READ ME

Credits:

Alexandra González

Victoria Komissarchik

Sofía Machado

Diego Mainardi

Sofía Olenick

Helen Zhang

Table of Contents

About the App

Böndi is a desktop application built with Python and Tkinter that helps users manage personal expenses, savings goals, and shared group expenses. It also includes a streak system that motivates the user to stay financially consistent.

Features Overview

Böndi brings personal finance management and shared expense tracking together in one simple, intuitive desktop app. Its main features include:

1. Savings Goals

- Create multiple goals with target amounts
- Optionally set deadlines
- Add savings over time and watch the progress bar update
- View your full savings history
- All goal data is stored automatically

2. Personal Expense Tracking

- Add, categorize, and manage your daily expenses
- Include optional notes for better organization
- View total spending at a glance
- All expense history is automatically saved and exported to CSV

3. Shared Expense Pods

- Create shared group "pods" where you manage your expenses, with friends, roommates, or travel groups
- Choose pod type: ongoing (roommates) or temporary (trips, events)
- Add members by username
- Split costs equally, by percentage, or by custom amounts
- View complete shared expense history per pod
- All pod data is saved and exported

4. Streak System (Motivation Feature)

- Earn streaks for consistent financial activity
- Actions that increase your streak:
- Adding an expense

- Adding savings
- Adding shared expenses
- Unlock fun badges as you maintain longer streaks

5. User Accounts & Recovery System

- Create secure user accounts
- Recover forgotten usernames using email + recovery word
- Recover forgotten passwords using username + recovery word
- All user data stored safely in JSON and CSV exports

. Clean, Simple Interface

- Tab-based navigation (Goals, Expenses, Shared, Streak)
- Built with Python and Tkinter
- Designed for clarity and ease of use

Project Overview

Böndi is a personal and shared money-management application designed for young adults who want a simple, intuitive, and fair way to track expenses, split bills, and stay financially organized. Unlike many existing finance apps that are either too basic or too overwhelming, Böndi focuses on clarity, fairness, and motivation, making money management something users can actually keep up with.

What the application does

Böndi combines personal budgeting, savings tracking, and shared expense management into one app. Users can manage daily expenses, work toward savings goals, collaborate on shared costs through pods, and stay motivated with a streak-based consistency system

Project Structure

The Böndi project folder contains the main application file, automatically generated data files, and a folder with screenshots used in the documentation.

The project structure should look like this (order does not matter):

```
FinalProject/
|
├── README.md                    # Full project documentation
├── Bondï_app.py                 # Main application file (run this to start the app)
|
├── users_data.json              # Main data storage file (auto-created on first run)
|
├── users.csv                    # Exported list of users
├── expenses.csv                 # Exported personal expenses
├── goals.csv                    # Exported savings goals and contributions
├── pods.csv                     # Exported list of shared pods
├── shared_expenses.csv          # Exported shared expenses for all pods
|
└── photos/                      # Folder containing all screenshots used in the README
    ├── InitialScreen.png
    ├── CreateAccount.png
    ├── ...
```

Installation & Setup

Follow these steps to install and run the Böndi application on your computer.
This project uses Python and Tkinter, so no external frameworks are required.

Download the Project (ZIP)

Download the Böndi project as a ZIP file and extract it on your computer.
After extracting, you will see the main project folder named FinalProject/.
Open this folder in PyCharm in the next steps.

Prerequisites

Before running the app, make sure you have:

Step 1: Install Python

Make sure you have Python 3.8 or higher installed.

Download Python here:
https://www.python.org/downloads/

Step 2: Open the Project in PyCharm

1. Open PyCharm
2. Click File FinalProject → Open

3. Select your project folder (the folder containing bondi_app.py)
4. Click OK

Step 3: Set the Python Interpreter in PyCharm

1. Go to File → Settings
2. Navigate to: Project: FinalProject → Python Interpreter
3. Choose a valid Python 3.x interpreter
4. Click Apply
5. This allows PyCharm to run your Python code correctly.

Step 4: Make Sure Tkinter Is Installed

Tkinter usually comes with Python.

