A PROJECT REPORT

on

LANGUAGE FLASHCARD APPLICATION

Submitted in partial fulfillment of requirements for the award of the course of

### ECA1121 – PYTHON PROGRAMMING

Under the guidance of

#### Dr.V.BRINDA , M.Tech,Ph.D.,

**Assistant Professor/ECE**

*Submitted By*

**HARSHATHA V P(8115U23EC035)**

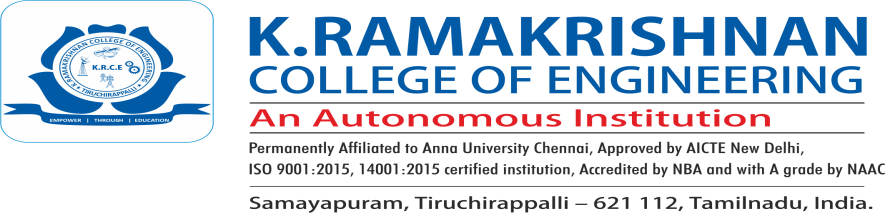
**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

# K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

**SAMAYAPURAM – 621 112**

MAY 2024

# K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

**SAMAYAPURAM – 621 112**

MAY 2024

## BONAFIDE CERTIFICATE

Certified that this project report titled **“LANGUAGE FLASHCARD APPLICATION”** is the bonafide work of **HARSHATHA V P (8115U23EC035),** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| **SIGNATURE**  **Dr.M.Maheshwari M.Tech, Ph.D.,**  **HEAD OF THE DEPARTMENT**  PROFESSOR,  Department of Electronics and Communication Engineering,  K. Ramakrishnan College of Engineering (Autonomous)  Samayapuram – 621 112 | **SIGNATURE**  **Dr. V. Brinda M.Tech, Ph.D.,**  **SUPERVISOR**  ASSISTANT PROFESSOR,  Department of Electronics and Communication Engineering,,  K. Ramakrishnan College of Engineering  (Autonomous)  Samayapuram – 621 112 |

Submitted for the End Semester Examination held on ………

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# DECLARATION

I jointly declare that the project report on **LANGUAGE FLASHCARD APPLICATION** is the result of original work done by us and best of our knowledge, similar work has not been submitted to “ANNA UNIVERSITY CHENNAI” for the requirement of Degree of BACHELOR OF ENGINEERING. This project report is submitted on the partial fulfillment of the requirement of the award of degree of BACHELOR OF ENGINEERING.

|  |
| --- |
| **Signature** |
| HARSHATHA V P |

Place: Samayapuram

Date:

**ACKNOWLEDGEMENT**

It is with great pride that we express our gratitude and indebtedness to our institution, “**K. Ramakrishnan College of Engineering (Autonomous)**”, for providing us with the opportunity to do this project.

We extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

We would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,**

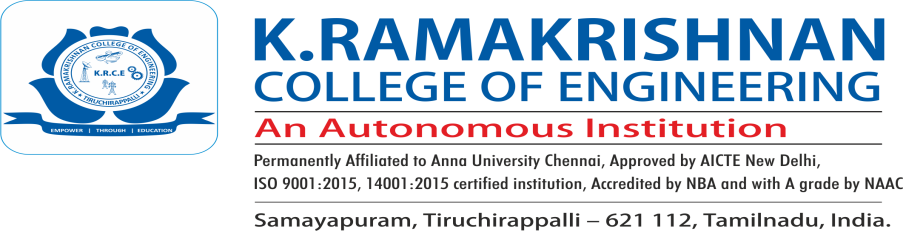
Principal, who gave the opportunity to frame the project to full satisfaction.

We thank **Dr. M. MAHESHWARI M.Tech, Ph.D.,** Head of the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING**, for providing her encouragement in pursuing this project.

We wish to convey our profound and heartfelt gratitude to our esteemed project guide **Dr. V. BRINDA M.Tech, Ph.D.,** Department of **ELECTRONICS AND COMMUNICATION ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

We render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

**DEPARTMENT OF ECE**

**VISION**

To be distinguished as a prominent program in Electronics and Communication Engineering Studies by preparing students for Industrial Competitiveness and Societal Challenges.

