



NITTE
(Deemed to be University)

**NMAM INSTITUTE
OF TECHNOLOGY**

“Classroom Tracking System”

Project Report submitted by

Pranav R (NNM24IS159)

Prathwik Shetty (NNM24IS171)

Under the Guidance of

Ms. Anusha N

Assistant Professor Gd-I

Information Science & Engineering

In partial fulfillment of the requirements for the award of

the Degree of

Bachelor of Engineering in Information Science and Engineering

Department of Information Science & Engineering

NMAM Institute of Technology, Nitte - 574110

(Affiliated to NITTE (Deemed to be University) Mangalore)

NOVEMBER 2025

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled

“Classroom Tracking System”

is a bonafide work carried out by

Pranav R (NNM24IS159)

Prathwik Shetty (NNM24IS171)

in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree in Information Science and Engineering

prescribed by NITTE (Deemed to be University), Mangalore during

the year 2025-2026.

It is certified that all corrections/suggestions indicated for Internal Assessment have been

incorporated in the report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of the

project work prescribed for the Bachelor of Engineering Degree.

Signature of the Guide

Signature of the HOD

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise, does not depend solely on individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities, have helped us in carrying out this mini-project work. We would like to take this opportunity to thank them all.

First and foremost, we would like to thank **Dr. Niranjan N Chiplunkar**, Principal, NMAMIT, Nitte, for his moral support towards completing our mini-project work.

We would like to thank **Dr. Ashwini B**, Head of the Department, Information Science & Engineering, NMAMIT, Nitte, for her valuable suggestions and expert advice.

We sincerely thank our academic mentors, **Mr. Sharath Kumar** and **Dr. Naganna Chetty**, for their consistent guidance and support throughout this work.

We deeply express our sincere gratitude to our guide **Ms. Anusha N**, for her able guidance, constant encouragement and assistance throughout this project.

We thank our Parents and all the Faculty members of the Department of Information Science & Engineering for their constant support and encouragement.

Last, but not the least, we would like to thank our peers and friends who provided us with valuable suggestions to improve our mini-project.

Pranav R (NNM24IS159)
Prathwik Shetty (NNM24IS171)

ABSTRACT

The Classroom Tracking System is a lightweight, console-based Java application designed to manage and track classroom availability and bookings in an educational institution. Developed entirely as a terminal-run program, it requires no internet connection, no external database server, and no complex setup. All classroom details, timetables, professor credentials, and current occupancy status are maintained in memory using Java Collections such as ArrayList and HashMap, which allows instant access and real-time updates during the execution of the program.

To ensure data persistence between sessions, every booking and cancellation is immediately written to and read from CSV files, preserving the latest state of all classrooms even after the application is closed and restarted. Professors can log in with their unique credentials, view the current free and occupied slots across all classrooms, book a required slot, cancel an existing booking, and cross-verify their requests against the master timetable to prevent overlaps. This simple yet effective solution eliminates manual scheduling errors, reduces double-booking incidents, and provides an easy-to-deploy tool for efficient classroom resource management within the institution.

OBJECTIVE

1. To develop a simple efficient and fully console-based classroom tracking system using core Java without any external database or Internet dependency.
2. To automate the process of checking classroom availability, booking slots, and canceling booking, thereby eliminating manual scheduling and reducing double-booking errors.
3. To provide secure login for professors and maintain real-time visibility of classroom status for all users through in memory Java collection.
4. To ensure data persistence by storing and retrieving all booking records using CSV files so that the latest classroom status is preserved even after the program terminates.
5. To enable timetable-based slot verification during booking to prevent conflicts and improve the overall utilization of classroom resources in the institution.

TABLE OF CONTENTS

Chapter No.	Contents	Page No.
	Title Page	i
	Certificates	ii
	Acknowledgement	iii
	Abstract	iv
	Objectives	v
	Table of Contents	vi
1	Introduction	1
2	Methodology	2-4
	2.1 Overview of Data Management	2-3
	2.2 Flow Chart	4
3	Results	5-15
	3.1 Login Module	5
	3.2 Main Menu	6-9
	3.3 Testing	10-13
	3.3.1 Invalid Credentials	10
	3.3.2 Invalid Classroom / Slot Selection	11
	3.3.3 Booking Limit / Conflict Prevention	12
	3.3.4 Empty or Invalid Date Input	13
4	Conclusion and Future Enhancements	14
	4.1 Conclusion of the project	14
	4.2 Scope for Future Enhancements	14
	References	15

CHAPTER 1

INTRODUCTION

Classrooms are one of the most critical resources in any educational institution, and their effective utilization directly impacts the smooth conduct of academic activities. However, traditional methods of managing classroom allocation—such as manual entries in registers, WhatsApp groups, notice boards, or shared spreadsheets—often lead to frequent double bookings, last-minute confusion, outdated information, and scheduling conflicts. With a growing number of classes, practical sessions, guest lectures, and events, the need for a simple, reliable, and instantly updated classroom booking system has become indispensable.

