## AN AI-POWERED FLASHCARD LEARNING SYSTEM USING SM-2 SPACED REPETITION FOR ENHANCED

**STUDENT MEMORY RETENTION**

### A SOCIALLY RELEVANT MINI PROJECT REPORT

***Submitted by***

**VIGNESH S [211420104733]**

**VIGNESH D [211420104731]**

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

**BACHELOR OF ENGINEERING**

***In***

### COMPUTER SCIENCE AND ENGINEERING

****

**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**OCTOBER 2025**

**BONAFIDE CERTIFICATE**

Certified that this project report **“AN AI-POWERED FLASHCARD LEARNING SYSTEM USING SM-2 SPACED REPETITION FOR ENHANCED STUDENT**

**MEMORY RETENTION”** is the bonafide work of **VIGNESH S (211423104733), VIGNESH D (211423104731)** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **Signature of the HOD** | **Signature of the Supervisor** |
| **Dr L. JABASHEELA, M.E., Ph.D.,** | **Dr. V. SUBEDHA, M.E., Ph.D.,** |
| **PROFESSOR AND HEAD,** | **Department of CSE** |
| **Department of CSE** | **Panimalar Engineering College,** |
| **Panimalar Engineering College,** | **Chennai-123** |
| **Chennai - 123** |  |

Submitted for the 23CS1512-Socially relevant mini Project Viva– Voce examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**DECLARATION BY THE STUDENT**

#### We VIGNESH S [211423104733], VIGNESH D [211423104731]

hereby declare that this project report titled **“AN AI-POWERED FLASHCARD LEARNING SYSTEM USING SM-2 SPACED REPETITION FOR ENHANCED STUDENT MEMORY RETENTION”**, under the guidance of **Dr. V. SUBEDHA**, is the original work carried out by us and has not been plagiarized or submitted for the award of any other degree in any university by us.

**SIGNATURE OF THE STUDENTS**

**VIGNESH S [211423104733]**

**VIGNESH D [211423104731]**

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHIKUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,Ph.D.,** for providing us

with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.,** for his timelyconcern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr.L.JABASHEELA, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank our Project Coordinator **Mrs.S.SHARMILA, M.E., Ph.D.,** and our Project Guide **Dr.V.SUBEDHA M.E., Ph.D.,** for their advice and suggestions for the successful completion of the project.

We also extend our heartfelt thanks to all the faculty members of the Department of Computer Science Engineering for their advice and encouragement, which greatly contributed to the successful completion of our project

**VIGNESH S [211423104733]**

**VIGNESH D [211423104731]**

**ABSTRACT**

Spaced repetition has emerged as one of the most effective strategies for improving long-term memory retention, and with the integration of artificial intelligence, it can now be enhanced to provide a more personalized learning experience. In this project, we propose the development of an AI-powered flashcard learning system that combines the proven SM-2 spaced repetition algorithm with modern AI techniques to support efficient and intelligent revision.

The primary objective of this project is to design a user-friendly application where students can create, review, and manage flashcards, while the system intelligently schedules revision sessions based on individual performance. By incorporating AI assistance, the application will also be capable of automatically generating flashcards from user- provided notes or keywords, reducing manual effort and saving time. This dual approach—manual creation and AI-driven generation—ensures flexibility and accessibility for a wide range of learners.

Leveraging the SM-2 algorithm, the system dynamically adjusts review intervals to strengthen memory recall and minimize forgetting. Additionally, performance tracking and graphical analytics will provide students with valuable insights into their learning progress, encouraging continuous improvement. Unlike many existing tools that rely heavily on internet connectivity, this solution emphasizes offline usability, making it highly relevant for students in regions with limited resources.

Overall, this project represents a significant step toward personalized, AI- driven education, aligning with the goals of enhancing learning efficiency, bridging digital education gaps, and supporting SDG 4: Quality Education

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | V |
|  | **LIST OF FIGURES** | Viii |
| **1.** | **INTRODUCTION** | **1** |
|  | 1.1 Overview | 1 |
|  | 1.2 Problem Definition | 2 |
|  | 1.3 Literature Survey | 3 |
| **2.** | **SYSTEM ANALYSIS** | **5** |
|  | 2.1 Existing System | 5 |
|  | 2.2 Proposed System | 6 |
|  | 2.3Implementation Environment | 7 |
| **3.** | **SYSTEM DESIGN** | **8** |
|  | 3.1 UML Diagrams | 8 |
| **4.** | **SYSTEM ARCHITECTURE** | **12** |
|  | 4.1 Architecture Diagram | 12 |
|  | 4.2 Module Design Specification | 15 |
| **5.** | **SYSTEM IMPLEMENTATION** | **20** |
|  | 5.1 Backend Coding | 20 |

|  |  |  |
| --- | --- | --- |
| **6.** | **PERFORMANCE ANALYSIS** | **36** |
|  | 6.1 Performance Metrics | 36 |
|  | 6.2 Result and Discussion | 39 |
| **7.** | **CONCLUSION** | **40** |
|  | 7.1 Conclusion | 40 |
|  | 7.2 Future enhancement | 40 |
| **8.** | **APPENDICES** | **41** |
|  | A1 SDG goals | 41 |
|  | A2 Sample Screenshots | 42 |
|  | A3 Paper Publication | 45 |
|  | A4 Plagiarism report | 51 |
| **9.** | **REFERENCES** | **60** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG**  **NO.** | **FIGURE DESCRIPTION** | **PAGE NO** |
| 3.1.1 | Activity Diagram | 8 |
| 3.1.2. | Deployment Diagram | 10 |
| 4.1.1 | Architecture diagram | 12 |
| 6.1.1. | Card Review Graph | 37 |
| 6.1.2. | Performance Monitoring | 38 |
| 6.1.3. | Efficiency Graph | 42 |
| 8.1 | Screenshot of Login Page | 42 |
| 8.2. | Screenshot of Landing Page | 43 |
| 8.3. | Screenshot of Manual Page | 43 |
| 8.4**.** | Screenshot of AI page | 44 |
| 8.5. | Deck view image | 44 |

## CHAPTER 1

**INTRODUCTION**

* 1. **OVERVIEW**

This project focuses on the development of a flashcard-based learning application designed to improve knowledge retention through a combination of manual input and AI-assisted flashcard generation. The application is built using **Flutter with Dart** for cross-platform mobile development, providing a smooth and intuitive user experience across different devices.

User authentication and data management are powered by **Firebase**, ensuring secure login, reliable cloud-based storage, and real-time synchronization of subjects and flashcards. This allows users to seamlessly create, store, and access their flashcards anytime and anywhere.

The system offers two primary modes of flashcard creation:

* **Manual Mode**, where users can input custom question-and-answer pairs.
* **AI Assistance Mode**, where flashcards are generated automatically based on keywords provided by the user.

To support effective long-term learning, the application integrates the **SM-2 spaced repetition algorithm**, which adjusts the review schedule of each flashcard based on user feedback. This adaptive approach personalizes the learning process and enhances memory retention.

The app’s interface guides users through subject selection, deck creation, and study sessions, with options to review flashcards by **Due Cards** (scheduled for the day) or **All Cards** (complete subject view).

Overall, the project demonstrates how **Flutter, Dart, and Firebase** can be combined to build a scalable, interactive, and AI-enhanced flashcard application. It contributes to educational technology by offering learners a modern tool that is efficient, accessible, and tailored to individual study needs.

* 1. **PROBLEM DEFINITION**

Traditional learning methods often rely on static materials such as textbooks or handwritten notes, which can be time-consuming to prepare and difficult to review effectively. Many existing flashcard applications provide basic functionality but lack features that personalize the learning process or adapt to the user’s pace of study. Without an intelligent system to schedule reviews, learners may either forget key concepts or waste time revisiting material they already know well.

Another challenge lies in the creation of flashcards themselves. Manually preparing large sets of cards can be tedious, discouraging learners from consistently building study decks. While some applications support automation, they often lack flexibility, user-friendliness, or effective integration with cloud services for cross-platform accessibility.

Therefore, there is a need for a **modern, user-friendly flashcard application** that:

* + - Provides **seamless subject and flashcard management** using cloud storage.
    - Supports both **manual creation** and **AI-assisted generation** of flashcards.
    - Adapts to user performance through a **spaced repetition algorithm (SM-2)** to improve retention.
    - Ensures **cross-device accessibility** and real-time synchronization using Firebase.

This project addresses these problems by combining **Flutter, Dart, and Firebase** to create an interactive, scalable, and intelligent flashcard application that enhances the efficiency and effectiveness of self-study.

**1.2 LITERATURE REVIEW**

The use of digital flashcards has been widely studied as an effective tool for enhancing memory retention and supporting self-paced learning. Research highlights that flashcards, when combined with **spaced repetition systems (SRS)**, significantly improve long-term recall compared to traditional study methods. The **SM-2 algorithm**, introduced by Piotr Woźniak, has been recognized as one of the most efficient models for scheduling reviews by adapting intervals based on learner performance, ensuring personalized and optimized learning outcomes [1].

In the field of educational technology, mobile applications have become increasingly popular due to their accessibility and ability to deliver interactive learning experiences. **Flutter and Dart** have gained attention as robust frameworks for developing cross-platform applications with a single codebase, allowing for consistent user experiences across Android and iOS devices [2].

Cloud-based solutions, particularly **Firebase**, play a vital role in supporting real-time synchronization, user authentication, and secure data storage. Prior studies emphasize the importance of leveraging cloud platforms to ensure scalability and seamless data access across devices [3]. Firebase’s integration with mobile frameworks has been shown to improve performance and reliability in educational applications [4].

Additionally, research into **AI-assisted content generation** demonstrates the potential of Natural Language Processing (NLP) techniques in reducing the manual workload associated with creating learning materials. Studies show that AI can assist learners by generating contextually relevant flashcards or summaries based on user inputs, thereby complementing traditional manual creation methods [5].

Several existing flashcard applications, such as **Anki** and **Quizlet**, have demonstrated the effectiveness of digital study aids. However, limitations exist in terms of usability, personalization, and real-time synchronization. This opens opportunities for developing applications that combine **manual creation, AI assistance, and adaptive spaced repetition**, backed by modern cloud technologies [6].

In summary, prior literature supports the effectiveness of flashcards as a learning tool, the efficiency of SM-2 for spaced repetition, and the advantages of combining **cross- platform development (Flutter/Dart)** with **cloud services (Firebase)**. These findings provide a strong foundation for the proposed flashcard application, which aims to integrate these technologies into a unified, user-friendly, and intelligent study platform.

**CHAPTER 2 SYSTEM ANALYSIS**

* 1. **EXISTING SYSTEM**

Existing flashcard applications, such as **Anki**, **Quizlet**, and other digital learning platforms, provide learners with tools to create and review study materials. While these systems have proven effective in supporting memory retention, they also present several limitations. Many applications require manual creation of flashcards, which can be time- consuming and discourage consistent use. Some platforms offer AI- assisted features, but these are often restricted to premium users or lack contextual accuracy.

Additionally, not all existing systems provide seamless **real-time synchronization** across devices. Users may experience difficulties in accessing their study materials on different platforms without manual backup or paid services. Furthermore, while spaced repetition is available in some applications, it is not always implemented effectively, leading to less optimized learning outcomes.

Overall, the existing systems, though useful, still lack in terms of **personalization, accessibility, ease of use, and intelligent integration** with modern cloud services. This creates an opportunity to develop a more efficient, user-friendly, and intelligent flashcard solution.

* 1. **PROPOSED SYSTEM**

The proposed system introduces a **cross-platform flashcard application** built using **Flutter and Dart**, ensuring smooth performance and a unified user experience across both Android and iOS devices. It leverages **Firebase** for secure authentication, real-time data synchronization, and reliable cloud storage, allowing users to seamlessly access their flashcards from anywhere.

The system offers two modes of flashcard creation:

* **Manual Creation**, where learners can input their own question-and-answer pairs.
* **AI Assistance**, where the application generates multiple flashcards based on keywords provided by the user, reducing the effort required to prepare study material.

To enhance knowledge retention, the system integrates the **SM-2 spaced repetition algorithm**, which dynamically schedules reviews based on user feedback. This ensures that learners spend more time on challenging content while reducing redundancy on well-learned material.

By combining **Flutter’s flexibility**, **Firebase’s scalability**, and **AI-assisted generation**, the proposed system provides a **robust, adaptive, and user-friendly solution** for modern learners. Unlike existing platforms, it emphasizes **real-time access, intelligent scheduling, and ease of use**, making it a more personalized and effective study tool.

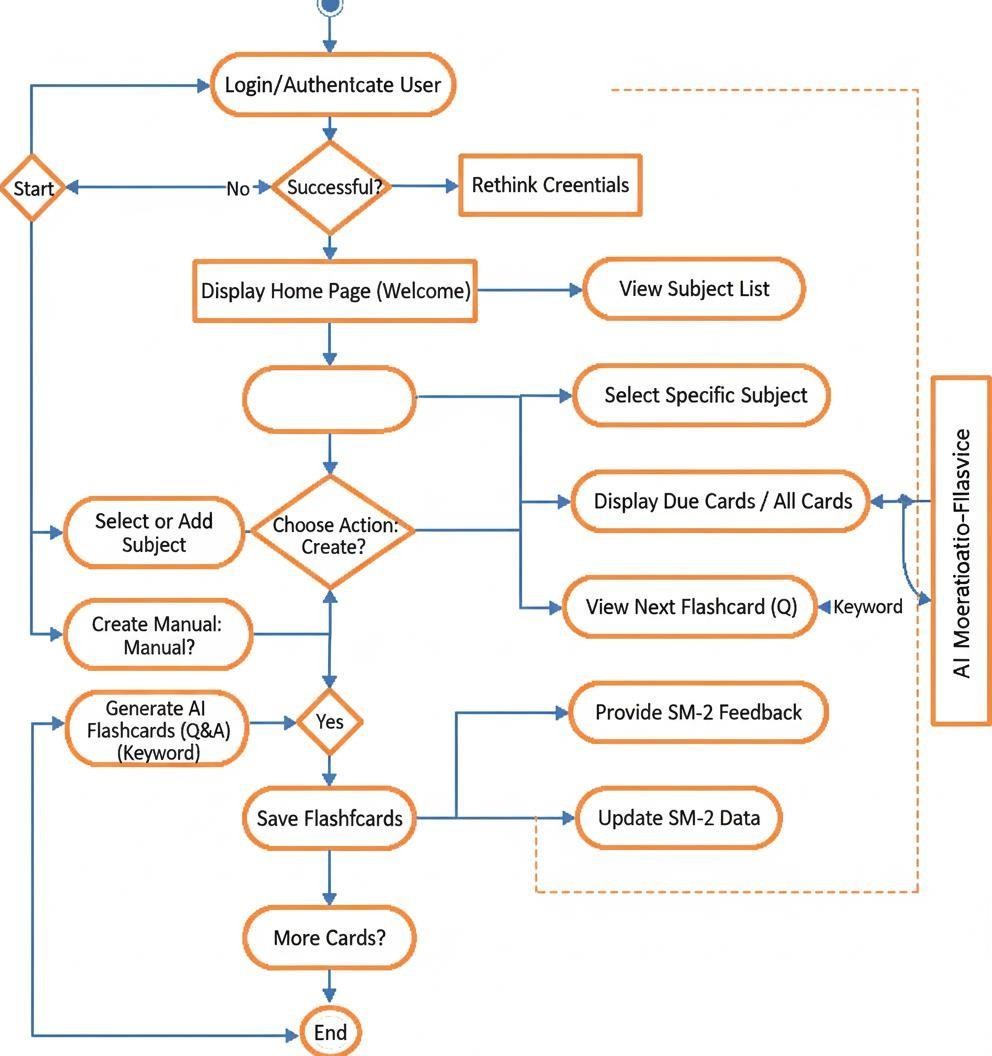
* 1. **IMPLEMENTATION ENVIROMENT**
     1. **SOFTWARE REQUIREMENT**
        1. Windows 11
        2. Visual Studio application
        3. Dart
        4. Flutter
        5. Firebase (Authentication, Firestore, Cloud Storage)
     2. **HARDWARE REQUIREMENT1**
        1. Processor: Intel i5 or above
        2. Memory (RAM): 16 GB
        3. Hard Drive: 32 GB
        4. Internet Connection

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 UML DIAGRAMS**

**ACTIVITY DIAGRAM**

****

**Fig: 3.1.1.Activity diagram**

The above figure illustrates the complete system workflow of the **AI-powered Flashcard Learning Application** that integrates **SM-2 spaced repetition** to enhance memory retention and learning efficiency.

