Міністерство освіти і науки України

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

Навчально-науковий інститут інформаційних технологій та робототехніки

Кафедра комп’ютерних та інформаційних технологій і систем

**Пояснювальна записка**

**до курсової роботи**

з дисципліни «Програмування»

на тему«Розробка додатку з використанням мови Python»

Виконав студент гр. 203-ТК

Масоха Максим Русланович

Керівник курсової роботи

к.ф-м.н., доцент

Двірна.О.А.

Дата захисту:

« » \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 р.

Оцінка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Полтава 2024 р.

ЗАВДАННЯ

Розробити та реалізувати мовою Python програму - телефонний довідник, що містить такі відомості: ПІБ, адреса, електронна пошта, телефон (мобільний, домашній). Дані зберігаються в csv файлі.

Функції роботи довідника:

- додавання нових контактів;

- редагування існуючих контактів. При цьому на одного абонента може припадати кілька номерів телефону;

- перегляд існуючих контактів. Контакти повинні виводитися у відсортованому порядку;

- видалення контакту;

- пошук за параметрами – Прізвище, номер телефону.

Програма повинна містити довідку, де описуються основні можливості програми та інструкція для користувача.

ЗМІСТ

ВСТУП……………………………………………………………………………………………………………………….4

РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА……………………………………………………………………..7

1.1. Технології розробки телефонного довідника……………………………..7

1.2. Мова створення додатку Python…………………………………………..9

РОЗДІЛ 2. ПРАКТИЧНА ЧАСТИНА……………………………………………………………………..12

2.1. Створення коду додатка…………………………………………………...12

2.2. Застосування додатку……………………………………………………...17

ВИСНОВКИ……………………………………………………………………………………………………………..24

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..……………………………………………………………….25

ВСТУП

Розробка телефонного довідника є актуальною незалежно від часу, оскільки збереження та керування контактами є важливою частиною організації інформації для багатьох людей та організацій. Незважаючи на те, що існують багато різних додатків телефонних довідників та онлайн-сервісів, самостійна розробка такого додатку має свої переваги, такі як можливість налаштування під власні потреби та контроль за збереженням особистих даних.

Крім того, навички розробки програмного забезпечення залишаються важливими і корисними навіть у сучасному світі, де існують різні інструменти та платформи для створення програм. Такі проекти допомагають вдосконалювати навички програмування, а також розвивати розуміння процесу розробки програмного забезпечення в цілому.

Таким чином, незважаючи на те, що телефонні довідники можуть здаватись старомодними в епоху смартфонів та онлайн-сервісів, їх розробка залишається актуальною і корисною для тих, хто хоче розвивати свої навички програмування та створювати корисні програми для себе або для інших.

Мета роботи полягає в створенні функціонального телефонного довідника з графічним інтерфейсом користувача, який дозволить користувачам зберігати, організовувати та керувати своїми контактами.

Основні цілі цього проекту включають:

* створення зручного і простого у використанні інтерфейсу: забезпечення можливості легкого введення, перегляду, редагування та видалення контактів без необхідності використання складних команд або процедур;
* збереження та організація даних: забезпечення зручного способу збереження та організації контактної інформації користувачів для подальшого використання;
* реалізація ключових функцій: Включення основних функцій, таких як додавання, редагування, видалення та пошук контактів, для забезпечення повноцінного функціоналу додатку;
* навчання та розвиток: набуття навичок програмування та розвиток розуміння процесу створення програмного забезпечення через практичний досвід розробки такого проекту;
* сприяння організації та продуктивності: надання користувачам зручного інструмента для керування своїми контактами, що допомагає зберігати, організовувати та знаходити необхідну інформацію швидко та ефективно.

Отже, мета цього проекту - створити простий, функціональний та корисний інструмент для керування контактами, який може бути використаний користувачами для покращення організації свого життя та підвищення продуктивності.

