

## PROJECT ABSTRACT

# Lab Equipment Booking System

*A reservation platform for shared lab instruments: live availability, slot booking with approval workflows, usage rules and no-show penalties. Delivered built-to-order with the full application, report, PPT and viva kit.*

## 01. Title

## 02. Introduction / Background

Oscilloscopes, 3D printers and GPU workstations are shared across hundreds of students, yet most labs still run on a paper register at the lab door — first-come chaos, no-shows wasting prime slots, and no record of who damaged what. This project builds a lab equipment booking system: live availability per instrument, slot reservations with lab-specific rules (approvals, training prerequisites, max hours), check-in discipline and no-show penalties — so scarce instruments serve the most students fairly.

## 03. Problem Statement

Paper registers cannot show availability before you walk to the lab, cannot prevent double-booking, and cannot enforce the rules that protect expensive instruments (training requirements, supervision, usage limits). The project addresses this with a reservation platform: each instrument has a bookable calendar governed by its policy, high-risk equipment needs approval or verified training, and repeated no-shows suspend booking rights — creating accountability the register never had.

## 04. Objectives

- Maintain a searchable inventory of instruments with specs, location and live status.
- Provide per-instrument slot booking governed by configurable policies.
- Enforce prerequisites: training completion or faculty approval for restricted equipment.
- Track check-ins, no-shows and damage reports with penalty rules.
- Give lab in-charges dashboards: utilization, peak hours, repeat offenders.
- Deliver the complete student kit: application, report, PPT and viva Q&A.;

## 05. Proposed Methodology

The system is developed in three stages. (1) Inventory: instruments registered with specifications, location, unit count, status (available/in-use/maintenance) and a booking policy (slot length, max hours, approval needs). (2) Reservation: students browse availability, book slots transactionally (no double-booking), and restricted items route through an approval queue. (3) Governance: check-in confirms usage, no-shows and damage reports feed a penalty engine, and utilization analytics guide the lab's purchase and maintenance planning.

## 06. System Architecture / Working

A role-based web application: students (browse, book, cancel), lab assistants (check-in, damage reports, maintenance blocks) and lab in-charges (policies, approvals, analytics). The backend exposes inventory, availability and booking APIs over a relational database with transactional slot allocation; a scheduler releases unclaimed slots after a grace period and applies no-show penalties.

## 07. Hardware / Software Requirements

- Server: shared hosting or small VPS
- Backend: Python (Django/Flask) or Node.js with PostgreSQL/MySQL
- Frontend: HTML5, CSS3, JavaScript (kiosk-friendly for lab check-in screens)
- Email/SMS for booking confirmations and reminders
- Optional: QR codes on instruments linking to their booking page

## 08. Expected Outcomes

A working system where instruments are discoverable, bookable and governed end-to-end. Expected behavior: concurrent bookings for the same slot resolve to exactly one holder, approval-gated items cannot be self-booked, and utilization dashboards reflect real bookings. Capacity expectations are design targets verified with the included concurrency test.

## 09. Applications

- Engineering college departmental labs
- Central instrumentation facilities
- Maker spaces and fabrication labs
- Research labs with shared equipment

## 10. Future Scope

- IoT interlocks: instrument powers on only for the booked user
- Usage-meter billing for consumables (filament, chemicals)
- Predictive maintenance from usage-hour logs
- Inter-college equipment sharing network