To verify: In PyCharm, open the built-in terminal

Run:

```
python -m tkinter
```

If a small Tk window opens, you're good.
If not, install Tkinter depending on your OS:

Windows:

Tkinter is included automatically.

macOS:

```
brew install python-tk
```

Ubuntu/Debian:

```
sudo apt-get install python3-tk
```

Step 5: Run the Application

In PyCharm:

1. Open Bondï_app.py
2. Right-click anywhere in the file
3. Select Run 'bondi_app'

OR use the built-in terminal:

```
python Bondï_app.py
```

Step 6: Automatic File Creation

When the app runs for the first time, it automatically creates:

- users_data.json – main storage file
- All CSV exports

You do not need to create these manually.

Step 7: Start Using Böndi

You can now:

- Create an account
- Log in
- Add goals, expenses, and shared pods
- Split expenses
- Track your streak
- View your data in CSV files
- Everything saves automatically as you use the app.

Step-by-Step Usage Instructions

Each step includes short explanations of the main features of the application, helping to understand how Böndi works in practice.

Create an Account

When the app opens, you will see a login screen.

Initial screen:

To create a new account:

- Click the *Create account* tab
- Enter:
    - Full name
    - Email
    - Username
    - Password (twice)
    - Recovery word (used for password reset)
    - Click Create account
- The account will be stored in users_data.json
- CSV files will be updated automatically

Sign in:
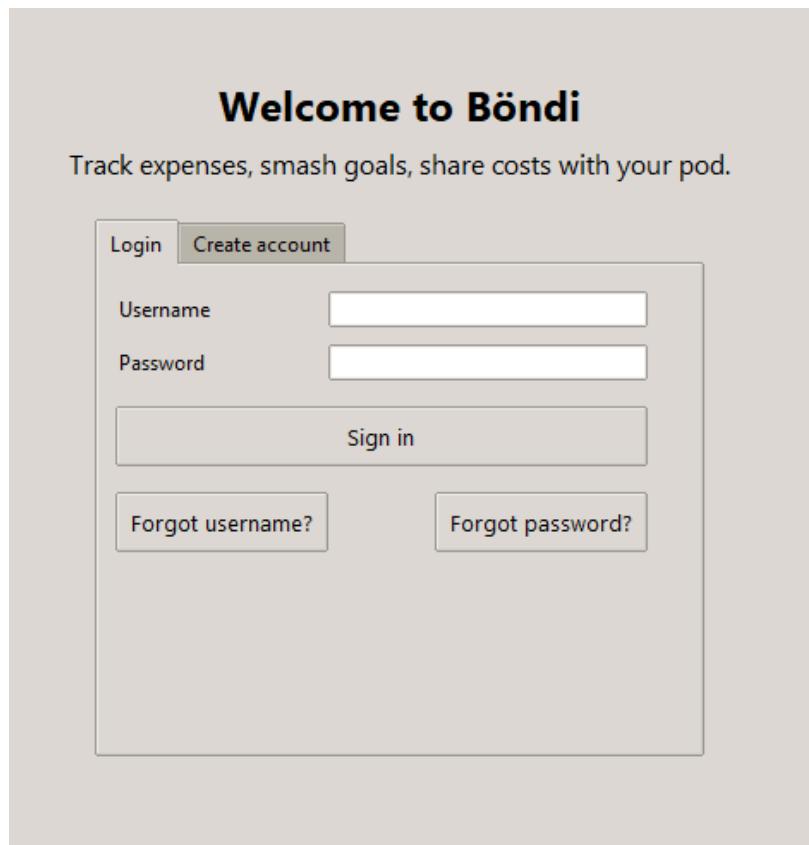
Log In:

Enter your username and password, then click Sign in.

