Page 1 — Title, Problem, Overview

TAConnect – Smart Office Hours & Scheduling Platform

Universities still manage a lot of TA office hours with spreadsheets, ad-hoc Zoom links, and long queues in crowded hallways. Students struggle to find available times, TAs waste time resolving conflicts manually, and coordinators have little visibility into how office hours are actually used. **TAConnect** is a self-hostable web application that centralizes office hour scheduling, booking, and analytics so that students can quickly find help, and TAs can run organized, conflict-free office hours with minimal overhead.

The system provides a modern React frontend and a Django REST backend, backed by a relational database and an email/push notification layer. TAs define recurring office hour slots with policies such as capacity limits, buffer times, and allowed-student lists; students then book into those slots through a clean, responsive UI. Under the hood, TAConnect enforces conflict-free bookings, tracks the full lifecycle of each session (pending, confirmed, completed, cancelled), and exposes analytics and CSV exports so TAs and departments can understand demand patterns over time. The entire platform is designed to be deployed on institutional infrastructure using Docker, with no dependency on third-party SaaS services.

Page 2 — System Architecture

High-Level Architecture

TAConnect follows a classic web client–server architecture:

- **Frontend (React 19 + Vite + Tailwind CSS)**
  - Implements the student and TA dashboards, booking flows, CSV upload for allowed students, analytics views, notification preferences, and theme toggle.
  - Talks to the backend exclusively via JSON over HTTPS, using an API client layer and JWT for authentication.

- **Backend (Django 5 + Django REST Framework)**
  - Exposes REST APIs for authentication, course/section and slot management, bookings, analytics, and notification preferences.
  - Organized into modular apps: accounts (users, roles, email verification), instructor (office hours, booking policies, allowed students, analytics), student (booking flows, student utilities), core (base models), and utils (email sending, push notifications, datetime helpers).
  - Automatically serves Swagger/OpenAPI documentation for all endpoints.
- **Database (PostgreSQL in production, SQLite in development/test)**
  - Stores all persistent data: users and roles, office hour slots, bookings, allowed-student lists, notification preferences, and audit timestamps.
  - Uses indexes to keep booking/slot queries fast, and encrypted fields (e.g. student IDs in AllowedStudents) for sensitive attributes.
- **Notification Layer (Email + Web Push)**
  - Email templates under backend/ta_connect/templates handle registration, verification, password reset, booking confirmation/cancellation, and bulk cancellations.
  - Web push notifications are implemented via service workers in the frontend and a push subsystem in backend/ta_connect/utils/push_notifications, using VAPID keys configured via environment variables.
- **Deployment & Self-Hosting (Docker + Docker Compose)**
  - A docker-compose.yml file plus backend and frontend Dockerfiles allow the full stack (frontend, backend, database) to be brought up with a single command.
  - All environment-specific configuration (DB credentials, email backend, JWT settings, encryption key, VAPID keys) is provided via .env files as documented in docs/SETUP.md and docs/ENVIRONMENT_VARIABLES.md.