Дослідження методів у контексті розробки телефонного довідника може включати наступні аспекти:

1. Оцінка існуючих рішень: дослідження різних існуючих телефонних довідників, які доступні на ринку, для аналізу їхніх функцій, інтерфейсу та користувацького досвіду. Це дозволить зрозуміти, які функції користувачі цінують найбільше та як можна покращити дизайн інтерфейсу.
2. Оцінка технічних можливостей: дослідження різних технологій та інструментів розробки програмного забезпечення для вибору найбільш підходящих для створення телефонного довідника. Це може включати аналіз мов програмування, фреймворків для розробки графічного інтерфейсу та методів зберігання даних.
3. Проведення тестування: розробка та виконання тестів для перевірки правильності роботи програми телефонного довідника. Це допоможе виявити та виправити будь-які помилки або недоліки в програмі перед її випуском.
4. Оцінка впливу на користувача: дослідження впливу використання телефонного довідника на продуктивність та організацію користувачів. Це може включати опитування користувачів для збору фідбеку щодо їхнього досвіду використання додатку.
5. Аналіз використання: моніторинг та аналіз поведінки користувачів в програмі для виявлення популярних функцій, частоти використання та інших показників ефективності. Це дозволить вдосконалити додаток на основі реальних потреб користувачів.

Дослідження цих аспектів дозволить краще зрозуміти потреби користувачів та покращити якість та корисність телефонного довідника.

РОЗДІЛ 1

ТЕОРЕТИЧНА ЧАСТИНА

1.1. Технології розробки телефонного довідника

Опис технологій для розробки телефонного довідника може включати такі складові:

Мова програмування: вибір мови програмування залежить від потреб проекту. Найпоширеніші мови для розробки десктопних додатків, зокрема з графічним інтерфейсом, це Python, Java, C# та C++.

Фреймворк для графічного інтерфейсу: у розробці графічних десктопних додатків часто використовують фреймворки, що спрощують створення інтерфейсу. Для Python це може бути Tkinter, PyQt або wxPython.

Зберігання даних: для зберігання даних в телефонному довіднику можна використовувати різні методи, такі як реляційні бази даних (наприклад, SQLite), файлові формати (наприклад, CSV або JSON) або навіть хмарні сервіси.

Тестування: для тестування програмного забезпечення можна використовувати різні інструменти та бібліотеки. У випадку Python це можуть бути бібліотеки, такі як unittest або pytest.

Розгортання: після розробки додатку важливо забезпечити його розгортання на різних платформах. Для десктопних додатків це може включати створення виконуваних файлів (exe для Windows, dmg для macOS, deb або rpm для Linux) або розміщення додатку в магазині додатків.

Додаткові інструменти: у розробці програм можуть використовуватися додаткові інструменти для контролю версій (наприклад, Git), редактори коду (наприклад, Visual Studio Code або PyCharm) та інші допоміжні інструменти.

Ці технології можуть бути використані для розробки телефонного довідника, проте кінцевий вибір залежить від конкретних потреб проекту та вподобань.

Для зберігання даних в телефонному довіднику можна використовувати різні методи, в залежності від складності проекту, потреб користувачів та інших факторів. Ось деякі методи зберігання даних, які можна розглянути:

Реляційні бази даних (SQL):

* SQLite: Легка вбудовувана база даних, яка зберігається в одному файлі. Вона ідеально підходить для невеликих додатків.
* MySQL, PostgreSQL, Oracle: Більш потужні серверні бази даних, які підходять для більш складних додатків з багатою функціональністю.

Файлові формати:

* CSV (Comma-Separated Values): Простий текстовий формат, у якому дані розділені комами або іншими роздільниками. Його легко обробляти та зберігати, але він не підтримує складні структури даних.
* JSON (JavaScript Object Notation): Формат обміну даними, який підтримує складні структури, такі як об'єкти та масиви. JSON зазвичай використовується для зберігання структурованих даних.