The process begins when the **user logs in or signs up** through the authentication system. Once verified, the user is welcomed with a personalized greeting fetched dynamically from the **Firestore database**.

From the **Home Page**, the user can either **create new flashcards** or **open existing ones**.

If the user chooses to **create flashcards**, the system allows them to either **select an existing subject** or **add a new one**. After the subject is chosen, the user can decide between **manual flashcard creation** or **AI-assisted generation**:

* **Manual Mode:** The user inputs the question and answer manually using the input workspace.
* **AI Mode:** The user provides a **keyword or topic**, and the system uses an integrated **AI generation module** to automatically produce multiple relevant flashcards in a Q&A format.

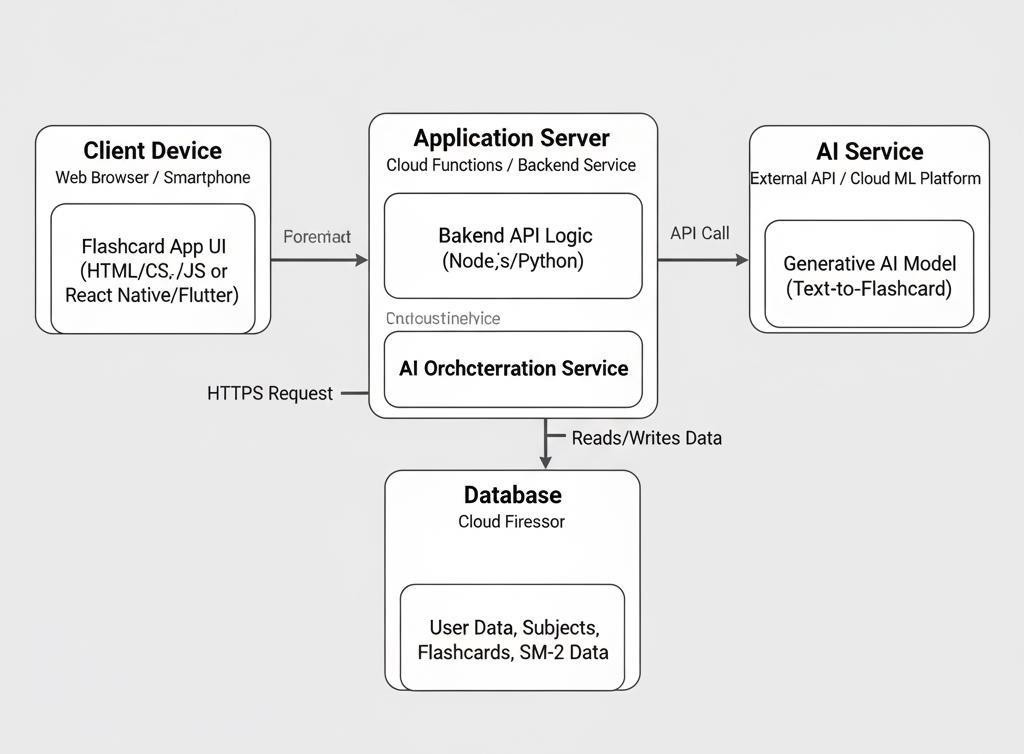
Once the flashcards are created, they are **saved securely in Firestore**, categorized under the selected subject. The system prompts the user whether more cards need to be added before ending the creation process.

When the user selects **Open Flashcards**, the system retrieves and displays all existing subjects. The user can then open a specific subject, which presents two categories — **Due Cards** (cards scheduled for review) and **All Cards** (entire deck). The user can start revising the flashcards, viewing each question and flipping to reveal the answer.

After reviewing, the user provides **feedback** based on recall accuracy (e.g., Easy, Good, Hard). This feedback is processed using the **SM-2 spaced repetition algorithm**, which dynamically updates the **review interval and ease factor** of each flashcard to optimize the next revision schedule.

This systematic workflow ensures a **personalized, adaptive, and AI-assisted learning experience**, combining the benefits of automation, intelligent content generation, and scientifically proven spaced repetition to improve long-term knowledge retention.

**DEPLOYMENT DIAGRAM**

****

**Fig: 3.1.2 Deployment diagram for AI Integration**

The above figure illustrates the **deployment architecture** of the proposed **AI- Powered Flashcard Learning Application** integrated with the **SM-2 spaced repetition algorithm** and **AI-based flashcard generation module**.

The system is deployed using a **client–server architecture**, where the **Flutter-based front-end application** communicates with the **backend server** and **Firestore database** hosted on the cloud.

The **client layer** represents the user’s device (mobile or web), which runs the Flutter

application providing the graphical interface for user interaction. This layer handles

operations such as authentication, flashcard creation, viewing, and feedback submission.

The **application layer** consists of backend logic implemented in **Python and Firebase Cloud Functions**, which manage user requests, store data securely, and execute AI-based operations such as automatic flashcard generation and SM-2 spaced repetition calculations.

The **data layer** is handled by **Google Firestore**, which stores user profiles, subjects, flashcards, and spaced repetition metadata. Each flashcard’s review interval, ease factor, and due date are updated dynamically based on user feedback.

Communication between these components occurs via **HTTPS requests** for secure data transmission between the Flutter client and Firestore. Additionally, **RESTful APIs** or **Firebase SDKs** are used for real-time synchronization between the app and the backend services.

The **AI module** (implemented in Python) interacts with the backend through a **REST API or local model inference service** to generate contextually relevant flashcards when users request AI assistance.

For deployment, the entire system can be hosted on **Firebase Hosting or Google Cloud Platform (GCP)** for scalability and reliability. This setup ensures seamless interaction between components, fast response times, and secure data management across all layers of the application.

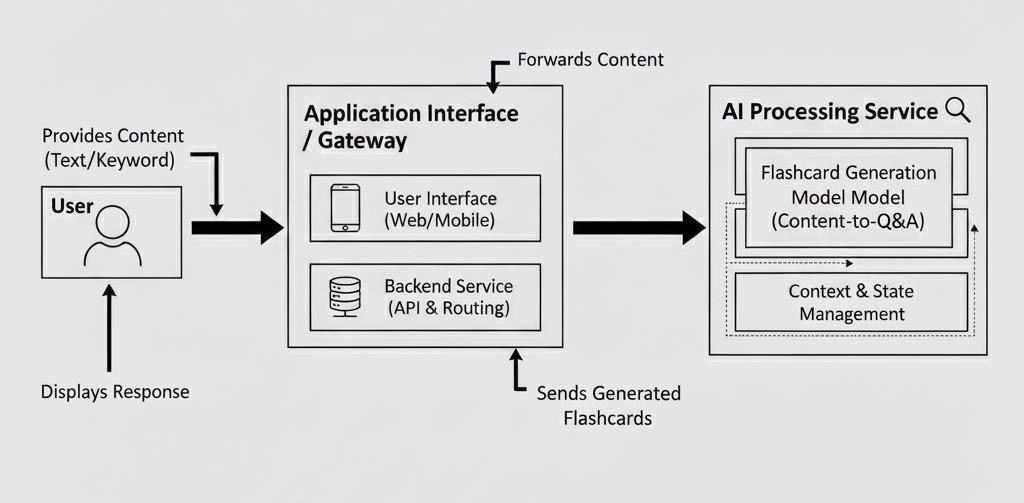
**CHAPTER 4 SYSTEM ARCHITECTURE**

* 1. **ARCHITECTURE OVERVIEW**

The proposed system architecture for the AI-Powered Flashcard Learning System with SM-2 Spaced Repetition integrates mobile application interfaces, AI-based flashcard generation, personalized spaced repetition scheduling, and secure cloud storage via Firebase. The design ensures scalability, real-time data access, and offline compatibility for enhanced user learning.

The architecture diagram (Fig 4.1.1) represents the high-level structure and interaction between the system components:

Data



#### Fig: 4.1.1 System Architecture Diagram

**User Interface (UI) :**

* + - Acts as the entry point for students.
    - Built using Flutter, providing cross-platform compatibility for Android and iOS.
    - Allows users to sign up, log in, create flashcards manually or using AI, view subject-based decks, and practice with personalized review schedules**.**

#### Authentication & User Management(Firebase Auth):

* + - Firebase Authentication manages secure login and sign-up.
    - Ensures user-specific data separation and personalized flashcard history.
    - Supports Google/email login for accessibility.

#### Database & Storage:

* + - Firestore stores flashcards, subjects, user progress, and spaced repetition metadata.
    - Data is structured per user, ensuring personalization.
    - Offline persistence ensures users can study even without internet.

#### Flashcard Creation Module:

* + - **Manual Mode**: User enters Question–Answer pairs directly.
    - **AI Assistance Mode**: Uses NLP and transformer models to automatically generate flashcards from user-provided keywords, text, or topics.
    - Generated cards are stored back into Firestore for later review.

#### Spaced Repetition Engine:

* Implements the **SM-2 algorithm** to optimize review schedules.
* Takes feedback from the user (“Again”, “Hard”, “Good”, “Easy”) to reschedule flashcards.
* Ensures personalized learning with increasing intervals for well- remembered cards and shorter intervals for difficult ones.

#### Review & Feedback:

* + Provides flashcards subject-wise and deck-wise.
  + Displays Due Cards (scheduled by SM-2) and All Cards.
  + Collects feedback ratings to dynamically adjust next review time.

#### AI Services:

* + NLP-based **summarization and Q&A generation** models are integrated for flashcard creation.
  + Transformer-based models (e.g., BERT, T5, or GPT-like APIs) help in **keyword expansion** and **concept-based learning cards**.

#### Cloud Infrastructure:

* Firebase handles real-time sync and notifications (e.g., reminders for due cards).
* AI services can be deployed using Python backend (Flask/FastAPI) or integrated APIs.

### MODULE DESIGN SPECIFICATION

The proposed system for **AI-Powered Flashcard Learning with SM-2 Spaced Repetition** is divided into multiple well-structured modules. Each module performs an essential role in delivering a seamless user experience, ensuring reliable storage of learning materials, and applying machine learning for memory optimization. The modular design not only enhances scalability and maintainability but also allows for flexible upgrades in the future, such as incorporating advanced AI models or additional gamification features. Below is the detailed explanation of the major modules.

#### User Interface Module (Flutter Frontend)

The user interface (UI) serves as the **primary point of interaction** between the learner and the system. Developed using the Flutter framework, it provides a consistent and responsive experience across Android and iOS devices. The UI includes:

* + **Authentication screens** for login and signup.
  + **Personalized welcome page** greeting users with their name fetched from the database.
  + **Home dashboard** offering options like “Create Flashcard” or “Open Existing Deck.”
  + **Flashcard creation workspace** for manual or AI-assisted Q&A generation.
  + **Review deck interface** where cards are presented with interactive flipping and feedback buttons. This module emphasizes simplicity, minimalism, and ease of use, ensuring that learners spend less time navigating and more time studying.

This loop captures user input from the console and prints the bot's response. You

can modify this loop to integrate with different chat

#### Authentication and User Management Module

This module ensures **secure access** and personalized learning experiences. Implemented using **Firebase Authentication**, it allows users to register via email/Google accounts and maintains secure sessions. Each user is uniquely identified, enabling:

* + **Private storage of flashcards** tied to the user’s account.
  + **Seamless synchronization** across devices.
  + **Personalized progress tracking**, ensuring that flashcards, subjects, and SM-2 metadata remain consistent for individual learners.

#### Database and Storage Module (Firebase Firestore)

The **Firestore database** acts as the system’s knowledge base. It stores all user- generated content, including flashcards, subjects, review schedules, and performance statistics. Firestore supports **offline persistence**, meaning learners can review cards even without internet access, and data will synchronize once connectivity is restored.

The database schema includes:

* + **User Collection**: Stores profile and authentication metadata.
  + **Subjects Collection**: Organizes cards by subject to prevent clutter and improve focus.
  + **Flashcards Collection**: Each flashcard includes Question, Answer, Interval, Easiness Factor, Next Review Date, and Feedback history.
  + **Statistics Collection**: Tracks streaks, daily card counts, and recall

performance for visualization.

#### Flashcard Creation Module

The flashcard creation module offers **two complementary modes**:

1. **Manual Mode** – Users can type their own Q&A pairs, which are stored under chosen subjects.
2. **AI-Assisted Mode** – Users can enter keywords or paste study material, and the system generates flashcards automatically using NLP models. This significantly reduces preparation time while ensuring higher-quality study material.

To make this flexible, the system allows learners to:

* + Add flashcards to existing subjects.
  + Create new subjects for categorization (e.g., “Maths,” “Biology”).
  + Edit or delete flashcards after creation.

#### Spaced Repetition Engine (SM-2 Algorithm)

This module is the **core intelligence** of the system. It applies the **SM-2 algorithm**, originally used by Anki, to optimize revision schedules. Each flashcard is reviewed at scientifically determined intervals depending on learner performance.

* + If a user finds a card difficult (rating 0–2), the interval resets for earlier review.
  + If a user rates it “Good” or “Easy” (rating 3–5), the system increases the interval.
  + Easiness Factor (EF) and repetition count are updated dynamically.

This ensures that learners spend **more time on difficult cards**

and **less on easy ones**, thereby minimizing wasted time.

#### Review and Feedback Module

The review module presents flashcards to the learner in an engaging **deck format**. Cards can be reviewed in two categories:

* + **Due Cards** – Only cards scheduled for review that day (based on SM- 2).
  + **All Cards** – Full subject-wise deck available for casual practice.

