

# Indian Journal of Modern Research and Reviews

This Journal is a member of the '*Committee on Publication Ethics*'

Online ISSN:2584-184X



*Research Article*

## Design and Development of Smart Notes Management System Using Firebase and AI

**Saurabh Gupta**

Assistant Professor, Department of Computer Science and Engineering  
IIMT Engineering College, Dr. A.P.J. Abdul Kalam Technical University (AKTU)  
Meerut, Uttar Pradesh, India

**Corresponding Author:** \*Saurabh Gupta

**DOI:** <https://doi.org/10.5281/zenodo.19329398>

### Abstract

The rapid adoption of digital learning resources in academic environments has significantly improved the distribution and accessibility of educational materials. However, existing methods of sharing academic notes through informal channels such as messaging platforms, emails, and cloud storage often lead to issues including file duplication, outdated materials, lack of version control, and poor organisation.

This paper presents the design and development of a Smart Notes Management System that provides a centralised platform for storing, organising, and managing academic notes using Firebase cloud services. The proposed system incorporates role-based access control for administrators, tutors, and students, ensuring structured management of academic resources based on academic year, semester, and subject.

Additionally, an AI-assisted module is integrated to generate concise summaries and extract metadata from uploaded documents to enhance document readability and navigation. Experimental evaluation demonstrates that the proposed system improves accessibility, reduces redundancy, and provides a scalable solution for academic note management.

### Manuscript Information

- **ISSN No:** 2584-184X
- **Received:** 20-02-2026
- **Accepted:** 26-03-2026
- **Published:** 30-03-2026
- **MRR:**4(3); 2026: 375-381
- **©2026, All Rights Reserved**
- **Plagiarism Checked:** Yes
- **Peer Review Process:** Yes

### How to Cite this Article

Gupta S. Design and Development of Smart Notes Management System Using Firebase and AI. Indian J Mod Res Rev. 2026;4(3):375-381.

### Access this Article Online



[www.multiarticlesjournal.com](http://www.multiarticlesjournal.com)

**KEYWORDS:** Smart Notes Management System, Firebase Cloud Storage, Artificial Intelligence, Document Summarization, Academic Note Management.

## 1. INTRODUCTION

The increasing use of digital academic resources has transformed the way educational content is shared and accessed in modern learning environments. Academic materials such as notes, assignments, and reference documents are now commonly distributed in digital formats rather than traditional printed copies. Despite this transition, the management and organization of academic notes in many institutions remain unstructured and inefficient.

Students frequently rely on informal methods such as messaging applications, shared cloud links, and peer-to-peer file sharing to obtain study materials. These approaches often result in issues including loss of important files, duplication of documents, outdated versions, and lack of proper authorship tracking. Such challenges create confusion among students and faculty members and reduce the effectiveness of knowledge dissemination.

Although existing Learning Management Systems and cloud storage platforms provide basic file-sharing capabilities, they often lack structured organization, real-time synchronization, and support for understanding large academic documents. Furthermore, these systems typically do not provide intelligent assistance for processing or summarizing lengthy learning materials.

To address these challenges, this study proposes a Smart Notes Management System that provides centralized storage, structured organization, and improved accessibility of academic

notes using Firebase cloud services. The system integrates artificial intelligence techniques to generate summaries and assist users in understanding uploaded documents. The proposed solution aims to enhance reliability, accessibility, and efficiency in academic note management.

## 2. OBJECTIVES

The main objectives of the proposed system are:

- To provide a centralized and secure platform for academic note management.
- To reduce duplication and confusion caused by multiple document versions.
- To enable structured access based on academic hierarchy.
- To support students through AI-generated document summaries.

## RELATED WORK

One of the major concerns for both scholars and students has always been note-taking academically. Academic notes must be stored in an appropriate manner, must be sharable, and must be updated frequently too. In recent years, several technological approaches have been developed to address these concerns successfully.

This section reviews the existing work related to academic note management systems. It also points out the limitations of existing solutions and the need for the development of a more effective solution.