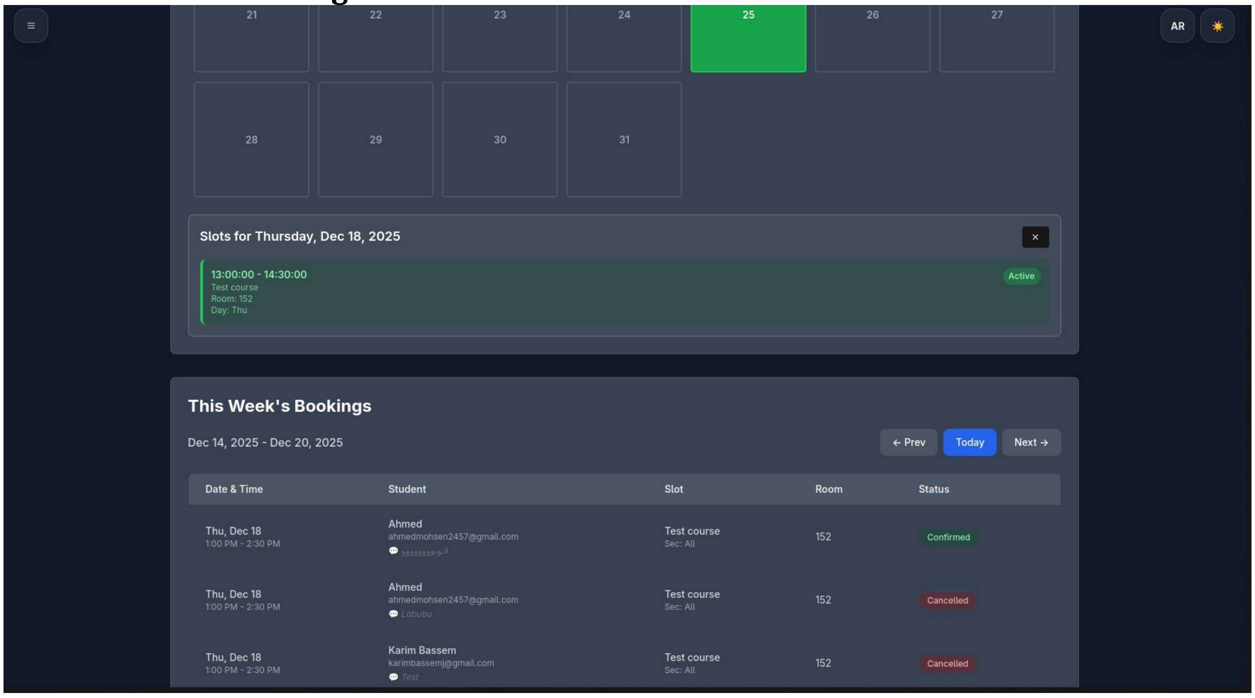
System Architecture Diagram

- You can base it on your existing diagrams in docs/diagrams/ (e.g., package or class diagram) but simplify to highlight:
  - Client (Student, TA) → React SPA
  - React SPA → Django REST API (JWT)
  - Django REST API → Database (Users, OfficeHourSlot, Booking, Policies, AllowedStudents)
  - Django REST API → SMTP email server and Web Push service.

Page 3 — Screenshots of Working Features

For this page, add real screenshots from your running app with short captions.

- **TA dashboard: creating office hours**



Slots for Thursday, Dec 18, 2025

13:00:00 - 14:30:00
Test course
Room: 152
Day: Thu
Active

This Week's Bookings

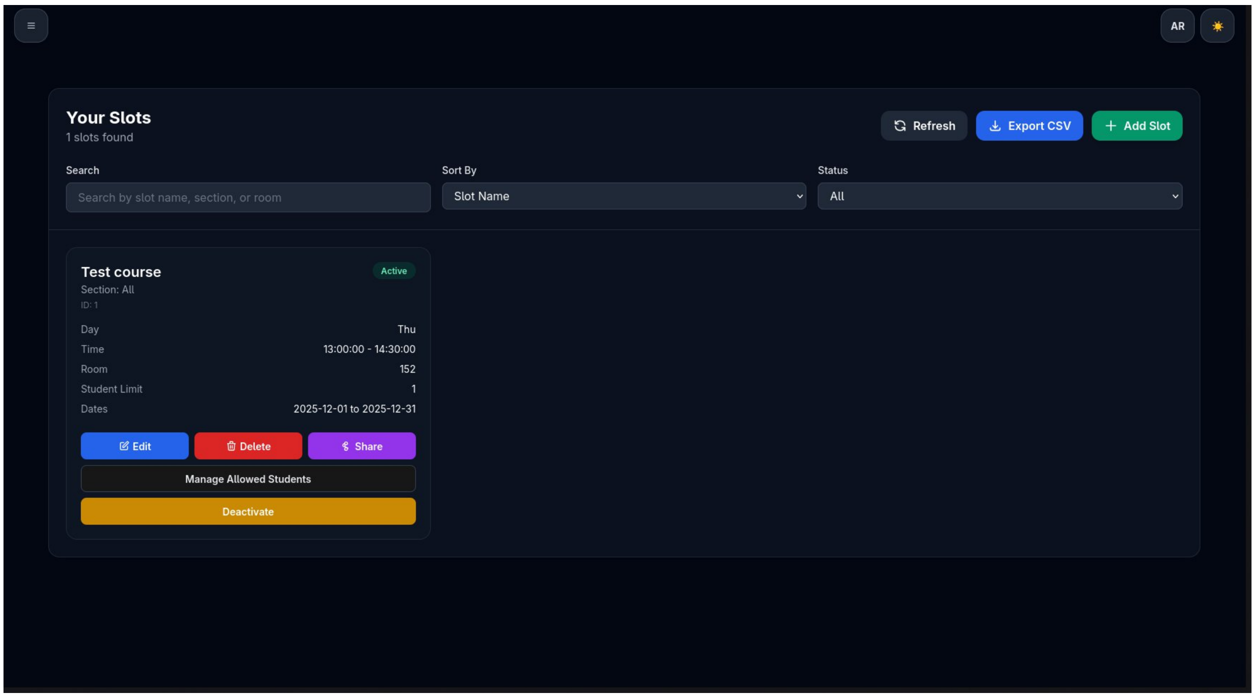
Dec 14, 2025 - Dec 20, 2025

← Prev | Today | Next →

Date & Time	Student	Slot	Room	Status
Thu, Dec 18 1:00 PM - 2:30 PM	Ahmed ahmedmohsen2457@gmail.com 💬 gggggggg[illegible]	Test course Sec: All	152	Confirmed
Thu, Dec 18 1:00 PM - 2:30 PM	Ahmed ahmedmohsen2457@gmail.com 💬 Labubu	Test course Sec: All	152	Cancelled
Thu, Dec 18 1:00 PM - 2:30 PM	Karim Bassem karimbassemj@gmail.com 💬 Test	Test course Sec: All	152	Cancelled