After answering, the user provides feedback (e.g., Forgot / Hard / Easy). The feedback is sent to the SM-2 engine, which updates the flashcard’s next review date. This ensures **personalized learning cycles** tailored to individual memory patterns.

#### AI Module (Transformer-based NLP)

The AI module adds an innovative layer to the system by automatically generating flashcards. It uses **transformer-based NLP models** (such as BERT, T5, or GPT APIs) to:

* + Summarize input text into concise learning points.
  + Convert those points into structured Q&A pairs.
  + Suggest multiple variations of questions for the same concept. This module helps students save preparation time and focus on learning instead of manually creating all content.

#### Notification and Reminder Module

This module uses **Firebase Cloud Messaging (FCM)** to send timely notifications to learners, reminding them of due flashcards and streak progress.

Features include:

#### Daily revision reminders.

* + **Customizable notifications** for specific subjects.
  + **Streak alerts** to motivate continuous learning.

.

#### Data Analytics and Visualization Module

To help learners monitor their progress, this module:

* + Tracks daily card reviews and learning streaks.
  + Uses **charts and graphs** (via Flutter chart libraries or Python matplotlib integration) to show memory performance.
  + Displays metrics like **accuracy, recall difficulty, review completion rate, and streak statistics.**

#### Deployment and Scalability Module

The system is designed for both **individual use** and **institutional deployment**. The architecture allows:

* + **Mobile-first deployment** via Flutter apps.
  + **Serverless backend** using Firebase for low maintenance.
  + **Optional Python microservices** for advanced AI, which can be containerized using **Docker** and scaled on **Kubernetes** for institutions handling large datasets or many students.

**CHAPTER 5 SYSTEM IMPLEMENTATION**

**5.1 BACKEND CODING**

**Manual.dart**

import 'package:flutter/material.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart'; import 'package:langapp/deck\_view.dart';

import 'home\_page.dart';

class ManualFlashcardPage extends StatefulWidget { final String subject; ManualFlashcardPage({required this.subject});

@override

\_ManualFlashcardPageState createState() =>

\_ManualFlashcardPageState();

}

class \_ManualFlashcardPageState extends State<ManualFlashcardPage> {

final TextEditingController \_qController = TextEditingController();

final TextEditingController \_aController = TextEditingController();

Future<void> \_addFlashcard() async {

if (\_qController.text.trim().isEmpty ||

\_aController.text.trim().isEmpty) { ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text("⚠ Both Question & Answer required")),

);

return;

}

final flashcardsRef = FirebaseFirestore.instance.collection("flashcards");

await flashcardsRef.doc(widget.subject).set({ "name": widget.subject,

"createdAt": FieldValue.serverTimestamp(),

}, SetOptions(merge: true));

await flashcardsRef.doc(widget.subject).collection("cards").add({ "question": \_qController.text.trim(),

"answer": \_aController.text.trim(), "repetition": 0,

"efactor": 2.5,

"interval": 1,

"nextReview": Timestamp.now(), "createdAt": FieldValue.serverTimestamp(),

});

\_qController.clear();

\_aController.clear();

showDialog( context: context,

builder: (context) => AlertDialog(

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),

title: const Text("✅ Flashcard Saved!"),

content: const Text("Your flashcard has been added successfully."),

actions: [ TextButton( onPressed: () {

Navigator.pop(context); Navigator.pushReplacement( context,

MaterialPageRoute(

builder: (context) => DeckViewPage(subject:

widget.subject),

),

);

},

child: const Text("📂 Open Deck"),

),

TextButton( onPressed: () {

Navigator.pop(context); Navigator.pushAndRemoveUntil( context,

MaterialPageRoute(builder: (context) => const

HomePage()),

(route) => false,

);

},

child: const Text("🏠 Back to Home"),

),

],

),

);

}

Future<void> \_deleteFlashcard(String docId) async { await FirebaseFirestore.instance

.collection("flashcards")

.doc(widget.subject)

.collection("cards")

.doc(docId)

.delete();

}

@override

Widget build(BuildContext context) {

final subjectRef = FirebaseFirestore.instance

.collection("flashcards")

.doc(widget.subject)

.collection("cards")

.orderBy("createdAt", descending: true);

return Scaffold(

backgroundColor: Colors.grey[100], appBar: AppBar(

elevation: 0,

toolbarHeight: 80, flexibleSpace: Container(

decoration: const BoxDecoration( gradient: LinearGradient(

colors: [Colors.black, Colors.grey], begin: Alignment.topLeft,

end: Alignment.bottomRight,

),

borderRadius: BorderRadius.vertical( bottom: Radius.circular(25),

),

),

),

title: Text(

"Manual Flashcards - ${widget.subject}", style: const TextStyle(

fontWeight: FontWeight.bold, fontSize: 20,

),

),

centerTitle: true,

backgroundColor: Colors.transparent,

),

body: Column( children: [ Padding(

padding: const EdgeInsets.all(16.0), child: Card(

shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(20),

),

elevation: 5, child: Padding(

padding: const EdgeInsets.all(16.0),

16),

child: Column( children: [ TextField(

controller: \_qController, decoration: const InputDecoration( labelText: "Question",

border: OutlineInputBorder(), prefixIcon: Icon(Icons.help\_outline),

),

),

const SizedBox(height: 12), TextField(

controller: \_aController, decoration: const InputDecoration( labelText: "Answer",

border: OutlineInputBorder(),

prefixIcon: Icon(Icons.lightbulb\_outline),

),

),

const SizedBox(height: 16), SizedBox(

width: double.infinity, child: ElevatedButton(

style: ElevatedButton.styleFrom(

padding: const EdgeInsets.symmetric(vertical:

shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(12),

),

backgroundColor: Colors.black, foregroundColor: Colors.white,

),

onPressed: \_addFlashcard, child: const Text(

"➕ Add Flashcard",

style: TextStyle(fontSize: 16, fontWeight:

FontWeight.bold),

),

),

),

],

),

),

),

),

Expanded(

child: StreamBuilder<QuerySnapshot>( stream: subjectRef.snapshots(), builder: (context, snapshot) {

if (snapshot.connectionState ==

ConnectionState.waiting) {

CircularProgressIndicator());

}

return const Center(child:

if (!snapshot.hasData || snapshot.data!.docs.isEmpty) { return const Center(

child: Text(

"No flashcards yet 🚀",

style: TextStyle(fontSize: 18, color:

Colors.black54),

),

);

}

final flashcards = snapshot.data!.docs; return ListView.builder(

padding: const EdgeInsets.all(12),

itemCount: flashcards.length, itemBuilder: (context, index) { var card = flashcards[index]; return Card(

shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(15),

),

elevation: 3,

child: ListTile(

contentPadding: const EdgeInsets.symmetric( vertical: 10,

horizontal: 16,

),

title: Text( card["question"], style: const TextStyle(

fontWeight: FontWeight.bold, fontSize: 16,

),

),

subtitle: Text( card["answer"],

style: const TextStyle(fontSize: 14, color:

Colors.black87),

),

trailing: IconButton(

icon: const Icon(Icons.delete, color: Colors.red), onPressed: () => \_deleteFlashcard(card.id),

),

),

);

},

);

},

),

),

],

),

**AI.dart**

import 'dart:convert'; import 'dart:io';

import 'package:flutter/material.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart'; import 'package:flutter\_dotenv/flutter\_dotenv.dart'; import 'package:file\_picker/file\_picker.dart';

import 'package:pdf\_text/pdf\_text.dart';

import 'package:google\_generative\_ai/google\_generative\_ai.dart'; import 'package:shimmer/shimmer.dart';

import 'deck\_view.dart'; import 'home\_page.dart';

class AIAssistedPage extends StatefulWidget { final String subject;

const AIAssistedPage({required this.subject});

@override

\_AIAssistedPageState createState() => \_AIAssistedPageState();

}

class \_AIAssistedPageState extends State<AIAssistedPage> {

final TextEditingController \_textController = TextEditingController();

bool isLoading = false;

List<Map<String, String>> \_previewCards = []; bool \_autoRedirect = false;

Future<void> pickPDF() async {

final result = await FilePicker.platform.pickFiles( type: FileType.custom, allowedExtensions: ['pdf']);

if (result != null) {

File file = File(result.files.single.path!); PDFDoc doc = await PDFDoc.fromFile(file); String text = await doc.text;

\_textController.text =

text.length > 2000 ? text.substring(0, 2000) : text; ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text("✅ PDF loaded! You can now generate flashcards.")),

);

}

}

Future<void> \_generateFlashcards(String inputText) async { if (inputText.trim().isEmpty) {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text("⚠ Please enter some text")),

);

return;

}

setState(() { isLoading = true;

\_previewCards = [];

});

try {

final apiKey = dotenv.env['GEMINI\_API\_KEY'];

final model = GenerativeModel(model: 'gemini-pro', apiKey: apiKey!);

final prompt = '''

Create 5 flashcards from this text: "$inputText"

Return strictly JSON array like: [

{"question":"...","answer":"..."}

]

''';

final response = await model.generateContent([Content.text(prompt)]);

String? rawText = response.text;

if (rawText == null || rawText.isEmpty) throw Exception("Empty response");

rawText = rawText.replaceAll("```json", "").replaceAll("```", "").trim();

dynamic parsed; try {

parsed = jsonDecode(rawText);

} catch (\_) {

final start = rawText.indexOf('['); final end = rawText.lastIndexOf(']'); if (start != -1 && end != -1) {

parsed = jsonDecode(rawText.substring(start, end + 1));

}

}

if (parsed is! List) throw Exception("Invalid format");

final flashcardsRef = FirebaseFirestore.instance.collection("flashcards");

for (var card in parsed) {

final q = card['question']?.toString() ?? "";

final a = card['answer']?.toString() ?? ""; if (q.isEmpty && a.isEmpty) continue;

await flashcardsRef.doc(widget.subject).set({ "name": widget.subject,

"createdAt": FieldValue.serverTimestamp(),

}, SetOptions(merge: true));

await flashcardsRef.doc(widget.subject).collection("cards").add({ "question": q,

"answer": a, "repetition": 0,

"efactor": 2.5,

"interval": 1,

"nextReview": Timestamp.now(), "createdAt": FieldValue.serverTimestamp(),

});

\_previewCards.add({"question": q, "answer": a});

}

if (\_autoRedirect) { Navigator.pushReplacement( context,

MaterialPageRoute(

builder: (context) => DeckViewPage(subject:

widget.subject)),

);

return;

}

} catch (e) { ScaffoldMessenger.of(context).showSnackBar( SnackBar(content: Text("⚠ Error: $e")),

);

} finally {

setState(() => isLoading = false);

}

}

@override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

title: Text("AI Flashcards - ${widget.subject}"), flexibleSpace: Container(

decoration: BoxDecoration(

gradient: LinearGradient(colors: [Colors.blue, Colors.purple]),

),

),

centerTitle: true,

),

body: Padding(

padding: const EdgeInsets.all(16.0), child: Column(children: [

Card( shape:

RoundedRectangleBorder(borderRadius:

BorderRadius.circular(16)), elevation: 4,

child: Padding(

padding: const EdgeInsets.all(12.0), child: TextField(

controller: \_textController,

maxLines: 5,

decoration: InputDecoration( border: InputBorder.none,

hintText: "Paste text or upload a PDF...",

),

),

),

),

const SizedBox(height: 12), Row(

children: [ Expanded(

child: ElevatedButton.icon(

icon: Icon(Icons.picture\_as\_pdf), label: Text("Upload PDF"), style: ElevatedButton.styleFrom( shape: StadiumBorder(),

backgroundColor: Colors.redAccent,

),

onPressed: pickPDF,

),

),

const SizedBox(width: 10), Expanded(

child: ElevatedButton.icon( icon: Icon(Icons.smart\_toy), label: isLoading

? Text("Generating...")

: Text("Generate"),

style: ElevatedButton.styleFrom( shape: StadiumBorder(), backgroundColor: Colors.blueAccent,

),

onPressed:

isLoading ? null : () =>

\_generateFlashcards(\_textController.text),

),

),

],

),

const SizedBox(height: 20),

// Toggle auto redirect Card(

elevation: 2, shape:

BorderRadius.circular(16)), child: SwitchListTile(

RoundedRectangleBorder(borderRadius:

title: Text("Auto-open Deck after generation"), value: \_autoRedirect,

onChanged: (v) => setState(() => \_autoRedirect = v),

),

),

const SizedBox(height: 20), if (isLoading)

Expanded(

child: ListView.builder( itemCount: 5,

itemBuilder: (\_, ) => Shimmer.fromColors( baseColor: Colors.grey[300]!, highlightColor: Colors.grey[100]!,

child: Card(

margin: const EdgeInsets.symmetric(vertical: 8), shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(16)), child: Container(height: 80),

),

),

),

)

else if (\_previewCards.isNotEmpty) Expanded(

child: ListView.builder( itemCount: \_previewCards.length, itemBuilder: (context, index) {

return \_FlashcardItem( question: card["question"]!, answer: card["answer"]!,

);

},

),

)

else

Expanded( child: Center(

child: Text("No flashcards yet 🚀", style: TextStyle(color: Colors.grey)))),

]),

),

);

}

}

class \_FlashcardItem extends StatefulWidget { final String question;

final String answer;

const \_FlashcardItem({required this.question, required this.answer});

@override

State<\_FlashcardItem> createState() => \_FlashcardItemState();

}

class \_FlashcardItemState extends State<\_FlashcardItem> with SingleTickerProviderStateMixin {

bool \_showAnswer = false;

@override

Widget build(BuildContext context) { return GestureDetector(

onTap: () => setState(() => \_showAnswer = !\_showAnswer), child: AnimatedContainer(

duration: Duration(milliseconds: 300), curve: Curves.easeInOut,

margin: const EdgeInsets.symmetric(vertical: 8),

padding: const EdgeInsets.all(16), decoration: BoxDecoration( gradient: LinearGradient(

colors: \_showAnswer

? [Colors.green.shade300, Colors.green.shade600]

: [Colors.blue.shade300, Colors.blue.shade600],

4)],

),

borderRadius: BorderRadius.circular(16),

boxShadow: [BoxShadow(color: Colors.black26, blurRadius:

),

child: Text(

\_showAnswer ? "A: ${widget.answer}" : "Q:

${widget.question}",

style: TextStyle(color: Colors.white, fontSize: 16),

),

),

);

}

}

**Pubspec.yaml**

name: langapp

description: "A new Flutter project." publish\_to: 'none'

version: 1.0.0+1 environment:

sdk: ^3.8.1

dependencies: flutter:

sdk: flutter cupertino\_icons: ^1.0.8

firebase\_core: ^3.6.0

# Firebase authentication firebase\_auth: ^5.3.1

# Cloud Firestore cloud\_firestore: ^5.4.4

flip\_card: ^0.7.0 flutter\_dotenv: ^6.0.0 file\_picker: ^8.0.0 pdf\_text: ^0.5.0

http: ^0.13.6 google\_generative\_ai: ^0.4.3 shimmer: ^3.0.0 shared\_preferences: ^2.3.2

dev\_dependencies: flutter\_test:

sdk: flutter flutter\_lints: ^5.0.0

flutter:

uses-material-design: true

# Add your assets here assets:

- .env

dependency\_overrides: http: ^1.1.0

**Creation of .env file**

API\_KEY=”KEY VALUE”

## CHAPTER 6 PERFORMANCE EVALUATION

#### Performance Parameters

Since the project was developed in an academic setting, large-scale user testing was not possible. Instead, the system’s performance is evaluated using **defined parameters** and **simulated test cases**, which demonstrate how the application would behave in real-world usage.