**Table 1:** Comparison of Existing Systems with Proposed Smart Notes Management System

Feature	Traditional Note Sharing Methods	Cloud Storage Platforms	AI-Based Tools	Proposed Smart Notes Management System
Centralized Storage	Not Available	Available	Not Available	Available
Version Control	Not Supported	Partially Supported	Not Supported	Fully Supported
Structured Academic Organization	Not Available	Limited Support	Not Available	Fully Supported
Role-Based Access Control	Not Available	Available	Not Available	Available
AI-Based Document Summarization	Not Available	Not Available	Available	Available
Duplicate File Detection	Not Supported	Not Supported	Not Supported	Supported
Real-Time Data Synchronization	Not Available	Available	Not Available	Available
Academic Workflow Integration	Not Supported	Limited Support	Not Supported	Fully Supported

### A. Mobile and Firebase-Based Academic Applications

Many researchers and developers exist who have developed different Android-based applications for sharing study materials like notes, assignments, and announcements. They use Firebase services such as Authentication, Realtime Database, and Cloud Storage.

#### Firestore helps these applications by:

- Providing secure login
- Allowing real-time data synchronisation
- Making files accessible from anywhere

Students can download these notes on their mobile devices at any time.

However, most Firebase-based academic applications only treat notes as simple files that can be uploaded and downloaded. They usually do not provide proper tools for managing different

versions of notes or preventing old and outdated notes from being shared. In many cases, organizing the notes is completely dependent on faculty members. Manual organisation can lead to duplicate files, missing updates, and confusion among students. Although Firestore improves availability, it does not fully solve the problem of reliable and organised note management.

### B. Cloud Document Management Systems

Cloud document management solutions are employed in an organisational setup to handle a large number of files with efficiency. The cloud document management solution consists of the following functions:

- Centralised storage
- Version control
- Access permissions
- Audit logs

Research shows that these systems help reduce data redundancy and improve document reliability.

However, most enterprise document management systems are designed for business use and are too complex for academic environments. They are not built with student workflows in mind. These systems do not support semester-wise or subject-wise management, which is required for educational organisations. Due to a lack of such support, enterprise management systems are inappropriate for academic note management for students.

### C. AI-Assisted Educational Tools

Increasingly, artificial intelligence is being adopted in educational institutions. The capabilities and uses of artificial intelligence include the ability to:

- Document summarisation
- Content recommendation
- Personalised learning

Text summarisation assists students in grasping large documents easily. Recommendation systems assist students in fetching them the material to learn. They are not directly integrated into academic note-sharing systems. In many cases, AI tools process only static text and do not support frequently updated academic notes with multiple versions. This limits their practical use in real academic workflows.

### D. Research Gap and Motivation

Based on the existing literature, the following limitations can be found:

- Most academic note-sharing applications only support file upload and download. They do not handle version control or duplicate files properly.
- Enterprise cloud document management systems are powerful but are not designed for academic environments.
- AI-based summarisation and recommendation tools are rarely integrated into academic note management platforms.
- No platform exists that incorporates cloud storage, a structured educational format, and artificial intelligence support for notes.

The above factors highlight the importance of a system that can be tailored exclusively to handle educational note-taking.

### E. Positioning of the Proposed System

A Smart Notes Management System is designed to deal with the above-discussed limitations. As opposed to other systems, it has a centralised and mobile-based platform that consists of:

- Firebase-based cloud infrastructure
- Structured academic organisation
- AI-assisted document support

The system focuses on practical implementation and real academic workflows rather than complex research-level

experimentation. This makes it suitable for real-world educational institutions and student-level project deployment.

## 3. PROPOSED SYSTEM / METHODOLOGY

I explain the design approach and the operational workflow of the Smart Notes Management System. I use steps, with Firebase cloud services and basic AI-assisted document processing. I make the management of notes easier. I make the accessibility of notes better.

### A. Overall System Approach

The Smart Notes Management System is designed as a cloud-based mobile application that facilitates efficient storage, organisation, and retrieval of academic notes. The system supports three user roles: administrator, tutor, and student, each with specific access privileges.

Firebase is used as the backend platform to manage authentication, cloud storage, and real-time database synchronisation. Tutors upload academic notes in PDF format, which are securely stored in cloud storage. An AI-assisted module processes uploaded documents to generate summaries and extract relevant information, thereby improving document accessibility.

**The Smart Notes Management System is designed as a cloud-based mobile application. It supports three types of users:**

- Administrator
- Tutor
- Student

Every type of user has its own role and privileges.

