**Интерактивный дашборд с использованием асинхронного программирования в Python**

Оглавление

[Введение 2](#_Toc185967898)

[1. Описание проекта 3](#_Toc185967899)

[1.1. Цели проекта 3](#_Toc185967900)

[1.2. Задачи проекта 3](#_Toc185967901)

[2. Структура проекта 4](#_Toc185967902)

[2.1. Файлы, включенные в проект 4](#_Toc185967903)

[2.2. Компоненты, включенные в проект 4](#_Toc185967904)

[3. Основной функционал 6](#_Toc185967905)

[3.1. Асинхронный подход (asyncio) 6](#_Toc185967906)

[3.2. Многопоточность (threading) 6](#_Toc185967907)

[3.3. Мультипроцессинг (multiprocessing) 6](#_Toc185967908)

[4. Сравнение библиотек 7](#_Toc185967909)

[4.1. Asyncio 7](#_Toc185967910)

[4.2. Threading 7](#_Toc185967911)

[4.3. Multiprocessing 7](#_Toc185967912)

[Заключение 8](#_Toc185967913)

# Введение

Данный проект представляет собой интерактивный дашборд, разработанный с использованием различных подходов к асинхронному программированию в Python: asyncio, threading и multiprocessing. Основная цель проекта — продемонстрировать возможности каждой из этих библиотек для создания динамических веб-приложений с визуализацией данных.

В проекте реализованы три подхода к обработке данных, каждый из которых имеет свои особенности и применения. Дашборд отображает графики, основанные на случайно сгенерированных данных о доходах, расходах и покупках в различных категориях.

# 1. Описание проекта

## 1.1. Цели проекта

- Обеспечить интерактивность и наглядность данных.

- Демонстрировать использование различных подходов к асинхронному программированию.

- Показать преимущества и недостатки каждого из подходов.

## 1.2. Задачи проекта

- Генерация случайных данных для визуализации.

- Обновление графиков в реальном времени.

- Сравнение производительности различных подходов.

# 

# 2. Структура проекта

## 2.1. Файлы, включенные в проект

- main.py - Главный файл запуска, который объединяет все подходы.

- asyncio\_module.py - Модуль для работы с подходом asyncio.

- threading\_module.py - Модуль для работы с подходом threading.

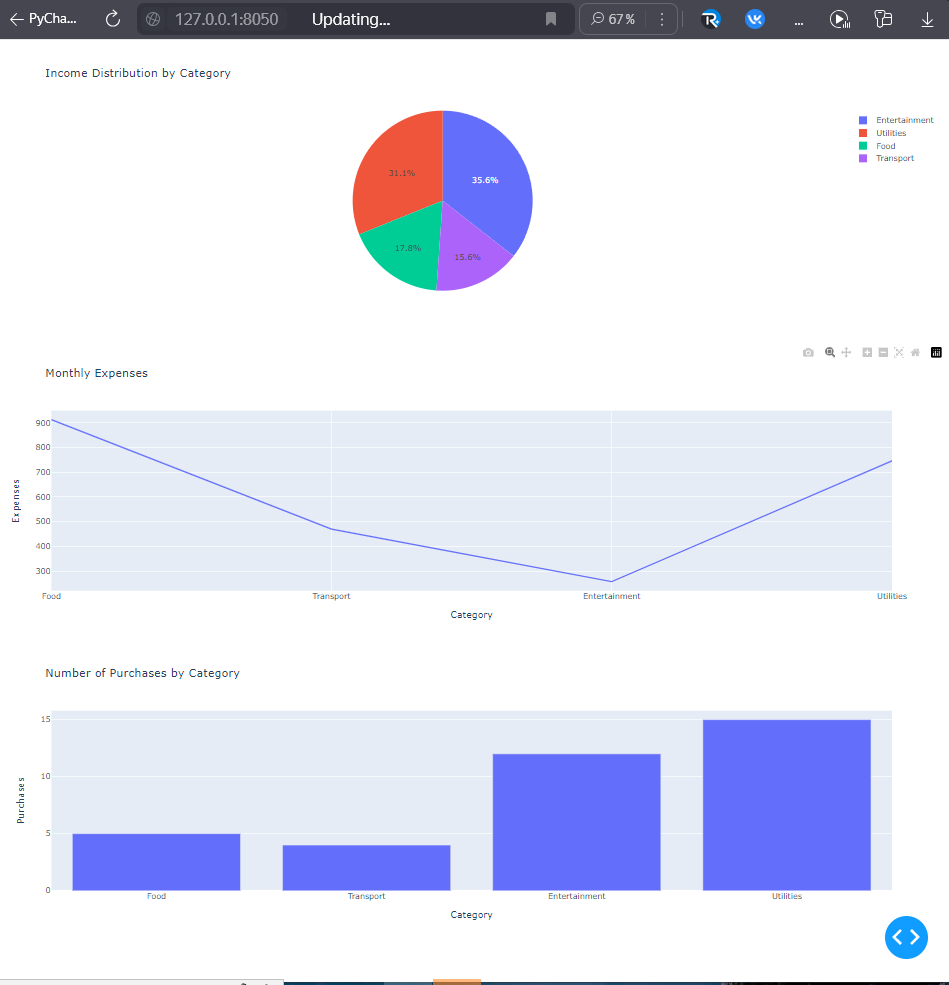
- multiprocessing\_module.py - Модуль для работы с подходом multiprocessing.