- In this view, a TA configures an OfficeHourSlot by specifying the course, section, day of week, time window, duration per booking, room/meeting link, and active date range. The UI supports recurring weekly slots with buffer times and validates that slots are well-formed before saving them to the backend.

- **Office hours list and schedule view**



Your Slots
1 slots found

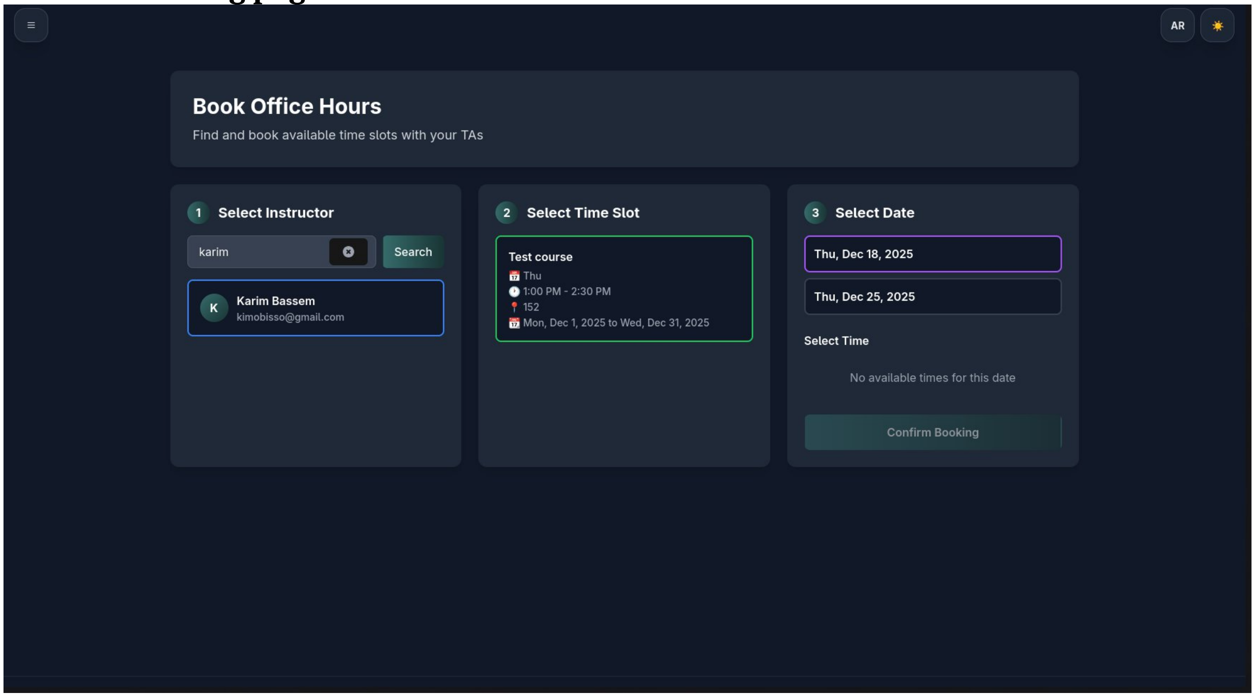
Search: Search by slot name, section, or room
Sort By: Slot Name
Status: All

Test course Active
Section: All
ID: 1

Day	Thu
Time	13:00:00 - 14:30:00
Room	152
Student Limit	1
Dates	2025-12-01 to 2025-12-31

This page lists all the TA's configured office hours, grouped by course/section. It pulls data from the OfficeHourSlot model and uses status flags to highlight which slots are currently active. From here, TAs can quickly enable/disable slots or drill into details for a specific course.

