Beta Testing Document

for

Aviators

(Automated Lecture Hall Booking Portal)

Prepared by

Group 13: **Group Name:** EE Got Latent

Trijal Srivastava	**221144**	trijals22@iitk.ac.in
Snehasis Satapathy	**221070**	ssnishasis22@iitk.ac.in
Gautam Arora	**220405**	garora22@iitk.ac.in
Naman Gupta	**220686**	namangupta22@iitk.ac.in
Divyam Agarwal	**220376**	divyamag22@iitk.ac.in
Udbhav Agarwal	**221149**	audbhav22@iitk.ac.in
Deham Rajvanshi	**220335**	dehamr22@iitk.ac.in
Saksham Parihar	**220939**	psaksham22@iitk.ac.in
Dharvi Singhal	**220354**	dharvis22@iitk.ac.in

Course: CS253

Mentor TA: Mr. Vikrant Chauhan

Date: April 20, 2025

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
v1.0	Trijal Srivastava Snehasis Satapathy Gautam Arora Naman Gupta Divyam Agarwal Udbhav Agarwal Deham Rajvanshi Saksham Parihar Dharvi Singhal	The first version of the Test Document	20/04/2025

1 Introduction

Overview:

This project proposes the development of an **Automated Lecture Hall Booking Portal** aimed at streamlining and digitizing the process of reserving shared campus spaces. The current system for booking lecture halls, seminar rooms, and auditoriums is often manual, inefficient, and prone to conflicts or underutilization. Our proposed solution seeks to eliminate these issues by offering a **centralized, transparent, and real-time platform** that automates room availability checks, handles booking requests, and notifies users of approval statuses, significantly reducing the need for human intervention.

The portal is designed to serve a wide range of campus stakeholders, including **professors, teaching assistants, student clubs and societies, administrative staff, event organizers, lecture hall coordinators, and research groups**. With features like **advanced search filters** (based on equipment, capacity, location, etc.), **automated notifications**, and **conflict resolution**, the system will support both academic and non-academic use cases efficiently. Additionally, it will generate **usage analytics and reports**, helping campus authorities monitor utilization trends, optimize resource allocation, and make data-informed decisions.

Built with **scalability** in mind, the system can easily be extended to include spaces like auditoriums and rooms across the RM, KD, and D-JAB buildings. By combining **automation, cloud integration, and user-friendly interfaces**, this project offers a smart and flexible solution for modern campus resource management, enhancing transparency, reducing administrative burden, and improving the overall efficiency of lecture hall allocations.

2 List of Reported Bugs

BUG 1:

Tested Feature: Feedback Form

Tester Name: Gautam Arora

Testing Date: 11/04/25

Bug Details: Feedback form not resetting after submitting another feedback

Bug Report Date: 11/04/25

Has the bug been fixed? : Yes

Date of Bug fixing: 12/04/25

Any other comment: Fixed the issue by changing the code in the backend by the team members

BUG 2:

Tested Feature: Admin Page

Tester Name: Dharvi Singhal

Testing Date: 11/04/25

Bug Details: Rejected booking still being displayed on the admin page until manual refresh

Bug Report Date: 11/04/25

Has the bug been fixed? : Yes

Date of Bug fixing: 12/04/25

Any other comment: They corrected the code, and now the booking automatically gets removed from the list without having to refresh

BUG 3:

Tested Feature: New Lecture Hall Booking

Tester Name: Saksham Parihar

Testing Date: 11/04/25

Bug Details: Rejected booking does not free up the lecture hall for new bookings

Bug Report Date: 11/04/25

Has the bug been fixed? : Yes

Date of Bug fixing: 12/04/25

Any other comment: The bug is now fixed, and the rejected bookings now free up the slot and become a valid available option for the user.