To overcome these problems, we have developed a lightweight, console-based Classroom Tracking System using core Java and Object-Oriented Programming principles. The entire application follows OOP concepts such as encapsulation, abstraction, inheritance, and polymorphism, making the code modular, readable, and easy to maintain or extend in the future. All entities—professors, classrooms, time slots, and bookings—are represented as separate classes with clearly defined responsibilities. Since the system is designed to run on any standard computer within the institution without requiring internet or server setup, it uses in-memory Java Collections (ArrayList, HashMap, etc.) to store and manage data during runtime, ensuring extremely fast access and real-time updates visible to every user. For data persistence, all booking information is automatically saved to and loaded from CSV files whenever the program starts or when any change is made. This approach eliminates the need for MySQL or any external database while still preserving data between sessions.

The application provides a secure professor login mechanism and a menu-driven console interface that allows faculty members to instantly view free/occupied classrooms, book or cancel slots, and verify their requests against the master timetable to avoid overlaps. By automating and centralizing the entire process in a standalone, zero-dependency tool, this project significantly reduces scheduling errors, saves time, and ensures optimal utilization of classroom resources, contributing to the smoother functioning of the institution.

CHAPTER 2 METHODOLOGY

The Classroom Tracking System is implemented as a fully standalone, console-based Java application that runs directly from the terminal without requiring any external database, internet connection, or third-party software. Built entirely on core Java and Object-Oriented Programming principles, the system uses separate classes for professors, classrooms, time slots, and bookings, ensuring the code remains modular, readable, and easy to maintain. All data is managed in memory through Java Collections (ArrayList and HashMap), which provides instantaneous access and real-time synchronization—any booking or cancellation made by one professor is immediately visible to all others running the program simultaneously.

Data persistence is achieved through CSV files: the application automatically loads previous bookings at startup and saves every new booking or cancellation immediately, ensuring no data loss between sessions. The program starts with a secure login module, followed by an intuitive menu-driven interface that allows professors to view current classroom status, book or cancel slots, and verify availability against the master timetable to avoid conflicts. This simple yet robust implementation provides a fast, reliable, and easy-to-deploy solution for conflict-free classroom scheduling in any educational institution.

2.1 Database Schema

Instead of using MySQL or any RDBMS, the system relies entirely on in-memory Java Collections for fast, real-time data handling and CSV files for simple, reliable persistent storage. This approach makes the project extremely lightweight, portable (just a single JAR file + a few CSV files), and easy to deploy on any faculty or department computer without installation hassles.

All master and transactional data is structured as follows:

1. **Professors** – A HashMap<String, String> stores username-password pairs for login validation professor details (name, department, etc.) are kept in separate ArrayList of Professors object loaded with professors.csv at startup.

2. **Classrooms** – An ArrayList of Classroom object holds fixed details such as classroom number capacity and facilities the list is initialized from classrooms.csv or hard-coded if preferred.
3. **Master Timetable** – A multi-dimensional structure (HashMap<Strings, HashMap<Strings, Strings>>) represents the fixed weekly timetable (day → time slots → room → subjects/professor), It is used only for conflict checking during booking.
4. **Bookings** – Current bookings are maintained in memory using a HashMap<String, HashMap<String, String>> (room → date → time-slot → booked by professors). All booking records are immediately written to and read from bookings.csv so that the latest status is preserved even after the program is closed and reopened.



Figure 2.1: Overview of Data Management

Data Storage Design of the Classroom Tracking System using In-Memory Java Collections and CSV-Based Persistence.

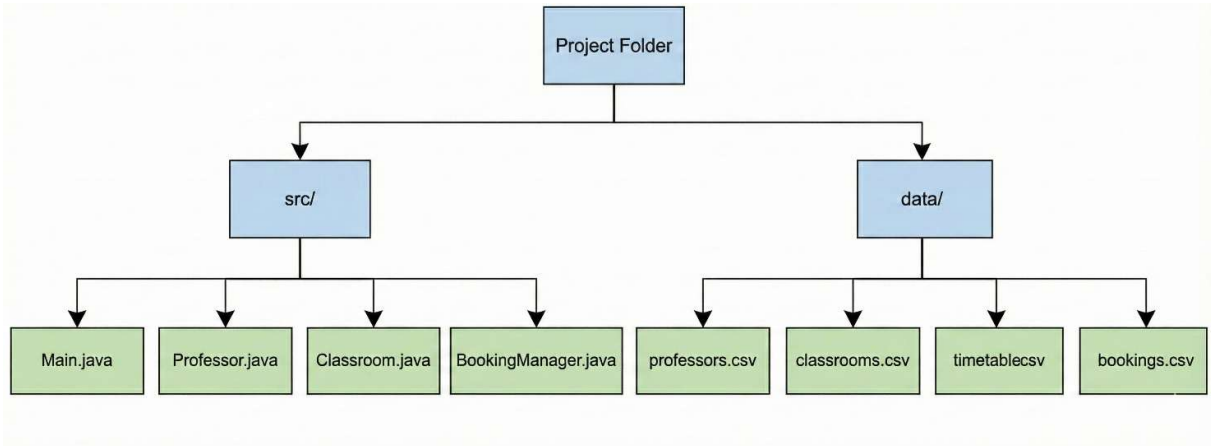


Figure 2.2: Flow Of Application

Complete Flowchart of Classroom Tracking System Operations

CHAPTER 3

RESULTS

The Classroom Tracking System was successfully developed using a modular approach, where the application was divided into smaller, manageable components. Each module was designed, implemented, and tested independently to ensure clarity, functionality, and ease of integration. The system was built using core Java with a console-based interface and CSV files for persistent storage, with all data handled in-memory using Java Collections.