- **Student booking page**


Book Office Hours

Find and book available time slots with your TAs

1 Select Instructor

karim

Search

K Karim Bassem kimobisso@gmail.com

2 Select Time Slot

Test course

📅 Thu

🕐 1:00 PM - 2:30 PM

📍 152

📅 Mon, Dec 1, 2025 to Wed, Dec 31, 2025

3 Select Date

Thu, Dec 18, 2025

Thu, Dec 25, 2025

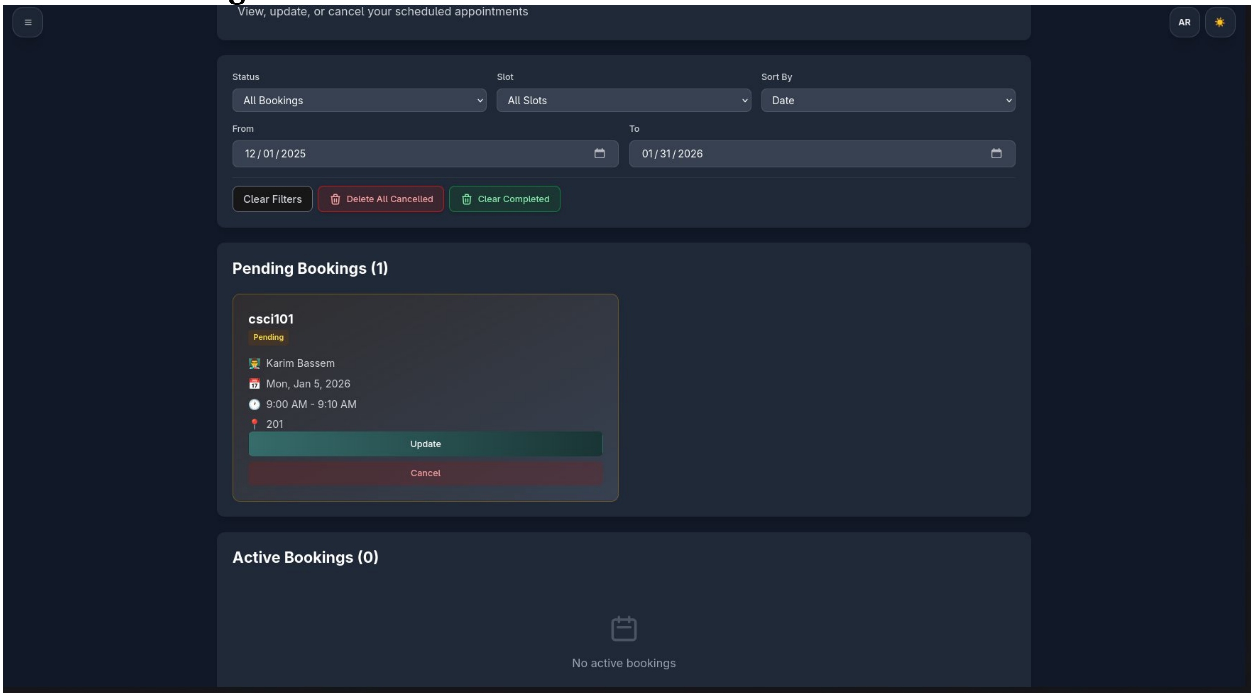
Select Time

No available times for this date

Confirm Booking

Students see only available slots that they are allowed to book, taking into account capacity limits, allowed-student restrictions, and existing bookings. The booking form typically includes the time, location/link, and an optional description so the student can indicate what they need help with.

- **Student booking confirmation**


View, update, or cancel your scheduled appointments

Status: All Bookings

Slot: All Slots

Sort By: Date

From: 12/01/2025

To: 01/31/2026

Clear Filters

Delete All Cancelled

Clear Completed

Pending Bookings (1)