**MISSION**

M1. To equip the students with latest technical, analytical and practical knowledge

M2. To provide vibrant academic environment and Innovative Research culture

M3. To provide opportunities for students to get Industrial Skills and Internships to meet out the challenges of the society.

**PROGRAM EDUCATIONAL OBJECTIVES (PEO’S)**

**PEO1**: Graduates will become experts in providing solution for the Engineering problems in Industries, Government and other organizations where they are employed.

**PEO2:** Graduates will provide innovative ideas and management skills to enhance the standards of the society by individual and with team works through the acquired Engineering knowledge.

**PEO3**: Graduates will be successful professionals through lifelong learning and contribute to the society technically and professionally.

.

**PROGRAM SPECIFIC OUTCOMES (PSO’S)**

**PSO1:** Students will qualify in National level Competitive Examinations for Employment and Higher studies

**PSO2:** Students will have expertise in the design and development of Hardware and Software tools to solve complex Electronics and Communication Engine erring problems in the domains like analog and digital electronics, embedded and communication systems

**PROGRAM OUTCOME**

**PO1: Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9: Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large. Some of the mare, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12: Lifelong learning:** Recognize the need for, and have the preparation and lifelong learning in the broadest context of technological change.

### ABSTRACT

A language flashcard application leveraging backend logic in Python offers an efficient and dynamic solution for language learning. The backend, developed using frameworks such as Flask or Django, manages user authentication, flashcard data storage, and retrieval. It supports CRUD operations to create, read, update, and delete flashcards, while implementing algorithms to track and optimize the learning process based on user performance. Python's robust libraries enable seamless integration of features like spaced repetition and machine learning for personalized study plans. The application ensures scalability and real-time updates, enhancing user engagement and learning outcomes.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGENO** |
| **1** | **INTRODUCTION**   * 1. Introduction   2. Objectives   3. Project Summarization |  |
| **2** | **PROJECT METHODOLOGY**   * 1. Introduction to System Architecture   2. Detailed System Architecture Diagram |  |
| **3** | **PYTHON PREFERENCE**   * 1. Explanation of why a dictionary is used   2. Comparison with other Data structures   3. Advantages and disadvantages of using a dictionary |  |
| **4** | **DATA STRUCTURES USED**    4.1 Tuple   * 1. List   4.3 Dictionary |  |
| **5** | **MODULES**   * 1. Flashcard declaration module   2. Review module   3. Input module   4. Quiz module   5. Testing or Display module |  |
| **6** | **ERROR MANAGEMENT**  6.1 Input Validation  6.2 Exception handling |  |
| **7** | **RESULT & DISCUSSION**  7.1 Result  7.2 Discussion |  |
|  | **CONCLUSION & FUTURE SCOPE**  8.1 Conclusion  8.2 Future Scope |  |
|  | **REFERENCES**  **APPENDIX – A** |  |

**LIST OF FIGURES**

**FIGURE NO FIGURE NAME PAGE NO**

2.1 Architecture Diagram

## CHAPTER 1 - INTRODUCTION

#### Introduction

#### simple language flashcard application built with Flask, SQLAlchemy, and WTForms. Users can add new flashcards through a web form, with each card having a front and back side for language learning. The application includes a Flask backend for handling HTTP requests, SQLAlchemy for database interactions, and WTForms for form validation. Data is stored in an SQLite database, and an HTML template renders the form for user input. This setup provides a straightforward and efficient way to manage and add language flashcards.

#### Purpose and Importance

The primary purpose of this project is to provide a simple and efficient web-based platform for users to create, store, and manage language flashcards, facilitating an effective method for language learning and memorization. Built with Flask, SQLAlchemy, and WTForms, the application allows users to easily add flashcards through a web form, with each card containing a front and back side. Data is stored in an SQLite database, ensuring easy access and management. The streamlined setup aims to enhance the learning experience by offering a user-friendly interface and backend support.