- README.md - Документация проекта.

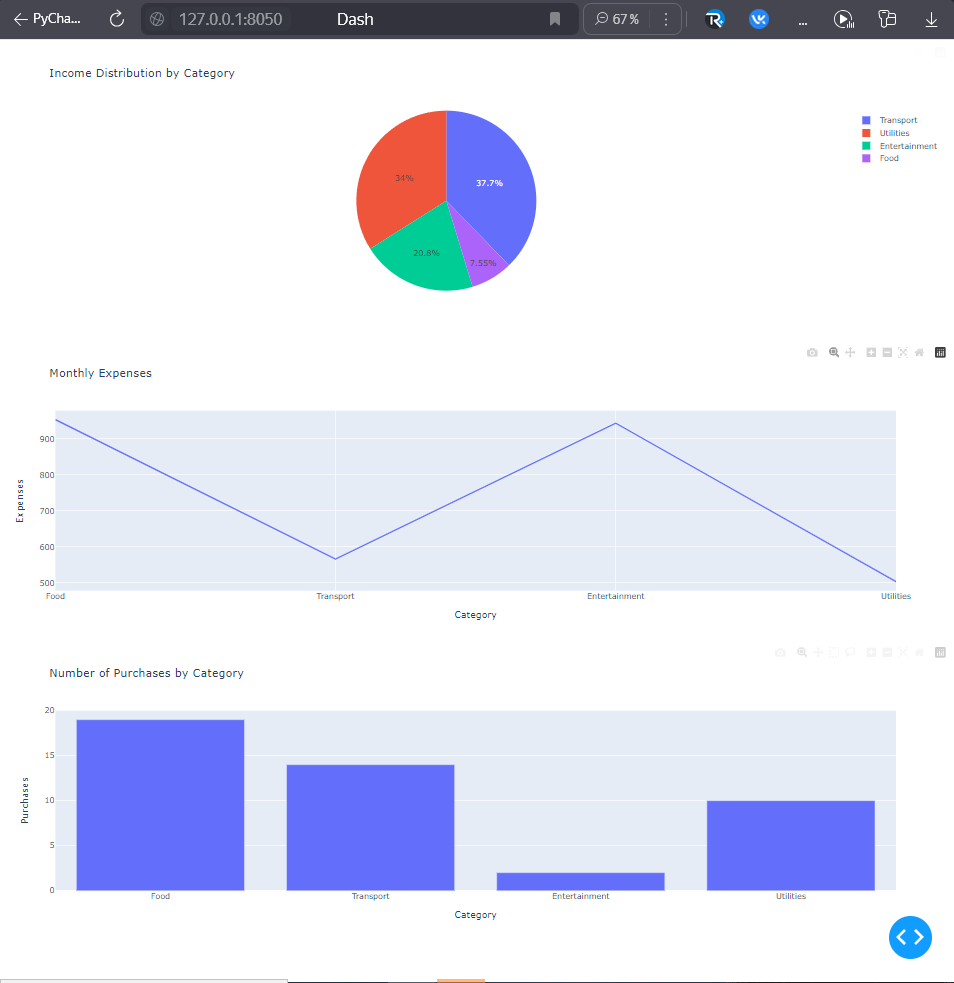
- requirements.txt - Список зависимостей.

