**Project Report**

**ON**

**FLASH CARD: A Memorization Technique.**

**Submitted by**

**Divya V. V.**

**Table of Contents**

**Chapter No Title Page**

**1 Abstract 3**

**2 Software Description 4**

**3 Code Segment 5**

**4 Sample Screenshots 9**

**5 Conclusion 15**

**6 Bibliography 16**

**1.Abstract**

This project is **titled “FLASH CARD: A Memorization Technique”** aims to develop a memorization technique. Flash cards are widely used to memorize new topics. A question or keyword or title is written in one side of card and its answer or description will be on the other side of the card. Then we need to go through the card repeatedly and test oneself to check if we learned the concept.

Objectives:

1. Creating Flashcard.

2. Adding Flashcard title

3. Adding description for the flashcard

4.Deleting Flashcard.

5.Updating Flashcard.

**2. Software Description**

**2.1 python:**

Python is a widely used programming language. It is easy to use and uses very simple syntax that adds to its popularity. Python efficiently handles high level data with its library developed for specific purposes.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Website, <https://www.python.org/>.

**2.2 Vscode :**

Visual Studio Code is one of most popular code editors, it is a lightweight but powerful editor which runs desktop and is available for Windows, macOS and Linux. It comes with built-in support many popular programming languages.

Vscode can be downloaded from <https://code.visualstudio.com/>

**2.3 Django:**

Django is a high-level Python web framework designed to help developers build complex web applications rapidly and securely.

The primary source for documentation, tutorial and information for Django is <https://www.djangoproject.com/>

**3. Code Segment**

**Views.py**

# Create your views here.

from django.shortcuts import render, get\_object\_or\_404, redirect

from .models import Flashcard

from .forms import FlashcardForm

def flashcard\_list(request):

    flashcards = Flashcard.objects.all()

    return render(request, 'flashcards/flashcard\_list.html', {'flashcards': flashcards})

def flashcard\_detail(request, pk):

    flashcard = get\_object\_or\_404(Flashcard, pk=pk)

    return render(request, 'flashcards/flashcard\_detail.html', {'flashcard': flashcard})

def flashcard\_create(request):

    if request.method == 'POST':

        form = FlashcardForm(request.POST)

        if form.is\_valid():

            form.save()

            return redirect('flashcard\_list')

    else:

        form = FlashcardForm()

    return render(request, 'flashcards/flashcard\_form.html', {'form': form})

def flashcard\_edit(request, pk):

    flashcard = get\_object\_or\_404(Flashcard, pk=pk)

    if request.method == 'POST':

        form = FlashcardForm(request.POST, instance=flashcard)

        if form.is\_valid():

            form.save()

            return redirect('flashcard\_list')

    else:

        form = FlashcardForm(instance=flashcard)

    return render(request, 'flashcards/flashcard\_form.html', {'form': form})

def flashcard\_delete(request, pk):

    flashcard = get\_object\_or\_404(Flashcard, pk=pk)

    flashcard.delete()

    return redirect('flashcard\_list')

**flashcard\_details.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>{{ flashcard.term }}</title>

    <style>

        .flashcard {

            width: 300px;

            height: 200px;

            border: 1px solid #000;

            display: flex;

            align-items: center;

            justify-content: center;

            cursor: pointer;

        }

    </style>

</head>

<body>

    <div class="flashcard" onclick="toggleDefinition()">

        <div id="card-content">{{ flashcard.term }}</div>

    </div>

    <script>

        function toggleDefinition() {

            var content = document.getElementById('card-content');

            if (content.innerHTML === "{{ flashcard.term }}") {

                content.innerHTML = "{{ flashcard.definition }}";

            } else {

                content.innerHTML = "{{ flashcard.term }}";

            }

        }

    </script>

    <a href="{% url 'flashcard\_list' %}">Back to list</a>

</body>

</html>

**flashcard\_list.html**

<html>

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Flashcards</title>

    <style>

        body {

            font-family: Arial, sans-serif; /\* Basic font styling \*/

            margin: 20px; /\* Add margin around the page \*/

        }

        .flashcard-item {

            display: flex;

            align-items: center; /\* Center align items vertically \*/

            margin-bottom: 10px; /\* Space between each flashcard item \*/

            border: 1px solid #ddd; /\* Light border around the flashcard item \*/

            border-radius: 8px; /\* Rounded corners \*/

            background-color: #f9f9f9; /\* Light background color \*/

            padding: 10px; /\* Padding inside the flashcard item \*/

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /\* Subtle shadow for 3D effect \*/