It uses Firebase as its backend platform. It handles:

- User authentication
- Cloud storage of notes
- Real-time database synchronisation

Tutors upload notes in PDF format. These PDFs are stored securely in Firebase Cloud Storage and become available to students. Once the upload has been conducted, there is a basic AI-assisted module ready to process the uploaded document in order to produce a short summary as well as perform basic information extraction.

This ensures the system is faster, secure, and user-friendly.

### B. System Architecture

The system architecture consists of four main components that work together to provide smooth note management.

**The proposed architecture consists of four major components:**

1. **Android Client Application:** Provides user interface for uploading, managing, and accessing notes.
2. **Firebase Authentication:** Ensures secure user login and role-based access control.
3. **Cloud Storage and Database:** Stores documents and associated metadata.

4. **AI Processing Module:** Generates summaries and extracts key information from uploaded files.

### C. Functional Workflow

The procedure consists of the following steps:

The system operates through the following workflow:

1. Users authenticate using Firebase authentication services.
2. The system verifies user roles and grants access accordingly.
3. Tutors upload PDF documents through the application interface.
4. Files are stored in cloud storage, and metadata is recorded in the database.
5. The AI module generates document summaries.
6. Students access and download notes along with summaries.

This workflow keeps notes organised and easy to access for all users.

### D. Duplicate Handling and Note Updates

The system includes a duplicate detection mechanism that identifies similar files based on subject and semester information. Users are notified when duplicate files are detected, ensuring version consistency and reducing redundancy.

#### When a tutor uploads a note:

- The system checks for existing notes of the same subject and semester
- If a similar file name or upload context is detected, the tutor is asked whether it is a new version or a separate document

This approach prevents unnecessary duplicate files and ensures students always access the latest version of notes.

### E. Design Rationale

The system is designed with a focus on:

- Simplicity
- Reliability
- Ease of use

#### Firestore is chosen because it provides:

- Real-time synchronization
- Scalability
- Easy integration with mobile applications

The AI-assisted features are kept lightweight so that the system runs fast even on low-end devices.

By combining cloud-based storage with basic AI support, the Smart Notes Management System provides an efficient and practical solution for managing academic notes at a student project level.

### Implementation

This section explains about the practical implementation of the Smart Notes Management System. The system is developed as

an Android mobile application and uses Firebase cloud services as the backend. Firebase helps in handling user login, real-time data sync, and secure storage of academic notes in a central location.

I designed the implementation to keep the system simple, easy to use and reliable. Role-based access control makes sure that authorised users can reach features. The system also supports content updates.

### A. Development Environment

The application was developed using standard mobile development tools and cloud platforms that are commonly used in academic and industry projects.

#### Software Tools Used

- Android Studio (latest stable version)
- Android SDK
- Firebase Console
- Java/Kotlin for Android development

#### Cloud Services Used

- Firebase Authentication
- Firebase Cloud Storage
- Firestore/RealtimeDatabase

We chose these tools because the tools are easy to integrate. The tools are well documented. The tools are good, for student level application development.

### B. User Roles and Access Control

I have seen the system support three user roles:

#### Administrator

I see the administrator manages user registrations. The administrator also assigns roles, like tutor or student. The administrator monitors the content.

#### Tutor

Tutors upload academic notes in PDF format. They can update notes whenever there are changes in the study material. Tutors organize notes by selecting the correct subject and semester during upload.

#### Student

Students can download notes for the subjects they are enrolled in. Students have read access but cannot upload any content. When I built the system, I made sure that role-based access control uses Firebase Authentication. I store the user role information in the database. I check the role after login. Role-based access control checks the users. Role-based access control gives the users the features that match the user's role.

### C. Notes Upload and Storage

Tutors upload academic notes through the Android application interface. When a PDF file is selected:

1. I upload the file to the Firebase Cloud Storage. The file now sits in the Firebase Cloud Storage.
2. The system creates the storage path. The unique storage path draws from the subject and semester.
3. The system creates a download URL. The secure download URL lets the file be accessed.
4. We store the note metadata in the database. The note metadata includes the title, the subject, the uploader name and the upload date.