Інші методи:

* XML (eXtensible Markup Language): Інший формат обміну даними, який дозволяє зберігати структуровані дані у вигляді тексту. Хоча XML менш популярний у сучасних додатках, він все ще використовується у деяких сценаріях.
* NoSQL бази даних: Для додатків з великим обсягом даних або потребою у високій швидкості обробки можна використовувати NoSQL бази даних, такі як MongoDB або Cassandra.

Вибір конкретного методу зберігання даних залежить від рівня складності додатку, обсягу даних, потреб користувачів та інших вимог проекту.

1.2. Мова створення додатку Python

Python – динамічна інтерпретована об’єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Розроблена в 1990 році голандським програмістом Гвідо ван Россумом.

Автор назвав мову на честь популярного британського комедійного серіалу 1970-х років «Літаючий цирк Монті Пайтона». Python (укр. Пайтон) — високорівнева мова програмування, посідає третє місце, поступаючись лише JavaScript та HTML/CSS, серед найпопулярніших технологій в опитуванні Stack Overflow. Однак вона не завжди була така популярна. Історія Python починається наприкінці 80-х з мови програмування ABC. Вона була створена в Амстердамі в Centrum Wiskunde & Informatica, щоб люди без досвіду могли швидко розпочати програмування. На жаль, вона не спрацювала належним чином. Пізніше Гвідо ван Россум взяв цю невдалу мову для створення Пайтона. Офіційна дата народження Python — 20 лютого 1991 року, коли Гвідо опублікував першу версію 0.9.0. У січні 1994 року було запущено версію 1.0 з інструментами для функціонального програмування.

Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення.

Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту.

У мови програмування динамічна типізація: є можливість передавати до функцій будь-який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Пайтоні оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об’єкта можна запитати під час виконання програми. Для кожного з об’єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

* у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
* гнучкість та масштабованість Пайтона дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність.
* Пайтон – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
* у Python — колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив’язаність» до системних бібліотек.

Мова програмування Пайтон використовується у наступних сферах:

* розробка програмних застосунків будь-якого напрямку;
* розробка серверної частини мобільних застосунків (найпопулярніший напрямок);
* ігри. Багато сучасних ігор для комп’ютерів (наприклад, World of Tanks) частково чи повністю написані на Python;
* вбудовані системи для різних пристроїв. Дуже часто Пайтон використовують для написання внутрішніх платформ управління банкоматами;
* скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень;
* тестування (автоматизація цього процесу).

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витрачати час на розписування дій, які повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проєкту.

Серед найпопулярніших фреймворків для Python:

Django — найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;

Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;

Web2py — підходить в першу чергу для вебзастосунків і може використовуватись на будь-яких архітектурах.

РОЗДІЛ 2

ПРАКТИЧНА ЧАСТИНА

2.1. Створення коду додатка

Імпорт бібліотек (Рисунок 2.1.1.):

* csv – Використовується для роботи з файлами CSV.
* os – Надає доступ до функцій операційної системи, таких як перевірка наявності файлу.
* tkinter – Використовується для створення графічного інтерфейсу користувача (GUI).
* ttk – Модуль ttk в Tkinter надає розширені віджети з більш сучасними стилями.
* messagebox – Використовується для відображення віконних повідомлень у GUI.

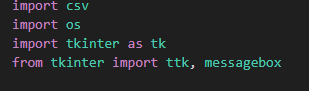


Рисунок 2.1.1 – Імпорт бібліотек

Шлях до файлу бази даних (Рисунок 2.1.2.) вказує шлях до файлу, в якому зберігаються дані про контакти.



Рисунок 2.1.2 – Шлях до файлу бази даних

Функція для створення бази даних (Рисунок 2.1.3.) перевіряє наявність файлу бази даних. Якщо файл не існує, створює новий файл CSV та записує заголовок.