csci101

Pending

Karim Bassem

📅 Mon, Jan 5, 2026

🕐 9:00 AM - 9:10 AM

📍 201

Update

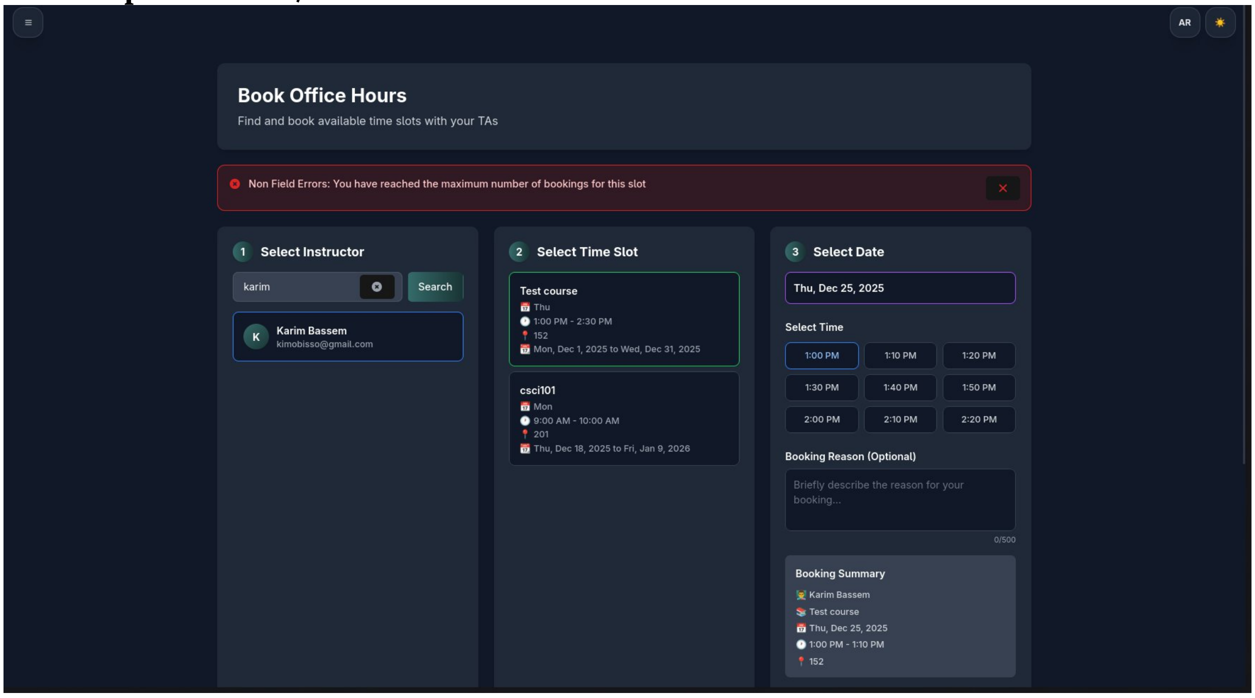
Cancel

Active Bookings (0)

No active bookings

After submitting, the student lands on a confirmation view that summarizes the booking details and status (initially “pending” or “confirmed” depending on the flow). Behind the scenes, the backend creates a Booking record linked to the selected OfficeHourSlot, and triggers email/push notifications using the appropriate templates.

- **Conflict prevention / validation**


≡ AR

Book Office Hours

Find and book available time slots with your TAs

Non Field Errors: You have reached the maximum number of bookings for this slot

1 Select Instructor

karim Search

K Karim Bassem
kimobisso@gmail.com

2 Select Time Slot

Test course
📅 Thu
🕐 1:00 PM - 2:30 PM
📍 152
📅 Mon, Dec 1, 2025 to Wed, Dec 31, 2025

csci101
📅 Mon
🕐 9:00 AM - 10:00 AM
📍 201
📅 Thu, Dec 18, 2025 to Fri, Jan 9, 2026

3 Select Date

Thu, Dec 25, 2025

Select Time

1:00 PM	1:10 PM	1:20 PM
1:30 PM	1:40 PM	1:50 PM
2:00 PM	2:10 PM	2:20 PM

Booking Reason (Optional)

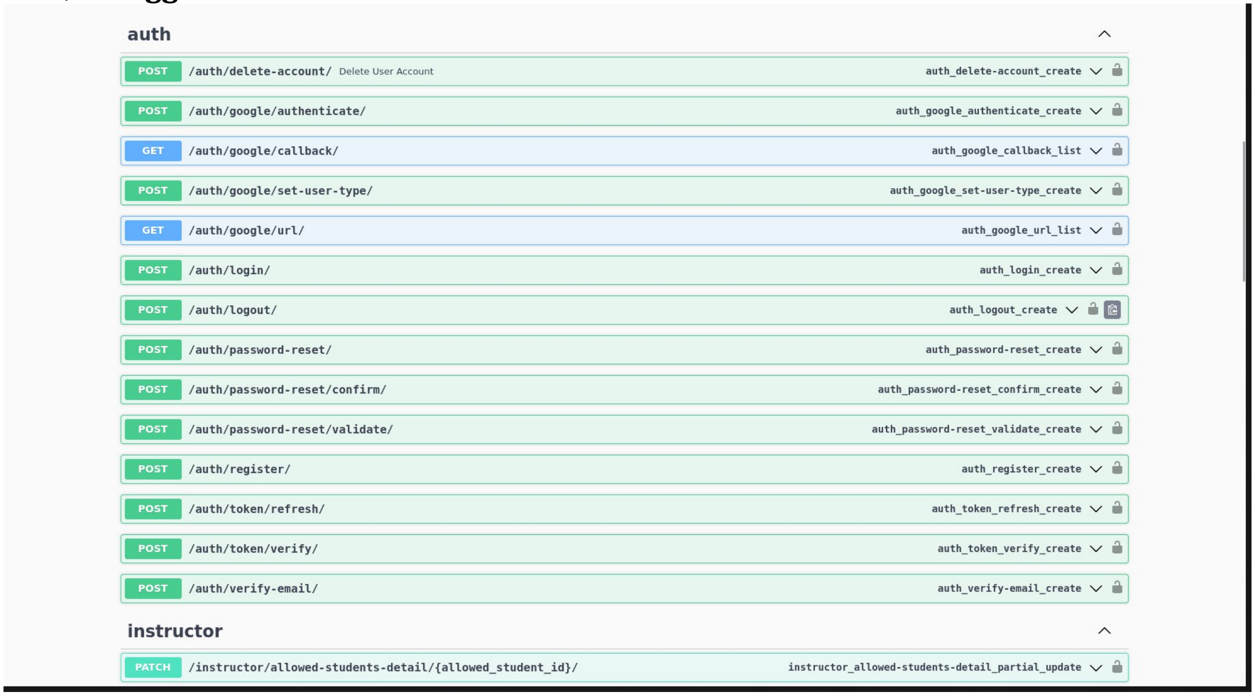
Briefly describe the reason for your booking...