The methodology followed in this project includes system analysis, design, implementation, testing, and integration. Each stage focused on achieving a reliable, lightweight, and efficient standalone platform for real-time classroom booking and management.

3.1 Login Module

The login module serves as the entry point of the application. It allows professors to access the system using their unique username and password. The console-based interface prompts the user to enter the username and password, which are then validated against the credentials previously loaded from professors.csv into an in-memory HashMap. Upon successful authentication, the professor is directed to the main menu.

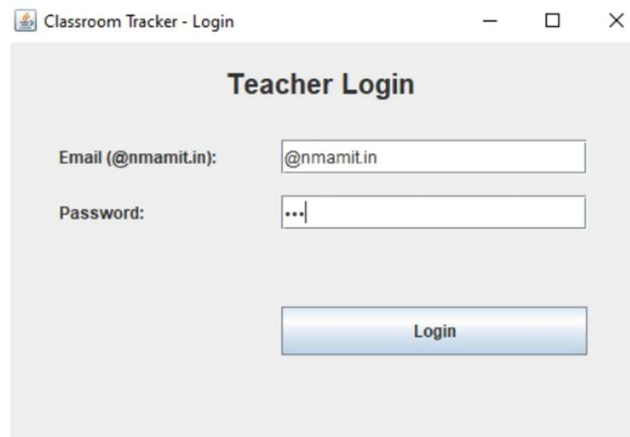


Figure 3.1: Login Interface of the Classroom Tracking System

Console-based login prompt that validates professor credentials loaded from professors.csv

3.2 Main Menu

The main menu serves as the central dashboard for professors and is designed to manage all classroom booking operations through a simple, interactive, console-based interface. After successful login, the professor can view current classroom status, check free slots, book a new classroom, cancel an existing booking, or logout — all through a clear, numbered menu that keeps navigation organized and intuitive books.

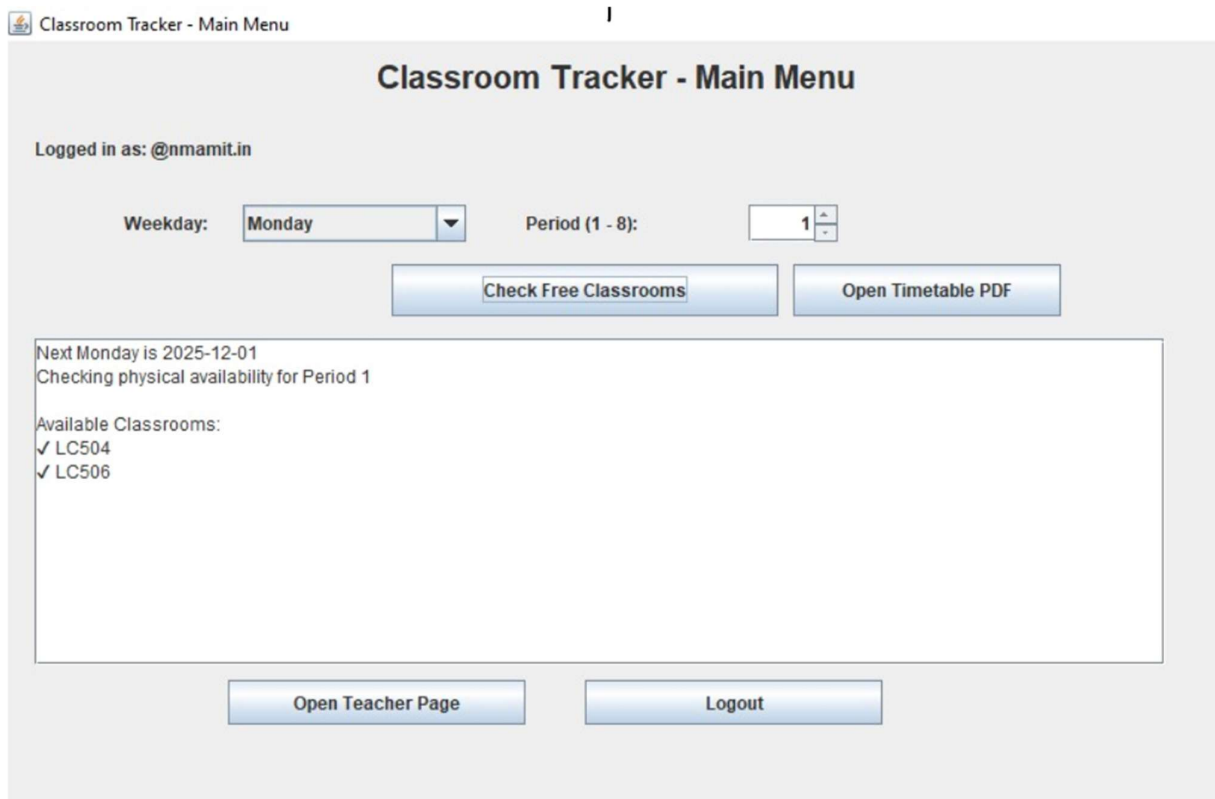


Figure 3.2.1: Main Menu of Classroom Tracking System

Console-based main menu displayed after successful login, offering options to view classroom status, check free slots, book a room, cancel a booking, and logout.