Forget your credentials:

- Forgot username? → recover using email + recovery word
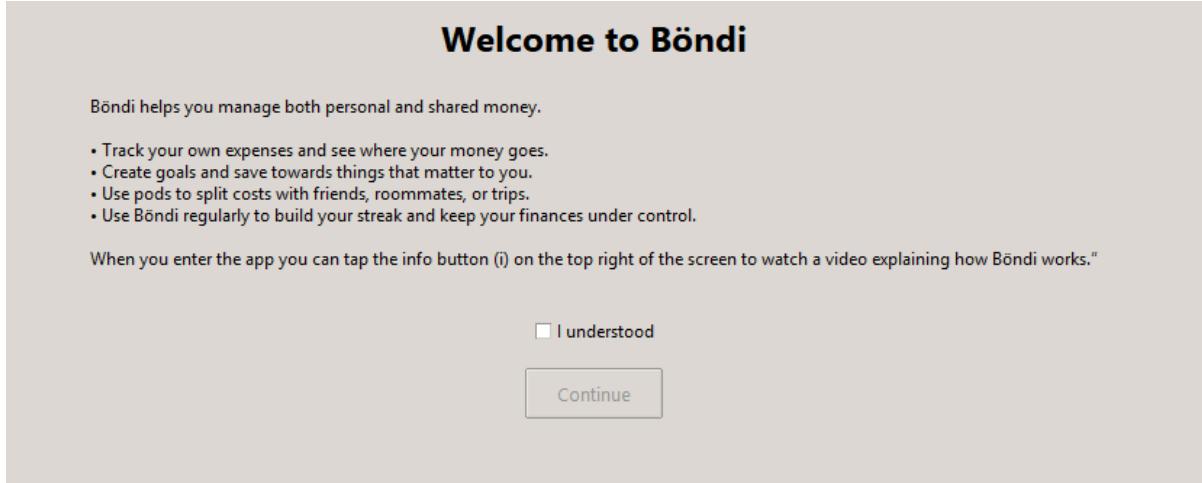
- Forgot password? → recover using username + the recovery word

## Information Screen:

After logging in for the first time, you will see a short introduction page explaining Böndi's features.

Click the check box *I understood*, then *Continue*.
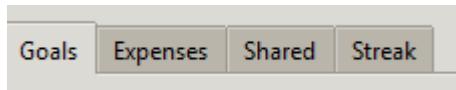
Welcome to Böndi

Böndi helps you manage both personal and shared money.

• Track your own expenses and see where your money goes.
• Create goals and save towards things that matter to you.
• Use pods to split costs with friends, roommates, or trips.
• Use Böndi regularly to build your streak and keep your finances under control.

When you enter the app you can tap the info button (i) on the top right of the screen to watch a video explaining how Böndi works."

☐ I understood

Continue

Using the Main Tabs:

Once inside the app, you will see four main tabs:

- ✔ Goals
- ✔ Expenses
- ✔ Shared
- ✔ Streak

| Goals | Expenses | Shared | Streak |

Each tab has its own functionality, explained below.

Goals Tab - Save Toward Goals

Create a new goal:

- Name Goal -> The goal you want to reach
- Target amount -> The money you need for it
- Optional deadline -> the end date for you to accumulate the amount
- Click the button: "Create Goal" to see it

EXAMPLE OF A CREATED GOAL:

Add savings to an existing goal:

- Select in the *Goal* slide bar what goal you are selecting
- Put in the *Amount* the money you want to add to achieve this goal
- Click the button *Add saving* to update the progress bar

Your savings history is also exported to CSV.

