UDC 004.89
DOI https://doi.org/10.32782/2521-6643-2025-1-69.7

Haitan O. M., Senior Lecturer at the Department of Computer and Information Technologies and Systems of National University "Yuri Kondratyuk Poltava Polytechnic"
ORCID: 0000-0002-7228-9937

Snytka I. V., Student at the Department of Computer and Information Technologies and Systems of National University "Yuri Kondratyuk Poltava Polytechnic"

INTEGRATED PLATFORMS FOR AUTOMATING PERSONAL FINANCIAL ACCOUNTING BASED ON CHATBOTS AND CLOUD TECHNOLOGIES

In the context of increasing financial complexity and the growing need for better personal finance management, virtual personal financial assistants have emerged as essential tools for individuals seeking efficient ways to track, manage, and analyze their finances. The widespread adoption of smartphones and cloud-based technologies has contributed to the growth of automated financial solutions that assist users in budgeting, expense tracking, and making financial decisions. This study examines the conceptual and practical foundations of developing a virtual personal financial assistant based on Telegram chatbot technology, integrated with cloud services such as Google Sheets. The research explores methods for automating financial accounting, enhancing user interaction with financial data, and ensuring secure, scalable, and accessible financial accounting tool for individual and family.

The paper investigates the integration of asynchronous communication models using Aiogram, implementation of cloud-based data management via the Google Sheets API, and the use of a lightweight database SQLite for handling service-related data. Particular attention is given to the interaction logic, user experience design, and security protocols based on OAuth2 for managing sensitive financial data.

Furthermore, the research analyzes the user experience, focusing on how the system supports both individual and shared financial management. It explores the functionality that enables multiple users to collaboratively manage a family budget.

The proposed solution represents an approach, combining elements of software engineering, cloud computing, and digital finance. The study highlights the advantages of chatbot-driven interfaces for personal finance management and outlines the potential for future enhancements, including greater automation, multi-user collaboration, and artificial intelligence integration.

Key words: *virtual financial assistant, financial accounting, Google Sheets, Telegram bot, chatbot.*

Гайтан О. М., Снитка І. В. Інтегровані платформи для автоматизації особистого фінансового обліку на основі чат-ботів та хмарних технологій

У контексті зростаючої складності фінансових процесів та потреби в ефективному управлінні особистими фінансами віртуальні персональні фінансові асистенти стають важливими інструментами для тих, хто шукає зручні способи відслідковування, управління та аналізу своїх фінансів. З поширенням смартфонів та хмарних технологій набувають популярності автоматизовані фінансові рішення, які допомагають користувачам у веденні бюджету, відслідковуванні витрат та прийнятті фінансових рішень. Це дослідження аналізує концептуальні та практичні аспекти розробки віртуального персонального фінансового асистента, побудованого на основі технології Telegram-бота та інтегрованого з хмарними сервісами, такими як Google Sheets. Метою дослідження є вивчення методів автоматизації фінансового обліку, покращення взаємодії користувача з фінансовими даними та створення безпечного, масштабованого і доступного інструменту для фінансового обліку як для окремих осіб, так і для сімей.

У статті детально розглядається інтеграція асинхронних моделей комунікації за допомогою Aiogram, впровадження хмарного управління даними через Google Sheets API та використання легкої бази даних SQLite для зберігання сервісних даних. Окрема увага приділяється логіці взаємодії користувача, дизайну інтерфейсу та питанням безпеки, зокрема застосуванню протоколу OAuth2 для захисту чутливої фінансової інформації.

Також досліджується досвід користувачів, особливо щодо того, як система підтримує як індивідуальне, так і спільне фінансове управління. Зокрема, розглядається функціональність, яка дозволяє кільком користувачам спільно управляти сімейним чи груповим бюджетом.

Запропоноване рішення поєднує елементи програмної інженерії, хмарних обчислень та цифрових фінансів. Дослідження підкреслює переваги чат-ботів для управління особистими фінансами та вказує на потенціал для подальших удосконалень, таких як підвищена автоматизація, розширена співпраця між користувачами та інтеграція штучного інтелекту.

Ключові слова: *віртуальний фінансовий помічник, фінансовий облік, Google Sheets, телеграм-бот, чат-бот.*

Formulation of the problem. Modern financial management systems play a pivotal role in the effective management of financial processes in both business and personal life. The advent of digital technologies and automation software solutions has profoundly transformed financial accounting, budget planning, and income and expense control. These functions have become easier, accessible, and more convenient for users.

A virtual financial assistant is a software tool or system designed to assist users in managing their finances. It is an application or a bot that performs a variety of functions to help users in managing their finances, maintaining records of income and expenses, planning budgets, analyzing expenses, and providing recommendations for achieving financial goals or investments. Presently, financial bots have become prevalent. These are computer programs that can automate some financial tasks using coded commands.

The importance of virtual financial assistants lies in their ability to provide users with convenience and efficiency in managing their finances. Virtual financial assistants can automate many routine tasks, such as recording expenses, categorizing transactions, and generating reports. This saves users significant time and effort, allowing them to focus more on strategic planning of their finances. Available on a variety of platforms, such as mobile apps or online services, virtual financial assistants allow users to have constant access to their financial data. They also offer a user-friendly and intuitive interface that allows even non-expert users to easily navigate their finances. Virtual assistants also use analytical algorithms and artificial intelligence to analyze a user's financial data. Based on this analysis, they provide personalized recommendations for the optimal allocation, savings, and investments, thereby assisting users in achieving their financial objectives. The programs pay close attention to the security of financial data. They use advanced encryption methods and security protocols to ensure the confidentiality and protection of financial data.

Key areas that require virtual assistance in managing personal finances include:

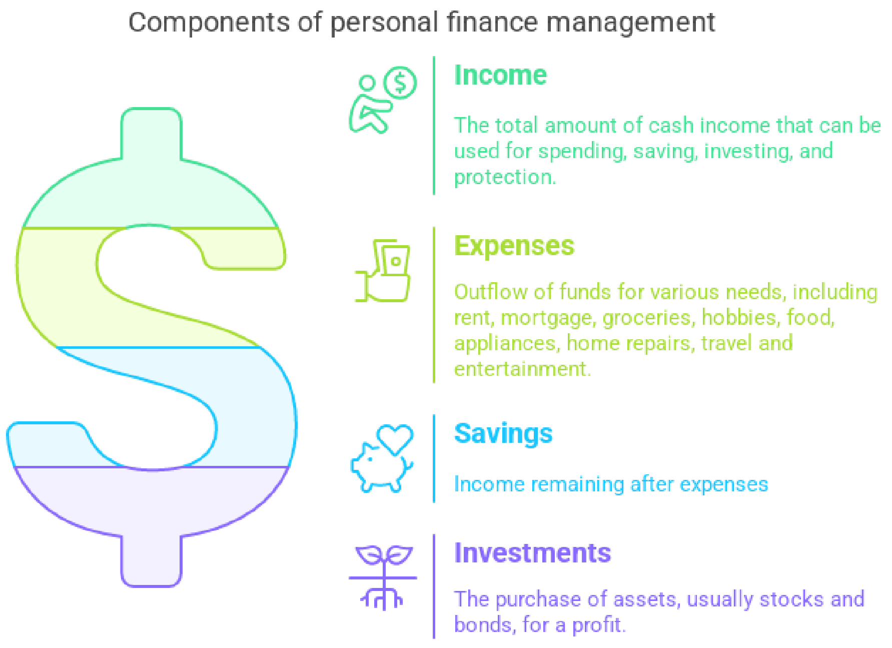

Components of personal finance management
Income
The total amount of cash income that can be used for spending, saving, investing, and protection.
Expenses
Outflow of funds for various needs, including rent, mortgage, groceries, hobbies, food, appliances, home repairs, travel and entertainment.
Savings
Income remaining after expenses
Investments
The purchase of assets, usually stocks and bonds, for a profit.

Fig. 1. Key areas of managing personal finances

Virtual financial assistants can be effectively used to manage family budgets by providing centralized accounting of financial transactions and access for all family members. This approach helps increase transparency, accountability, and consistency in family financial management. Sharing expenses helps to avoid unnecessary spending, optimize the use of resources, and develop healthy financial habits. Furthermore, a transparent accounting system stimulates discussion of common financial goals and improves communication among family members about budget decision-making. All family members can see their contribution to the family budget and understand the overall financial situation, how funds are being used, and which expenses are being prioritized.