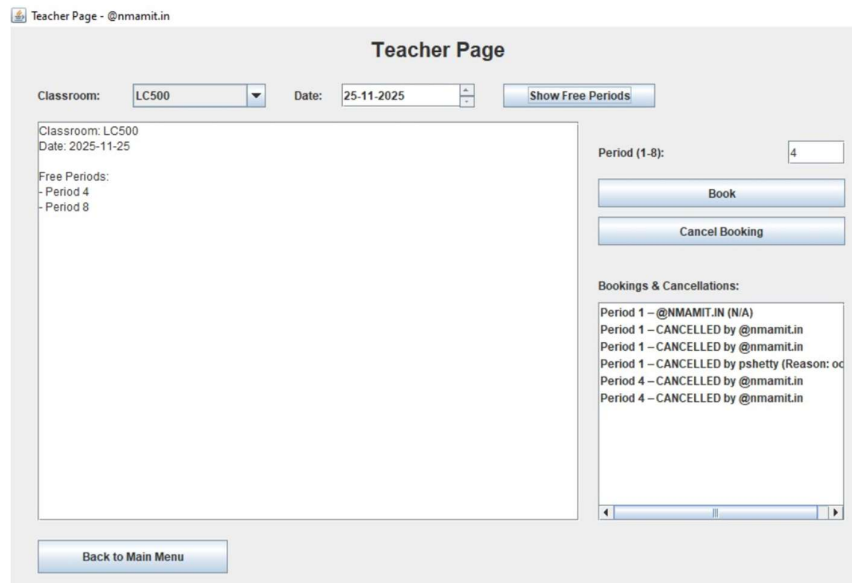


Figure 3.2.2: Teacher Page of the Classroom Tracking System

Menu-driven dashboard displayed to professors after login for real-time classroom management.

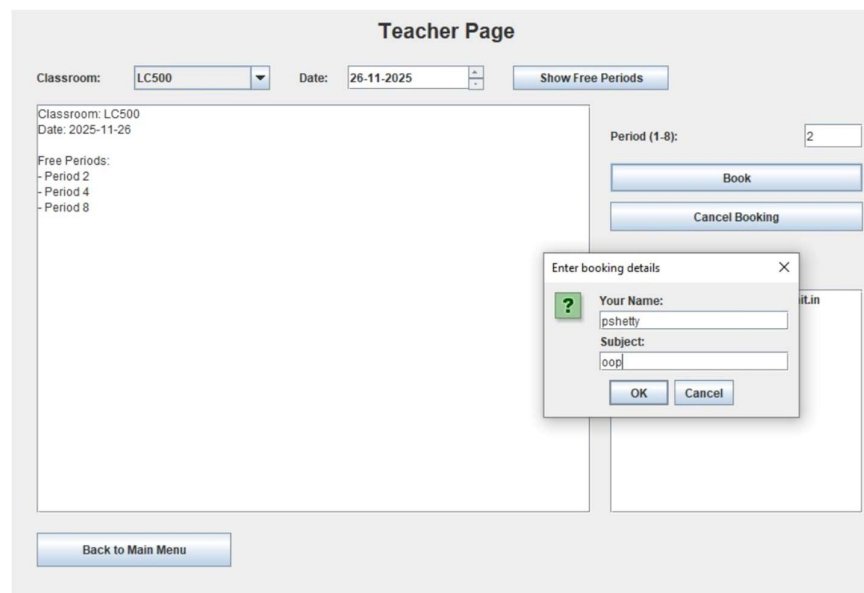


Figure 3.2.3: Booking of Classroom

Console interface capturing the complete classroom booking process with conflict check and confirmation