Tuples, being immutable, are ideal for storing fixed data like configuration settings and constant flashcard attributes, ensuring data integrity and faster access. Dictionaries, with their key-value structure, provide efficient storage and retrieval of flashcard data, user preferences, and application settings. They allow for quick lookups and updates, enhancing the overall performance and scalability of the application. Utilizing these data structures ensures robust and efficient data management, critical for a smooth user experience.

#### Objectives

1. Flashcard Creation and Management

2. Effective Learning Experience

3. User Progress Tracking

4. User Authentication and Data Security:

#### 1.3 Project Summarization

The Language Flashcard Application is a Python-based tool designed to facilitate efficient language learning through customizable flashcard management and adaptive study modes. Users can create, edit, and categorize flashcards containing vocabulary, definitions, and examples, enhancing their ability to memorize and comprehend language concepts. The application features interactive quiz modes that employ spaced repetition algorithms to optimize learning retention based on user performance. Progress tracking metrics provide insights into individual learning achievements, supported by secure authentication and data encryption for user privacy. Accessible across devices via a web-based platform, the application aims to provide a seamless and personalized learning experience for language learners of all levels.

This project demonstrates several key principles and capabilities in software development and educational technology:

**Effective Learning Tools:** By implementing spaced repetition algorithms and adaptive quiz modes, the project showcases effective learning techniques that are backed by cognitive science research.

**Backend Logic and Data Management:** Utilizing Python with frameworks like Flask or Django for the backend, the project demonstrates robust data management practices. It incorporates features for creating, editing, and organizing flashcards, as well as tracking user progress.

**Cross-Platform Accessibility:** Designed as a web-based application, the project underscores the importance of accessibility across different devices and platforms.

**Educational Impact:** Ultimately, the project highlights the potential of technology to facilitate language acquisition and educational progress. By offering a structured and interactive learning environment, it supports users in mastering vocabulary and language concepts effectively, fostering continuous improvement and achievement in language skills.

**User-Centered Design:** Through customizable features such as flashcard categorization and personalized progress tracking, the project emphasizes a user-centered approach.

# CHAPTER 2

# PROJECT METHODOLOGY

#### Introduction to System Architecture

System architecture refers to the conceptual model that defines the structure, behavior, and more views of a system. It serves as a blueprint for both the system and the project developing it. In the context of Language Flashcard Application , the system architecture outlines how various components interact and work together to achieve the desired functionality.

#### High-Level System Architecture

The high-level system architecture for the language flashcard application typically consists of several key components:

1. User Interface (UI)
2. Application Logic

#### Components of the System Architecture

1. **User Interface (UI)**

The User Interface is the layer with which the end-users interact. It can be a command-line interface (CLI) or a graphical user interface (GUI). The UI component handles user inputs and displays the results. The input handling is the processes user inputs such as Creation and Editing, Progress Tracking .The Output display shows the results of user content such as Progress Tracking and Review Mode Navigation.

1. **Application logic**

The Application Logic is the core of the application where all processing is done. This layer handles several core functionalities that are crucial for its operation and usability.

**Flashcard Management:** Logic for creating new flashcards with fields such as term, definition, example sentence, and optional media.

**Learning Modes:** Logic to present flashcards in a quiz format, manage user responses, and provide feedback on correctness.

1. **Data Management Layer**

The Data Management Layer is responsible for the language flashcard application includes tables for storing flashcards, user information, and progress tracking, managed through Python functions interfacing with an SQLite database.

**1.2 Detailed System Architecture Diagram**

Include a diagram that visually represents the system architecture. The diagram should depict how each component interacts with the others. For example, it can show the User Interface sending requests to the Application Logic, which in turn interacts with the Data Management Layer and the Storage Layer.