Users can customize bots individually. For instance, financial bots can send notifications regarding loan payments, debts, and other reminders that the user needs.

The following are the advantages of Virtual financial assistants for users: the ability to control financial income, expenses, and keep records; the formation of a plan and income calculations, the ability to build a model of interaction with finances in advance; the ability to generate reports; the ability to view statistics; access to information 24/7; and customization of the program according to the user needs [1].

Analysis of recent research and publications. Mohanan A. et al. [2] reviewed the current literature on intelligent personal assistants. They outlined the key benefits of personal financial assistants, which increase productivity and improve accessibility by allowing users to manage finances, plan expenses, obtain information, and perform routine tasks using voice or text commands. However, along with these benefits, attention has been drawn to security and privacy threats, particularly in the case of third-party apps that could lead to unauthorized access or misuse of user data.

Iovine A. et al. [3] classified the types of conversational agents into three categories: informational, transactional, and advisory systems. Khudolii Yu.S. and Kosolapenko V.S. [4] added lead generators and feedback chatbots to this list.

Iovine A. et al. studied the development of key components of virtual customer assistants (VCA) in the financial industry depending on the category, reviewed templates and best practices for creating Conversational User Interfaces for VCA.

The prospects for the development of virtual financial assistant technologies lie in the use of generative Artificial Intelligence, Deep Machine Learning, and the integration of quantum algorithms into financial planning. Jain M. and Srihari A. [5], considering personal financial management tools, compared traditional approaches and solutions based on artificial intelligence. Pavlyuchenko D. M. [6] conducted a study on the impact of Artificial Intelligence and Machine Learning on banking services.

The market for financial management software is expanding rapidly due to the development of technology and the growing interest in personal finance. Numerous software solutions have been developed to assist users in managing their finances.

"**Money Manager Expense and Budget**" is a financial management app developed by RealbyteInc. that helps users track their expenses, plan their budgets, and control their financial resources [7].

Key functions of «Money Manager Expense and Budget» include:

1. Input and tracking of users' incomes and expenses with categorization of expenses and their descriptions.

2. Budget planning with the definition of monthly expenditure limits for each category. The ability to set budget goals and receive notifications when approaching the limit, ensuring expense control.

3. Tracking reports and statistics in the form of charts and diagrams for expense control, displaying incomes and expenses by various categories and over a specific period of time.

4. Synchronization across different devices and backup [8].

"Money Lover" is a financial app created by Finsify in 2011 for personal finance tracking and resource management. The app is available for use on various platforms, such as mobile devices with Android and iOS [8].

Key functions of "Money Lover" include:

1. Transaction and expense tracking. Adding transactions by categories to track user expenses. The ability to link bank accounts for tracking transaction histories.

2. Financial planning. The ability to categorize expenses, set limits for each category, and receive notifications when approaching the limit.

3. Financial management for special events. The ability to create separate financial plans for events with planned expenses, such as vacations, holidays, or gifts.

4. Automatic addition of recurring transactions, such as weekly purchases or monthly bills, into the system.

5. Account management. The ability to add various bank accounts, credit cards, and other financial resources to the system. The ability to track balances, transfers, expenses, and income from all accounts for a complete overview of financial status.

In addition to applications that require downloading and installation, there are also chatbots integrated into messaging platforms such as Facebook, Viber, and Telegram. These chatbots offer the following advantages for users:

1. Convenience. Users are not required to download or install a separate application, as the chatbots are directly embedded within platforms like Facebook, Viber, and Telegram. This enables users to carry out financial transactions and access information in a convenient manner without leaving the messaging app.

2. Quick access. Users can obtain essential financial information directly within the messenger chat. This saves time by eliminating the need to switch to standalone applications or websites.

3. Interactivity. Chatbots enable user interaction through text messages or interactive buttons, which simplifies communication. Users can easily specify their requests and receive the appropriate responses and actions.

"WhereIsMyMoney" is a Telegram-based financial management bot designed to help users track their expenses and manage their financial resources [9].

The main functions of "WhereIsMyMoney" include:

1. Tracking of expenses and income, with the ability to input amounts and descriptions for both.

2. Expense statistics over a specified period.

3. Report export in selected formats, such as csv or xlsx.

Based on the analysis of existing software solutions in the field of virtual financial assistants, the development of a virtual financial assistant in the form of a chatbot was deemed appropriate solution, and its core functionality was defined.

The purpose of the article is to explore the development of a virtual financial assistant in the form of a Telegram bot integrated with Google Sheets, designed to support users in managing and tracking their personal finances.

Based on the analysis of the subject area and taking into account user needs for a convenient tool for managing both personal and shared financial records, the functional requirements for a chat-bot were defined as follows:

- User interaction initialization: launching the bot, providing instructions, creating an individual accounting spreadsheet, and supporting invitations for shared access.
- Implementation of a main menu with buttons: entering transactions (income and expenses), viewing recent entries, aggregating by categories, viewing balance over a selected period (recent, monthly, or total income/expenses), accessing the spreadsheet, and accessing help information (getting a link to the spreadsheet and a help section).
- System configuration: managing reminders and invitations, and setting up shared (family) finance tracking.
- Transactions: adding, categorizing, commenting on, cancelling, editing, and deleting financial records.
- Spreadsheet operations: synchronized access to the “Expenses”, “Income”, and “Reports” sheets, with the ability to edit or delete transactions.
- Table deletion and user management: deleting the spreadsheet (for the owner) and removing invited users.

Non-functional requirements include maintaining informal conversational interactions, preserving the user session between interactions, and ensuring continuous bot operation without requiring repeated registration.

Presenting main material. The platform selected for the development of a virtual financial assistant defines the foundation for its functionality and the mode of interaction with users. There are various platforms on which a virtual financial assistant can be developed, each offering its own advantages and specific features. Several popular platforms are considered and compared below.

Messagers Facebook, Viber, and Telegram offer powerful capabilities for chatbot development. They have a broad user base and provide convenient communication through text messages, buttons, and graphical elements.

Mobile platforms (iOS, Android) allow users to interact with the virtual assistant via a mobile application on their devices. This ensures portability and accessibility for users.

Web platforms enable the creation of a financial assistant accessible through a browser, offering flexibility and convenient interactive access to financial features.

For development of the virtual financial assistant, the **Telegram** platform was chosen due to its popularity, user-friendly interface, and robust tools for building bots. Telegram has a large and steadily growing number of active users, which ensures broad access to the target audience (Fig. 2) [10].

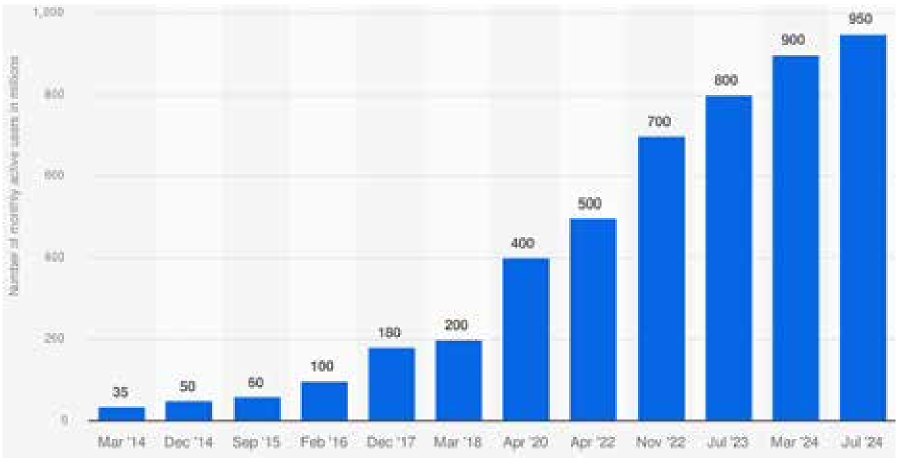


Fig. 2. Increase in the Number of Telegram Users

Source: © Statista 2025

The platform supports BotAPI and provides a clear, well-documented programming interface that simplifies functionality integration. Telegram’s simple and intuitive interface facilitates interaction with financial features and services. Robust security measures and modern encryption techniques ensure the protection of confidential data, which is critical for handling sensitive financial user data.

In development of the virtual financial assistant for automation of financial accounting and planning, several key software technologies were selected to ensure system efficiency, flexibility, and integration capability. The primary requirements for these technologies are ease of integration with existing services, user-friendliness, reliability, and scalability. The selection process considered the following aspects: user convenience, automation capabilities, and the technical implementation of interactions between system components.