EXAMPLE OF ADD SAVING TO GOAL:
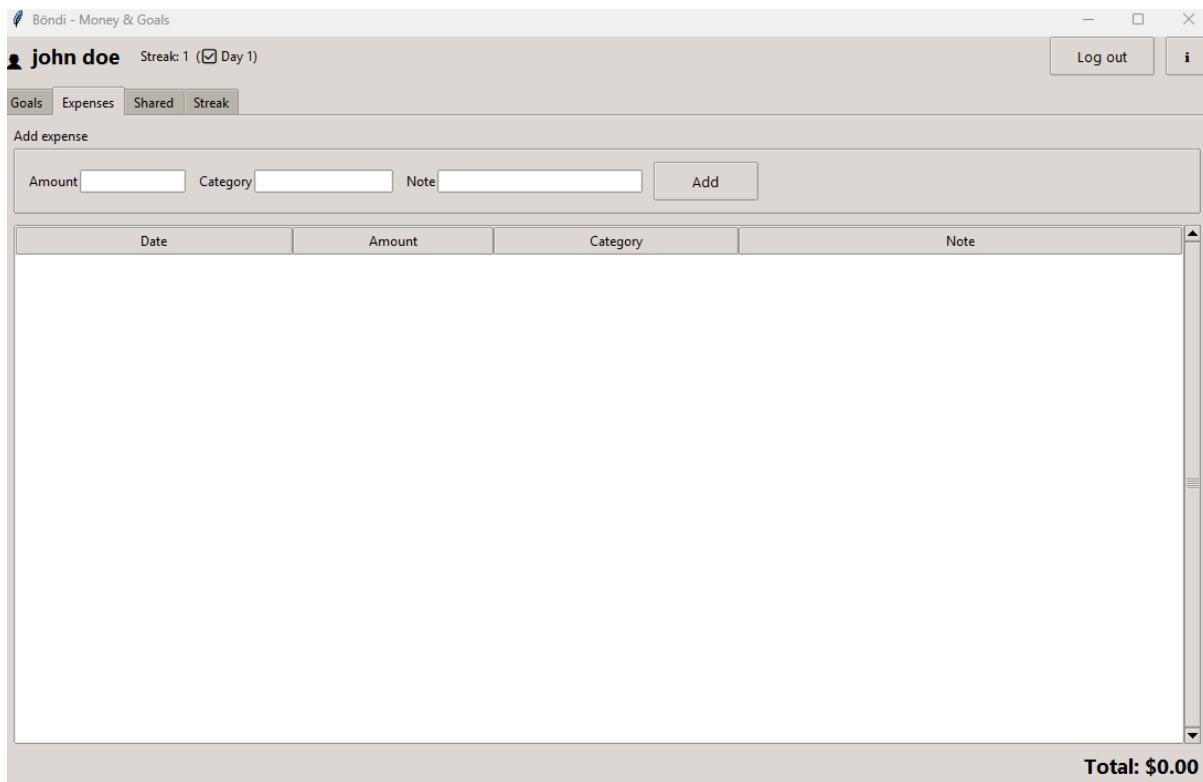
Expenses Tab - Track Personal Expenses

Here you can:

- Enter an amount
- Choose a category (e.g. transport, restaurant, shopping)
- Add a note
- Save the expense