I think the method keeps the notes safe. I trust the method. The method lets the authorized users get the notes fast.

#### D. AI-Assisted Summary Generation

Students need notes that're easy to read. The AI-assisted feature makes notes easier to read. The AI-assisted feature saves time for the students. The AI-assisted feature creates summaries of the uploaded notes. Short summaries of the notes let the students understand the material faster. After the tutor uploads a document:

- The AI module extracts important text from the PDF.
- A short summary is generated automatically.
- The summary is stored along with the note metadata.

Students read the summary before students download the document. Students understand the topic quickly. Students decide whether students need the file.

#### E. User Interface Design

The application interface is simple and user-friendly.

##### Main Screens

- **Login Screen:** Secure login using email and password
- **Dashboard:** Displays available subjects and recently uploaded notes
- **Upload Screen:** Allows tutors to upload PDF notes
- **Notes View Screen:** Allows students to browse and download notes

I notice the interface has navigation. I notice the interface also has layouts and works smoothly on Android devices.

#### F. Testing and Validation

The system was tested using sample notes uploaded by tutors. Functional testing was carried out to verify:

- Successful user authentication
- Correct role-based access control
- Proper upload and download of PDF files
- Accurate display of metadata and summaries

Testing results confirmed that the system works as expected and provides reliable access to academic notes.

## 4. RESULTS AND OBSERVATION

The proposed system was evaluated through functional testing and user feedback. The results indicate that the system

successfully supports secure authentication, structured note organization, and efficient document retrieval.

Compared with traditional note-sharing methods such as messaging platforms and email distribution, the system provides improved organization, reduced duplication, and enhanced accessibility. User feedback indicated that the system is easy to use and improves academic resource management.

However, the system performance depends on internet connectivity, and large file uploads may require additional processing time. The quality of generated summaries also depends on the clarity of text within uploaded documents.

I tested the system using the sample notes that the tutors uploaded. The students opened the sample notes through the Android application.

#### A. Functional Testing Results

Test engineers performed tests. The functional tests checked each module of the system. The functional tests showed that every module of the system worked correctly.

##### 1. User Authentication

I saw the login system work with Firebase Authentication. I saw users enter their email and password, and the system identified each user's role and directed them to the dashboard.

##### 2. Notes Upload and Storage

I saw tutors upload PDF files without any errors. Firebase Cloud Storage stored the files securely. The system generated the download links. I saw students download the files without any access issues.

##### 3. Metadata Management

I noticed that the system correctly stored the subject name, semester, tutor name, and upload date in the database. I saw the system display the notes by subject and semester. The perspective was simple.

##### 4. Display the summary.

You upload a file. The summary generated by the AI-assisted feature. Shown with the note. The students read the summary before the students download the document.

#### B. Comparison with Traditional Methods

The team compared the system with note-sharing methods. Traditional note-sharing methods included WhatsApp groups, email sharing, and shared cloud folders.

We saw the following improvements:

- The proper categorization makes the notes easier to find.
- I see that students do not rely on the files or the expired links
- We reduced the risk of downloading material
- Academic content was available from a single platform

#### C. Usability Feedback

I saw a small group of students and tutors test the application. The students and tutors gave feedback.

- I observed that students found the application easy to use
- Tutors liked the upload process. The upload process was simple.

- Users liked the way notes were organized by subject

The interface looks clean. The interface is easy to understand for first-time users.

#### D. Performance Observations

We tested the application under internet conditions. The application worked as expected. I saw that upload and download operations run smoothly. I saw that upload and download operations work well for the PDF file sizes. The sync made notes appear instantly on the student devices. I used the app many times and it stayed stable each time.

#### E. Observed Limitations

We ran the test. The test revealed the limitations:

- The upload speed depends on the internet connection. The internet connection sets the upload speed.
- I notice that the large PDF files take time to upload.
- The quality of the text within the PDF determines the quality of the summary. I see that when the text quality is clear the summary quality improves.

I believe the system's limitations can be improved by the developers.

## 5. CONCLUSION AND FUTURE SCOPE

### A. CONCLUSION

The Smart Notes Management System provides an efficient solution for managing academic notes through centralized storage, structured organization, and AI-assisted document processing. The system improves accessibility, reduces redundancy, and enhances academic resource management.