One of the core technologies used for developing the virtual assistant is the **Telegram API**, which enables the creation of bots for automated user interaction via chat. The **Aiogram** library was chosen for this purpose. Aiogram is a high-level, asynchronous Python library for building Telegram bots, capable of efficiently handling numerous concurrent requests. It supports essential functionality and provides a convenient interface for interacting with the Telegram API and developing complex, interactive bots. Aiogram enables flexibility of user interaction logic and is reliable in terms of scalability. Its key advantages lie in its simplicity and flexibility. Thanks to its extended functionality, Aiogram facilitates the creation of bots with complex user scenarios and interactions through commands and messages. It provides convenient tools for processing incoming messages, managing finite-state machines, and implementing response logic. It also supports various message types, custom keyboards with buttons, and user interaction events.

Python is selected as the main programming language for development of the virtual assistant due to its high performance, flexibility, and simplicity in implementing interaction logic among system components. Python's elegant syntax, efficient data structures, and dynamic typing contribute to its usability. As an interpreted language, Python is well-suited for scripting and rapid bot development. Moreover, it includes a vast number of standard libraries that offer ready-made solutions for tasks ranging from text processing to data analysis. Python's cross-platform nature allows it to run on various operating systems and devices.

Google Sheets was chosen as the cloud-based platform for storing and processing financial data in the table view. It provides a convenient interface for working with large volumes of data and enables users to effectively manage their finances through integration with other services. Interaction with Google Sheets is implemented using the **gspread** library, which allows connections to a Google account to read, write, and update spreadsheet data without manual document modifications. The library's extensive functionality supports operations on various data types, creation of new spreadsheets, editing and deletion of existing ones, and management of access rights to Google Sheets directly from the Telegram bot. This significantly reduces user workload and automates most data-related operations.

Google Sheets also enables automated data processing through built-in formulas. This functionality allows the financial assistant to perform calculations, analyze data, and provide relevant reports to the users. Therefore, choosing Google Sheets as the tool for storing users' financial data is a rational decision, offering user-friendly data handling, multi-device access, and automation capabilities.

Each user has an individual Google Sheets spreadsheet to which the bot has access. Expense and income data entered by users via the Telegram bot are automatically recorded in the corresponding spreadsheet. This allows for convenient storage and tracking of the user's financial history.

To work with Google Sheets via the Telegram bot, a Google service account is required. It facilitates interaction with various Google services, including the Sheets and Drive APIs. This account provides automatic authentication and access to functions that enable interaction with stored data, including file creation, reading, updating, and other operations within the Google services. Using this service account, the financial assistant uses Sheets API to read and write data to Google Sheets. This enables the automatic update of financial data, execution of calculations, and generation of reports based on the spreadsheet content. Additionally, the service account grants access to the Drive API, allowing file operations such as uploading, saving, searching, and synchronizing files in the Google Drive cloud storage.

To ensure security and authentication when accessing Google Sheets, the **OAuth2** standard is used. This protocol provides secure access to user data without requiring password storage. The **oauth2client** library is employed for interaction with OAuth2, offering tools for authorization and authentication necessary for accessing Google API. The library provides secure access to users' financial data by retrieving access tokens for service accounts used to interact with Google API directly from Python. This approach ensures security and access control to financial data and automates the process of accessing Google Sheets.

The use of the gspread and oauth2client libraries allows the virtual financial assistant to access Google Sheets and perform read/write operations. In addition, other libraries are utilized: **schedule** for task planning, **decimal** for high-precision decimal operations, and **datetime** for working with dates and times.

For automating tasks such as periodic data updates or end-of-day reminders, the **schedule** library in Python is used. It enables the configuration and execution of scheduled tasks, supporting the development of reminder functionality for financial operations and user expense tracking. This tool allows for planning tasks, events, and actions at specific time intervals, as well as setting up recurring tasks. The library also supports different time zones and is useful for automating processes such as data collection, message sending, and event planning.

To store general information about the financial assistant, the SQLite database was selected. SQLite is a lightweight relational database engine that is well-suited for small-scale projects. It provides fast data access and a simple SQL syntax for interaction with the database.

The architecture of the chatbot represents a multi-component system that enables interactive communication between the user and the application. It is built on a microservices-based approach, where each component is responsible for a specific function. The structure of the virtual financial assistant comprises three main components: Telegram bot, SQLite database, and Google Sheets. The Telegram bot serves as the user interface, allowing users to input their financial data and receive reports. It interacts with users by collecting and processing requests, then transfers the data to Google Sheets for storage. Google Sheets are utilized to organize and store users' financial data. The bot also accesses financial data via the Google Sheets API, enabling automated operations such as balance calculations, expense tracking, and report generation. The SQLite database stores and provides access to user-related information (such as chat ID, email addresses, and spreadsheet link). The bot interacts with the database to retrieve and update user information. This architecture allows the virtual financial assistant to efficiently interact with users, securely store their financial data, and ensure confidentiality and integrity of the information.

Integration between components is implemented via RESTful APIs for communication between the Telegram bot and Google Sheets. This ensures a high level of scalability, security, and user convenience.

The operation algorithm of the virtual financial assistant is based on the interaction between the user, the Telegram chatbot, and Google Sheets (Fig. 3).

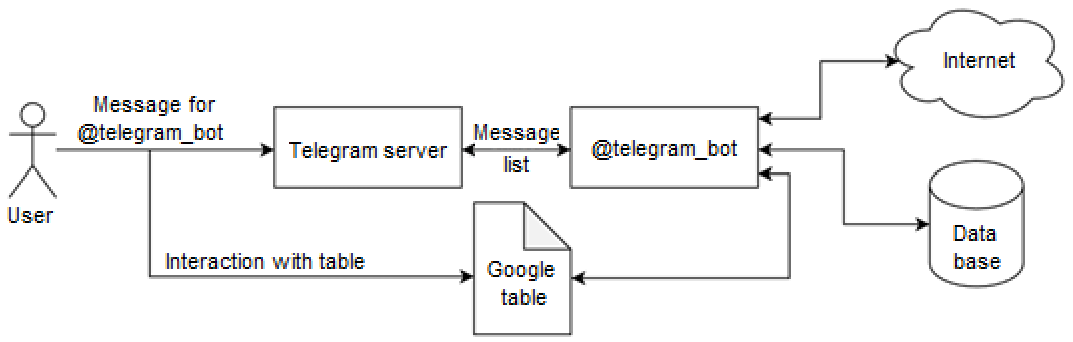


Fig. 3. The operation algorithm of the virtual financial assistant

The Telegram bot operation algorithm can be divided into several stages. The bot constantly monitors the incoming stream of messages and waits for a message from the user. Upon receiving a message, the bot performs semantic and structural analysis to determine the type of the request or command sent. It examines the message text, its structure, and the presence of keywords to identify the required operation. Following the analysis, the bot executes the corresponding action or request. This may include providing information, performing a specific function, conducting a search, or interacting with Google Sheets or the database. Upon completion, the bot generates an appropriate response for the user in the form of text, a link, or other format that best addresses the user's request. Depending on the action performed, the bot may save the obtained data for future interactions. Additionally, the user has access to the linked Google Sheet, which they can view or modify.

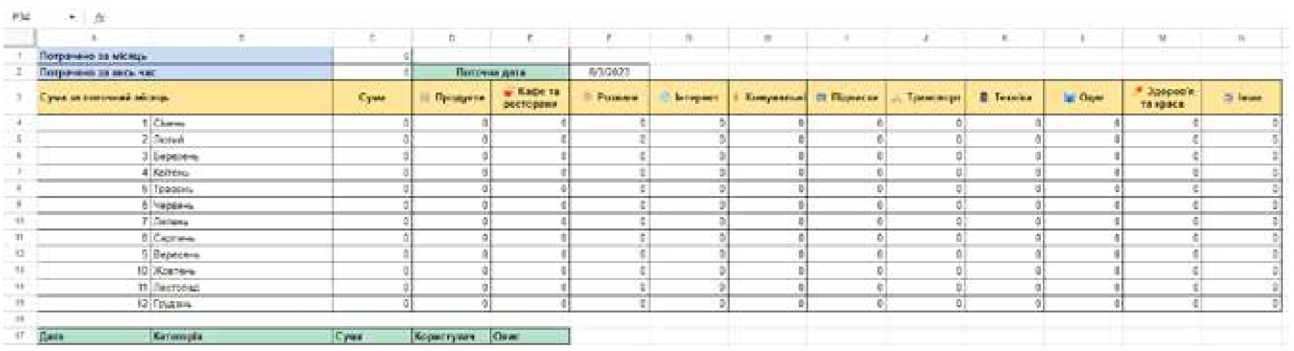
The development of a Telegram chatbot begins with registration via the Telegram Bot API through the official BotFather, which facilitates the registration process and provides an API token for interaction with the Telegram Bot API.