+-------------------------------------------+

| User Interaction |

| (Frontend UI - HTML/CSS/JS) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| HTTP Request |

| (Client to Server) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| Web Server |

| (Flask/Django) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| API Endpoint Handler |

| (Routing & Request Handling) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| Application Logic |

| (Python Functions - Backend) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| Data Access Layer |

| (DAL - SQL Queries/ORM) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| Database |

| (SQLite/MySQL) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| Data Storage/Retrieval |

| (SQL Commands) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| HTTP Response |

| (Server to Client) |

+-----------------------+-------------------+

|

v

+-----------------------+-------------------+

| Updated User Interaction |

| (Frontend UI - Updated) |

+-------------------------------------------+

**Fig 2.1 : Architecture Diagram**

# CHAPTER 3

**PYTHON PREFERENCE**

#### Explanation of why a dictionary was chosen

#### Python is a dynamic, high-level, free open source, and interpreted programming language. It supports object-oriented programming as well as procedural-oriented programming. The dictionary data structure was likely chosen for the language flashcard application due to its efficiency in storing and retrieving key-value pairs, which closely aligns with the needs of managing flashcards where each card typically consists of a term (key) and its corresponding definition or example (value). This choice offers fast access times for both reading and updating data, essential for a responsive and scalable application handling potentially large volumes of flashcard entries.

#### Features in Python

#### 1. Free and Open Source: Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword. Download Python Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

#### 2. Easy to code: Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language. 3. Easy to Read

#### 4. Object-Oriented Language: One of the key features of Python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.

#### 5. GUI Programming Support: Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in Python. PyQt5 is the most popular option for creating graphical apps with Python.

#### 6. High-Level Language: Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

#### 7. Large Community Support: Python has gained popularity over the years. Our questions are constantly answered by the enormous StackOverflow community. These websites have already provided answers to many questions about Python, so Python users can consult them as needed.

#### 8. Easy to Debug: Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program’s issues once you understand how to interpret Python’s error traces. Simply by glancing at the code, you can determine what it is designed to perform.

#### 9. Python is a Portable language: Python language is also a portable language. For example, if we have Python code for Windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

#### 10. Python is an Integrated language: Python is also an Integrated language because we can easily integrate Python with other languages like C, C++, etc.

#### 11. Interpreted Language: Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile Python code this makes it easier to debug our code. The source code of Python is converted into an immediate form called bytecode.

#### 12. Large Standard Library : Python has a large standard library that provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in Python such as regular expressions, unit-testing, web browsers, etc.

#### 13. Dynamically Typed Language: Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don’t need to specify the type of variable.

#### 14. Frontend and backend development: With a new project py script, you can run and write Python codes in HTML with the help of some simple tags <py-script>, <py-env>, etc. This will help you do frontend development work in Python like javascript. Backend is the strong forte of Python it’s extensively used for this work cause of its frameworks like Django and Flask.

#### 15. Allocating Memory Dynamically: In Python, the variable data type does not need to be specified. The memory is automatically allocated to a variable at runtime when it is given a value.

# CHAPTER -4

**DATA STRUCTURE METHODOLOGY**

#### 4.1 Dictionary:

#### A dictionary is a collection that stores data in key-value pairs, allowing fast retrieval of values based on unique keys. Suitable for storing flashcards where each term (key) is associated with its definition or example (value). Provides efficient lookup times for rapid access and modification of flashcard content.

#### 4.2 Sets:

#### Sets store unique elements without duplicates, offering operations such as union, intersection, and difference.

#### Useful for managing unique tags or categories associated with flashcards. Ensures each tag is unique and facilitates efficient grouping and filtering operations.

#### 4.3 Lists:

#### Lists are ordered collections that can contain elements of varying data types, providing flexibility in managing sequential data.

#### Useful for organizing lists of flashcard sets, user progress logs, or history. Supports operations like appending, slicing, and iterating through elements.

# CHAPTER-5 MODULES

**1.Adding Flashcards Module**

Function Name: add\_flashcard

Description: Allows users to create and add new flashcards to their collection. Users input the term, definition, example sentence, pronunciation, and optional media (such as images or audio). The function validates inputs and stores the new flashcard in the database.

**2.Reviewing Module**

Function Name: review\_flashcards

Description: Facilitates the review of flashcards based on a spaced repetition algorithm. Retrieves flashcards from the database, prioritizing cards based on the user's performance and schedule. Displays flashcards for review and allows users to mark their recall confidence, adjusting the next review time accordingly.

