PROJECT TITLE :-

University Library Management System (ULMS)

Course Code: CSE2006

Submitted By: Shreshth Gupta

Registration Number: 24BCE11307

Submitted To : VISHAL S. BHATI

Date : 24 NOVEMBER 2025

2. Introduction

The Notes Management System is a simple text-based Java application designed to store and manage user notes efficiently. It allows the user to create, update, delete, and search notes while maintaining persistent storage using file handling.

This project demonstrates OOP principles, modular programming, and basic data management.

---

3. Problem Statement

Traditional note-taking becomes unorganized, messy, and difficult to search. Digital alternatives may be complex or require internet access.

This project provides a minimal, offline, and efficient Java-based tool for managing notes.

---

4. Functional Requirements

1. Add Note

2. Edit Note

3. Delete Note

4. View Note

5. List All Notes

6. Search Notes

7. Filter Notes by Category

8. File-based Persistence

---

5. Non-Functional Requirements

- **Usability:** Simple text menu for interaction

- **Performance:** Fast file operations & search

- **Maintainability:** Modular class design

- **Reliability:** Notes are stored safely in text file

- **Portability:** Runs on any system with Java

---

6. System Architecture

Layers:

- Presentation Layer → Main.java

- Business Logic Layer → NoteManager.java

- Data Layer → Storage.java

- Model Layer → Note.java

Diagram (describe):

- User → Main → NoteManager → Storage → File

---

7. UML Diagrams

⍰ Use Case Diagram

- Add Note
- Edit Note
- Delete Note
- List Notes
- Search Notes
- Filter Notes

⍰ Class Diagram

Classes:

- Note
- NoteManager
- Storage

Relationships:

- NoteManager uses Note

- NoteManager depends on Storage

⍰ Sequence Diagram (Add Note)

User → Main → NoteManager → Storage → File

---

8. Design Decisions & Rationale

- Chose text file storage for simplicity
- Separated modules for cleaner design
- Used ID system to uniquely identify notes
- Used separators (`|||`) for safe file parsing

---

9. Implementation Details

- Language: Java
- IDE: IntelliJ IDEA
- Files: Note.java, NoteManager.java, Storage.java, Main.java
- Storage: data/notes.txt

---

10. Screenshots / Results

(Add IntelliJ Run Window screenshots here)

---

11. Testing Approach

- Manual testing of each menu option
- Edge cases:
  - Empty input
  - Invalid ID
  - Search with no results
  - Long notes

---

12. Challenges Faced

- Managing multiline input
- Ensuring proper file formatting
- Handling parsing and escaping newline characters

---

13. Learnings & Key Takeaways

- Java OOP concepts (classes, objects, methods)
- File handling with BufferedReader/Writer
- Designing modular and maintainable code
- CRUD implementation logic
- UML diagrams and documentation

---

14. Future Enhancements

- Swing or JavaFX GUI
- MySQL/SQLite database
- User login
- Dark mode / custom themes
- Export notes as PDF/TXT

---

15. References

- Oracle Java Documentation
- Course materials
- Class notes