After receiving the API token, to integrate with Google Sheets, it is necessary to create a dedicated service account in the Google Cloud Platform. This entails initiating a new project, enabling the Google Sheets API and Google Drive API services, and configuring credentials with "Application Data" access type. The service account is granted the "Owner" role to permit the creation and modification of Google Sheets documents. Once these settings are configured, a JSON-formatted key must be generated. JSON is a language-independent data exchange format based on JavaScript, used to transmit structured information between heterogeneous systems. It represents data as key–value pairs, and supports objects (including associative arrays), arrays, numbers, strings, Boolean values, and nulls.

The use of a predefined template in Google Sheets allows for the creation of a standardized, preformatted document, which serves as a basis for storing user-provided financial data via the chatbot. To configure such a template, a spreadsheet must be created and the service account granted editor permissions (Fig. 4–6).

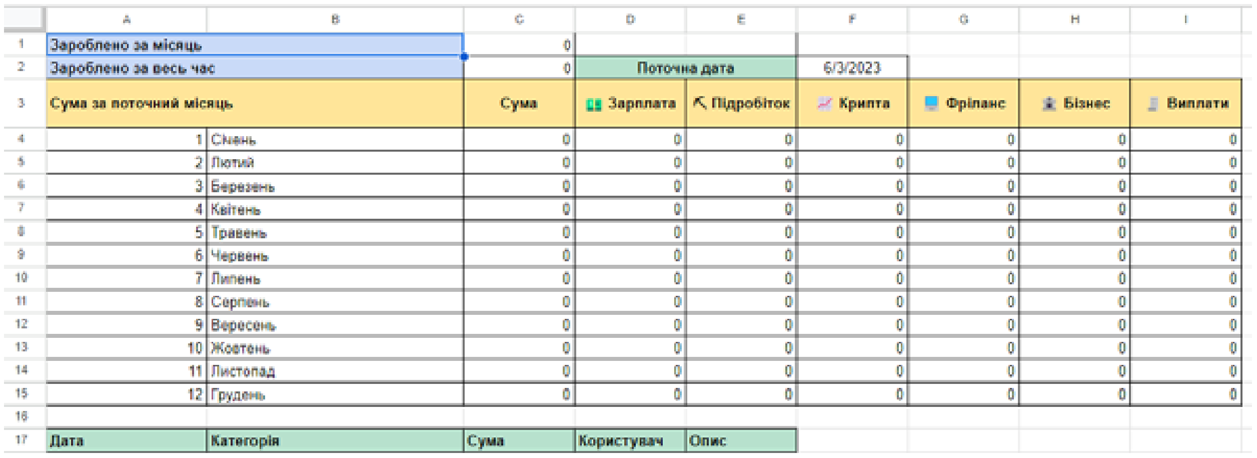
The sqlite3 is a built-in module included in the standard Python distribution starting from version 2.5. Within the framework of the virtual financial assistant, it is used exclusively for storing service information about users and data that facilitates modifications to the spreadsheet, since the primary financial data (income, expenses, reports) is stored in Google Sheets. The attributes of the database are listed below:

- user_id stores the user's identifier in the database; data type: INTEGER; set as the PRIMARY KEY;
- email stores the user's email address; data type: TEXT;
- spreadsheet_link stores the link to the Google Sheets; data type: TEXT;
- last_added_row_sheet1 stores the index of the last added row in the first sheet; data type: INTEGER;
- last_added_row_sheet2 stores the index of the last added row in the second sheet; data type: INTEGER;



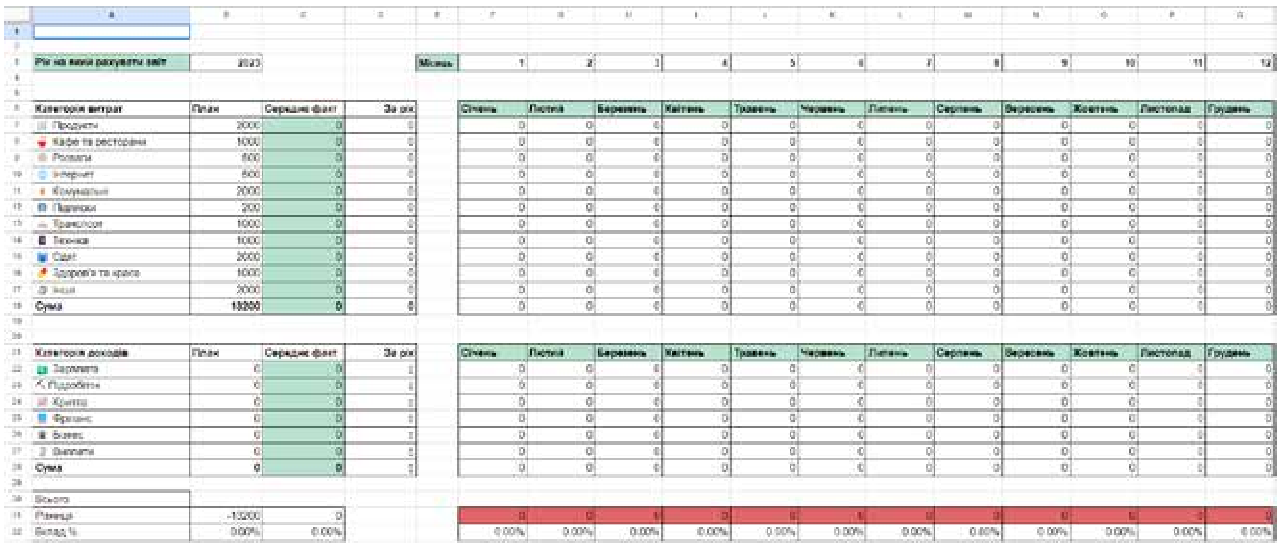
Потрачено за місяць		0											
Потрачено за весь час		0	Поточна дата		6/3/2023								
Сума за поточний місяць		Сума	Продукти	Кафе та ресторани	Розваги	Інтернет	Комунальні	Підписки	Транспорт	Техніка	Одяг	Здоров'я та краса	Інше
1	Січень	0	0	0	0	0	0	0	0	0	0	0	0
2	Лютий	0	0	0	0	0	0	0	0	0	0	0	0
3	Березень	0	0	0	0	0	0	0	0	0	0	0	0
4	Квітень	0	0	0	0	0	0	0	0	0	0	0	0
5	Травень	0	0	0	0	0	0	0	0	0	0	0	0
6	Червень	0	0	0	0	0	0	0	0	0	0	0	0
7	Липень	0	0	0	0	0	0	0	0	0	0	0	0
8	Серпень	0	0	0	0	0	0	0	0	0	0	0	0
9	Вересень	0	0	0	0	0	0	0	0	0	0	0	0
10	Жовтень	0	0	0	0	0	0	0	0	0	0	0	0
11	Листопад	0	0	0	0	0	0	0	0	0	0	0	0
12	Грудень	0	0	0	0	0	0	0	0	0	0	0	0
Дата	Категорія	Сума	Користувач	Опис									

Fig. 4. Template for the "Expenses" sheet



Зароблено за місяць		0						
Зароблено за весь час		0	Поточна дата		6/3/2023			
Сума за поточний місяць		Сума	Зарплата	Підробіток	Крипта	Фріланс	Бізнес	Виплати
1	Січень	0	0	0	0	0	0	0
2	Лютий	0	0	0	0	0	0	0
3	Березень	0	0	0	0	0	0	0
4	Квітень	0	0	0	0	0	0	0
5	Травень	0	0	0	0	0	0	0
6	Червень	0	0	0	0	0	0	0
7	Липень	0	0	0	0	0	0	0
8	Серпень	0	0	0	0	0	0	0
9	Вересень	0	0	0	0	0	0	0
10	Жовтень	0	0	0	0	0	0	0
11	Листопад	0	0	0	0	0	0	0
12	Грудень	0	0	0	0	0	0	0
Дата	Категорія	Сума	Користувач	Опис				