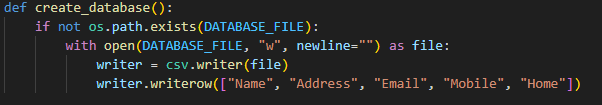


Рисунок 2.1.3 – Функція для створення бази даних

Функція для додавання нового контакту (Рисунок 2.1.4.) отримує дані з текстових полів введення та додає новий запис до файлу бази даних CSV.

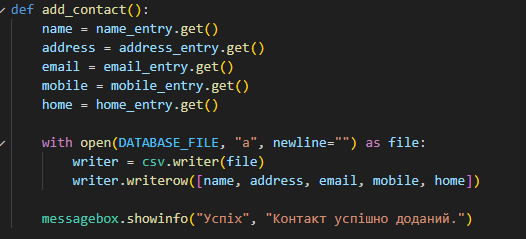


Рисунок 2.1.4 – Функція для додавання нового контакту

Функція для редагування існуючого контакту (Рисунок 2.1.5.) отримує ім'я контакту, новий мобільний та домашній телефони, зчитує всі контакти з файлу, редагує відповідний запис та зберігає зміни.

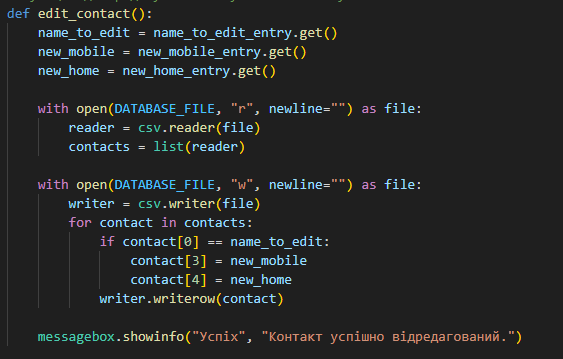


Рисунок 2.1.5 – Функція для редагування існуючого контакту

Функція для перегляду всіх контактів (Рисунок 2.1.6.) зчитує дані з файлу бази даних та відображає їх у вікні повідомлень. Якщо довідник порожній, виводиться відповідне повідомлення.

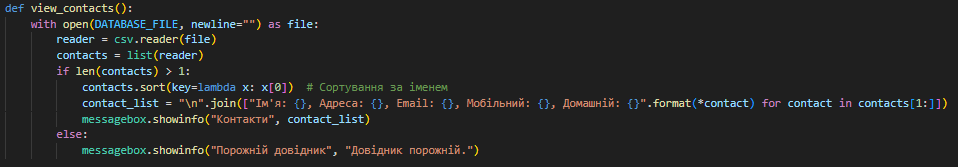


Рисунок 2.1.6 – Функція для перегляду всіх контактів

Функція для видалення контакту (Рисунок 2.1.7.) отримує ім'я контакту для видалення, зчитує всі контакти з файлу, видаляє відповідний запис та зберігає зміни.

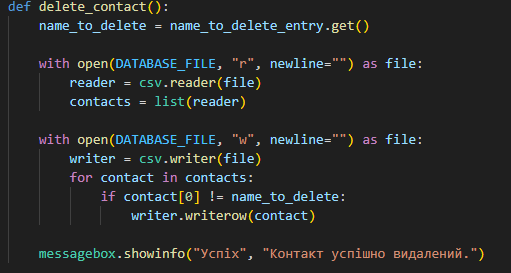


Рисунок 2.1.7 – Функція для видалення контакту

Функція для пошуку контакту за параметрами (Рисунок 2.1.8.) отримує параметр пошуку, зчитує всі контакти з файлу та шукає контакти, що відповідають критеріям. Виводить знайдені контакти у вікні повідомлень.

