

PROJECT ABSTRACT

Faculty Office-Hour Scheduler

An appointment system for student-faculty meetings: live availability, 15-minute slot booking, agenda sharing and reminders. Delivered built-to-order with the full application, report, PPT and viva kit.

01. Title

02. Introduction / Background

Students needing five minutes with a professor face a familiar ritual: hovering outside a cabin, knocking during a meeting, or sending emails that drown in inbox. Faculty, meanwhile, get interrupted at random hours. This project builds an office-hour scheduler: faculty publish weekly availability, students book 15-minute slots with an agenda attached, both sides get reminders, and cancellation or rescheduling is one tap — replacing corridor queues with a calendar.

03. Problem Statement

Ad-hoc meetings waste both sides' time: students wait outside busy cabins, faculty lose focus to interruptions, and project-guidance discussions happen without preparation because no agenda was shared. The project addresses this with structured scheduling: visible availability with leave and busy blocks, slot booking with a stated purpose, agenda notes the faculty reads beforehand, and no-show tracking that keeps the system fair.

04. Objectives

- Let faculty publish recurring weekly office hours with breaks, leaves and per-slot capacity.
- Let students discover faculty by name, department or subject and book free slots.
- Attach purpose and agenda to every booking so meetings start prepared.
- Send reminders and support one-tap cancellation/rescheduling.
- Track no-shows and ratings to keep both sides accountable.
- Deliver the complete student kit: application, report, PPT and viva Q&A.;

05. Proposed Methodology

The system is developed in three stages. (1) Availability: faculty define weekly templates (e.g. Tue/Thu 4–6 PM, 15-min slots) plus exceptions for leave and meetings; the engine expands these into bookable slots. (2) Booking: students pick a slot, state a purpose and agenda; the booking is confirmed or queued for faculty approval per faculty preference. (3) Meeting lifecycle: reminders fire before the slot, either side can cancel/reschedule within policy, and completion prompts mutual feedback.

06. System Architecture / Working

A role-based web application: faculty (availability, approvals, appointment list), students (search, booking, history) and department view (aggregate load). The backend manages slot generation, transactional booking (no double-booking under concurrency), and notification scheduling over a relational database; the frontend is a calendar-centric responsive UI.

07. Hardware / Software Requirements

- Server: shared hosting or small VPS
- Backend: Python (Django/Flask) or Node.js with PostgreSQL/MySQL
- Frontend: HTML5, CSS3, JavaScript with calendar components
- Email/SMS service for reminders
- Optional: calendar (ICS) export for personal calendars

08. Expected Outcomes

A working scheduler where faculty availability converts into booked, reminded, completed meetings. Expected behavior: double-booking is impossible at the data layer, leave blocks remove slots automatically, and agendas reach the faculty before the meeting. Adoption expectations are design targets; the report includes a pilot plan for one department.

09. Applications

- Student-faculty doubt-clearing and mentoring
- Final-year project guidance meetings
- Placement-cell counselor appointments
- Lab and workshop in-charge consultations

10. Future Scope

- Video-meeting links for remote consultations
- Group slots for team project reviews
- Analytics on consultation load per faculty
- Integration with the college timetable to auto-block class hours