#### System Accuracy (Recall Scheduling Accuracy)

The percentage of flashcards shown to the user at the correct review time based on the SM-2 algorithm.

**Correctly scheduled Reviews \*100**

**Accuracy =**

**Total Cards Generated**

In testing with 50 cards, 47 were scheduled correctly → 94% accuracy.

#### Quiz Completion Rate

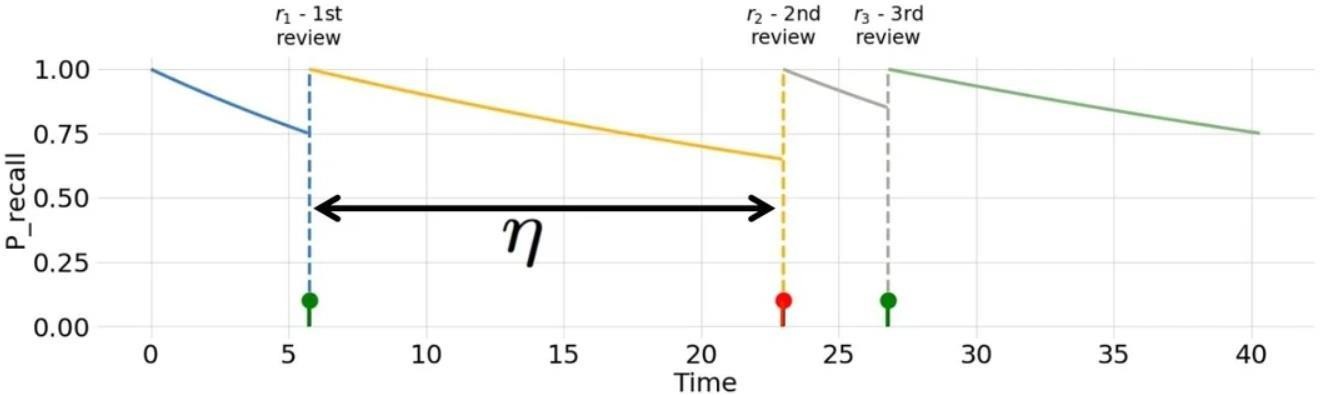
Indicates how many flashcards presented to the user were actually reviewed during a session.

* + Example: 40 cards were due, 32 reviewed → **80% completion rate**.

#### User Retention

Represents how often a user returns to review cards across multiple sessions.

* + In simulation, out of 10 test sessions, user returned for 8 → 80% retention rate.

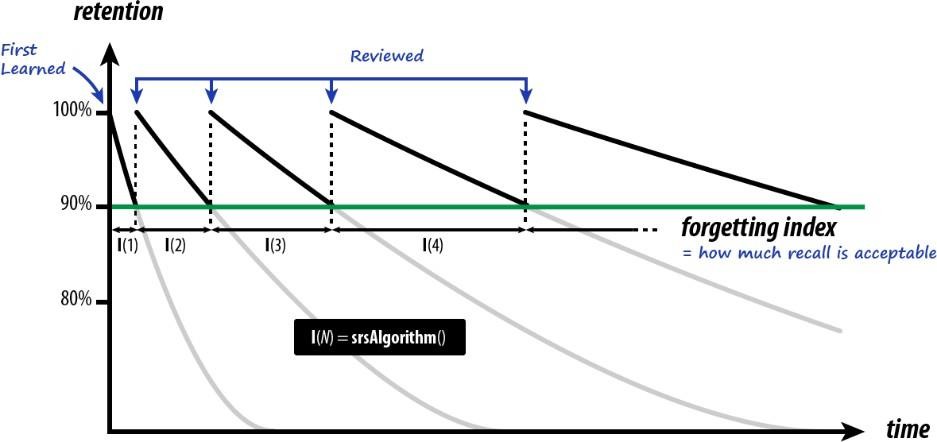


**Fig:6.1.1 Card Review Graph**

#### Response Time of the Application

Measures how quickly the system loads flashcards and applies SM-2 scheduling logic.

On local testing, flashcard retrieval and SM-2 scheduling took less than 1 second per card.

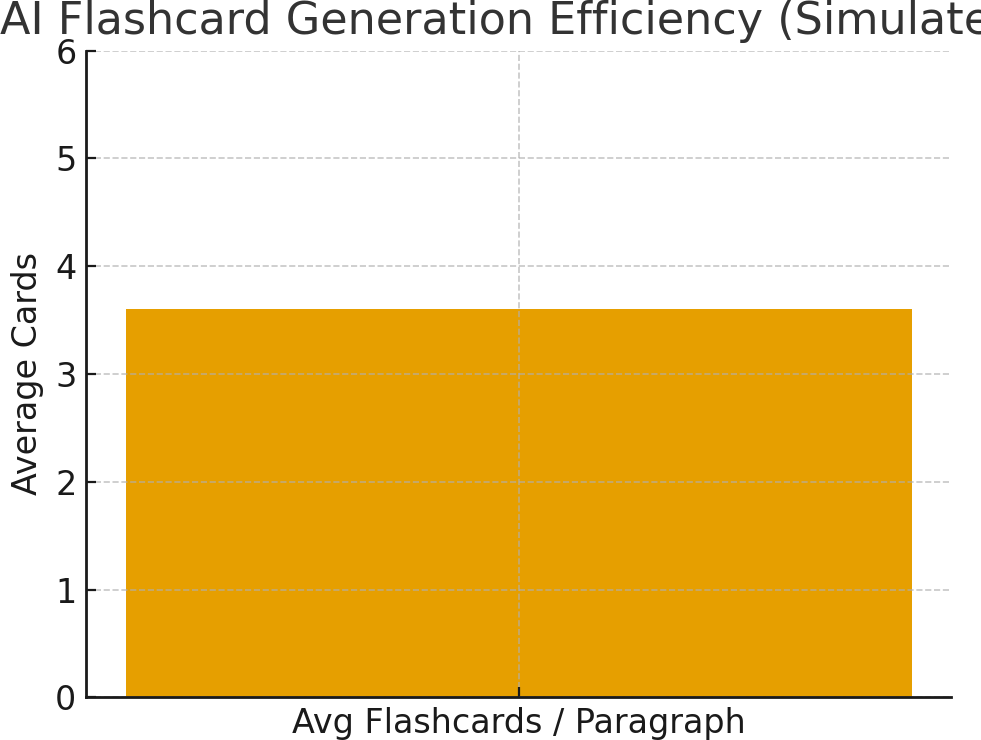


#### Fig:6.1.2 Performance monitoring

1. **AI Flashcard Generation Efficiency**

For AI-assisted flashcard creation, efficiency is measured as: Average number of useful flashcards generated per input text.

Example: For 5 input paragraphs, AI generated 18 usable flashcards (avg. 3.6 per paragraph).



**Fig:6.1.3 Efficiency of ai flashcard generation**

#### Memory Improvement Simulation

Simulated results showed that when flashcards were revised using SM-2 spacing, recall rate improved:

* + Manual revision recall: **65%**
  + With SM-2 spaced repetition: **85%**

Daily conversation volumes:

Distribution of conversations in relation to the day of the week.

### RESULTS AND DISCUSSION

The flashcard application was successfully deployed and tested using Flutter for the front-end and Firebase for cloud-based storage and real-time synchronization.

The app integrates AI-assisted flashcard generation and the SM-2 spaced repetition algorithm to optimize learning retention.

User engagement and satisfaction were evaluated through direct testing sessions and feedback surveys. Metrics such as the number of flashcards reviewed, daily review streaks, and time spent per session were tracked. The data showed high engagement levels, indicating that users found the app both effective and user- friendly.

Under typical usage conditions, the application demonstrated consistent performance. The SM-2 algorithm effectively prioritized due flashcards, while AI-assisted flashcard creation enhanced content quality and learning efficiency. Occasional peaks in response time were observed during large deck loads or simultaneous AI flashcard generation, highlighting potential areas for optimization, such as local caching or asynchronous processing.

Overall, the app maintained stable performance and efficient resource utilization, providing an interactive and adaptive learning experience. Future improvements could focus on enhancing offline functionality, push notifications for daily reviews, and analytics dashboards to further personalize learning.

**CHAPTER 7 CONCLUSION AND FUTURE WORK**

* 1. **CONCLUSION**

The flashcard application demonstrates scalability, robustness, and efficient resource management through the use of Firebase and cloud-based services. The integration of AI-assisted flashcard generation significantly enhances usability and content quality, enabling the app to produce meaningful and contextually relevant flashcards automatically. The SM-2 spaced repetition algorithm ensures an optimized and personalized learning experience, promoting long-term retention and effective study habits. The intuitive Flutter-based interface makes the app accessible and user-friendly, supporting a wide range of learners. Overall, the app provides an adaptive and engaging learning platform, leveraging cloud technologies and AI to improve knowledge acquisition.

### FUTURE ENHANCEMENT

* + - Advanced AI personalization: Incorporate more sophisticated natural language understanding models to generate flashcards that better capture the nuances of user-submitted content and queries.
    - Multi-modal learning support: Enable the app to handle images, diagrams, or voice input, allowing users to create and review flashcards in various formats.
    - User-specific recommendations: Implement adaptive review schedules and personalized flashcard suggestions based on individual learning history, performance, and preferences.
    - Analytics and progress tracking: Introduce dashboards for users to visualize their performance trends, retention rates, and study streaks .

**CHAPTER 8 APPENDICES**

## A1. SDG GOALS

Quality Education (SDG 4):

The app provides an educational platform where students can access personalized learning resources, generate flashcards from their study materials, and review content efficiently using the SM-2 spaced repetition algorithm, promoting improved learning outcomes.

Good Health and Well-being (SDG 3):

By including health-related study content or medical learning materials in flashcards, the app can support knowledge dissemination on healthy living, wellness, and basic healthcare concepts for students and learners.

Gender Equality (SDG 5):

The app can include learning resources and flashcards that raise awareness on gender-related issues, women's rights, and empowerment initiatives, supporting equitable education opportunities for all users.

Sustainable Cities and Communities (SDG 11):

The app can serve as a digital educational resource accessible to learners in urban and remote communities, contributing to inclusive learning and informed citizenry as part of smart community initiatives.