0/500

Booking Summary
Karim Bassem
Test course
📅 Thu, Dec 25, 2025
🕐 1:00 PM - 1:10 PM
📍 152

TAConnect prevents conflicting bookings by checking existing Booking records for overlaps using the slot duration and configured policies (e.g., set_student_limit, buffer times, and per-section rules). The UI surfaces these rules as clear error messages, so students understand why a particular time is not available.

- **API / Swagger documentation**


auth

```
POST   /auth/delete-account/  Delete User Account        auth_delete-account_create
POST   /auth/google/authenticate/                        auth_google_authenticate_create
GET    /auth/google/callback/                            auth_google_callback_list
POST   /auth/google/set-user-type/                       auth_google_set-user-type_create
GET    /auth/google/url/                                 auth_google_url_list
POST   /auth/login/                                      auth_login_create
POST   /auth/logout/                                     auth_logout_create
POST   /auth/password-reset/                             auth_password-reset_create
POST   /auth/password-reset/confirm/                     auth_password-reset_confirm_create
POST   /auth/password-reset/validate/                    auth_password-reset_validate_create
POST   /auth/register/                                   auth_register_create
POST   /auth/token/refresh/                              auth_token_refresh_create
POST   /auth/token/verify/                               auth_token_verify_create
POST   /auth/verify-email/                               auth_verify-email_create
```

instructor

```
PATCH  /instructor/allowed-students-detail/{allowed_student_id}/   instructor_allowed-students-detail_partial_update
```

The backend publishes an OpenAPI specification and an interactive Swagger UI, which lists endpoints for authentication, office hour management, bookings, analytics, and more. Developers and integrators can use this page to test endpoints, inspect request/response schemas, and obtain JWT tokens for manual API calls.

.