Fig. 5. Template for the "Income" sheet



Рік на який рахувати звіт	2023		Місяць	1	2	3	4	5	6	7	8	9	10	11	12
Категорія витрат	План	Середній факт	За рік	Січень	Лютий	Березень	Квітень	Травень	Червень	Липень	Серпень	Вересень	Жовтень	Листопад	Грудень
Продукти	2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Кафе та ресторани	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Розваги	500	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Інтернет	500	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Комунальні	2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Підписки	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Транспорт	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Техніка	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Одяг	2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Здоров'я та краса	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Інше	2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Сума	13200	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Категорія доходів	План	Середній факт	За рік	Січень	Лютий	Березень	Квітень	Травень	Червень	Липень	Серпень	Вересень	Жовтень	Листопад	Грудень
Зарплата	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Підробіток	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Крипта	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Фріланс	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Бізнес	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Виплати	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Сума	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Всього															
Різниця	-13200	0		0	0	0	0	0	0	0	0	0	0	0	0
Вклад %	0.00%	0.00%		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Fig. 6. Template for the "Reports" sheet

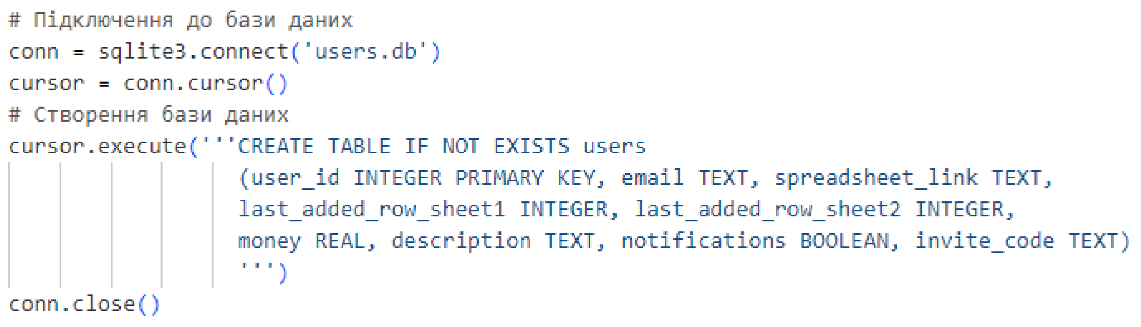
− money stores the last amount entered by the user; data type: REAL;

− description stores the last description entered by the user; data type: TEXT;

− notifications stores information on whether notifications are enabled for the given user; data type: BOOLEAN;

− invite_code stores the invitation code used for managing family budgets; data type: TEXT.

Fig. 7 presents the database initialization used for storing user data.



```
# Підключення до бази даних
conn = sqlite3.connect('users.db')
cursor = conn.cursor()
# Створення бази даних
cursor.execute('''CREATE TABLE IF NOT EXISTS users
                  (user_id INTEGER PRIMARY KEY, email TEXT, spreadsheet_link TEXT,
                  last_added_row_sheet1 INTEGER, last_added_row_sheet2 INTEGER,
                  money REAL, description TEXT, notifications BOOLEAN, invite_code TEXT)
                  ''')
conn.close()
```

Fig. 7. The database initialization

At the initial stage of implementing the Telegram bot, a handler for the /start command is created. This handler analyzes the arguments passed with the command using the message.get_args() method. Depending on the presence of an argument, the corresponding function is invoked: handle_normal_start for a standard launch, or handle_invite_start when a user is invited to use the system jointly.

The handle_normal_start method processes the message that initiates interaction with the bot. Its core functionality depends on whether the user is registered in the database:

- establishes a connection to the database and retrieves the user record using the user_id;
- if the user already exists in the database, sends a message containing a link to their spreadsheet, retrieving the spreadsheet key using get_spreadsheet_key;
- if the user is not present in the database, sends a welcome message describing the bot's features and prompts the user to enter their actual Gmail address;
- sets the state to UserEmail.email in order to handle the entered email address in the next step.

The handle_invite_start method handles messages that initiate the process of inviting a user to join a shared financial accounting system. Its main actions include:

- sends a message to the user requesting their email address;
- sets the state to UserEmail.invite_email to process the entered email.

The UserEmail.email and UserEmail.invite_email states validate the correctness of the entered email address and subsequently invoke the create_sheet and share_sheet functions respectively.

The create_sheet function performs the following tasks:

- creates a spreadsheet based on a template and grants access to the provided email address;
- sends a message to the user in the chat with a link to the created spreadsheet and usage instructions;
- generates a menu for further interaction with the bot;
- stores the user's data in the database for future identification and generates an invitation code for inviting additional users.

The share_sheet function performs the following operations:

- connects to the database and retrieves the spreadsheet link using the invitation code;
- if the spreadsheet link exists, extracts the email from the user's message;
- opens the spreadsheet using the retrieved link and grants editing access to the specified email;
- stores the user's data in the database, including user ID, email, spreadsheet link, and other relevant information;
- sends the user a message with the spreadsheet link and instructions for using the financial assistant bot.

The handle_text function processes user messages that are not bot commands, ensuring smooth interaction with the bot. Its main operations are:

- retrieves the text message and user ID;
- splits the message into two parts – amount and description – using a space;
- checks whether the first part is a float-type number;
- if valid, performs the required operations; if the first part is not a number, sends an error message indicating incorrect input;
- formats the number to two decimal places;
- stores the amount and description in the database;
- displays category selection buttons to the user.

The handle_text_message_sheet function, which manages data insertion into the "Expenses" and "Income" sheets in Google Sheets, executes the following tasks:

- authorizes access to Google Sheets using provided credentials;
- opens the spreadsheet using the key retrieved from the database;
- retrieves the money amount and description from the database;

- obtains additional user details, such as the user’s name;
- formats the current timestamp;
- creates a data list including the formatted timestamp, message text, money amount, user name, and description;
- searches the first empty row in the spreadsheet, starting from row 18;
- updates this row with the prepared data using the update method;
- updates the database with information about the last added entry.

To display information about the added transaction and provide a “Cancel” button, the cancel_button functions are implemented with the following functionality:

- deletes the previous message containing buttons;
- retrieves the current time and user ID;
- obtains the user’s recorded money amount from the database;
- creates a new message confirming the recorded money amount into a specific category;
- adds a “Cancel transaction” button to the message.

Functions such as balance and balance_month are implemented to display the balance for all time / for the current month. They perform the following operations:

- call the get_data_from_cell function to retrieve data from a specific cell in the SHEET for expenses and income respectively;
- calculate the balance by subtracting expenses from income;
- convert resulting income, expenses, and balance data in format for display;
- generate a formatted message showing the financial balance, including income, expenses, and balance for the selected period;
- remove previously displayed buttons from the message; generate a button labeled either “View Monthly” or “View All-Time” for further navigation;
- send the new message with the balance data and the navigation button.

The report_by_category_of_expenses, report_by_category_of_income (for categorized transaction summaries), report_recent_transactions, and report_recent_earnings (for 15 recent transactions / earnings) functions implement reporting mechanisms for the user’s financial activity.

The report_by_category_of_expenses function performs the following tasks:

- authorizes access to Google Sheets using the provided credentials;
- opens the spreadsheet;
- determines the current month;
- defines the range for expense categories and amounts;
- retrieves category and amount values from the spreadsheet;
- formats report text by adding category names and corresponding amounts;
- sends the generated report to the chat.

The /delete_sheet command implements conditional logic based on the user’s role: if the spreadsheet was created by the user, the command initiates complete deletion; if the user was invited to the shared sheet, the command removes them from access. The delete_sheet function performs the following tasks:

- checks the presence of an invite_code in the database for the current user;
- depending on the presence of the invite_code, generates confirmation buttons for deleting the sheet (confirm_delete) or exiting the shared access (confirm_remove);
- sends a confirmation message with the relevant buttons.