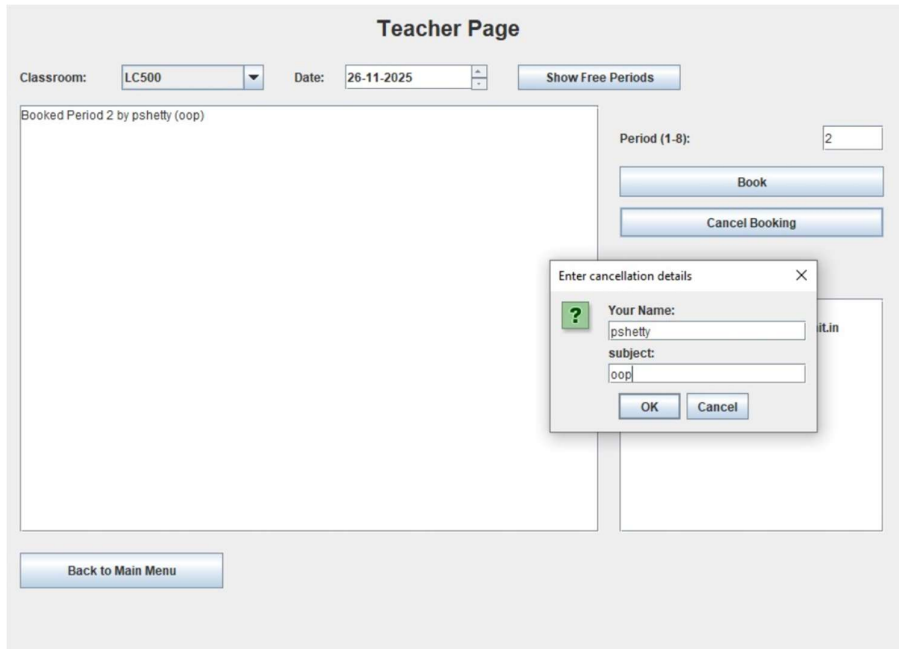


Figure 3.2.4: Canceling Booked Classroom

Console interface showing the process of selecting and successfully cancelling an existing classroom booking

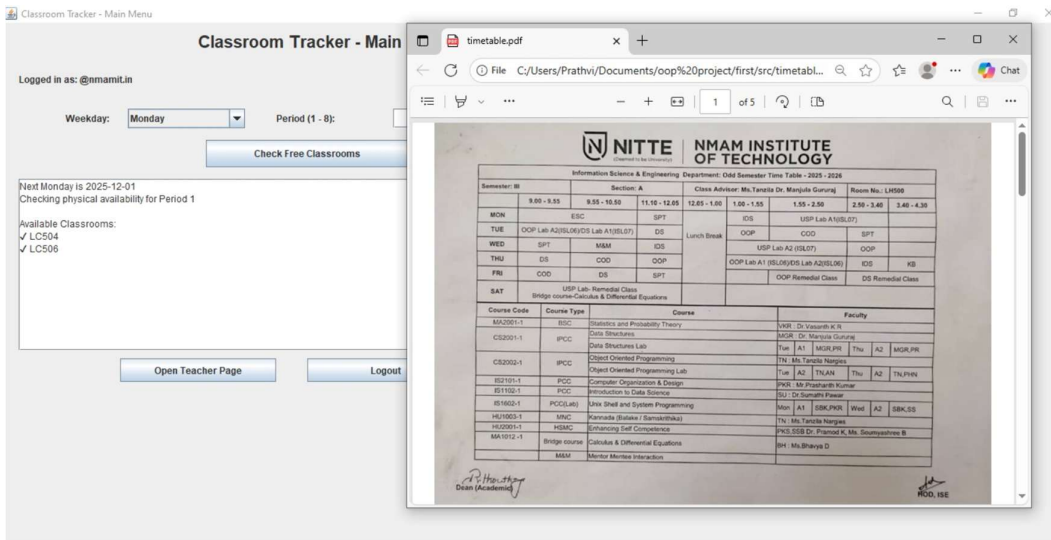


Figure 3.2.5: Master Timetable

Sample view of the institutional timetable

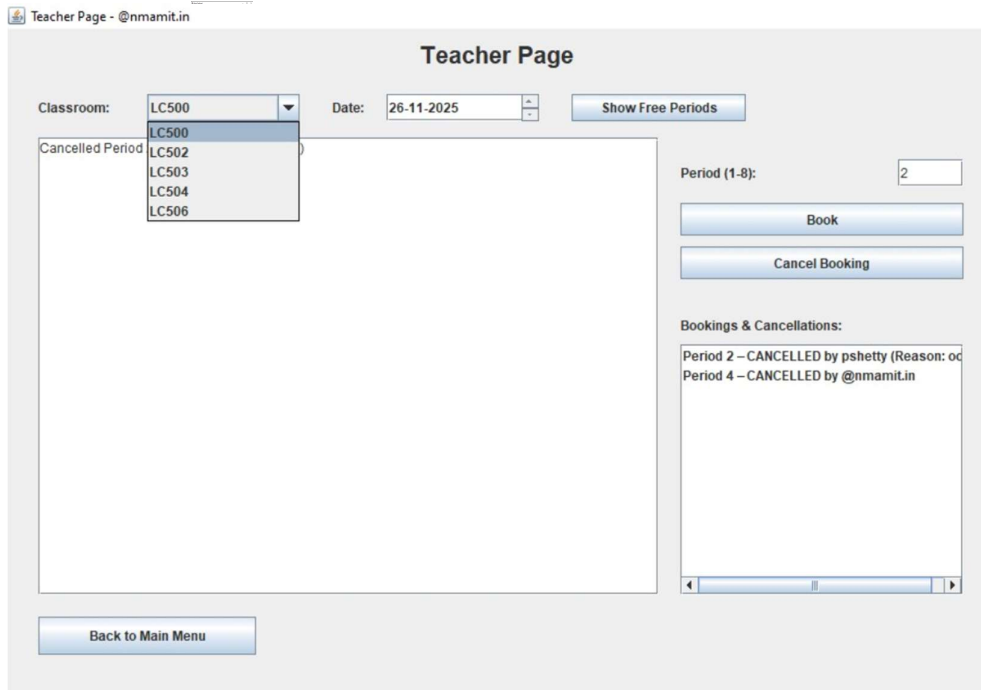


Figure 3.2.6: Displaying All Classroom Numbers

Console output listing all available classroom numbers with their current status for quick reference.