## 2.2. Компоненты, включенные в проект

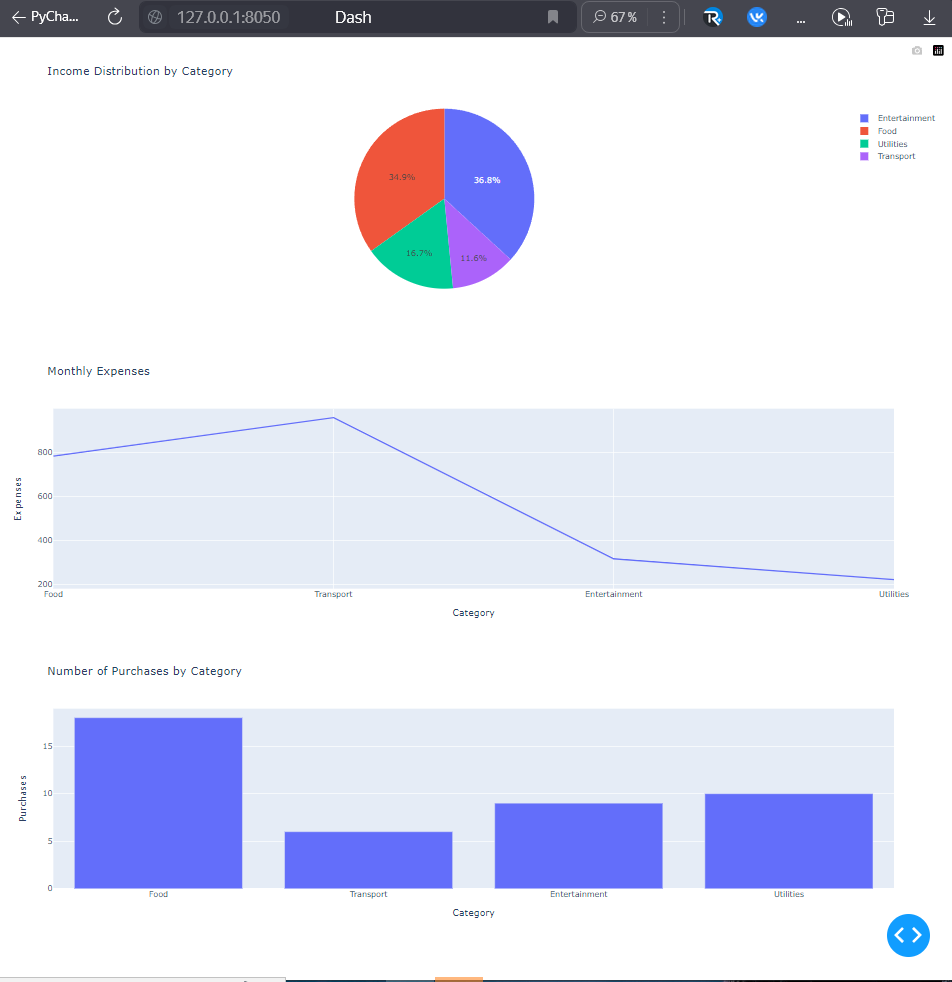
Дашборд на библиотеке asyncio:



Дашборд на библиотеке threading:



Дашборд на библиотеке multiprocessing:



# 3. Основной функционал

## 3.1. Асинхронный подход (asyncio)

- Генерация данных с использованием asyncio.

- Обновление графиков с помощью асинхронных вызовов.

## 3.2. Многопоточность (threading)

- Генерация данных в отдельном потоке.

- Обновление графиков с использованием глобальных переменных.

## 3.3. Мультипроцессинг (multiprocessing)

- Генерация данных в отдельном процессе.

- Использование очереди для передачи данных между процессами.

# 4. Сравнение библиотек

## 4.1. Asyncio

Преимущества:

- Высокая производительность для асинхронных операций.

- Читаемость кода с использованием async и await.

Недостатки:

- Сложность реализации для новичков.

- Неэффективен для CPU-bound задач.

## 4.2. Threading

Преимущества:

- Легкость в использовании и быстрота развертывания.

Недостатки:

- Ограничения GIL (Global Interpreter Lock).

- Возможные проблемы с синхронизацией.

## 4.3. Multiprocessing

Преимущества:

- Масштабируемость с использованием нескольких ядер процессора.

Недостатки:

- Сложность в реализации и управление межпроцессным взаимодействием.

# Заключение

Проект демонстрирует использование различных подходов к асинхронному программированию в Python для создания интерактивного дашборда. Каждый подход имеет свои преимущества и недостатки.