The bottom of the tab shows your total spending.
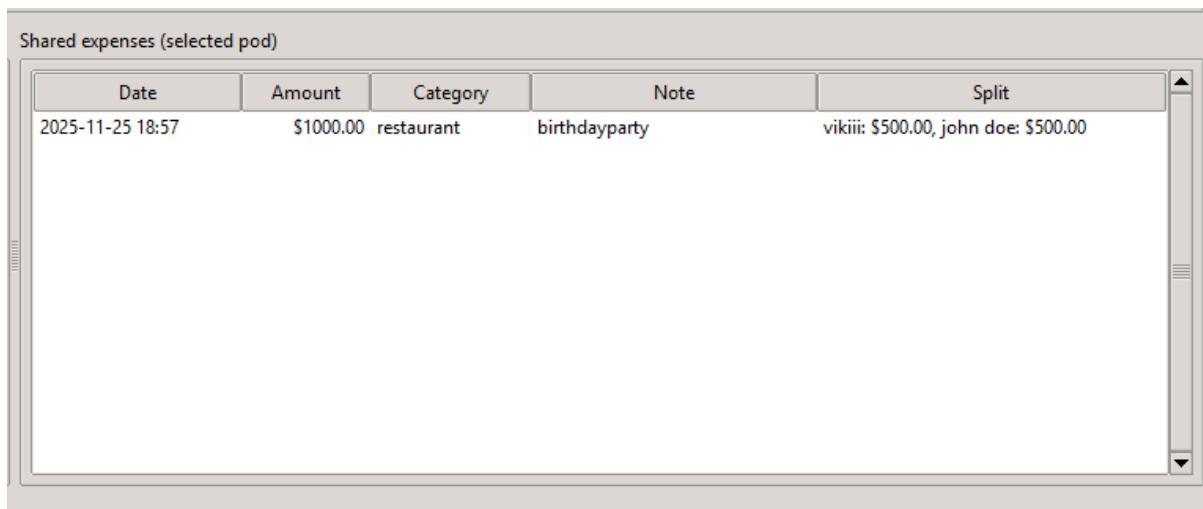
EXAMPLE OF ADD EXPENSE:
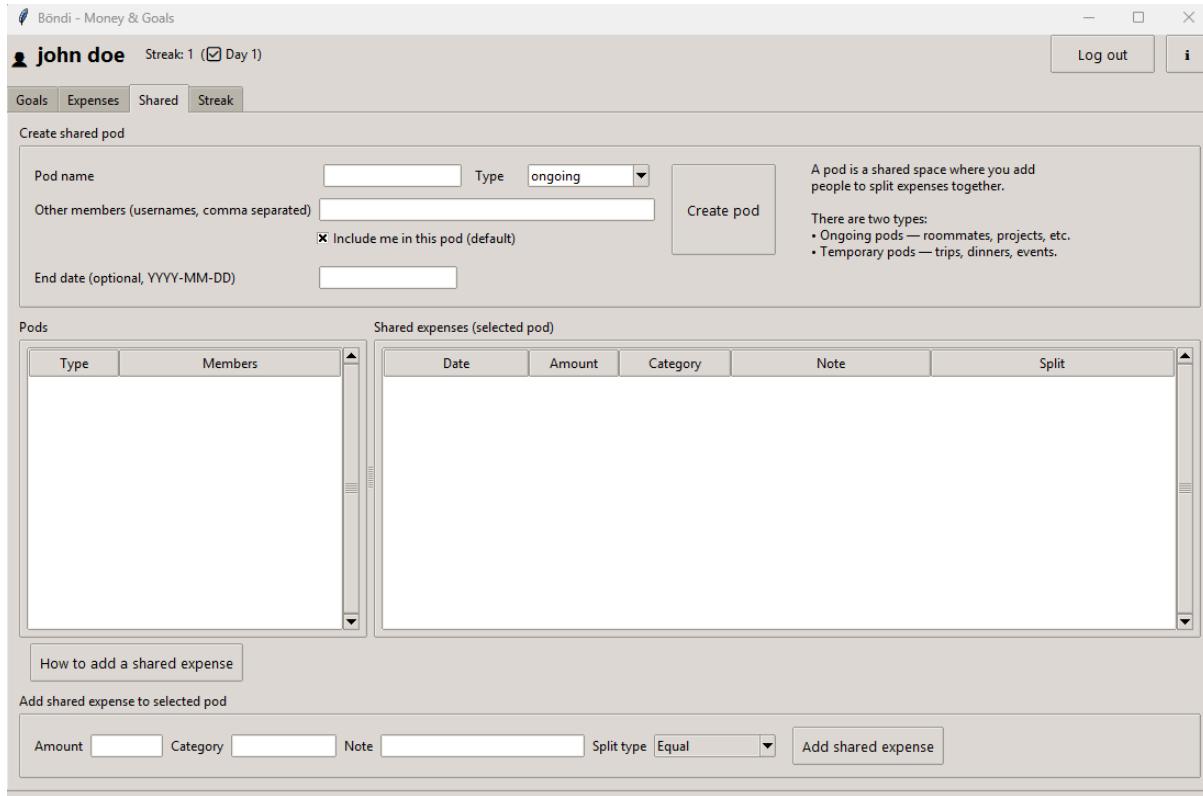
Shared Tab - Pods & Shared Expenses

The image below shows the Shared tab, where users create pods and manage shared expenses.

"Create shared pod" section (top-left)

This section allows the user to create a new pod, a group of people who will share expenses.

Fields:

- Pod name
  A custom name for the pod (e.g., "Roommates", "Trip to Paris", "Project Team").
- Type (ongoing / temporary)
  Determines whether the pod is long-term (roommates, partners) or short-term (dinners, vacations, events).
- Other members (comma separated)
  Add additional usernames to the pod.
  Duplicate entries are automatically prevented using a hash table.
- Include me in this pod (default)
  Checkbox that includes the current user in the pod.
  If unchecked, the user must add at least one other member.
- End date (optional)
  Only for temporary pods. After this date, the pod stops appearing in the list.