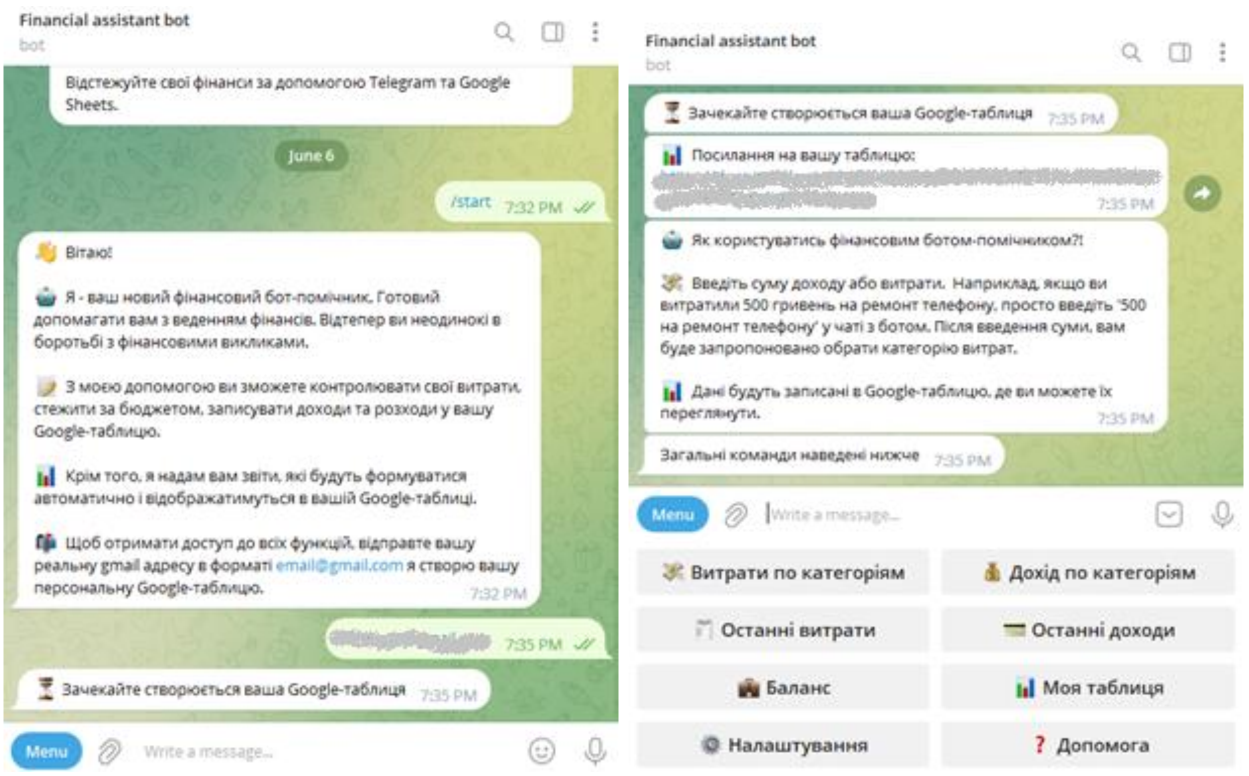
The confirm_delete function performs the following tasks:

- retrieves the user_id from the request;
- calls the del_spreadsheet function via google_client to delete the spreadsheet using the key from the database;
- deletes the associated database record based on the spreadsheet link;
- sends a message confirming the deletion and provides instructions for entering a new email address to restart the process;
- sets the state to UserEmail.email to collect the user’s email.

The confirm_remove function performs the following tasks:

- retrieves the user_id from the request;
- queries the database to obtain the user’s email and spreadsheet link;
- opens the spreadsheet using the link;
- removes the user’s access rights based on their email address;
- deletes the corresponding record in the database;
- sends a message confirming successful removal of the email from the spreadsheet, along with instructions for entering a new email to restart the process;
- sets the state to UserEmail.email to collect the user’s email.

Upon first activation, the virtual financial assistant provides a brief guide to the chatbot and prompts the user for their email address to generate a personal spreadsheet. Once the address is submitted, the bot displays a message indicating that the spreadsheet is being created and asks the user to wait. After the table has been created, the assistant sends a link to it accompanied by a brief usage guide (Fig. 8).


Financial assistant bot
bot

Відстежуйте свої фінанси за допомогою Telegram та Google Sheets.

June 6

/start 7:32 PM

👋 Вітаю!

🤖 Я - ваш новий фінансовий бот-помічник. Готовий допомагати вам з веденням фінансів. Відтепер ви неодинокі в боротьбі з фінансовими викликами.

📝 З моєю допомогою ви зможете контролювати свої витрати, стежити за бюджетом, записувати доходи та розходи у вашу Google-таблицю.

📊 Крім того, я надам вам звіти, які будуть формуватися автоматично і відображатимуться в вашій Google-таблиці.

👫 Щоб отримати доступ до всіх функцій, відправте вашу реальну gmail адресу в форматі email@gmail.com я створю вашу персональну Google-таблицю. 7:32 PM

[redacted] 7:35 PM

⏳ Зачекайте створюється ваша Google-таблиця 7:35 PM

📊 Посилання на вашу таблицю: [redacted] 7:35 PM

🤖 Як користуватись фінансовим ботом-помічником?!

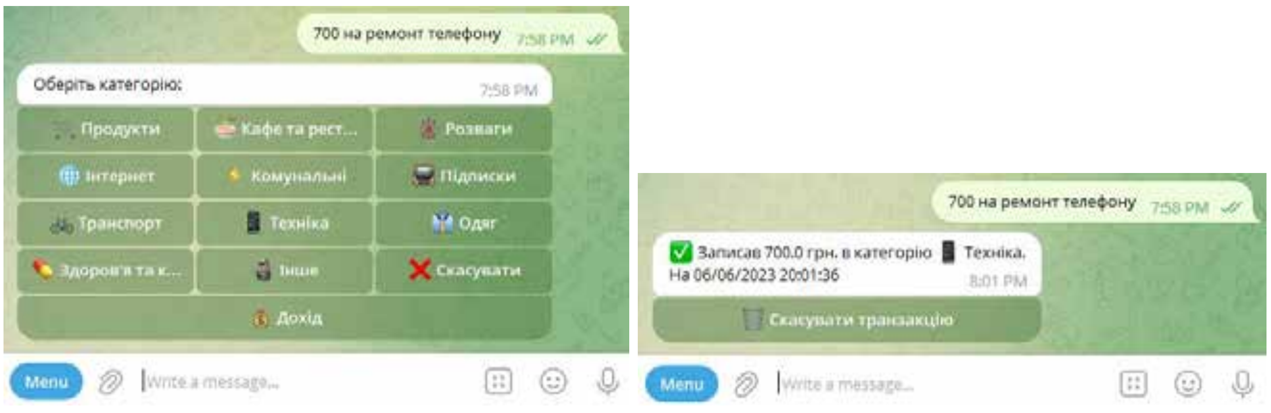
💸 Введіть суму доходу або витрати. Наприклад, якщо ви витратили 500 гривень на ремонт телефону, просто введіть '500 на ремонт телефону' у чаті з ботом. Після введення суми, вам буде запропоновано обрати категорію витрат.

📊 Дані будуть записані в Google-таблицю, де ви можете їх переглянути. 7:35 PM

Загальні команди наведені нижче 7:35 PM

Fig. 8. The registration process

Upon completing of the registration process and personal table generation a menu appears, presented as a set of interactive buttons that assist the user in managing finances and configuring the personal virtual assistant. To initiate interaction, the user inputs an expense, for example: “700 for phone repair.” The bot then prompts the user to select a category for the expense or income. As a result, a confirmation message is displayed showing the transaction details (Fig. 9).


700 на ремонт телефону 7:58 PM

Оберіть категорію: 7:58 PM

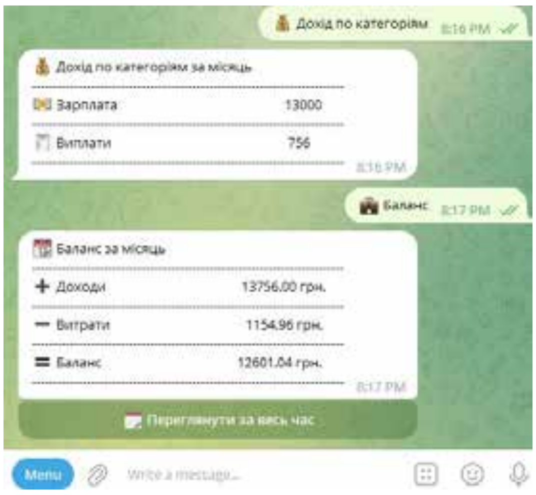
700 на ремонт телефону 7:58 PM

✅ Запис на 700.0 грн. в категорію 📱 Техніка. На 06/06/2023 20:01:36 8:01 PM

Fig. 9. The recording of transaction

In a similar manner, the user input both income and expenditure data into the table. After transactions have been recorded, the user can review them either by accessing the spreadsheet via the registered email address or directly within the Telegram bot through the corresponding menu options (Fig. 10).