**3.Quiz Module**

Function Name: start\_quiz

Description: Initiates a quiz session where users are presented with flashcards in a randomized order. The quiz module retrieves flashcards from the database, presents the term or definition, and prompts users to recall or recognize the corresponding information. Tracks user responses, provides immediate feedback on correctness, and calculates quiz scores.

# CHAPTER – 6 ERROR MANAGEMENT

#### 6.1. Input Validation

Input validation plays a pivotal role in software development, ensuring the reliability, security, and stability of applications. In the context of error management, robust input validation mechanisms are crucial for handling and preventing potential issues arising from incorrect, malformed, or malicious user inputs. Within the realm of software development using tools like Visual Studio, implementing effective input validation strategies involves scrutinizing and verifying user inputs to ensure they meet predefined criteria and conform to expected formats before processing.

This process involves various techniques such as range checks, data type validation, length validation, format validation (e.g., email addresses, phone numbers), and input sanitization to prevent injection attacks like SQL injection or cross-site scripting (XSS). For instance, Visual Studio supports the integration of validation libraries and frameworks, enabling developers to perform comprehensive checks on user inputs, reducing the likelihood of vulnerabilities and improving the overall robustness of the application.

By applying stringent input validation mechanisms throughout the codebase, developers can fortify their applications against potential errors, exceptions, and security threats stemming from invalid inputs. Furthermore, incorporating error handling routines and providing informative feedback to users when invalid inputs are detected not only enhances user experience but also aids in diagnosing and rectifying input-related issues, contributing to the overall reliability and resilience of software systems developed within the Visual Studio environment.

Ultimately, meticulous input validation serves as a critical component of error management, preemptively addressing potential pitfalls related to user inputs and reinforcing the integrity and security of software applications.

#### Exception handling

Exception handling in data structures is a critical aspect of software development, addressing unforeseen errors that may occur during operations. It encompasses various error types, such as array out-of-bounds access or operations on empty data structures. By throwing exceptions in response to errors, developers can prevent runtime failures and enable graceful recovery. Utilizing try-catch blocks allows for the isolation of error-prone code and the implementation of custom strategies for handling exceptions. This not only enhances the overall stability of the application but also facilitates debugging and maintenance by providing informative error messages. Additionally, custom exceptions can be defined for specific scenarios, offering more granular control over error handling. Well-documented exception handling practices guide developers on effectively addressing errors, contributing to the creation of robust, reliable, and user-friendly software systems.

