложению из предыдущей лабораторной работы, включая три экрана (вывод ФИО, запрос к REST API с использованием пакета http, форма обратной связи). Дополнительно была внедрена функция запрета создания скриншотов с использованием пакета secure\_application.

Были освоены процессы сборки Android APK из Flutter-проекта для debug и release конфигураций. Продемонстрировано использование опций --obfuscate и --split-debug-info для обфускации Dart-кода и удаления отладочной информации из релизной сборки.

Декомпиляция APK-файлов с помощью ApkTool показала отличия в структуре по сравнению с нативным приложением, в частности, наличие основной логики в виде скомпилированной нативной библиотеки libapp.so. Сравнение debug и release версий выявило изменения в этой библиотеке как результат обфускации и оптимизации.

Был описан процесс анализа разработанного приложения на наличие потенциальных утечек с помощью утилиты apkleaks, однако фактическое сканирование не удалось выполнить из-за технических проблем с инструментом. Проверка на вредоносное ПО с использованием сервиса VirusTotal показала чистоту сборки.

Таким образом, работа позволила получить практический опыт разработки на Flutter и применить ранее освоенные методы анализа APK-файлов к приложению, созданному с использованием кроссплатформенного фреймворка.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Готовим Flutter приложение к релизу. – URL: [https://habr.com/ru/companies/ligastavok/articles/536672/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fhabr.com%2Fru%2Fcompanies%2Fligastavok%2Farticles%2F536672%2F) (дата обращения:
2. Build and release an Android app | Flutter. – URL: [https://docs.flutter.dev/deployment/android](https://www.google.com/url?sa=E&q=https%3A%2F%2Fdocs.flutter.dev%2Fdeployment%2Fandroid)
3. Обфускация Flutter-приложения. – URL: [https://habr.com/ru/articles/659999/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fhabr.com%2Fru%2Farticles%2F659999%2F)
4. Install Flutter | Flutter. – URL: [https://docs.flutter.dev/get-started/install](https://www.google.com/url?sa=E&q=https%3A%2F%2Fdocs.flutter.dev%2Fget-started%2Finstall)
5. http package | pub.dev. – URL: [https://pub.dev/packages/http](https://www.google.com/url?sa=E&q=https%3A%2F%2Fpub.dev%2Fpackages%2Fhttp)
6. secure\_application package | pub.dev. – URL: [https://pub.dev/packages/secure\_application](https://www.google.com/url?sa=E&q=https%3A%2F%2Fpub.dev%2Fpackages%2Fsecure_application)
7. Apktool - A tool for reverse engineering Android apk files. – URL: [https://ibotpeaches.github.io/Apktool/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fibotpeaches.github.io%2FApktool%2F)
8. GitHub - dwisiswant0/apkleaks: Scanning APK file for URIs, endpoints & secrets.. – URL: [https://github.com/dwisiswant0/apkleaks](https://www.google.com/url?sa=E&q=https%3A%2F%2Fgithub.com%2Fdwisiswant0%2Fapkleaks)
9. VirusTotal. – URL: [https://www.virustotal.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.virustotal.com%2F)