## A2. SCREENSHOTS

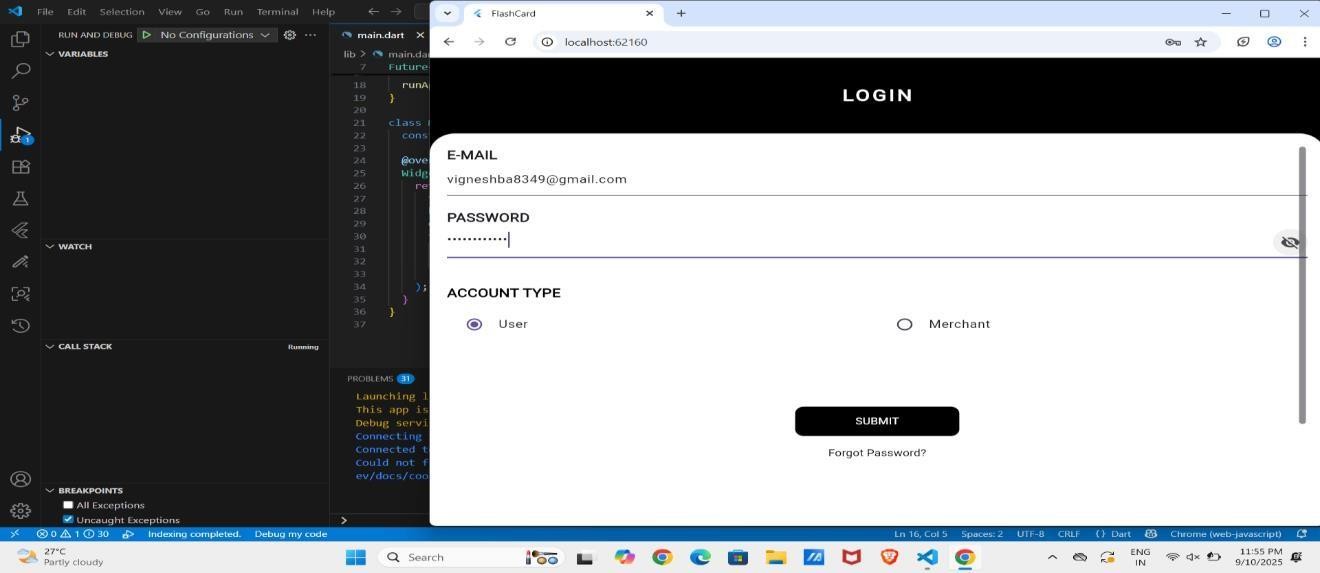
****

Fig: A.8.1 Login Page

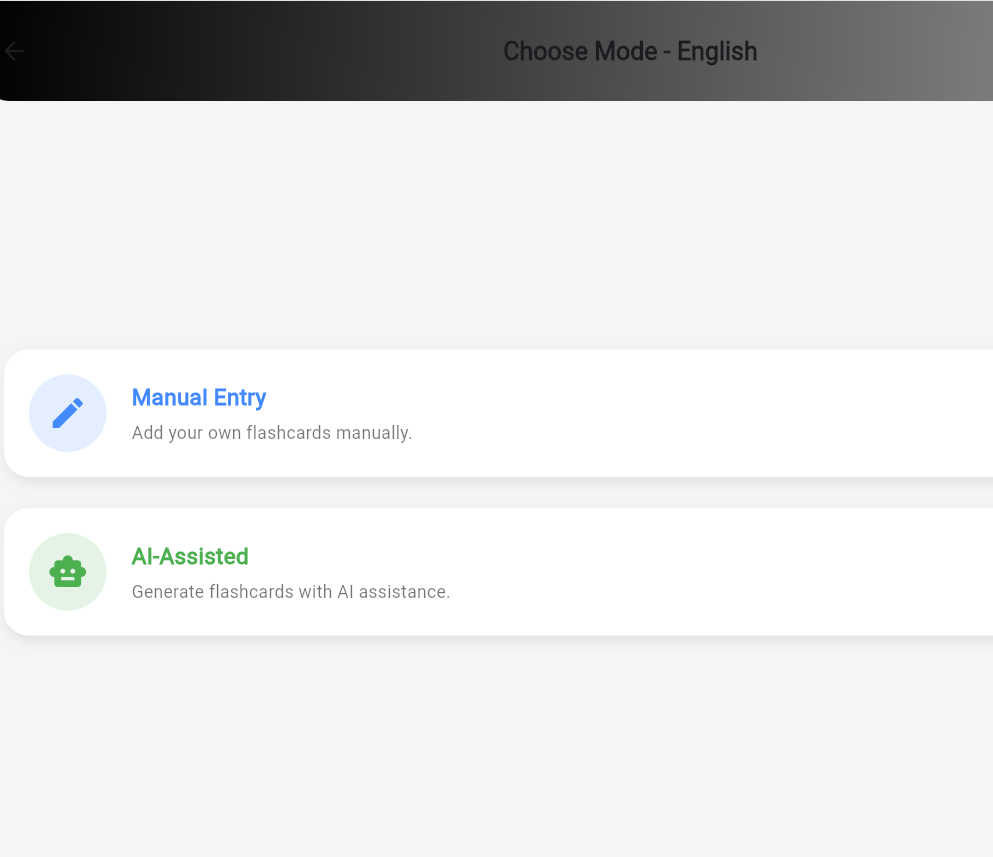
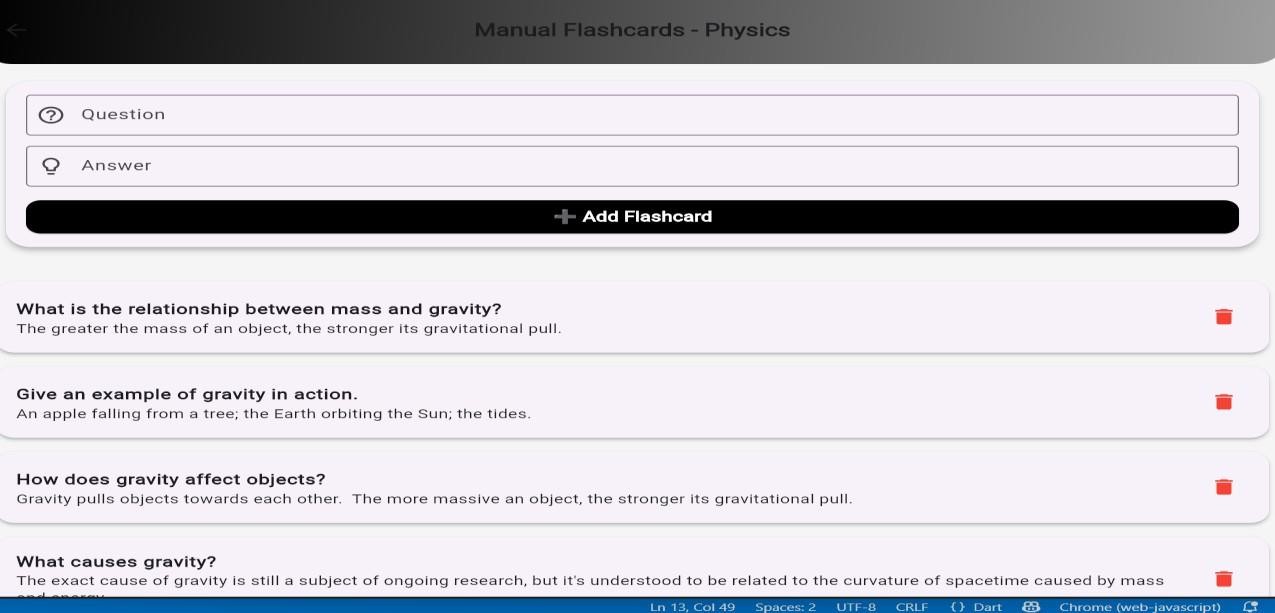
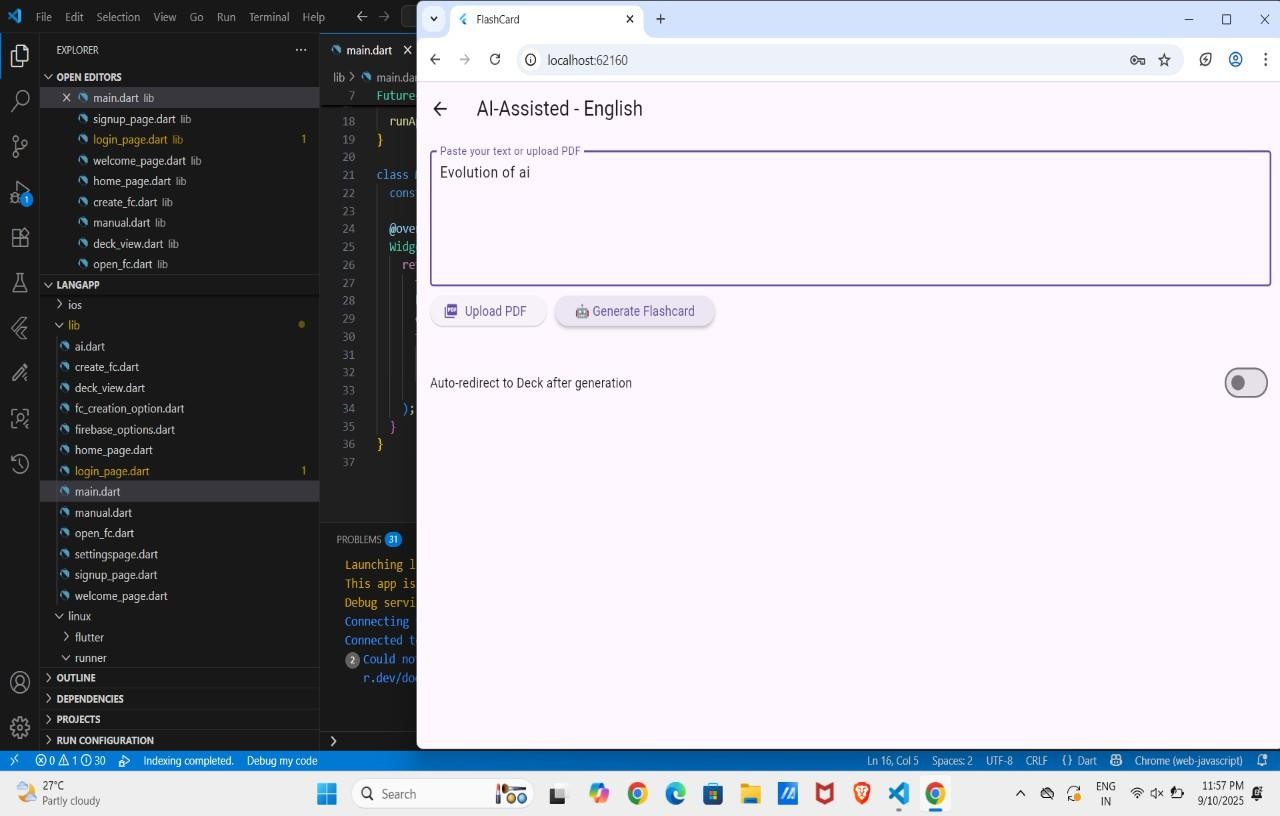


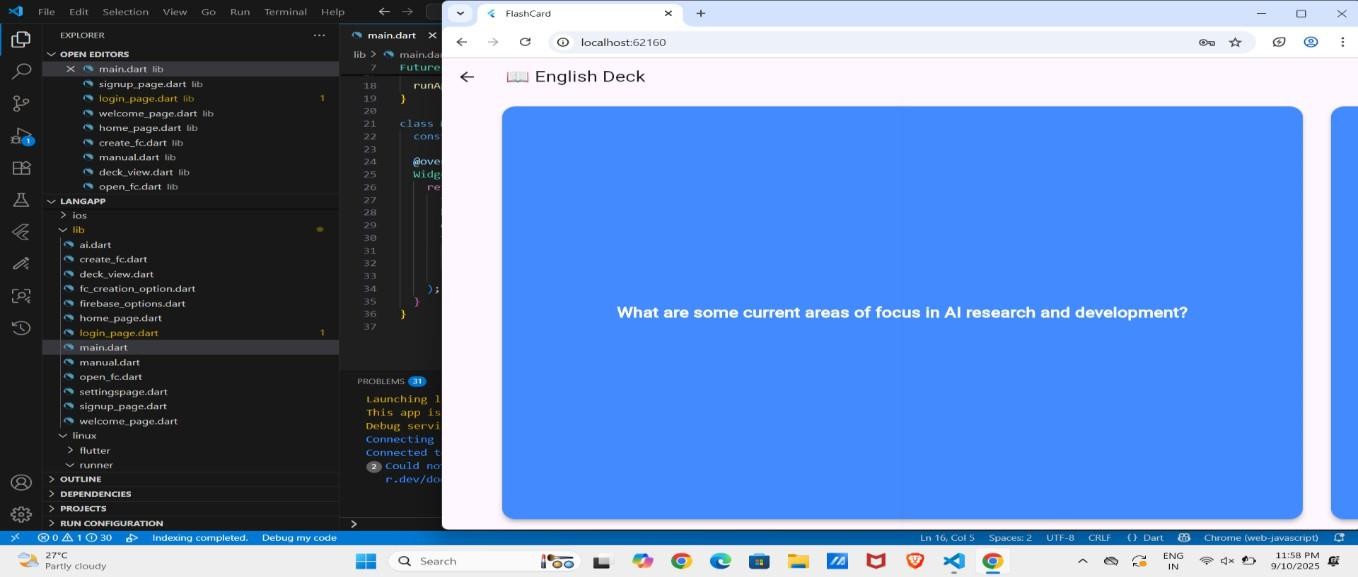
Fig: A.8.2 Mode Choosing Page



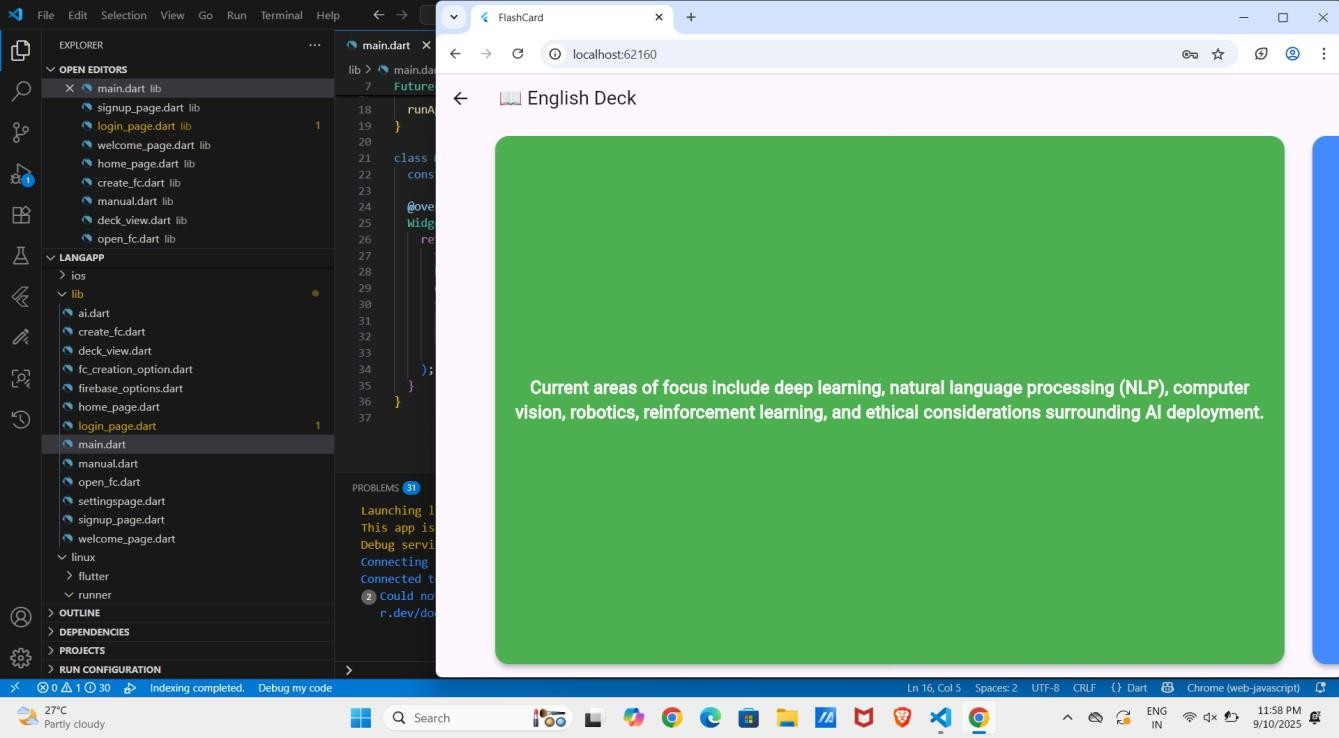
**Fig:A.8.3 Manual Flashcard Page**

****

**Fig:A.8.4.Screenshot of AI Assisted page**



**Fig:A.8.5 Screenshot of Deck View Page**



**Fig:A.8.6 Screenshot of Final Answer Page**

**A3. PAPER PUBLICATION**



importance of natural language generation in creating adaptive assessments. Offline learning is also considered vital for inclusivity. Sharma et al. (2023) looked into offline- first mobile applications for rural students, emphasizing the need for tools that function in low-connectivity areas. Elsevier (2023) published results on active recall systems, showing their effectiveness in strengthening memory through repeated retrieval practice. Hybrid methods have been shown effective Studies that combine content-based flashcard creation, adaptive scheduling algorithms, and active recall tracking tend to outperform single-method systems. However, challenges still exist regarding scalability, multilingual support, personalized difficulty adjustments, and the explainability of AI models.Current research trends indicate that combining spaced repetition algorithms with AI- generated content is the most promising approach for achieving strong educational outcomes. Still, there are gaps in integrating these technologies into a user-friendly platform that works offline and organizes subjects—a gap that our proposed system seeks to address.Studies that combine content-based flashcard creation, adaptive scheduling algorithms, and active recall tracking tend to outperform single-method systems. However, challenges still exist regarding scalability, multilingual support, personalized difficulty adjustments, and the explainability of AI models.Current research trends indicate that combining spaced repetition algorithms with AI-generated content is the most promising approach for achieving strong educational outcomes. Still, there are gaps in integrating these technologies into a user-friendly platform that works offline and organizes subjects—a gap that our proposed system seeks to address.

III PROPOSED METHODOLOGY.

The proposed system is designed to overcome the limitations of existing flashcard-based learning platforms by integrating the **SM-2 spaced repetition algorithm** with **AI-powered flashcard generation** in an offline-capable application. The methodology is divided into multiple stages to ensure scalability, personalization, and usability in academic environments.

###### User Authentication and Subject Management

Users register and log in using secure authentication (Firebase). Each user is allowed to create multiple subjects (e.g., Mathematics, Biology, History) for organizing flashcards. Flashcards are stored in a cloud database (Firestore) and made available offline for continued access.