BUG 4:

Tested Feature: Overall UI

Tester Name: Snehasis Satapathy

Testing Date: 11/04/25

Bug Details: Cross-Platform Issues

Bug Report Date: 11/04/25

Has the bug been fixed? : Yes

Date of Bug fixing: 12/04/25

Any other comment: The group members clarified that, as mentioned in the user manual, the portal is better suited for use on a PC or laptop. It wasn't designed with mobile phones in mind, so some layout issues on smaller screens are expected.

3 Overall Quality of the Software

1. How good is the user manual? Was it easy to install and run the software?

The user manual provided reasonably clear instructions for installing and using the software. While most of the steps were easy to follow and helped guide the installation process, a few sections could have been more detailed or clarified for first-time users. Despite this, we were able to get the software up and running without major issues. Overall, the manual was helpful and made the installation process manageable, though there's some room for improvement in terms of clarity and thoroughness.

2. How is the quality of the code? Is it well documented? Is the code modular? Are the variables and the function names meaningful?

The code quality in this project was generally good. It was fairly well-organized, and most variables and functions had meaningful names that made the code easier to follow. The comments included were helpful, though some sections could have benefited from more detailed explanations. The modular structure of the code made it manageable and easier to update, which is a definite strength. While the documentation was present and covered the basics of each section, there's still room for improvement in terms of clarity and depth. Overall, the codebase was functional and understandable, supporting ease of use and future development.

3. Do you think that the software lacks some major features that may make it less acceptable to the user?

While the Aviators lecture hall booking portal claims to offer essential functionalities required for managing bookings effectively, it falls short in several important aspects. Although it allows basic room reservation tasks, the software lacks certain critical features that would enhance its usability and overall acceptance among users. Several functionalities mentioned in the initial requirements appear either partially implemented or missing, limiting the software's practical utility.

- Recurring Bookings: A major missing feature is the ability to create recurring bookings. Teachers often require lecture halls for regular, repeating classes or meetings, and the absence of this functionality forces manual entry for each session, significantly reducing efficiency.
- Room Capacity Details: The software does not currently provide clear details regarding maximum room capacities or seating arrangements. This missing

information complicates the booking process, especially when users need to match class sizes with appropriate rooms.

- Cancellation and Modification Flexibility: Another critical functionality missing from Aviators is the ease of modifying or canceling existing bookings. The current interface requires cumbersome processes to make even minor adjustments, which can be highly inconvenient for users.
- Notification System: A crucial yet overlooked feature is the absence of automated notifications or alerts confirming bookings, changes, or cancellations. Users currently have no immediate confirmation or reminder system integrated, resulting in confusion and potential scheduling conflicts.
- Incomplete Lecture Hall Listings: The software currently includes only a limited number of lecture halls, missing several key halls, which significantly restricts scheduling options and availability for users.

4. Does the software not satisfy any major non-functional requirements?

While Aviators provides a structured framework for automated lecture hall booking, it currently faces several usability and performance challenges**.** At times, pressing the *Login* button yields no response, preventing access to the system altogether. Additionally, the platform experiences noticeable slowdowns and occasional freezing during peak usage, especially when multiple users are accessing it concurrently. Core operations—such as loading the dashboard, checking hall availability, or switching between modules—can become unresponsive or delayed. These issues impact the overall reliability and user experience. Addressing these concerns through robust optimization and error handling will be essential for Aviators to function effectively in a real-world academic environment.

Appendix A - Group Log

S.No	Date	Timings	Venue	Description
1	05/04/2025	20:00 - 21:00	RM Building	Initial team meeting to discuss the testing responsibilities and assignments.
2	08/04/2025	21:00 - 22:00	Google Meet	Online meeting to go through the shared SRS, user manual, and test documentation.
3	11/04/2025	16:00 - 17:00	Hall 12	Collaborative session to finalize bugs and publish test results on GitHub.
4	18/04/2025	21:00 - 22:00	Hall 12	Team met to distribute work for preparing the beta testing report.
5	20/04/2025	20:00 - 21:00	Google Meet	Final meeting to compile and complete the formal testing report as a team.