The system supports collaborative financial tracking. To enable this functionality, the user must navigate to the “Settings” section, select the “Shared Accounting” option, and send an invitation to another user (Fig. 11).


💰 Дохід по категоріям 8:16 PM

💰 Дохід по категоріям за місяць

💵 Зарплата	13000
Виплати	756

8:16 PM

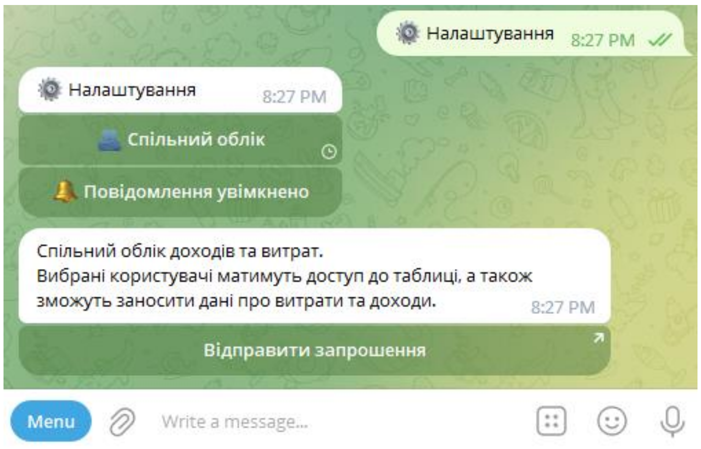
💼 Баланс 8:17 PM

📅 Баланс за місяць

➕ Доходи	13756.00 грн.
➖ Витрати	1154.96 грн.
🟰 Баланс	12601.04 грн.

8:17 PM

Fig. 10. Representation of income data by categories and overall balance

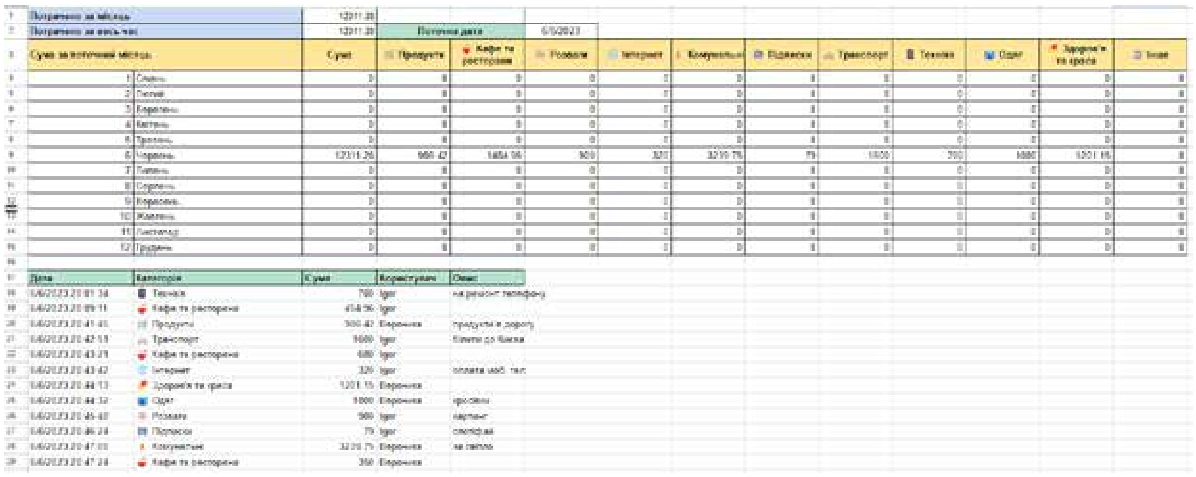

⚙️ Налаштування 8:27 PM

⚙️ Налаштування 8:27 PM

Спільний облік доходів та витрат.
Вибрані користувачі матимуть доступ до таблиці, а також зможуть заносити дані про витрати та доходи. 8:27 PM

Fig. 11. Sending an invitation for collaborative financial tracking

Upon following the invitation link and executing the /start command, the invited user receives a message from the assistant, indicating that they have been invited to a shared accounting. To proceed, the user is prompted to enter their email address. Once submitted, the new participant is added into the owner's spreadsheet, allowing for joint financial management. As multiple users interact with the financial assistant over time, the spreadsheet reflects collaborative activity (Fig. 12–13).

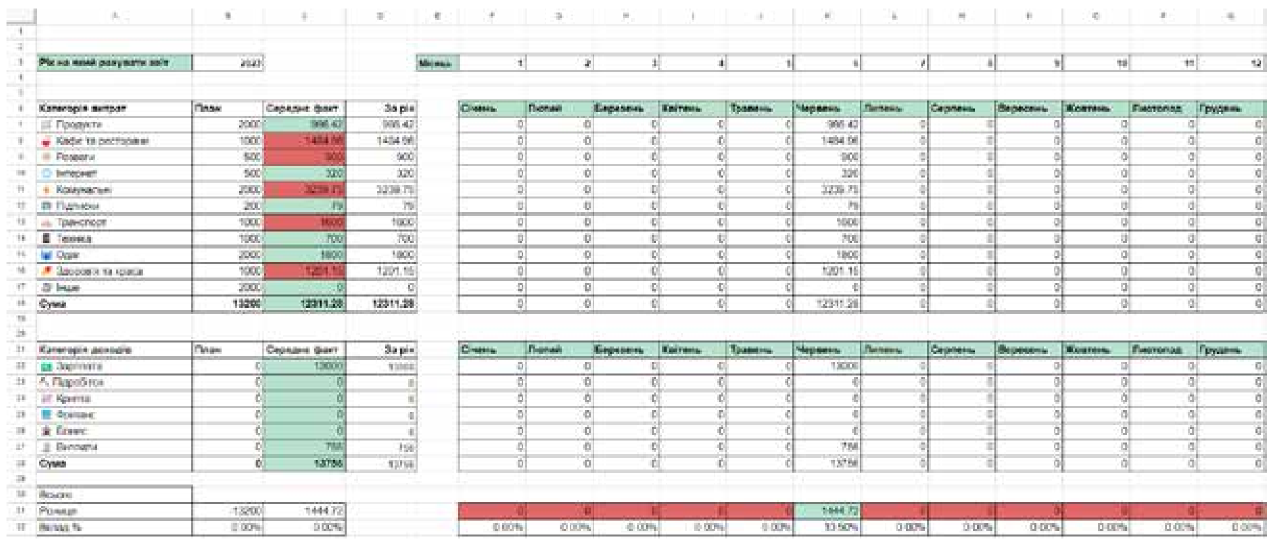


Витрачено за місяць		12311.28			
Витрачено за весь час		12311.28	Поточна дата		6/6/2023

Сума за поточний місяць		Сума	Продукти	Кафе та ресторани	Розваги	Інтернет	Комунальні	Підписки	Транспорт	Техніка	Одяг	Здоров'я та краса	Інше
1	Січень	0	0	0	0	0	0	0	0	0	0	0	0
2	Лютий	0	0	0	0	0	0	0	0	0	0	0	0
3	Березень	0	0	0	0	0	0	0	0	0	0	0	0
4	Квітень	0	0	0	0	0	0	0	0	0	0	0	0
5	Травень	0	0	0	0	0	0	0	0	0	0	0	0
6	Червень	12311.28	900.42	1454.96	900	320	3219.75	79	1600	700	1000	1201.15	0
7	Липень	0	0	0	0	0	0	0	0	0	0	0	0
8	Серпень	0	0	0	0	0	0	0	0	0	0	0	0
9	Вересень	0	0	0	0	0	0	0	0	0	0	0	0
10	Жовтень	0	0	0	0	0	0	0	0	0	0	0	0
11	Листопад	0	0	0	0	0	0	0	0	0	0	0	0
12	Грудень	0	0	0	0	0	0	0	0	0	0	0	0