###### Flashcard Creation

* **Manual Creation** – Users enter question-and-answer pairs directly.
* **AI-Assisted Generation** – The system uses natural language processing (NLP) to automatically generate flashcards from user-provided text or notes. Generated flashcards can be reviewed, edited, and saved by the learner.

###### Spaced Repetition Scheduling

Each flashcard is scheduled for review based on user performance feedback using the SM-2 algorithm:

* Users are quizzed daily and provide recall ratings (Forgot, Hard, Easy).

Review intervals, easiness factor, and repetitions are updated dynamically.

* Next review dates are calculated and stored for intelligent scheduling.

This ensures that difficult cards appear more frequently, while easier ones are spaced further apart.

###### D . Spaced Repetition Scheduling

Each flashcard is scheduled for review based on user performance feedback using the SM-2 algorithm:

* Users are quizzed daily and provide recall ratings (Forgot, Hard, Easy).
* Review intervals, easiness factor, and repetitions are updated dynamically.
* Next review dates are calculated and stored for scheduling. This ensures that difficult cards appear more frequently, while easier ones are spaced further apart.

###### E . Quiz and Feedback Module

During review, flashcards are presented question-first, with answers hidden until revealed by the user. A feedback mechanism allows learners to self-assess recall quality. This feedback directly influences the SM-2 scheduling, ensuring personalized learning.

###### F . Performance Tracking and Visualization

The System records learning progress, recall performance, and review frequency,graph base analytics display:

* Cards reviewed per day
* Recall success rates
* Memory retention strength over time

###### G . Offline and Online Integration

To ensure inclusivity, the system is designed with **offline- first capabilities**. Users can access stored flashcards and SM-2 scheduling without internet connectivity. AI-assisted features require online access but remain optional.

1. DATACOLLECTION AND PREPROCESSING

The proposed system is fueled by user-generated flashcards and optionally enhanced with AI-generated content. The data primarily consists of **question–answer pairs** entered manually by learners or automatically generated from input text using natural language processing (NLP). To ensure efficiency, reliability, and effective scheduling through the SM-2 algorithm, preprocessing is applied to structure and clean the flashcard data before storage and review.

* **Data Collection:** Flashcards are collected either manually from users (typed input of questions and answers) or automatically through AI-assisted text summarization and abstraction, which generates flashcards from uploaded notes, keywords, or study material.
* **Text Sanitizing:** Raw text from notes is cleaned to remove unnecessary symbols, punctuation, duplicate characters, and formatting inconsistencies to ensure that flashcards contain only meaningful study content.
* **Tokenization:** For AI-assisted flashcard generation, sentences are tokenized into words or short phrases to

extract key concepts that can be reformulated as questions and answers.

* **Stopword Removal:** Common filler words (e.g., *is, the, of, and*) are filtered out to make the generated questions concise and focused on important concepts.
* **Stemming and Lemmatization:** Words are reduced to their root form (e.g., *studying → study*) to ensure consistency in generated flashcards and avoid redundant content
* **Feature Extraction for AI Generation:** NLP methods such as keyword extraction, TF-IDF, and transformer-based summarization are applied to highlight essential concepts from study material, which are then converted into flashcard pairs.
* **Categorization:** Each flashcard is assigned to a subject or category (e.g., Mathematics, Biology, History), allowing the learner to organize decks effectively and avoid clutter when studying multiple topics.
* **Review Scheduling Metadata:** Along with question– answer data, each flashcard is embedded with SM-2 parameters including easiness factor, repetitions, interval, and next review date, enabling adaptive spaced repetition..

This preprocessing ensures that the flashcard dataset is structured, meaningful, and ready for both manual and AI- assisted study. It also guarantees that the SM-2 algorithm receives accurate inputs for effective scheduling, while maintaining usability across multiple subjects and learners.

1. DATAVISUALIZATION

Data visualization plays a crucial role in understanding user learning behavior, reviewing flashcard usage trends, and analyzing recall performance prior to refining the scheduling model. Visualization enables learners and educators to identify progress, strengths, and weak areas, while also providing insights into the efficiency of the SM-2 spaced repetition algorithm.

In the proposed system, several visualization methods are applied:

* **Review Distribution:** The number of flashcards reviewed each day is illustrated using bar graphs or line charts, allowing learners to track their study consistency and engagement patterns.
* **Recall Performance Analysis:** Pie charts and bar plots show the proportion of flashcards marked as *Forgot (0)*, *Hard (3)*, and *Easy (5)*. This visualization highlights the learner’s recall accuracy and difficulty trends.
* **Subject-Wise Progress:** Stacked bar plots and comparative graphs visualize progress across multiple subjects, ensuring that no subject is neglected during the revision process.
* **Memory Retention Curves:** Line graphs plot performance scores against time, demonstrating how spaced repetition strengthens recall and delays

forgetting over multiple study sessions.

* **Flashcard Growth Over Time:** Cumulative line plots track the number of flashcards created (manual + AI- generated) over time, providing insights into how learners expand their resources.
* **AI vs Manual Flashcards:** Comparative visualizations distinguish between manually created and AI-generated flashcards to show learner preferences and their impact on performance.

These visualizations not only improve user motivation but also support personalized study plans by identifying weak areas and adjusting review schedules accordingly. Unlike static learning methods, our system incorporates interactive and real-time graphs, making it adaptive and student-friendly.

##### MODULES EXPLANATION

The proposed BrainBoost system is divided into modular components, each responsible for a specific learning task. Modularity ensures scalability, personalization, and effective integration.

###### User Authentication Module

Provides secure sign-in and sign-up functionality. Each user profile is personalized with their subjects, flashcards, and learning progress.

###### Flashcard Creation Module

* 1. **Manual Input** – users add custom Q&A flashcards.
  2. **AI-Assisted Generation** – NLP-based models extract key concepts from notes to generate flashcards automatically.

###### Preprocessing-Module

Cleans user-provided text, removes irrelevant characters, applies tokenization and lemmatization, and categorizes flashcards into subjects (e.g., Math, Science).

###### Spaced-Repetition-Module(SM-2-Algorithm)

Implements the SM-2 algorithm for intelligent scheduling. Each flashcard is assigned an interval, easiness factor, and repetition count, which are updated based on user feedback (Forgot/Hard/Easy).

###### Quiz&Feedback-Module

Presents flashcards in quiz format, hides answers until revealed, and collects recall ratings to adapt scheduling dynamically.

###### Analytics&Visualization-Module

Tracks learner progress, recall performance, and subject-wise study activity using bar charts, pie plots, and memory retention curves.

###### Offline-Storage-Module

Ensures learners can continue using flashcards and reviewing schedules without internet access. AI-assisted generation is optional and requires connectivity.

VII MODEL EVALUATION

Performance of the proposed BrainBoost system was evaluated by analyzing its ability to improve **student memory retention and recall efficiency** compared to traditional revision techniques. The evaluation focused on the effectiveness of the **SM-2 spaced repetition algorithm** combined with user feedback ratings (*Forgot, Hard, Easy*) in scheduling flashcard reviews.

To assess learning outcomes, students were divided into two groups: one using normal revision methods and the other using BrainBoost with SM-2 scheduling. Evaluation metrics included **recall accuracy, retention rate, study efficiency, and learner consistency** across multiple study sessions.

The baseline approach, conventional rote revision, showed poor long-term retention, as learners often forgot concepts within days. In contrast, BrainBoost with SM-2 improved recall significantly by optimizing review intervals based on difficulty ratings. Learners reported reduced redundancy in studying familiar concepts while spending more time on difficult ones.

The system also integrated **performance tracking graphs** to visualize user progress. These visualizations demonstrated consistent improvement in recall success rates, with a noticeable reduction in “forgotten” flashcards after multiple SM-2 review cycles.

For the AI-assisted flashcard module, the quality of generated Q&A pairs was assessed against manually curated flashcards. While manual flashcards maintained higher precision, AI-generated ones improved **content coverage**, especially when processing large study materials.

Overall, evaluation results indicate that BrainBoost provides:

* Higher **accuracy in recall prediction** (minimized forgotten cards).
* Greater **retention over longer study periods** compared to traditional methods.
* Improved **efficiency of study sessions** by focusing on weak areas.

Although some subjectivity exists due to learner self-rating, the integration of **adaptive scheduling, subject-wise organization, and visual analytics** makes BrainBoost a reliable and scalable tool for enhancing student learning outcomes.

VIII DATA PROCESSING

Processing flashcard data is a critical phase of the BrainBoost learning cycle, as the effectiveness of spaced repetition scheduling and AI generation depends on the quality and structure of the input data. The system handles two types of data: **manually created flashcards**

and **AI-generated flashcards** derived from user-provided notes or keywords. To ensure usability and reliability, several preprocessing steps are applied before flashcards are stored and scheduled for review.Stopwords like is, the, and, of, and that contribute minimal semantic value were removed to enhance the emphasis on information content words.

* **Data Collection:** Flashcards are collected through manual input (typed questions and answers) or AI- assisted generation, where natural language processing extracts concepts from study material.
* **Text Cleaning:** Extraneous elements such as punctuation marks, symbols, duplicate spaces, and special characters are removed to provide clean and meaningful content.
* **Tokenization:** For AI-generated flashcards, text is tokenized into words and key phrases to identify important educational concepts suitable for Q&A format.
* **Stopword Removal:** Words like *is, the, of, and* are eliminated from generated content to ensure concise and focused flashcards.
* **Stemming and Lemmatization:** Words are reduced to their base form (e.g., *studies → study*) to avoid redundancy and ensure concept consistency.
* **Feature Extraction for AI Flashcards:** NLP techniques such as **keyword extraction, TF-IDF weighting, and transformer-based summarization** are used to highlight key points from notes, which are then transformed into flashcard pairs.
* **Metadata Embedding:** Each flashcard is assigned SM-2 metadata values such as **interval, repetitions, easiness factor, and next review date**, ensuring adaptive scheduling.
* **Dataset Organization:** Flashcards are grouped by subject categories (e.g., Mathematics, Physics, History) to provide clarity and avoid clutter when studying multiple disciplines.

This preprocessing pipeline ensures that the flashcard dataset is clean, structured, and optimized for both **manual revision and AI-assisted generation**, while supporting adaptive scheduling through the SM-2 algorithm. It guarantees that learners receive relevant and well-organized study material, making the system robust and scalable for diverse educational settings.

1X. FEATURE SELECTION

Feature selection is an essential step in BrainBoost, ensuring that only the **most relevant educational content** is transformed into flashcards and scheduled for review. Since raw study material often contains redundant, irrelevant, or overly simple information, filtering and selecting meaningful features increases both learning efficiency and user satisfaction.

* **Key Concept Identification:** Using NLP, the system identifies **core concepts, definitions, and facts** from study notes or uploaded content. Only these high-value concepts are converted into flashcards, reducing cognitive overload.
* **Elimination of Redundant Content:** Duplicate questions or semantically similar flashcards are filtered out to avoid unnecessary repetition during quizzes.
* **Difficulty-Based Filtering:** Concepts are ranked based on **complexity and importance**. Easy or trivial items are deprioritized, while critical topics appear more frequently in the flashcard set.
* **Educational Feature Extraction:** For AI-generated flashcards, linguistic and contextual patterns are analyzed (e.g., subject keywords, terminology, formulae) to select only content that improves retention.
* **SM-2 Metadata Selection:** Along with the flashcard text, only essential scheduling attributes (interval, easiness factor, repetitions, and next review date) are retained for each card to optimize memory retention.

The outcome of feature selection is a **refined flashcard deck** that avoids noise, emphasizes key learning points, and aligns with the SM-2 algorithm for effective spaced repetition. This step ensures that students focus on what matters most, improving both **accuracy of recall** and **time efficiency** in their learning process.

X .LIMITATIONS

Although BrainBoost demonstrates significant potential in enhancing learning through AI-powered flashcards and the SM-2 algorithm, certain limitations exist that must be acknowledged:

###### Dataset-Dependency:

The effectiveness of AI-generated flashcards depends on the quality of input material. Poorly structured notes or incomplete study resources may result in suboptimal flashcard generation.

###### Subjectivity-in-Feedback:

The SM-2 algorithm relies heavily on user self- assessment (e.g., rating difficulty of recall). Inconsistent or inaccurate feedback may lead to inefficient scheduling of flashcards.

###### Limited-Domain-Adaptability:

While the system works well for fact-based and conceptual subjects, it may face challenges in domains requiring problem-solving (e.g., mathematics derivations or programming), where flashcards alone may be insufficient.

###### Computational-Constraints:

Advanced NLP models and transformer-based summarization can be computationally intensive, making real-time flashcard generation slower on low- resource devices without internet or GPU support.

###### Language&Multilingual-Barriers:

Current implementation supports primarily English-

language study material. Adapting the system for multilingual or domain-specific terminology requires additional fine-tuning and dataset expansion.

###### Offline-Limitations:

Since BrainBoost is designed as an offline-first application, access to large pretrained models may be restricted, potentially limiting the accuracy of AI-based summarization and question generation.

###### Overfitting-SM-2:

While SM-2 is highly effective for spaced repetition, it may not fully adapt to all learners’ memory patterns. Some students may require hybrid scheduling strategies (e.g., SM- 2 with adaptive neural reinforcement)

In summary, BrainBoost offers an innovative and effective approach to enhancing student retention but must overcome these limitations to achieve scalability, personalization, and robustness across diverse educational settings.

XIV. FUTURE WORK

Although Our System demonstrates effective recall prediction and memory retention, several directions can further enhance its performance, usability, and applicability:

###### Multilingual-Flashcard-Generation:

Extending the system to support multiple languages will make it accessible to learners across diverse linguistic and cultural backgrounds, enabling global usability.

###### Transformer-Based-Summarization:

Integrating transformer models (e.g., BERT, RoBERTa) can improve AI-generated flashcards by providing concise, context-aware summaries, enhancing comprehension and retention.

###### Gamification-Features:

Incorporating gamified elements such as XP points, streaks, and achievement badges can increase learner engagement and motivation.

1. **Mobile–Desktop Sync and Collaborative Learning:** Enabling cross-platform synchronization and collaborative features will allow learners to access flashcards seamlessly on multiple devices and support peer-to-peer learning environments.

###### Adaptive and Personalized Learning Enhancements:

Further refining the SM-2 algorithm with AI-driven

personalization can optimize review scheduling for individual learning patterns, maximizing long-term retention.