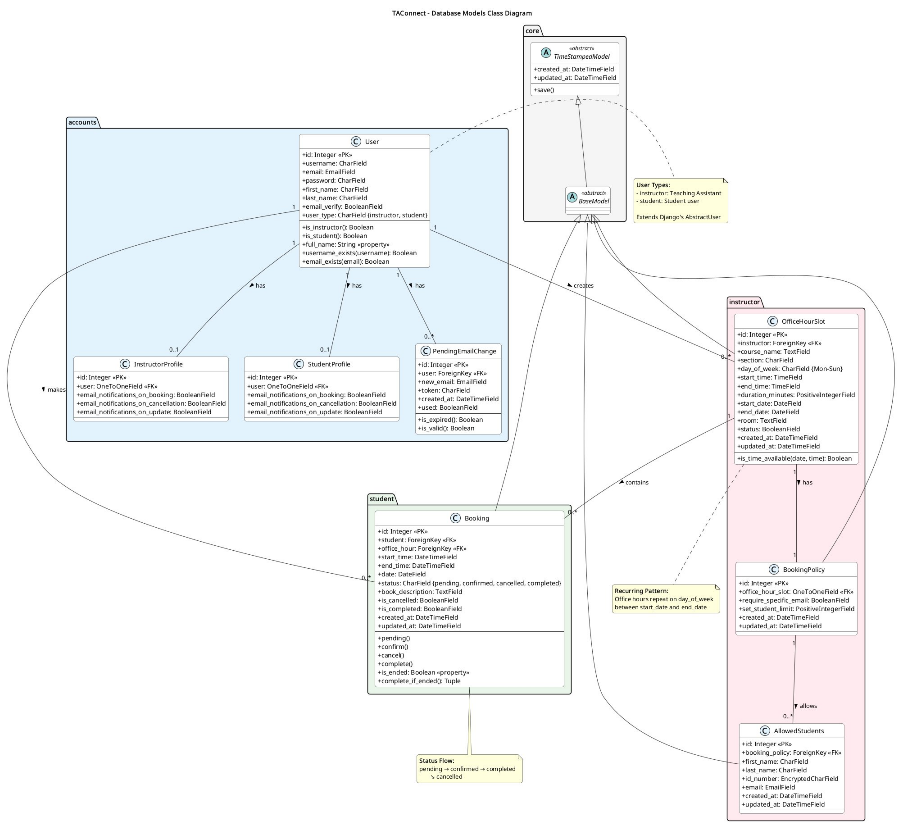
Page 4 — Database / UML / Flow Diagram

Core Data Model

At the center of TAConnect are a handful of core models:

- **User / Profiles**
  - User extends Django's AbstractUser and adds fields for user_type (instructor vs. student) and email_verify for verification status.
  - Each user may have an InstructorProfile or StudentProfile that stores email-notification preferences for different booking events.
- **OfficeHourSlot (Instructor side)**
  - Represents a recurring office hour definition: course name, section, day of week, start/end time, duration per booking, active date range, room/meeting link, and status.
  - Has a one-to-one BookingPolicy, which controls per-slot rules such as whether only specific emails can book and the maximum number of concurrent bookings per student.
  - AllowedStudents records link to BookingPolicy and hold an encrypted student ID, name, and email, forming the "whitelist" for restricted sections.
- **Booking (Student side)**
  - Stores the actual reservations students make against an OfficeHourSlot, including student reference, start/end time, date, description, and status (pending, confirmed, cancelled, completed).
  - Derived helpers ensure that the boolean flags (is_cancelled, is_completed) and the status field stay consistent, and that end times are automatically computed from the associated slot's duration when needed.

Model Relations Diagram


TAConnect - Database Models Class Diagram
core
«abstract»
TimeStampedModel
+created_at: DateTimeField
+updated_at: DateTimeField
+save()
«abstract»
BaseModel
User Types:
- instructor: Teaching Assistant
- student: Student user
Extends Django's AbstractUser
accounts
User
+id: Integer «PK»
+username: CharField
+email: EmailField
+password: CharField
+first_name: CharField
+last_name: CharField
+email_verify: BooleanField
+user_type: CharField {instructor, student}
+is_instructor(): Boolean
+is_student(): Boolean
+full_name: String «property»
+username_exists(username): Boolean
+email_exists(email): Boolean
has
has
has
makes
creates
InstructorProfile
+id: Integer «PK»
+user: OneToOneField «FK»
+email_notifications_on_booking: BooleanField
+email_notifications_on_cancellation: BooleanField
+email_notifications_on_update: BooleanField
StudentProfile
+id: Integer «PK»
+user: OneToOneField «FK»
+email_notifications_on_booking: BooleanField
+email_notifications_on_cancellation: BooleanField
+email_notifications_on_update: BooleanField
PendingEmailChange
+id: Integer «PK»
+user: ForeignKey «FK»
+new_email: EmailField
+token: CharField
+created_at: DateTimeField
+used: BooleanField
+is_expired(): Boolean
+is_valid(): Boolean
instructor
OfficeHourSlot
+id: Integer «PK»
+instructor: ForeignKey «FK»
+course_name: TextField
+section: CharField
+day_of_week: CharField {Mon-Sun}
+start_time: TimeField
+end_time: TimeField
+duration_minutes: PositiveIntegerField
+start_date: DateField
+end_date: DateField
+room: TextField
+status: BooleanField
+created_at: DateTimeField
+updated_at: DateTimeField
+is_time_available(date, time): Boolean
contains
has
BookingPolicy
+id: Integer «PK»
+office_hour_slot: OneToOneField «FK»
+require_specific_email: BooleanField
+set_student_limit: PositiveIntegerField
+created_at: DateTimeField
+updated_at: DateTimeField
allows
AllowedStudents
+id: Integer «PK»
+booking_policy: ForeignKey «FK»
+first_name: CharField
+last_name: CharField
+id_number: EncryptedCharField
+email: EmailField
+created_at: DateTimeField
+updated_at: DateTimeField
Recurring Pattern:
Office hours repeat on day_of_week between start_date and end_date
student
Booking
+id: Integer «PK»
+student: ForeignKey «FK»
+office_hour: ForeignKey «FK»
+start_time: DateTimeField
+end_time: DateTimeField
+date: DateField
+status: CharField {pending, confirmed, cancelled, completed}
+book_description: TextField
+is_cancelled: BooleanField
+is_completed: BooleanField
+created_at: DateTimeField
+updated_at: DateTimeField
+pending()
+confirm()
+cancel()
+complete()
+is_ended: Boolean «property»
+complete_if_ended(): Tuple
Status Flow:
pending → confirmed → completed
↘ cancelled