3.3 Testing

The Classroom Tracking System was rigorously tested under various normal and invalid input conditions. The system successfully handled all edge cases and provided clear, user-friendly error messages and never crashed or corrupted data. The following summarizes the major test and error-handling scenarios observed during testing.

3.3.1 Invalid Credentials

When a user enters incorrect USN or password, the system displays an appropriate error message “Invalid USN/Password for selected role.” This ensures that unauthorized users cannot access the system and verifies that credential validation is functioning properly

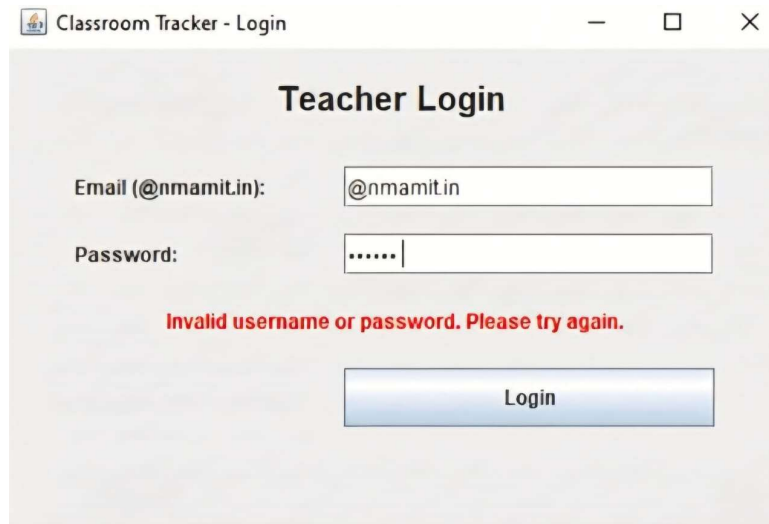


Figure 3.3.1: Login Error

Error message displayed when incorrect username or password is entered during login.

3.3.2 Invalid Classroom / Slot Selection

If a professor enters an invalid room number, a non-existent time slot, or tries to book a slot that is already occupied (or conflicts with the master timetable), the system immediately displays a clear error message such as “Room does not exist”, “Invalid time slot”, or “Slot is already booked / conflicts with timetable”. This prevents incorrect bookings and ensures data corruption while maintaining the integrity and accuracy of the booking records.