Future enhancements may include advanced AI summarization techniques, multilingual support, offline access capabilities, advanced search features, and integration with existing learning management systems.

### B. Future Scope

Though the present implementation meets the basic requirements, there are some improvements to be incorporated in the future versions to enhance the efficiency of the above process.

1. **Enhanced AI Summarization:** More advanced summarization techniques can be integrated to generate exam-oriented short notes, important questions, and key concepts.
2. **Multilingual Support:** The software can be developed to include several regional languages so that students speaking different languages find no difficulty in fetching study materials
3. **Offline Access:** Offline access can be provided by implementing local caching so that students can read previously downloaded notes without an internet connection.

4. **Advanced Search and Filtering:** Search options based on keywords, topics, subjects, and upload dates can be added for faster navigation.
5. **Student Feedback and Ratings:** A feedback and rating system can help tutors improve the quality and relevance of uploaded notes.
6. **Analytics Dashboard:** Usage analytics can be implemented to help faculty understand which notes are accessed frequently and identify gaps in academic resources.
7. **Integration with Learning Platforms:** The System can be interfaced with existing Learning Management Systems to form a comprehensive e-Academic Environment. These improvements can make the system more powerful, intelligent, and capable in a large-scale educational setting.

These enhancements can make the system more powerful, intelligent, and suitable for large-scale academic environments.

## REFERENCES

1. Tiwari A, Kumar S. A Firebase-based secure cloud storage system for academic document management. *IEEE Access*. 2021;9:112234-112245.
2. Chowdhury MZ, Shahjalal M, Ahmed S, Jang YM. A cloud-based architecture for e-learning using mobile devices. In: *Proceedings of the IEEE International Conference on Consumer Electronics*. 2020. p.130-135.
3. Jain S, Patel R. Android-based mobile application for academic content sharing. *International Journal of Computer Applications*. 2020;176(23):1-7.
4. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies*. 2019. p.4171-4186.
5. Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*. 2019. p.3982-3992.
6. Zhang Y, Wang X, Huang F. Hybrid extractive-abstractive summarization using transformer networks. *IEEE Transactions on Neural Networks and Learning Systems*. 2022;33(7):3104-3115.
7. Kumar R, Yadav A. Duplicate detection in academic documents using semantic similarity. In: *Proceedings of the IEEE International Conference on Data Mining Workshops*. 2021. p.550-557.
8. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, *et al*. Attention is all you need. In: *Proceedings of the Neural Information Processing Systems Conference*. 2017. p.5998-6008.
9. Dean J, Barroso LA. The tail at scale: Cloud architecture and service reliability. *Communications of the ACM*. 2013;56(2):74-80.

10. Google Firebase. Firebase documentation: authentication, storage, and cloud Firestore. Google Developers; 2023. Available from: <https://firebase.google.com/docs>
11. Mihalcea R, Tarau P. TextRank: bringing order into texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2004. p.404-411.
12. Lin H, Bilmes J. A class of submodular functions for document summarization. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics – Human Language Technologies. 2011. p.510-520.
13. Gupta S, Jain AK. Efficient PDF text extraction and pre-processing techniques for NLP tasks. Journal of Information Science. 2022;48(2):210-223.
14. Aggarwal P, Sharma D. Mobile cloud computing for education: performance and usability analysis. IEEE Transactions on Cloud Computing. 2022;10(1):105-118.
15. Mohanty S, Singh P. A smart content delivery system using AI-based text processing for universities. In: Proceedings of the IEEE International Conference on Smart Computing. 2021. p.220-227.

#### Creative Commons License

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution–Non-commercial–No Derivatives 4.0 International (CC BY-NC-ND 4.0) License. This license permits users to copy and redistribute the material in any medium or format for non-commercial purposes only, provided that appropriate credit is given to the original author(s) and the source. No modifications, adaptations, or derivative works are permitted.

#### About the corresponding author



**Saurabh Gupta** is an Assistant Professor in the Department of Computer Science and Engineering at IIMT Engineering College, affiliated with Dr. A.P.J. Abdul Kalam Technical University (AKTU), Meerut. He is dedicated to teaching, research, and mentoring students, with interests in advancing computing technologies and fostering academic excellence in engineering education.