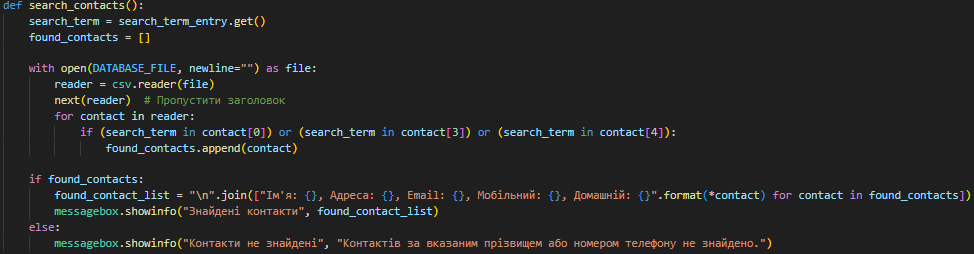


Рисунок 2.1.8 – Функція для пошуку контакту за параметрами

Створення вікна Tkinter та розміщення елементів (Рисунок 2.1.9., Рисунок 2.1.10.)

* створює вікно Tkinter з назвою "Телефонний довідник";
* створює та розміщує елементи у вікні, такі як мітки (Label), текстові поля введення (Entry), кнопки (Button) та навігаційні панелі (Frame).

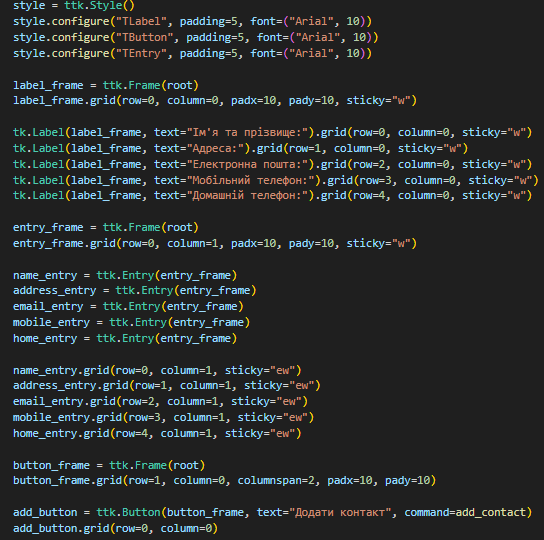


Рисунок 2.1.9 – Створення вікна Tkinter та розміщення елементів

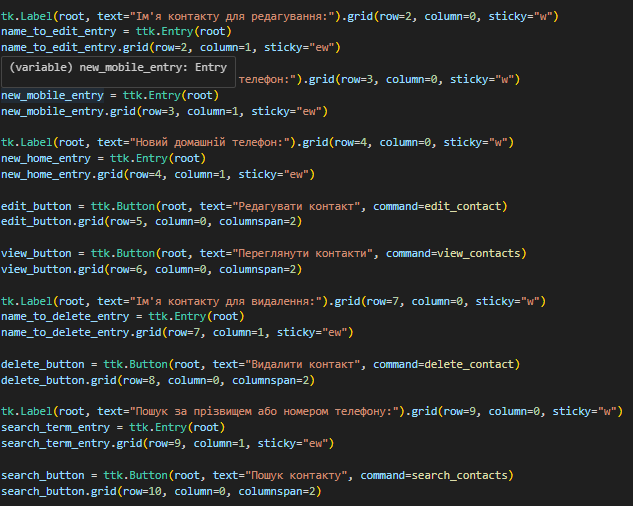


Рисунок 2.1.10 – Створення вікна Tkinter та розміщення елементів

Запуск головного циклу Tkinter (Рисунок 2.1.11.) запускає головний цикл Tkinter, який очікує подій користувача та обробляє їх.



Рисунок 2.1.11 – Запуск головного циклу Tkinter

2.2. Застосування додатку

Запуск додатку проводиця через бібліотеку Tkinter.

Додавання нового контакту:

* користувач вводить ім'я, адресу, електронну пошту, мобільний та домашній телефон (Рисунок 2.2.12.);
* натискає кнопку "Додати контакт";
* дані про нового контакту зберігаються у файлі бази даних CSV;
* відображається вікно повідомлення з підтвердженням успішного додавання контакту (Рисунок 2.2.13.).