- Create pod (button)

  Saves the new pod and updates the list on the left.
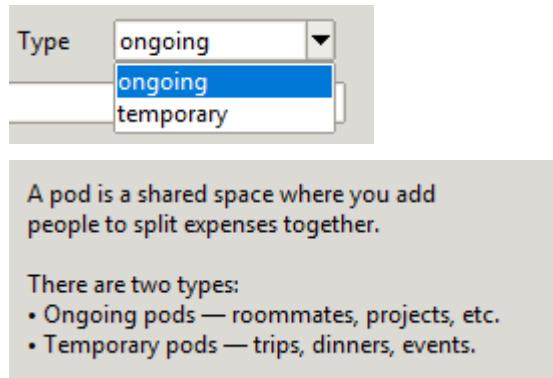
Pod explanation box (top-right)

- A description panel that explains what pods are and the difference between:
- Ongoing pods → roommates, repeat expenses
- Temporary pods → trips, events, dinners
- This helps guide users who are new to shared expense systems.

"Pods" panel (left middle)

This is the list of all pods the user or any account is part of.

Each row shows:

- Pod type
- Members

Selecting a pod here loads its expenses on the right.
If no pods exist, the list will be empty.

EXAMPLE OF POD CREATED:

Pods

Type	Members
ongoing	vikiii, john doe

"How to add a shared expense" button (bottom-left)

Opens a detailed help dialog explaining all the steps needed to know how to add a Shared Expense:
This button exists to help first-time users understand how splitting works.

Shared Expense Help ✕

How to add a shared expense:

1. Select your pod.
Click on a pod in the list. When selected, it will highlight in blue.

2. Enter the total amount.
Fill in the amount, category, and note in the fields below.

3. Choose how you want to split the expense:

a) Percentages
This option lets you assign different percentages to each member.
To use it:
• Select 'Percentages' in the Split Type menu.
• Click 'Add shared expense'.
• You will be prompted to enter the percentage for each member.

b) Equal
This option automatically splits the total evenly among all pod members.

c) Custom amounts
This option lets you choose the exact amount each member pays.
To use it:
• Select 'Custom amounts' in the Split Type menu.
• Click 'Add shared expense'.
• You will be prompted to enter the amount for each member.
The total of these amounts must match the total you entered.

Aceptar

"Add shared expense to selected pod" section (bottom)
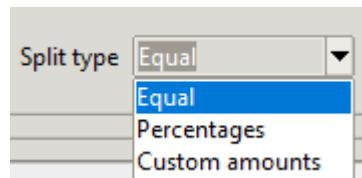
Add shared expense to selected pod

| Amount | Category | Note | Split type | Equal ▾ | Add shared expense |

This is where users add new shared expenses.

Fields:

- Amount:
  Total expense to be split between pod members.
- Category:
  e.g. groceries, tickets, utilities, dinner.
- Note:
  Additional context describing the expense.
- Split type (dropdown):
  Options include:

- Equal: automatic even split
- Percentages – user enters percent per member
- Custom amounts – user enters exact amounts

- Add shared expense (button)
  Saves the entry and updates the pod expense list.

"Shared expenses (selected pod)" panel (right middle)

When the user selects a pod (to know it is marked in blue), all expenses recorded in that pod will appear here.

Columns:

- Date
- Amount
- Category
- Note
- Split (shows how much each member pays)
  This acts as a shared expense history for the pod.

EXAMPLE OF SHARED EXPENSE BY EQUAL AMOUNT

Streak Tab - Motivation System

The streak tab shows:

A streak counts how many days in a row you have been active.
You increase your streak by doing any of the following:
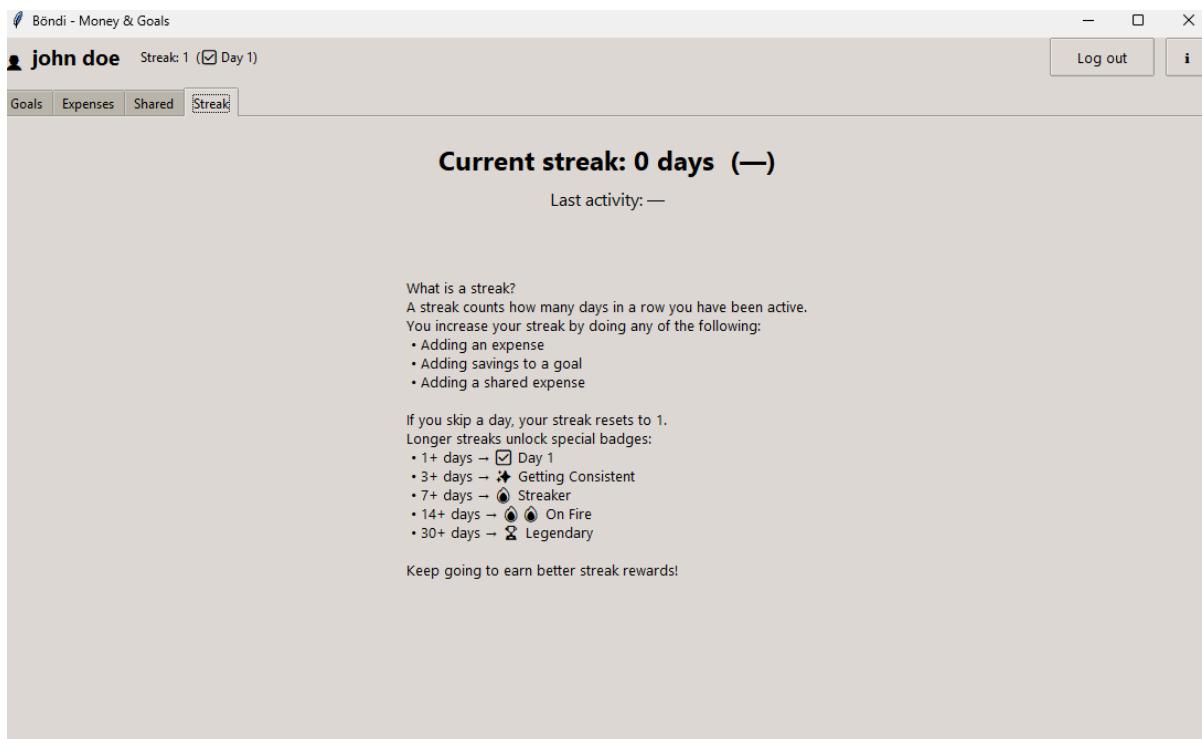
- Adding an expense

- Adding savings to a goal
- Adding a shared expense

If you skip a day, your streak resets to 1.

Longer streaks unlock special badges:

- 1+ days → ✅ Day 1
- 3+ days → ✨ Getting Consistent
- 7+ days → 🔥 Streaker
- 14+ days → 🔥 🔥 On Fire
- 30+ days → 🏆 Legendary

Log Out Button

The Log out button ends the current user session and returns the user to the login screen.

When clicked:

- The app clears the current user
- The main interface (Goals, Expenses, Shared, Streak) is hidden
- The login/signup screen is shown again

No data is deleted, all user information remains safely stored.
This simply ends the session so another user can log in.

Info Button What It Does

The (i) button opens a small pop-up window that provides a video explaining about the application.

Tutorial video

Böndi includes a built-in tutorial video that explains how to use the main features:

1. Log in to Böndi
2. Click the (i) button in the top-right corner of the app.
3. The app will automatically open the video located inside the /video/ folder.

https://www.loom.com/share/40bf6bcbc34a4d8faa210a130b201cb4

Data File Overview

Böndi automatically creates and manages several data files to store user information, expenses, goals, pods, and shared transactions. Each file has a specific purpose:

user_data.json:

```
users_data.json
```

Purpose: Main data storage file for the entire application.
Stores:

- User accounts (name, email, username, password)
- Recovery words
- User-specific data such as goals, expenses, and pods
  This file is created automatically the first time the app runs.

users.csv:

```
shared_expenses.csv
```

Purpose: Export of all users in the system.

Includes:

- Full name
- Email
- Username

Useful for external viewing or administrative review.

expenses.csv:

```
shared_expenses.csv
```

Purpose: Export of every personal expense added by users.
Contains:

- Username
- Amount
- Category
- Note
- Date

Useful for tracking trends or reviewing total spending outside the app.

goals.csv:

```
goals.csv
```

Purpose: Export of all savings goals and goal contributions.
Stores:

- Goal name
- Target amount
- Saved amount
- Deadline (optional)
- History of contributions

Helps users view long-term saving patterns.

pods.csv:

```
pods.csv
```

Purpose: Export of all shared groups ("pods") created in the app.

