## Personal Budget Tracker

## Introduction

## This project is a web-based personal finance tracker that allows users to monitor and analyze their spending across multiple bank accounts. Users can connect their banking data using TrueLayer, view detailed statistics, and gain insights into their financial habits.

## Problem Statement

## In today's digital world, managing finances across multiple accounts and banks can be confusing and fragmented. Users often lack a centralized system to track spending, identify patterns, or visualize their budget. This project addresses the need for a unified, user-friendly solution for financial awareness and control.

## Objectives

## To centralize financial data from multiple bank accounts.

## To provide interactive dashboards for transaction analytics.

## To automate categorization and visualization of expenses.

## To support secure authentication and data privacy using industry standards.

## Technology Stack

* List the main tools, frameworks, libraries, and languages used.
* **Frontend: HTML5, Tailwind CSS, Chart.js, Jinja2 Templates**
* **Backend: Python, Flask, Flask-Login, SQLAlchemy**
* **Database: PostgreSQL**
* **Others: TrueLayer (Open Banking API), GitHub**

**Installation Instructions**

# 1. Clone the repository

git clone <https://github.com/Tnamor/AdvancedProjectPersonal_Budget_Tracker_13P.git>

# 2. Navigate into the project directory

cd PersonalBudgetTracker

# 3. Create and activate virtual environment

python -m venv venv

source venv/bin/activate # On Windows: venv\Scripts\activate

# 4. Install dependencies

pip install -r requirements.txt

# 5. Set up environment variables

# Create a .env file and add:

# FLASK\_APP=app.py

# FLASK\_ENV=development

# DATABASE\_URL=postgresql://user:password@localhost/dbname

# TRUELAYER\_CLIENT\_ID=...

# TRUELAYER\_CLIENT\_SECRET=...

# 6. Initialize database

flask db init

flask db migrate

flask db upgrade

# 7. Run the application

flask run

**Usage Guide**

## Register and log in to your account.

## Connect your bank using TrueLayer.

## The dashboard displays overall statistics including:

## Spending by category

## Monthly summaries

## Transaction history

## Click on a bank card to view details specific to that account.

**Testing**

This project uses pytest for testing Flask routes, database models, and basic authentication logic.

Setting up tests

## Create a virtual environment and install dev dependencies:

## pip install -r requirements.txt

## pip install pytest pytest-flask

## Ensure a separate test database is configured (e.g., test.db or test PostgreSQL URL).

## The tests/ directory contains:

## test\_auth.py — tests for login/logout/register

## test\_routes.py — tests for dashboard and account views

## test\_models.py — tests for database model behavior

## Running tests:

## pytest

## You can also run tests with coverage:

## pytest --cov=src

## Known Issues / Limitations (Optional)

## Sandbox bank data from TrueLayer is limited.

## Currently supports only banks available in TrueLayer Sandbox.

## Mobile responsiveness may be limited on some views.

## References

* TrueLayer Developer Docs
* Flask Documentation
* Chart.js
* Tailwind CSS

# Team Members

Write the FULL NAMES of each team member, with their Student ID, and the practice group number. Format:

* Tursynbay Roman, 220103062, 14-P
* Bigeldi Bekzat, 220103328, 13-P
* Bigeldi Bekzat, 220103328, 13-P
* Bakhi Abylaikhan, 220103117, 13-P
* Nurbol Zhomart, 220103226, 13-P