* 1. **Test Cases**
     1. **Positive Test Cases**

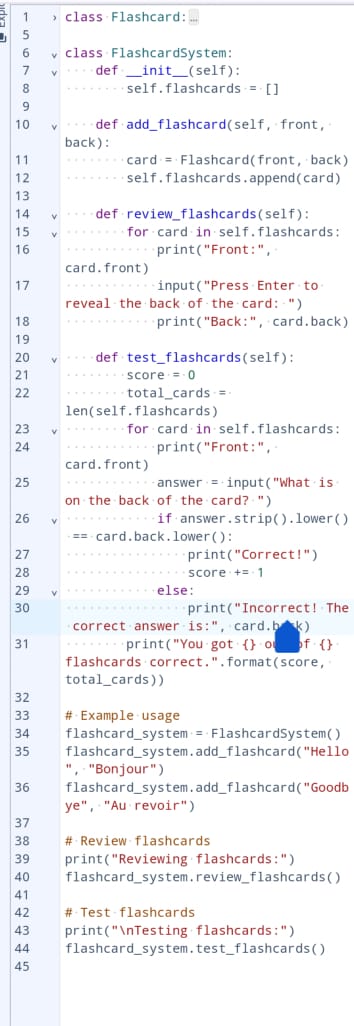
Adding a Flashcard

* + 1. **Negative Test Cases**

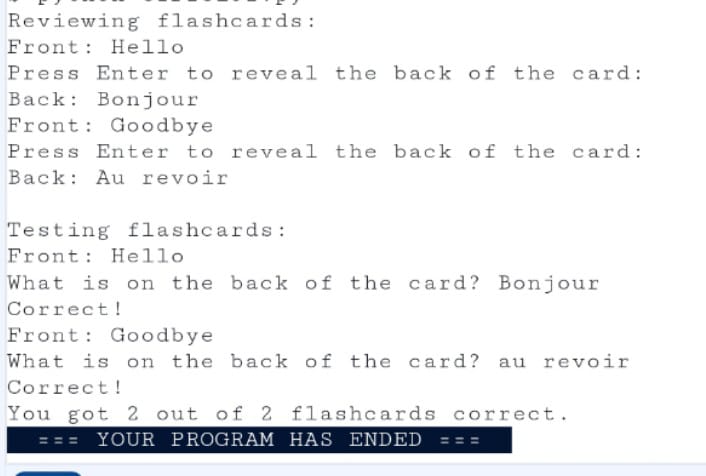
Reviewing a Flashcard

### CHAPTER – 7 RESULT AND DISCUSSION

* 1. **Results**

****

**OUTPUT**

****

* 1. **Discussion**

This language flashcard application project serves as a versatile tool for enhancing language learning through interactive and systematic study techniques. By leveraging features such as adding flashcards, reviewing modules based on spaced repetition algorithms, and interactive quizzes, the application supports users in building vocabulary, improving comprehension, and retaining language concepts effectively. The modular design ensures flexibility and scalability, allowing for seamless integration of new features and enhancements. With a robust backend implemented in Python, utilizing data structures like dictionaries and SQL databases, the application ensures efficient data management and retrieval. User-friendly interfaces facilitate intuitive navigation and engagement, promoting an optimal learning experience. Overall, this project not only aids in structured language learning but also demonstrates effective use of technology to foster continuous improvement and mastery of language skills.

#### Future Scope

In the future, this language flashcard application could expand by incorporating advanced features such as multimedia support for richer learning experiences, personalized spaced repetition algorithms leveraging machine learning, and integration with social learning aspects for collaborative study. Enhancements could also include gamification elements to enhance user motivation, mobile accessibility with offline capabilities, and analytics for tracking progress and offering personalized learning insights. By integrating AI and NLP technologies, the application could further enhance its functionality with automated language assessment and real-time translation support. Continuous updates based on user feedback and trends in language education would ensure the application remains relevant and effective in facilitating comprehensive language learning and skill development.

REFERENCES

1. Anurag Gupta, IPS Jharkhand, GP Biswass, “ Python Programming, Problem Solving, Packages and Libraries”,McGraw Hill Education (India) Private Limited, 2019 ISBN-13:978-93-5316-800-1

2. Reema Theraja , “Python Programming: Using Problem Solving Approach” Oxford Higher Education,2017.

3. Gowrishankar S, Veena A, “Introduction to Python Programming”, 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372

4.​ https//:w3schools.com

# APPENDIX

# class Flashcard

# def review\_flashcards(self)

# for card in self.front

# print("Front:", card.front)

# input("Press Enter to reveal the back of the card: ")

# print("Back:", card.bdef test\_flashcards(self):

# score = 0

# total\_cards = len(self.flashcards)

# for card in self.flashcards:

# print("Front:", card.front)

# answer = input("What is on the back of the card? ")

# if answer.strip().lower() == card.back.lower():

# print("Correct!")

# score += 1

# else:

# print("Incorrect! The correct answer is:", card.back)

# print("You got {} out of {} flashcards correct.".format(score, total\_cards))

# # Example usage

# flashcard\_system = FlashcardSystem()

# flashcard\_system.add\_flashcard("Hello", "Bonjour")

# flashcard\_system.add\_flashcard("Goodbye", "Au revoir")

# # Review flashcards

# print("Reviewing flashcards:")

# flashcard\_system.review\_flashcards()

# # Test flashcards

# print("\nTesting flashcards:")

# flashcard\_system.test\_flashcards()

# 