Includes:

- Pod name
- Pod type (ongoing or temporary)

- Members
- End date (if temporary)

Useful for reviewing group structures and shared activity.

shared_expenses.csv:

```
shared_expenses.csv
```

Purpose: Export of all shared expenses submitted inside pods.
Contains:

- Pod name
- Amount
- Category
- Note
- Date
- Split method (equal, percentage, custom)
- Individual member shares

Makes it easy to externally analyze shared spending.

photos/ folder:

Purpose: Contains all screenshots used in the README.
This folder is only for documentation and is not required for the app to run.

Technologies Used

Böndi is built using simple, reliable technologies that focus on accessibility and ease of use. The project uses:

Programming Language

- Python 3.8+ — core language used to build the entire application

User Interface

- Tkinter — Python's built-in GUI library used to create all screens, tabs, buttons, and dialogs

Data Storage

- JSON — stores user accounts and all persistent application data (users_data.json)
- CSV — exports user data, expenses, goals, pods, and shared expenses (users.csv, expenses.csv, goals.csv, etc.)

Development Environment

- PyCharm — recommended IDE for running and testing the project

Additional Tools

- Python Standard Library modules:
- datetime — for dates, streak tracking, and deadlines
- csv — exporting data
- hashlib — ensuring secure handling of duplicates in pods
- tkinter.messagebox — alerts and confirmations

Further Improvements

Dark Mode / Theme Customization

Add support for light/dark themes to improve accessibility and modernize the UI.

Delete & Edit Options

Allow users to:

- Edit goals
- Delete goals
- Edit expenses
- Delete expenses
- Edit pods
- Delete shared expenses
- This is one of the most common missing features in Tkinter apps.

Improved UI Layout

Enhance the interface with:

- Better spacing

- Consistent styling
- A more modern look

Pie Charts & Visual Analytics

Add spending insights such as:

- Category charts
- Monthly summaries
- Savings vs. expenses graphs

This would make the app feel more complete and visually informative.

Notifications & Reminders

Include reminders for:

- Daily usage (to keep streaks alive)
- Upcoming goal deadlines
- Shared expense updates

Cloud Sync / Online Storage

Allow users to sync through:

- Google Drive
- Dropbox
- A custom backend

Multi-language Support

Support multiple languages like:

- English
- Spanish
- French
- Portuguese

Export to PDF

Allow export of:

- Expense reports

- Goal summaries
- Pod balances

Mobile Version (Future)

Create a mobile-friendly version using:

- Kivy
- Flutter
- React Native

This shows vision and long-term planning.

Bibliography / Webgraphy

Bibliography / Webgraphy Books

- Bhargava, Aditya Y. Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People. Manning Publications, 2016.
- Van Rossum, Guido & Drake, Fred L. The Python Language Reference Manual. Python Software Foundation.
- Zimmermann, T. Python GUI Programming with Tkinter. Packt Publishing.

Official Documentation

- Python Official Documentation – https://docs.python.org/3/
- Tkinter GUI Documentation – https://docs.python.org/3/library/tkinter.html
- CSV Python Module – https://docs.python.org/3/library/csv.html
- JSON Python Module – https://docs.python.org/3/library/json.html
- Datetime Module – https://docs.python.org/3/library/datetime.html

Web Resources

- Real Python – Tutorials on Python, Tkinter, and GUI design
  https://realpython.com
- Stack Overflow – General troubleshooting and best practices
  https://stackoverflow.com
- GeeksforGeeks – Reference articles on data structures & algorithms
  https://geeksforgeeks.org
- W3Schools – Beginner-friendly Python references
  https://www.w3schools.com/python/

Credits

- Victoria Komissarchik Trubitsyna
- Helen Zhang
- Alexandra Roxana Gónzalez de la Flor
- Diego Ignacio Mainardi Mañu
- Sofía Machado Restrepo
- Sofía Olenick Otero