– You can adapt docs/diagrams/TAConnect - Class Diagram (Models Only).png or docs/diagrams/classes_TAConnect.png and highlight just the main entities:
  - User → InstructorProfile / StudentProfile
  - User → Booking
  - OfficeHourSlot → BookingPolicy → AllowedStudents
  - OfficeHourSlot → Booking

Brief Flow Description

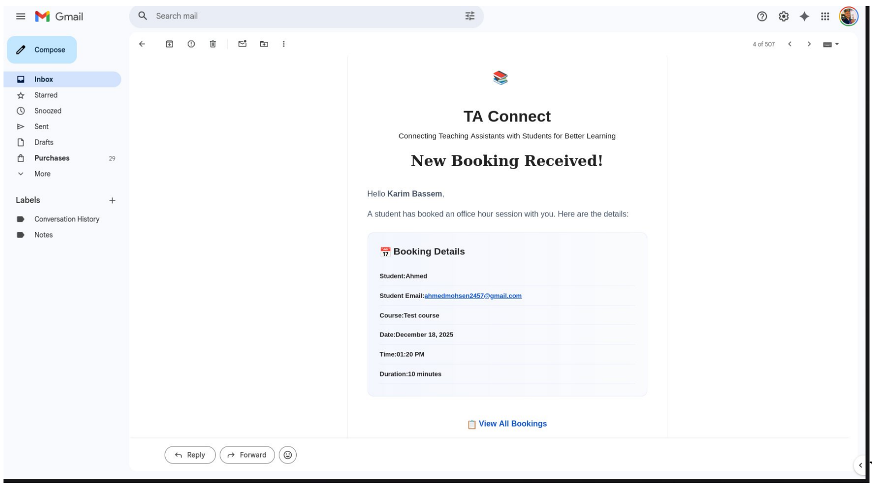
- An instructor creates one or more OfficeHourSlot records and, optionally, a BookingPolicy with an AllowedStudents list (often imported via CSV).
- Students browse available OfficeHourSlots that match their section and policy and submit bookings.
- Each booking's lifecycle is tracked through status transitions (pending → confirmed → completed / cancelled), and the system updates derived flags, enforces time-based rules, and feeds analytics dashboards with aggregated booking data.

.

Page 5 — Additional Results (Optional)

Use this page to showcase extra evidence of system completeness.

- **Email notification example**



TA Connect

Connecting Teaching Assistants with Students for Better Learning

New Booking Received!

Hello **Karim Bassem**,

A student has booked an office hour session with you. Here are the details:

📅 **Booking Details**

Student:Ahmed

Student Email:ahmedmohsen2457@gmail.com

Course:Test course

Date:December 18, 2025

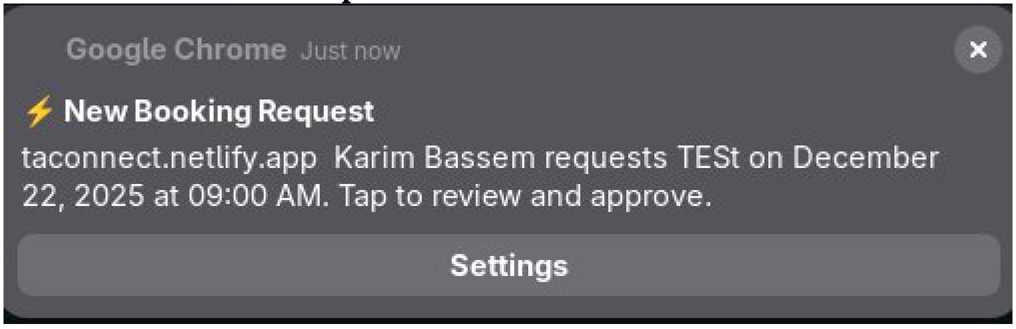
Time:01:20 PM

Duration:10 minutes

📋 **View All Bookings**

When a booking is created, updated, or cancelled, the system renders a corresponding HTML email using templates from the backend (booking_confirmation_email_Student.html, booking_cancellation_email_TA.html, etc.) and sends it through the configured SMTP backend.

- **Push notification example**



⚡ **New Booking Request**

taconnect.netlify.app Karim Bassem requests TESt on December 22, 2025 at 09:00 AM. Tap to review and approve.

- For users who opt in, TAConnect sends real-time browser notifications on key booking events. The frontend service worker (public/service-worker.js / sw.js) and the backend push subsystem work together to deliver these alerts.