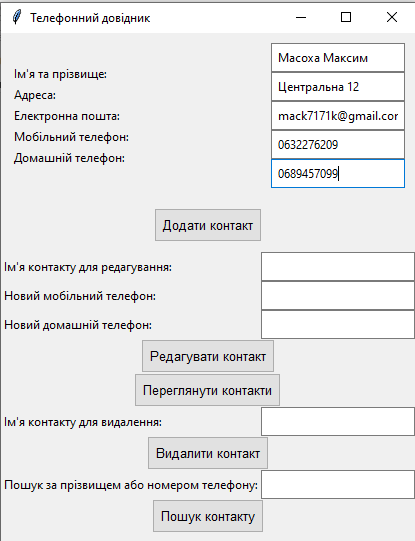


Рисунок 2.2.12 – Введення даних

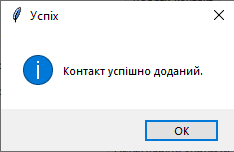


Рисунок 2.2.13 – Вікно успішного додавання контакту

Редагування існуючого контакту (edit\_contact):

* користувач вводить ім'я контакту для редагування, новий мобільний та домашній телефон (Рисунок 2.2.14.);
* натискає кнопку "Редагувати контакт";
* зчитуються всі контакти з файлу бази даних;
* відповідний запис редагується згідно введених даних;
* зміни зберігаються у файлі бази даних CSV;
* відображається вікно повідомлення з підтвердженням успішного редагування контакту (Рисунок 2.2.15.).

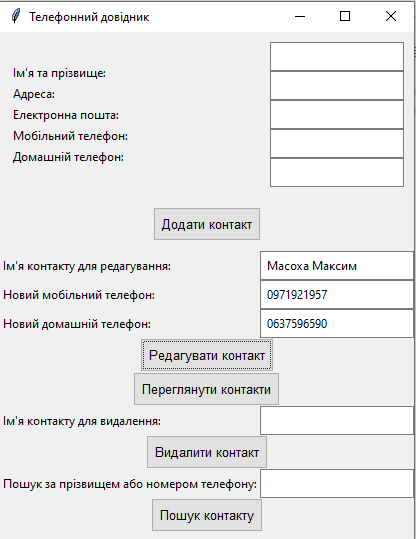


Рисунок. 2.2.14 – Введення даних для редагування

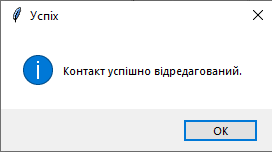


Рисунок 2.2.15 - Вікно успішного редагування контакту

Перегляд усіх контактів:

* натискається кнопка "Переглянути контакти";
* зчитуються всі контакти з файлу бази даних;
* контакти сортуються за ім'ям;
* виводяться дані про всі контакти у вікні повідомлення (Рисунок 2.2.16.).

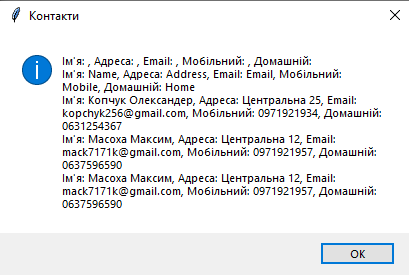


Рисунок 2.2.16 – Контакти сортуються за ім'ям та виводяться дані про всі контакти

Видалення контакту:

* користувач вводить ім'я контакту для видалення (Рисунок 2.2.17.);
* натискається кнопка "Видалити контакт";
* зчитуються всі контакти з файлу бази даних;
* видаляється відповідний запис;
* зміни зберігаються у файлі бази даних CSV;
* відображається вікно повідомлення з підтвердженням успішного видалення контакту (Рисунок 2.2.18.).