Дата	Категорія	Сума	Користувач	Опис
6/6/2023 21:01:34	Техніка	700	Igor	на ремонт телефону
6/6/2023 21:09:11	Кафе та ресторани	414.96	Igor	
6/6/2023 21:41:45	Продукти	900.42	Вероніка	продукти в дорогу
6/6/2023 21:42:13	Транспорт	1600	Igor	білети до Києва
6/6/2023 21:43:29	Кафе та ресторани	680	Igor	
6/6/2023 21:43:42	Інтернет	320	Igor	оплата моб. тел
6/6/2023 21:44:13	Здоров'я та краса	1201.15	Вероніка	
6/6/2023 21:44:32	Одяг	1000	Вероніка	кросівки
6/6/2023 21:45:40	Розваги	900	Igor	картинг
6/6/2023 21:46:24	Підписки	79	Igor	спотіфай
6/6/2023 21:47:03	Комунальні	3219.75	Вероніка	за тепло
6/6/2023 21:47:28	Кафе та ресторани	350	Вероніка	

Fig. 12. Expenses sheet after data entry by multiple users



Рік на який розрахувати звіт	2023			Місяць	1	2	3	4	5	6	7	8	9	10	11	12

Категорія витрат	План	Середнє факт	За рік	Січень	Лютий	Березень	Квітень	Травень	Червень	Липень	Серпень	Вересень	Жовтень	Листопад	Грудень
Продукти	2000	996.42	996.42	0	0	0	0	0	996.42	0	0	0	0	0	0
Кафе та ресторани	1000	1484.96	1484.96	0	0	0	0	0	1484.96	0	0	0	0	0	0
Розваги	500	900	900	0	0	0	0	0	900	0	0	0	0	0	0
Інтернет	500	320	320	0	0	0	0	0	320	0	0	0	0	0	0
Комунальні	2000	3239.75	3239.75	0	0	0	0	0	3239.75	0	0	0	0	0	0
Підписки	200	79	79	0	0	0	0	0	79	0	0	0	0	0	0
Транспорт	1000	1600	1600	0	0	0	0	0	1600	0	0	0	0	0	0
Техніка	1000	700	700	0	0	0	0	0	700	0	0	0	0	0	0
Одяг	2000	1800	1800	0	0	0	0	0	1800	0	0	0	0	0	0
Здоров'я та краса	1000	1201.15	1201.15	0	0	0	0	0	1201.15	0	0	0	0	0	0
Інше	2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Сума	**13200**	**12311.28**	**12311.28**	0	0	0	0	0	12311.28	0	0	0	0	0	0

Категорія доходів	План	Середнє факт	За рік	Січень	Лютий	Березень	Квітень	Травень	Червень	Липень	Серпень	Вересень	Жовтень	Листопад	Грудень
Зарплата	0	13000	13000	0	0	0	0	0	13000	0	0	0	0	0	0
Підробітки	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Крипта	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Фріланс	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Бізнес	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Виплати	0	758	758	0	0	0	0	0	758	0	0	0	0	0	0
Сума	**0**	**13758**	13758	0	0	0	0	0	13758	0	0	0	0	0	0

Всього														
Різниця	-13200	1444.72	0	0	0	0	0	1444.72	0	0	0	0	0	0
Вклад %	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	10.50%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Fig. 13. Report sheet after data entry by multiple users

To ensure proper operation, the virtual financial assistant must be deployed either on a hosting platform or a local server. For local deployment, the following steps are required. Before launching the project, a virtual environment should be created using the command *python -m venv venv*, and then activated via *\venv\Scripts\activate*. All required dependencies must be installed using the command *pip install -r requirements.txt* within the activated environment. Next, the database should be initialized by executing the script *python db.py*. If necessary, configuration parameters – such as the Telegram bot token and service account credentials – can be modified in the *settings.py* file. Once configuration is complete, the project can be launched using the command *python bot.py*. This approach ensures that the assistant functions correctly in both local and cloud-based environments.

Conclusions. Virtual personal financial assistants serve as effective tools for enhancing financial literacy and ensuring stability at both individual and family levels. They simplify budgeting processes, facilitate income and expense tracking, and support informed financial decision-making.

This study presents the implementation of a Telegram bot integrated with Google Sheets, providing users with convenient and secure access to the functions of a personal financial assistant. The results demonstrate the practical value of such an approach for everyday financial management.

The developed tool has significant potential for further enhancement, particularly through expanding functionality, increasing automation, and integrating artificial intelligence. The proposed solution represents a meaningful step toward the digital transformation of personal finance management.

Bibliography:

1. Розробка фінансових додатків та сервісів від Wezom. URL: https://wezom.com.ua/ua/blog/razrabotka-finansovyh-prilozhenij.

2. Mohanan A., Sabu A., Shinu S.S., Surendran S., ChandraBabu G. Assessing the Development and Use of Virtual Personal Assistants. *International Journal for Research Trends and Innovation (IJRTI)*. 2025. Vol. 10, Issue 1. Pp. a870–a879. URL: https://ijrti.org/papers/IJRTI2501104.pdf.

3. Iovine A., Narducci F., Musto C., de Gemmis M., Semeraro G. Virtual Customer Assistants in Finance: From State of the Art and Practices to Design Guidelines. *Computer Science Review*. 2023. Vol. 47. Article 100534. DOI: 10.1016/j.cosrev.2023.100534.

4. Худолій Ю. С., Косолапенко В. С. Особливості застосування чат-ботів на основі штучного інтелекту у фінансовій сфері. *Економіка і регіон*. 2023. № 3 (90). С. 97–103. DOI: 10.26906/EiR.2023.3(90).3036.

5. Jain M., Srihari A. AI Driven Personal Finance Management Tools. *International Journal of Novel Research and Development* (*IJNRD*). 2024. Vol. 9, Issue 12. P. b822–b831. URL: https://ijrti.org/papers/IJNRD2412187.pdf.

6. Павлюченко Д. М. Вплив штучного інтелекту та машинного навчання на банківські послуги. *Академічні візії*. 2024. Вип. 32. 14 с. DOI: 10.5281/zenodo.12936652.

7. Money Manager Expense & Budget. URL: https://www.realbyteapps.com/

8. 10 фінансових додатків для домашньої бухгалтерії. URL: https://new.minfin.com.ua/ua/ideabank/10-prilozhenij-kotorye-pomogut-razbogatet.

9. WHEREISMYMONEY – PERSONAL FINANCE BOT – TELEGRAM BOT. URL: https://tdirectory.me/bot/whereismymoneybot.dhtml.

10. Telegram global MAU 2024. URL: https://www.statista.com/statistics/234038/telegram-messenger-mau-users.

References:

1. Rozrobka finansovykh dodatkiv ta servisiw vid Wezom. URL: https://wezom.com.ua/ua/blog/razrabotka-finansovyh-prilozhenij.

2. Mohanan, A., Sabu, A., Shinu, S.S., Surendran, S., ChandraBabu, G. (2025). Assessing the Development and Use of Virtual Personal Assistants. *International Journal for Research Trends and Innovation* (*IJRTI*). Vol. 10. Issue 1. 2025. a870 – a879. URL: https://ijrti.org/papers/IJRTI2501104.pdf.

3. Iovine, A., Narducci, F., Musto, C., de Gemmis, M., Semeraro, G. (2023). Virtual Customer Assistants in Finance: From State of the Art and Practices to Design Guidelines. *Computer Science Review*. Vol. 47. 100534. https://doi.org/10.1016/j.cosrev.2023.100534.

4. Khudolii, Yu. S., Kosolapenko, V. S. (2023). Osoblyvosti zastosuvannia chat-botiv na osnovi shtuchnoho intelektu u finansovii sferi. Ekonomika i rehion. 2023. № 3 (90). Pp. 97–103. DOI: 10.26906/EiR.2023.3(90).3036

5. Jain, M., Srihari, A. (2024). AI Driven Personal Finance Management Tools. *International Journal of Novel Research and Development (IJNRD)*, Vol. 9, Issue 12. 2024. b822-b831. URL: https://ijrti.org/papers/IJNRD2412187.pdf.

6. Pavliuchenko, D. M. (2024). Vplyv shtuchnoho intelektu ta mashynnoho navchannia na bankivski posluhy. Akademichni Vizii. Vol. 32, Zenodo. 14 pp. DOI:10.5281/zenodo.12936652.

7. Money Manager Expense & Budget. URL: https://www.realbyteapps.com/

8. 10 finansovykh dodatkiv dlia domashnoi bukhhalterii. URL: https://new.minfin.com.ua/ua/ideabank/10-prilozhenij-kotorye-pomogut-razbogatet.

9. WHEREISMYMONEY – PERSONAL FINANCE BOT – TELEGRAM BOT. URL: https://tdirectory.me/bot/whereismymoneybot.dhtml.

10. Telegram global MAU 2024. URL: https://www.statista.com/statistics/234038/telegram-messenger-mau-users/.