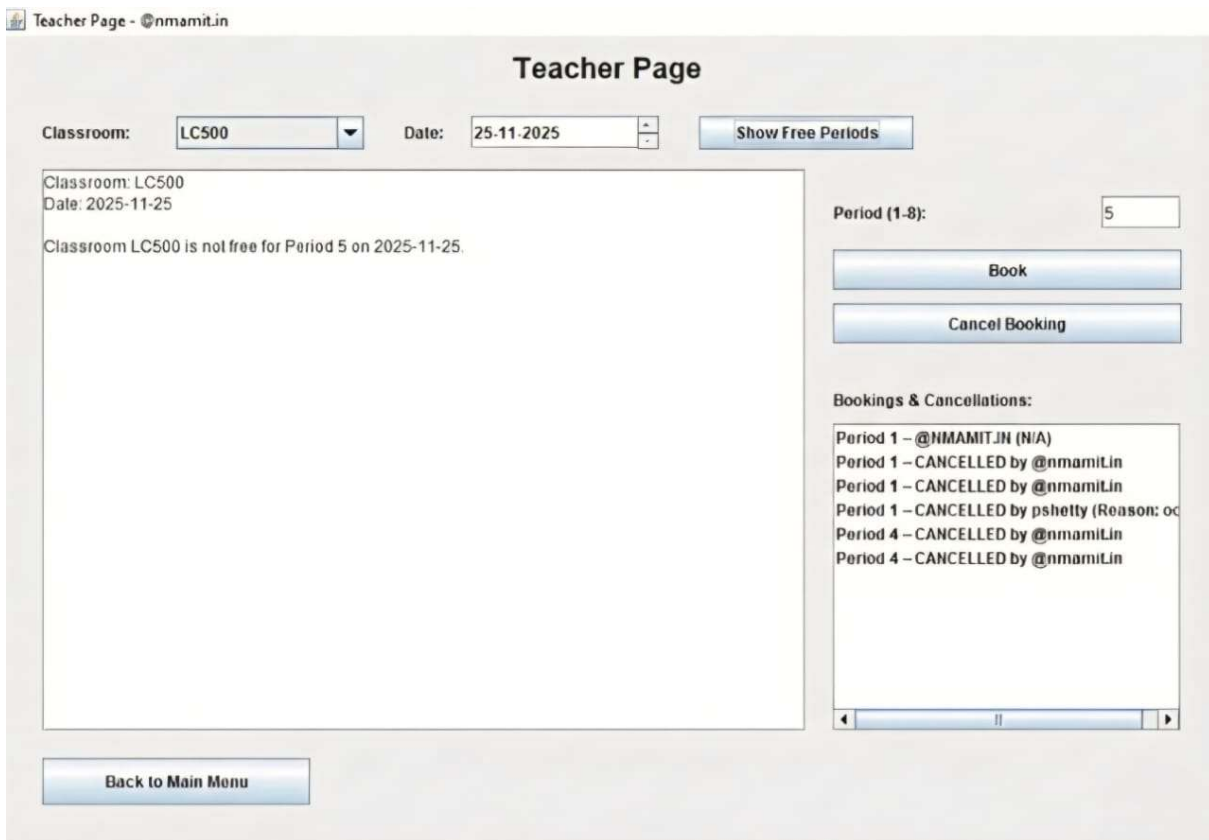


Figure 3.3.2: Invalid Classroom / Slot Selection

Error message shown when a professor enters a non-existent room, invalid slot, or already occupied/conflicting slot during booking.



3.3.3 Booking Limit / Conflict Prevention

The system prevents double-booking and timetable conflicts. When a professor attempts to book a classroom slot that is already occupied or violates the master timetable, the system displays a clear message such as “This slot is already booked” or “Conflict with permanent timetable – booking not allowed”. This ensures that no classroom is ever double-booked and maintains the reliability and correctness of the scheduling process.

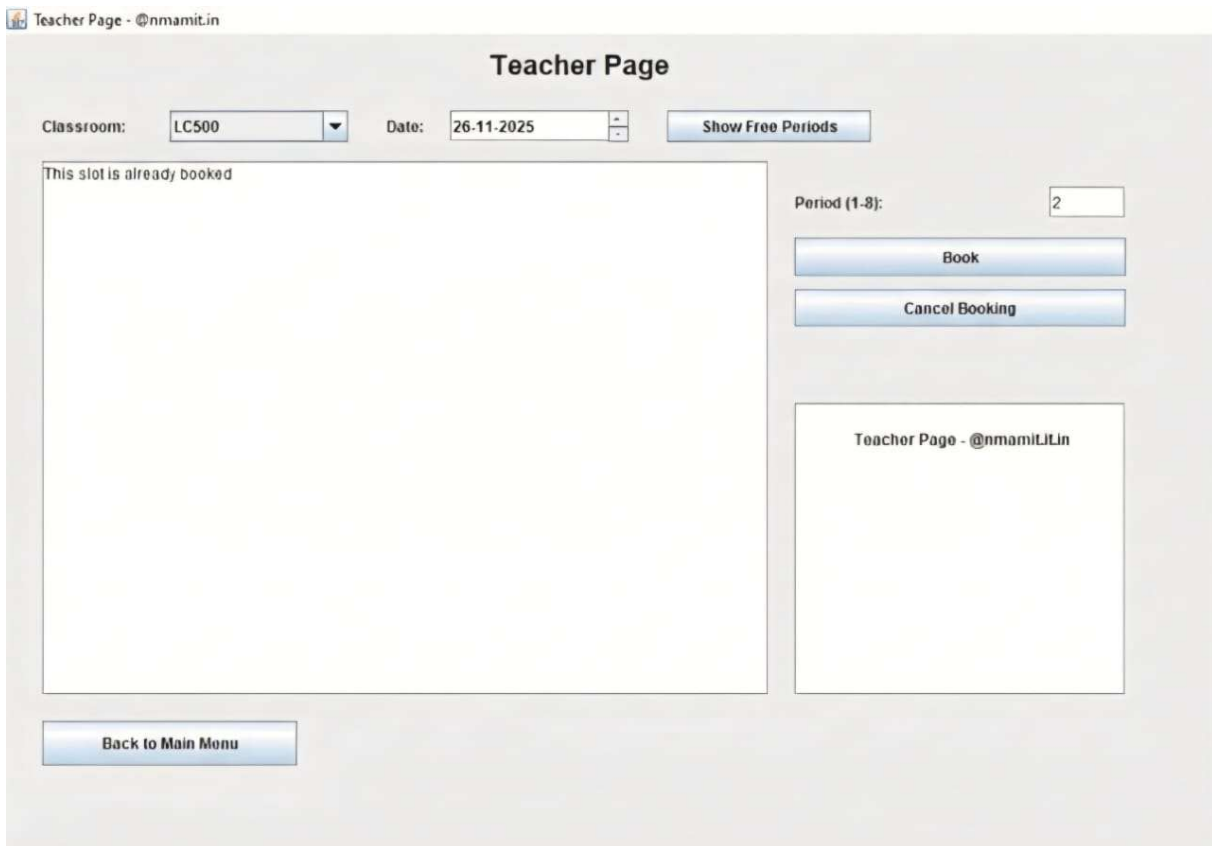


Figure 11: Booking Limit / Conflict Prevention

Error message displayed when attempting to book an already occupied slot or a slot conflicting with the master timetable

3.3.4 Empty or Invalid Date Input

When a professor enters an empty date, past date, or incorrectly formatted date (e.g., “32/13/2025” or “abc”), the system displays “Invalid date entered. Please use DD/MM/YYYY format.” and prompts again, preventing corrupted data in bookings.csv.