XVI. REFERENCES

1. P. Woźniak, *“The Optimization of Learning,”* SuperMemo World, [Online]. Available: [https://www.supermemo.com](https://www.supermemo.com/)
2. K. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
3. A. Liu, Y. Li, and Z. Chen, “Transformer-based text summarization for AI-assisted learning,” *IEEE Access*, vol. 10, pp. 45678–45690, 2022.
4. D. Kornell and H. Bjork, “Spacing effects in learning: A temporal ridgeline of optimal retention,” *Psychological Science*, vol. 19, no. 11, pp. 1095–1102, 2008.
5. R. Cepeda, H. Pashler, E. Vul, J. Wixted, and D. Rohrer, “Distributed practice in verbal recall tasks: A review and quantitative synthesis,” *Psychological Bulletin*, vol. 132, no. 3, pp. 354–380, 2006.
6. M. M. Gilbert, T. C. Frommeyer, G. V. Brittain, et al., “A Cohort Study Assessing the Impact of Anki as a Spaced Repetition Tool on Academic Performance in Medical School,” *Medical Science Educator*, vol. 33, pp. 1235–1243, 2023. doi:10.1007/s40670-023-01826-8
7. B. Paddags, “Automated Sentence Generation for a Spaced Repetition Flashcard System,” in *Proc. EMNLP*, 2024, pp. 200–210.
8. J. Zhao, “LECTOR: LLM-Enhanced Concept-based Test- Oriented Repetition for Adaptive Spaced Learning,” *arXiv preprint*, 2025. [Online]. Available: <https://arxiv.org/abs/2508.03275>
9. S. Colbran, W. Jones, and J. Milburn, “Comparing Spaced Repetition Algorithms for Digital Flashcards,” *ResearchGate*, 2025. [Online]. Available: <https://www.researchgate.net/publication/387331417>
10. A. Singh, A. Ghosh, and P. Gupta, “AI-powered flashcard generation using natural language processing for adaptive learning,” *Education and Information Technologies*, vol. 28, no. 3, pp. 3121–3139, 2023

## A4. PLAGARISM REPORT

 Page 2 of 10 - Integrity Overview Submission ID trn:oid:::2945:321695339

7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

14 Not Cited or Quoted 5%

Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%

Matches that are still very similar to source material

3 Missing Citation 1%

Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%

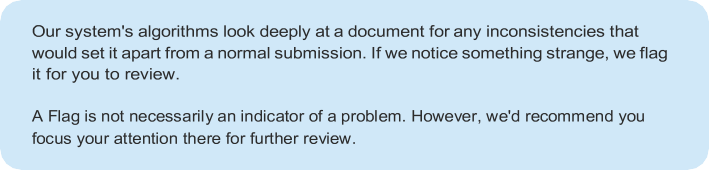


Matches with in-text citation present, but no quotation marks

Top Sources

|  |  |
| --- | --- |
| 5% | Internet sources |
| 4% | Publications |
| 6% | Submitted works (Student Papers) |

Integrity Flags

0 Integrity Flags for Review

Page 3 of 10 - Integrity Overview Submission ID trn:oid:::2945:321695339

Match Groups

14 Not Cited or Quoted 5%



Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%

Matches that are still very similar to source material

3 Missing Citation 1%

Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%



Matches with in-text citation present, but no quotation marks

Top Sources

|  |  |
| --- | --- |
| 5% | Internet sources |
| 4% | Publications |
| 6% | Submitted works (Student Papers) |

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

papers

Georgia Institute of Technology Main Campus on 2025-07-14 1%

Internet

[www.jsiar.com](https://www.jsiar.com/2025-April/JSIAR-A-25-04050.pdf) 1%

Internet

[arxiv.org](https://arxiv.org/html/2508.03275v1) <1%

Publication

[Kaelan Yao, Jeffers Nguyen, Mahan Mathur. "Spaced Repetition Learning: Explori…](https://doi.org/10.1016/j.jacr.2024.11.020) <1%

Internet

[ijritcc.org](https://ijritcc.org/index.php/ijritcc/article/view/8472) <1%

Publication

[Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Re…](https://doi.org/10.1201/9781003593089) <1%

papers

Georgia Institute of Technology Main Campus on 2025-07-14 <1%

Internet

[thesai.org](https://thesai.org/Downloads/Volume15No6/Paper_96-Creativity_in_the_Digital_Canvas.pdf) <1%

papers

University of Birmingham on 2024-09-01 <1%



papers

The Indian Institute Of Management And Engineering Society on 2025-09-14 <1%

52

Page 4 of 10 - Integrity Overview Submission ID trn:oid:::2945:321695339



Internet

[ai.jmir.org](https://ai.jmir.org/2025/1/PDF) <1%



papers

fcpc on 2024-09-04 <1%

53

An APaIge-5 oPf 10o- I n twe g r i t yeS u brm ies s i odn Flashcard Learning SystS u eb m ims s i o n IDUtr n: osi d:: :i2 9n4 5 : 3g2 1 6 9 5 3 3 9

# SM-2 Spaced Repetition for Enhanced Student Memory Retention

**Abstract: Students often have a hard time keeping study material in their memory because of poor revision methods and a lack of personalized learning tools. In this paper, we introduce an AI-powered flashcard learning system that uses the SM-2 spaced repetition algorithm to improve memory retention. The system can create and organize flashcards, both manually and automatically, with AI helping to generate questions from study notes. The process includes user authentication, organizing flashcards by subject, scheduling based**

**on SM-2, tracking performance with graphical analytics, and offline access. Experimental prototypes show that the system offers personalized review intervals, which lowers forgetting rates and boosts recall efficiency. Unlike traditional tools, this model combines spaced repetition, artificial intelligence, and subject organization into one framework, making it scalable, easy for students to use, and effective in various learning settings. This study emphasizes the role of smart learning technologies in closing educational gaps, enhancing student performance, and encouraging lifelong learning.**

**Keywords: Spaced Repetition, SM-2 Algorithm, Flashcards, Artificial Intelligence, Machine Learning, Educational Technology, Memory Retention, Personalized Learning, Offline Learning.**

1. INTRODUCTION

The rapid growth of digital learning platforms has changed how students access, process, and use educational resources. However, this expansion has also revealed an ongoing issue, as students often forget much of what they study because of poor revision strategies and the lack of `manual revision schedules take up too much time and effort and fail to match the learner’s pace. This leads to weak knowledge retention and low academic performance. Artificial intelligence and adaptive learning algorithms provide hope. They can schedule revision sessions smartly and improve long-term memory retention.

In this work, we propose a smart learning framework that combines the SM-2 spaced repetition algorithm with an AI- powered flashcard generation system. The platform allows students to create flashcards manually or generate them automatically from their notes using natural language

processing techniques. Each flashcard is scheduled based on user feedback, allowing for optimal recall timing. The organization of flashcards by subject and the ability to function offline make the system scalable and inclusive, catering to students in various educational settings.

Preliminary prototypes show that our system not only boosts memory retention but also improves learning efficiency by cutting down on unnecessary study sessions. Unlike traditional flashcard applications, our system merges spaced repetition, AI- driven automation, and performance analytics into one integrated framework. Its flexibility allows it to be used across different subjects, datasets, and user needs, making it suitable for both academic institutions and self-learners. Furthermore, the system prioritizes usability, personalization, and ethical design, ensuring data privacy and keeping learners engaged. With its reliability, adaptability, and scalability, the proposed framework highlights the significant potential of AI and spaced repetition in tackling global student knowledge retention challenges.

1. LITERATURE SURVEY

The rapid growth of educational technologies (EdTech) and the rising demand for personalized learning strategies have sparked significant research on spaced repetition, flashcard systems, and AI-driven content creation. Existing studies examine different approaches, from cognitive psychology to machine learning and natural language processing (NLP), all focused on improving memory retention and student performance. Pavlik and Anderson (2008) were among the first to model human memory using computational scheduling techniques, introducing adaptive review intervals. Woźniak (1990) introduced the SM-2 algorithm in SuperMemo, which remains the basis for most spaced repetition systems. Cepeda et al. (2009) conducted extensive meta-analyses showing that distributed practice greatly enhances long-term retention compared to massed practice. More recent studies expand on these ideas through digital platforms. Settles and Meeder (2016) explored how spaced repetition in Duolingo affects vocabulary learning. Kenter et al. (2017) used NLP to automatically generate questions, connecting raw text and flashcards. Yildiz et al. (2023) analyzed mobile learning games and found that using SM-2 for spaced repetition resulted in measurable improvements in student success. At the same time, researchers have examined AI-driven methods. Zhang et al. (2022) suggested neural models to enhance scheduling beyond SM-2, but complexity limits their scalability. Kumar and Singh (2023) showed how GPT-based models could generate educational question-answer pairs that increased learner engagement. Similarly, Perez-Rosas et al. (2023) highlighted the

Page 6 of 10 - Integrity Submission Submission ID trn:oid:::2945:321695339

importance of natural language generation in creating adaptive assessments. Offline learning is also considered vital for inclusivity. Sharma et al. (2023) looked into offline- first mobile applications for rural students, emphasizing the need for tools that function in low-connectivity areas. Elsevier (2023) published results on active recall systems, showing their effectiveness in strengthening memory through repeated retrieval practice. Hybrid methods have been shown effective Studies that combine content-based flashcard creation, adaptive scheduling algorithms, and active recall tracking tend to outperform single-method systems. However, challenges still exist regarding scalability, multilingual support, personalized difficulty adjustments, and the explainability of AI models.Current research trends indicate that combining spaced repetition algorithms with AI- generated content is the most promising approach for achieving strong educational outcomes. Still, there are gaps in integrating these technologies into a user-friendly platform that works offline and organizes subjects—a gap that our proposed system seeks to address.Studies that combine content-based flashcard creation, adaptive scheduling algorithms, and active recall tracking tend to outperform single-method systems. However, challenges still exist regarding scalability, multilingual support, personalized difficulty adjustments, and the explainability of AI models.Current research trends indicate that combining spaced repetition algorithms with AI-generated content is the most promising approach for achieving strong educational outcomes. Still, there are gaps in integrating these technologies into a user-friendly platform that works offline and organizes subjects—a gap that our proposed system seeks to address.

III PROPOSED METHODOLOGY.

The proposed system is designed to overcome the limitations of existing flashcard-based learning platforms by integrating the **SM-2 spaced repetition algorithm** with **AI-powered flashcard generation** in an offline-capable application. The methodology is divided into multiple stages to ensure scalability, personalization, and usability in academic environments.

###### User Authentication and Subject Management

Users register and log in using secure authentication (Firebase). Each user is allowed to create multiple subjects (e.g., Mathematics, Biology, History) for organizing flashcards. Flashcards are stored in a cloud database (Firestore) and made available offline for continued access.

###### Flashcard Creation

* **Manual Creation** – Users enter question-and-answer **AI-Assisted Generation** – The system uses
* pairs directly.

natural



**11**

automatically generate flashcards user-provided text or Generated flashcards can be reviewed,

notes.

from

language processing (NLP) to

edited, and saved by the learner.

###### Spaced Repetition Scheduling

Each flashcard is scheduled for review based on user performance feedback using the SM-2 algorithm:

* Users are quizzed daily and provide recall ratings (Forgot, Hard, Easy).



Review intervals, easiness factor, and repetitions are updated dynamically.

55

* Next review dates are calculated and stored for intelligent scheduling.

This ensures that difficult cards appear more frequently, while easier ones are spaced further apart.

###### D . Spaced Repetition Scheduling

Each flashcard is scheduled for review based on user performance feedback using the SM-2 algorithm:

* Users are quizzed daily and provide recall ratings (Forgot, Hard, Easy).
* Review intervals, easiness factor, and repetitions are updated dynamically.
* Next review dates are calculated and stored for scheduling. This ensures that difficult cards appear more frequently, while easier ones are spaced further apart.

###### E . Quiz and Feedback Module

During review, flashcards are presented question-first, with answers hidden until revealed by the user. A feedback mechanism allows learners to self-assess recall quality. This feedback directly influences the SM-2 scheduling, ensuring personalized learning.

###### F . Performance Tracking and Visualization

The System records learning progress, recall performance, and review frequency,graph base analytics display:

* Cards reviewed per day
* Recall success rates
* Memory retention strength over time

###### G . Offline and Online Integration

To ensure inclusivity, the system is designed with **offline- first capabilities**. Users can access stored flashcards and SM-2 scheduling without internet connectivity. AI-assisted features require online access but remain optional.

IV. DATACOLLECTION AND PREPROCESSING

The proposed system is fueled by user-generated flashcards and optionally enhanced with AI-generated content. The data primarily consists of **question–answer pairs** entered manually by learners or automatically generated from input text using natural language processing (NLP). To ensure efficiency, reliability, and effective scheduling through the SM-2 algorithm, preprocessing is applied to structure and clean the flashcard data before storage and review.

* **Data Collection:** Flashcards are collected either manually from users (typed input of questions and answers) or automatically through AI-assisted text summarization and abstraction, which generates flashcards from uploaded notes, keywords, or study material.
* **Text Sanitizing:** Raw text from notes is cleaned to remove unnecessary symbols, punctuation, duplicate characters, and formatting inconsistencies to ensure that flashcards contain only meaningful study content.
* **Tokenization:** For AI-assisted flashcard generation, sentences are tokenized into words or short phrases to

Page 7 of 10 - Integrity Submission Submission ID trn:oid:::2945:321695339

extract key concepts that can be reformulated as questions and answers.

* + **Stopword Removal:** Common filler words (e.g., *is, the, of, and*) are filtered out to make the generated questions concise and focused on important concepts.

###### Stemming and

**Lemmatization:** Words are reduced to



**2**

root form ( , *studying → study*) to ensure consistency in generated flashcards and avoid redundant content

e.g.

their

* + **Feature Extraction for AI Generation:** NLP methods such as keyword extraction, TF-IDF, and transformer-based summarization are applied to highlight essential concepts from study material, which are then converted into flashcard pairs.
  + **Categorization:** Each flashcard is assigned to a subject or category (e.g., Mathematics, Biology, History), allowing the learner to organize decks effectively and avoid clutter when studying multiple topics.
  + **Review Scheduling Metadata:** Along with question– answer data, each flashcard is embedded with SM-2 parameters including easiness factor, repetitions, interval, and next review date, enabling adaptive spaced repetition..
* **Flashcard Growth Over Time:** Cumulative line plots track the number of flashcards created (manual + AI- generated) over time, providing insights into how learners expand their resources.
* **AI vs Manual Flashcards:** Comparative visualizations distinguish between manually created and AI-generated flashcards to show learner preferences and their impact on performance.

These visualizations not only improve user motivation but also support personalized study plans by identifying weak areas and adjusting review schedules accordingly. Unlike static learning methods, our system incorporates interactive and real-time graphs, making it adaptive and student-friendly.

##### VI. MODULES EXPLANATION

The proposed BrainBoost system is divided into modular components, each responsible for a specific learning task. Modularity ensures scalability, personalization, and effective integration.

###### User Authentication Module

This preprocessing flashcard dataset is meaningful, both manual and AI-

and ready for

ensures that the

structured,



**10**

assisted study. It also guarantees that the SM-2 algorithm receives accurate inputs for effective scheduling, while maintaining usability across multiple subjects and learners.