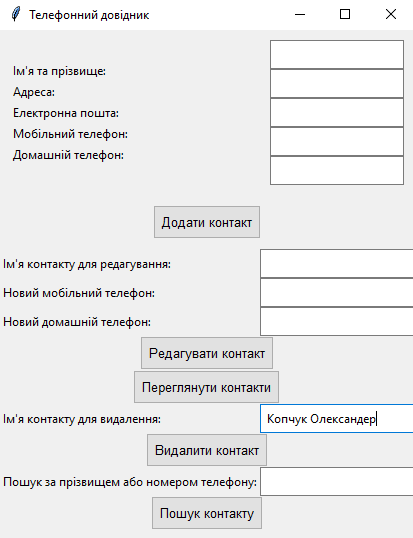


Рисунок 2.2.17 - Вводим ім'я контакту для видалення

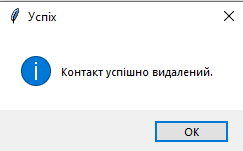


Рисунок 2.2.18 – вікно успішного видалення контакту

Пошук контакту за параметрами:

* користувач вводить параметр пошуку (ім'я або номер телефону) (Рисунок 2.2.19.);
* натискається кнопка "Пошук контакту";
* зчитуються всі контакти з файлу бази даних;
* шукаються контакти, які відповідають введеному параметру;
* виводяться знайдені контакти у вікні повідомлення (Рисунок 2.2.20.).

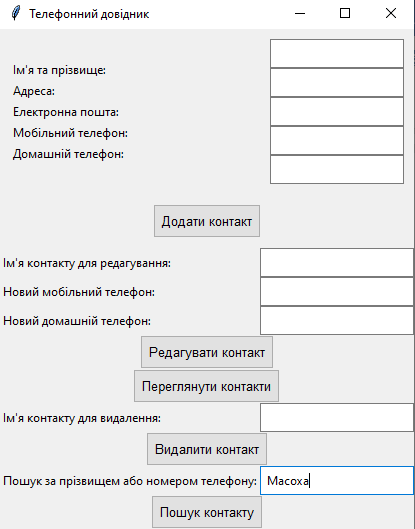


Рисунок 2.2.19 – Параметр пошуку

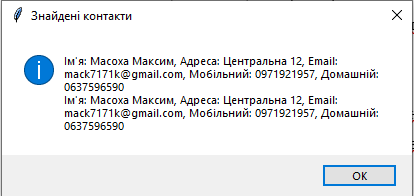


Рисунок 2.2.20 – Виводяться знайдені контакти

ВИСНОВКИ

У даній курсовій роботі був розроблений телефонний довідник з графічним інтерфейсом, який дозволяє користувачам додавати, редагувати, переглядати, видаляти та шукати контакти.

Додаток було реалізовано з використанням мови програмування Python та бібліотеки Tkinter для створення графічного інтерфейсу.

Основні функції додатку включають додавання нових контактів, редагування існуючих контактів, перегляд існуючих контактів, видалення контактів та пошук контактів за прізвищем або номером телефону. Для зберігання даних використовується файл бази даних у форматі CSV.

Програма надає зручний інтерфейс для управління контактами, що дозволяє користувачам легко керувати своїм списком контактів.

Додаток може бути використаний як основа для подальших розширень та розвитку, наприклад, додавання функцій сортування, фільтрації або імпорту/експорту даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is Python? [Електронний ресурс] – Режим доступу до ресурсу: https://www.python.org/doc/essays/blurb/.

2. Alternative Python Implementations [Електронний ресурс] – Режим доступу до ресурсу: https://www.python.org/download/alternatives/.

3. Python data model [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/reference/datamodel.html#the-standard-type-hierarchy>

4. NodeMCU Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://nodemcu.readthedocs.io/en/master/.

5. Docker Docs [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.docker.com/>.

6. Vue.js Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://vuejs.org/v2/guide/>.

7. SQLAlchemy 1.3 Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://docs.sqlalchemy.org/en/13/.

8. Flask-Migrate extension [Електронний ресурс] – Режим доступу до ресурсу: <https://flask-migrate.readthedocs.io/en/latest/>