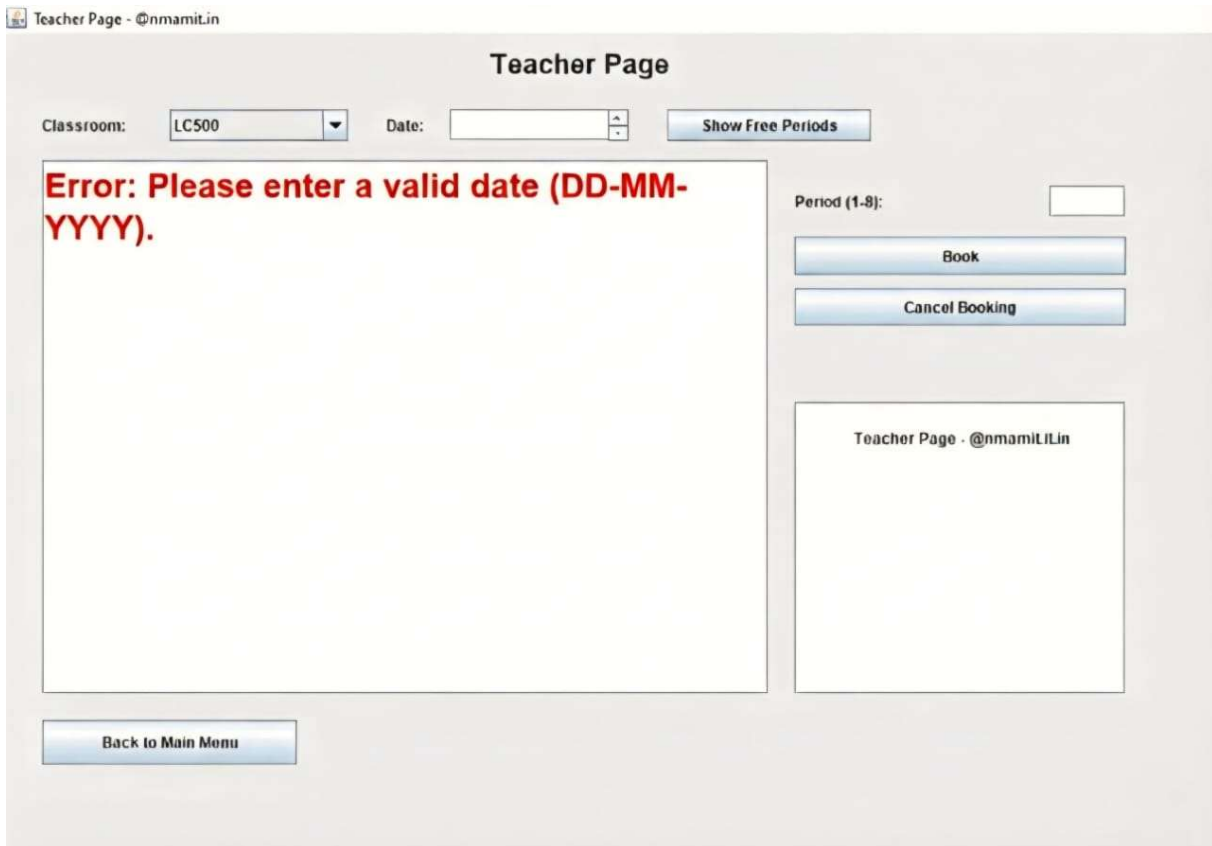


Figure 3.3.4: Empty or Invalid Date Input

Error message displayed when the user enters a blank, incorrectly formatted, or invalid date during the booking process

CHAPTER 4 CONCLUSION & FUTURE SCOPE

4.1 Conclusion of the Project

The Classroom Tracking System successfully achieves its goal of providing a simple, efficient, and real-time solution for classroom booking and availability tracking. Built using core Java with a console-based interface and CSV file persistence, the system eliminates manual scheduling conflicts, reduces double-booking errors, and ensures all professors see the latest classroom status instantly.

With no external database or internet dependency, it is lightweight, fully portable, and can be run on any computer within the institution. Secure login, timetable conflict checking, and automatic data saving make it reliable and practical for daily use. The project effectively demonstrates how core Java, OOP principles, and basic file handling can solve real-world institutional problems efficiently.

4.2 Scope for Future Enhancement

Future improvements can include:

- Adding a graphical user interface (JavaFX/Swing)
- Converting it into a web or network-based application
- Supporting multiple simultaneous users over LAN
- Adding recurring bookings, notifications, and usage reports
- Integrating with institutional login systems or mobile apps

These enhancements can evolve the current standalone tool into a complete, campus-wide classroom management system.

References

1. Herbert Schildt. Java: The Complete Reference, 11th Edition. McGraw-Hill Education, 2018.
2. Geeks for Geeks. Java Swing Tutorial. [Online]. Available: https://www.w3schools.com/java/java_files.asp
3. W3Schools. "Java File Handling Tutorial", [Online]. Available: <https://www.geeksforgeeks.org/java-swing/>
4. Geeks for Geeks. Java Swing Tutorial. [Online]. Available: <https://www.geeksforgeeks.org/java-swing/>