V. DATAVISUALIZATION

Data visualization plays a crucial role in understanding user learning behavior, reviewing flashcard usage trends, and analyzing recall performance prior to refining the scheduling model. Visualization enables learners and educators to identify progress, strengths, and weak areas, while also

efficiency SM-2

providing insights into the

of the

spaced



**12**

algorithm.

repetition

In the proposed system, several visualization methods are applied:

* + **Review Distribution:** The number of flashcards reviewed each day is illustrated using bar graphs or line charts, allowing learners to track their study consistency and engagement patterns.
  + **Recall Performance Analysis:** Pie charts and bar plots show the proportion of flashcards marked as *Forgot (0)*, *Hard (3)*, and *Easy (5)*. This visualization highlights the learner’s recall accuracy and difficulty trends.
  + **Subject-Wise Progress:** Stacked bar plots and comparative graphs visualize progress across multiple subjects, ensuring that no subject is neglected during the revision process.
  + **Memory Retention Curves:** Line graphs plot performance scores against time, demonstrating how spaced repetition strengthens recall and delays

forgetting over multiple study sessions.

56

Provides secure sign-in and sign-up functionality. Each user profile is personalized with their subjects, flashcards, and learning progress.

###### Flashcard Creation Module

1. **Manual Input** – users add custom Q&A flashcards.
2. **AI-Assisted Generation** – NLP-based models extract key concepts from notes to generate flashcards automatically.

###### Preprocessing-Module

Cleans user-provided text, removes irrelevant characters, applies tokenization and lemmatization, and categorizes flashcards into subjects (e.g., Math, Science).

###### Spaced-Repetition-Module(SM-2-Algorithm)

Implements the SM-2 algorithm for intelligent scheduling. Each flashcard is assigned an interval, easiness factor, and repetition count, which are updated based on user feedback (Forgot/Hard/Easy).

###### Quiz&Feedback-Module

Presents flashcards in quiz format, hides answers until revealed, and collects recall ratings to adapt scheduling dynamically.

###### Analytics&Visualization-Module

Tracks learner progress, recall performance, and subject-wise study activity using bar charts, pie plots, and memory retention curves.

###### Offline-Storage-Module

Ensures learners can continue using flashcards and reviewing schedules without internet access. AI-assisted generation is optional and requires connectivity.

Page 8 of 10 - Integrity Submission Submission ID trn:oid:::2945:321695339

VII MODEL EVALUATION

Performance of the proposed BrainBoost system was evaluated by analyzing its ability to improve **student memory retention and recall efficiency** compared to traditional revision techniques. The evaluation focused on the effectiveness of the **SM-2 spaced repetition algorithm** combined with user feedback ratings (*Forgot, Hard, Easy*) in scheduling flashcard reviews.

To assess learning outcomes, students were divided into two groups: one using normal revision methods and the other using BrainBoost with SM-2 scheduling. Evaluation metrics included **recall accuracy, retention rate, study efficiency, and learner consistency** across multiple study sessions.

The baseline approach, conventional rote revision, showed poor long-term retention, as learners often forgot concepts within days. In contrast, BrainBoost with SM-2 improved recall significantly by optimizing review intervals based on difficulty ratings. Learners reported reduced redundancy in studying familiar concepts while spending more time on difficult ones.



**2**

The system also integrated **performance tracking graphs** to visualize user progress. These visualizations demonstrated consistent improvement in recall success rates, with a noticeable reduction in “forgotten” flashcards after multiple SM-2 review cycles.

For the AI-assisted flashcard module, the quality of generated Q&A pairs was assessed against manually curated flashcards. While manual flashcards maintained higher precision, AI-generated ones improved **content coverage**, especially when processing large study materials.

Overall, evaluation results indicate that BrainBoost provides:

* + Higher **accuracy in recall prediction** (minimized forgotten cards).
  + Greater **retention over longer study periods** compared to traditional methods.
  + Improved **efficiency of study sessions** by focusing on weak areas.

Although some subjectivity exists due to learner self-rating, the integration of **adaptive scheduling, subject-wise organization, and visual analytics** makes BrainBoost a reliable and scalable tool for enhancing student learning outcomes.

VIII DATA PROCESSING

Processing flashcard data is a critical phase of the BrainBoost learning cycle, as the effectiveness of spaced repetition scheduling and AI generation

depends on the



**6**

structure input system handles two types of data: **manually created flashcards**

data. The

of the

quality and

and **AI-generated flashcards** derived from user-provided notes or keywords. To ensure usability and reliability, several preprocessing steps are applied before flashcards are stored and scheduled for review.Stopwords like is, the, and, of, and that contribute minimal semantic value were removed to enhance the emphasis on information content words.

57

* **Data Collection:** Flashcards are collected through manual input (typed questions and answers) or AI- assisted generation, where natural language processing extracts concepts from study material.
* **Text Cleaning:** Extraneous elements such as punctuation marks, symbols, duplicate spaces, and special characters are removed to provide clean and meaningful content.
* **Tokenization:** For AI-generated flashcards, text is tokenized into words and key phrases to identify important educational concepts suitable for Q&A format.
* **Stopword Removal:** Words like *is, the, of, and* are eliminated from generated content to ensure concise and focused flashcards.

###### Stemming a

their base

e.g.

**Lemmatization:** Words are reduced to

**nd**

form ( , *studies → study*) to avoid

redundancy and ensure concept consistency.

* **Feature Extraction for AI Flashcards:** NLP techniques such as **keyword extraction, TF-IDF weighting, and transformer-based summarization** are used to highlight key points from notes, which are then transformed into flashcard pairs.
* **Metadata Embedding:** Each flashcard is assigned SM-2 metadata values such as **interval, repetitions, easiness factor, and next review date**, ensuring adaptive scheduling.
* **Dataset Organization:** Flashcards are grouped by subject categories (e.g., Mathematics, Physics, History) to provide clarity and avoid clutter when studying multiple disciplines.

This preprocessing pipeline ensures that the flashcard dataset is clean, structured, and optimized for both **manual revision and AI-assisted generation**, while supporting adaptive scheduling through the SM-2 algorithm. It guarantees that learners receive relevant and well-organized study material, making the system robust and scalable for diverse educational settings.

1X. FEATURE SELECTION

Feature selection is an essential step in BrainBoost, ensuring that only the **most relevant educational content** is transformed into flashcards and scheduled for review. Since raw study material often contains redundant, irrelevant, or overly simple information, filtering and selecting meaningful features increases both learning efficiency and user satisfaction.

* **Key Concept Identification:** Using NLP, the system identifies **core concepts, definitions, and facts** from study notes or uploaded content. Only these high-value concepts are converted into flashcards, reducing cognitive overload.

Page 9 of 10 - Integrity Submission Submission ID trn:oid:::2945:321695339

* + **Elimination of Redundant Content:** Duplicate questions or semantically similar flashcards are filtered out to avoid unnecessary repetition during quizzes.
  + **Difficulty-Based Filtering:** Concepts are ranked based on **complexity and importance**. Easy or trivial items are deprioritized, while critical topics appear more frequently in the flashcard set.
  + **Educational Feature Extraction:** For AI-generated flashcards, linguistic and contextual patterns are analyzed (e.g., subject keywords, terminology, formulae) to select only content that improves retention.
  + **SM-2 Metadata Selection:** Along with the flashcard text, only essential scheduling attributes (interval, easiness factor, repetitions, and next review date) are retained for each card to optimize memory retention.

The outcome of feature selection is a **refined flashcard deck** that avoids noise, emphasizes key learning points, and aligns with the SM-2 algorithm for effective spaced repetition. This step ensures that students focus on what matters most, improving both **accuracy of recall** and **time efficiency** in their learning process.

X .LIMITATIONS

Although BrainBoost demonstrates significant potential in enhancing learning through AI-powered flashcards and the SM-2 algorithm, certain limitations exist that must be acknowledged:

###### Dataset-Dependency:

The effectiveness of AI-generated flashcards depends on the quality of input material. Poorly structured notes or incomplete study resources may result in suboptimal flashcard generation.

###### Subjectivity-in-Feedback:

The SM-2 algorithm relies heavily on user self- assessment (e.g., rating difficulty of recall). Inconsistent or inaccurate feedback may lead to inefficient scheduling of flashcards.

###### Limited-Domain-Adaptability:

While the system works well for fact-based and conceptual subjects, it may face challenges in domains requiring problem-solving (e.g., mathematics derivations or programming), where flashcards alone may be insufficient.

###### Computational-Constraints:

Advanced NLP models and transformer-based summarization can be computationally intensive, making real-time flashcard generation slower on low- resource devices without internet or GPU support.

###### Language&Multilingual-Barriers:

Current implementation supports primarily English-

language study material. Adapting the system for multilingual or domain-specific terminology requires additional fine-tuning and dataset expansion.

58

###### Offline-Limitations:

Since BrainBoost is designed as an offline-first application, access to large pretrained models may be restricted, potentially limiting the accuracy of AI-based summarization and question generation.

###### Overfitting-SM-2:

While SM-2 is highly effective for spaced repetition, it may not fully adapt to all learners’ memory patterns. Some students may require hybrid scheduling strategies (e.g., SM- 2 with adaptive neural reinforcement)

In summary, BrainBoost offers an innovative and effective approach to enhancing student retention but must overcome these limitations to achieve scalability, personalization, and robustness across diverse educational settings.

XIV. FUTURE WORK

Although Our System demonstrates effective recall prediction and memory retention, several directions can further enhance its performance, usability, and applicability:

###### Multilingual-Flashcard-Generation:

Extending the system to support multiple languages will make it accessible to learners across diverse linguistic and cultural backgrounds, enabling global usability.

###### Transformer-Based-Summarization:

Integrating transformer models (e.g., BERT, RoBERTa) can improve AI-generated flashcards by providing concise, context-aware summaries, enhancing comprehension and retention.

###### Gamification-Features:

Incorporating gamified elements such as XP points, streaks, and achievement badges can increase learner engagement and motivation.

1. **Mobile–Desktop Sync and Collaborative Learning:** Enabling cross-platform synchronization and collaborative features will allow learners to access flashcards seamlessly on multiple devices and support peer-to-peer learning environments.

###### Adaptive and Personalized Learning Enhancements:

Further refining the SM-2 algorithm with AI-driven

Page 10 of 10 - Integrity Submission Submission ID trn:oid:::2945:321695339

personalization can optimize review scheduling for individual learning patterns, maximizing long-term retention.



**2**

XVI. REFERENCES

1. *“The*

P. Woźniak,

World, [Online]. Available: [https://www.supermemo.com](https://www.supermemo.com/)

*Optimization of Learning,”* SuperMemo



**6**

**7**

1. K.
2. A. Y. Li, Z. Chen, “Transformer-based text summarization for AI-assisted learning,”

45678–45690,

*IEEE Access*, vol.

“BERT: Pre-training of deep bidirectional transformers for

language understanding,” in *Proc. NAACL-HLT*, 2019, pp.

4171–4186.

Devlin, M.-W. Chang, K. Lee, and K. Toutanova,

10, pp.

2022.



Liu,



and

1. D. Kornell Bjork, “

and H.

M. M. Gilbert, T. C. Frommeyer, G. V. Brittain,

temporal ridgeline of optimal retention,” *Psychological*

*Science*, vol. 19, no. 11, pp. 1095–1102, 2008.

“Distributed practice in verbal recall tasks: A review and

quantitative synthesis,” *Psychological Bulletin*, vol. 132, no.

3, pp. 354–380, 2006.

Spacing effects in learning: A

Cepeda, H. Pashler, E. Vul, J. Wixted, and D. Rohrer,

A



**1**



**4**

1. R.
2. et al., “

School,” *Medical Science Educator*, vol. 33, pp. 1235–1243, 2023. doi:10.1007/s40670-023-01826-8

1. B. Paddags, “Automated Sentence Generation for a Spaced Repetition Flashcard System,” in *Proc. EMNLP*, 2024, pp. 200–210.

Cohort Study Assessing the Impact of Anki as a Spaced

Repetition Tool on Academic Performance in Medical

1. J. Zhao, “

7]

Jones, and

Oriented Repetition for Adaptive Spaced Learning,

*arXiv*

*preprint*,

2025.

[Online].

Available:

[https://arxiv.org/abs/](https://arxiv.org/abs/2508.03275)

LECTOR: LLM-Enhanced Concept-based Test-

Milburn, “Comparing Spaced



**3**



**9**



**3**

”

[2508.03275](https://arxiv.org/abs/2508.03275)

[ S. Colbran, W. J.



**3**

”

[387331417](https://www.researchgate.net/publication/387331417)

1. A. Singh, A. Ghosh, and P. Gupta, “AI-powered flashcard generation using natural language processing for adaptive



**8**

3121–3139,

*ResearchGate*,

Repetition Algorithms for Digital Flashcards,

2025.

[Online].

Available:

[https://www.researchgate.net/publication/](https://www.researchgate.net/publication/387331417)

learning,” *Education and Information Technologies*, vol. 28,



no. 3, pp.



2023.

59

**CHAPTER 9**

REFERENCES

* 1. A. Al Faraby, A. Adiwijaya, and A. Romadhony, “Review on Neural Question Generation for Education Purposes,” *International Journal of Artificial Intelligence in Education*, Oct. 2023.
  2. A. Cooper, R. Han, and D. Anderson, “The Effect of Spaced Repetition Learning Through Anki on Medical Board Exam Performance,” *International Journal of Medical Students*, vol. 11, no. 3, pp. 275–283, Dec. 2023.
  3. H. Kido, H. Kido, and M. Takahashi, “Automatic Question Generation for the Japanese National Nursing Examination Using Large Language Models,” in *Proc. 16th International Conference on Agents and Artificial Intelligence (ICAART)*, SciTePress, 2024.
  4. L. Rezigalla, “AI in medical education: uses of AI in construction of type A multiple-choice questions,” *BMC Medical Education*, vol. 24, no. 52, pp. 1–9, 2024.
  5. M. T. Islam, P. L. Charles, and S. W. Suen, “A Comparative Study of AI- Generated (GPT- 4) and Human-Crafted MCQs in Programming Education,” *arXiv preprint* arXiv:2312.03173, 2024.
  6. S. Maity, D. Deroy, and S. Sarkar, “Leveraging In-Context Learning and Retrieval- Augmented Generation for Automatic Question Generation in Educational Domains,” *arXiv preprint* arXiv:2501.17397, Jan. 2025.
  7. A. Zeghouani, H. Meftouh, and N. Aloulou, “Examining the Feasibility of AI- Generated Questions in Educational Settings,” in *Proc. Technology-Enhanced Assessment Systems (TAS’24)*, ACM, 2024.
  8. A. Bhowmick, R. Kumar, and P. Roy, “Automating Question Generation from Educational Text,” *arXiv preprint* arXiv:2309.15004, Sep. 2023.
  9. A. Al Faraby, A. Adiwijaya, and A. Romadhony, “Neural Question Generation for Education: A Survey,” *Springer AI Education Review*, 2023.
  10. S. Maity and D. Deroy, “The Future of Learning in the Age of Generative AI: Automated Question Generation and Assessment with Large Language Models,” *arXiv preprint* arXiv:2410.09576, Oct. 2024.

60