        }

        .flashcard-term-box {

            flex: 0.1; /\* Takes up more space \*/

            padding: 10px; /\* Padding inside the term box \*/

            border: 1px solid #ddd; /\* Light border around the term box \*/

            border-radius: 8px; /\* Rounded corners \*/

            background-color: #d1e7dd; /\* Light green background color \*/

            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /\* Subtle shadow for 3D effect \*/

            margin-right: 10px; /\* Space between term box and actions \*/

        }

        .flashcard-actions {

            display: flex;

            gap: 5px; /\* Space between action boxes \*/

        }

        .flashcard-action-box {

            flex: 1; /\* Each action box takes equal space \*/

            padding: 5px; /\* Padding inside the action box \*/

            border: 1px solid #ddd; /\* Light border around the action box \*/

            border-radius: 8px; /\* Rounded corners \*/

            background-color: #e9ecef; /\* Light gray background color for action box \*/

            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /\* Subtle shadow for 3D effect \*/

            text-align: center; /\* Center text inside the action box \*/

            font-size: 14px; /\* Font size for action text \*/

        }

        .flashcard-action-box:nth-child(1) {

            background-color: #f8d7da; /\* Light red background color for View Details \*/

        }

        .flashcard-action-box:nth-child(2) {

            background-color: #fff3cd; /\* Light yellow background color for Edit \*/

        }

        .flashcard-action-box:nth-child(3) {

            background-color: #d4edda; /\* Light green background color for Delete \*/

        }

        .flashcard-action-box a {

            display: block;

            text-decoration: none; /\* Remove underline from links \*/

            color: #007bff; /\* Link color \*/

        }

        .flashcard-action-box a:hover {

            text-decoration: underline; /\* Underline on hover for better UX \*/

        }

    </style>

</head>

<body>

    <h1>Flashcards</h1>

    <a href="{% url 'flashcard\_create' %}">Add New Flashcard</a>

    <ul style="list-style: none; padding: 0;">

        {% for flashcard in flashcards %}

            <li class="flashcard-item">

                <div class="flashcard-term-box">

                    {{ flashcard.term }}

                </div>

                <div class="flashcard-actions">

                    <div class="flashcard-action-box">

                        <a href="{% url 'flashcard\_detail' flashcard.pk %}">View Details</a>

                    </div>

                    <div class="flashcard-action-box">

                        <a href="{% url 'flashcard\_edit' flashcard.pk %}">Edit</a>

                    </div>

                    <div class="flashcard-action-box">

                        <a href="{% url 'flashcard\_delete' flashcard.pk %}">Delete</a>

                    </div>

                </div>

            </li>

        {% endfor %}

    </ul>

</body>

</html>

**4. Sample Screenshots**

**A screenshot of a computer

Description automatically generated**

**Fig 1: Home page**

**A screenshot of a computer

Description automatically generated**

**Fig 2: Adding card details**

**A screenshot of a computer

Description automatically generated**

**Fig 3: Viewing front side of the flashcard.**

**A screenshot of a computer

Description automatically generated**

**Fig 4: Viewing the backside of the flashcard**

**A screenshot of a flashcard

Description automatically generated**

**Fig 5: Editing Flashcard**

**5. Conclusion and Future Scope**

The project is completed within the stipulated time. All the functionalities are tested and found to be working properly.

Future enhancement

1. Paragraph summarization, such that system itself generate flashcard based on the data provided.
2. Grouping flashcard manually and automatically based on the category or interest or relevance or relationship or topics.
3. Categorize flashcard based on if the content is learned or not.

**6.** **Bibliography**

**1.** [**https://docs.python.org/3.14/library/index.html**](https://docs.python.org/3.14/library/index.html)

**2.** [**https://www.djangoproject.com/**](https://www.djangoproject.com/)

**3.** [**https://www.geeksforgeeks.org/build-a-flashcards-using-django/#**](https://www.geeksforgeeks.org/build-a-flashcards-using-django/)

**4.** [**https://www.w3schools.com/css/**](https://www.w3schools.com/css/)