Bilkent University

Department of Computer Engineering

TA Management System Project

Final Report

Deliverable 5

Spring 2025

Team 12

Ahmet Kenan Ataman <22203434>
Berfin Örtülü <21802704>
Erdem Uğurlu <22203391>
Gülferiz Bayar <21901442>
Mehmet Emre Şahin <22201765>

Instructor: *Eray TÜZÜN*
Teaching Assistant(s): *Utku Boran TORUN, Yahya ELNOUBY*

Table of Contents

1 INTRODUCTION

TA Management System is a web-based platform developed for the Faculty of Engineering at Bilkent University, specifically targeting the CS and IE departments in its first release. The system is designed to monitor the workloads of teaching assistants (TAs), manage course-related task assignments, and ensure the fair and efficient distribution of exam proctoring duties.

TAs are responsible for a wide range of academic activities including grading, conducting recitations, supervising laboratory sessions, holding office hours, and proctoring examinations. However, the level of TA involvement varies across courses, often resulting in imbalanced workloads. These discrepancies can lead to inefficiencies and dissatisfaction among TAs. The TA Management System addresses this issue by centralizing the tracking of all TA responsibilities and distributing proctoring assignments based on each TA's current workload and availability.

The system supports multiple user roles, including TAs, course instructors, authorized departmental staff, dean office, and admin. Each user role in the system is equipped with tailored functionalities that align with their responsibilities. TAs manage their availability, workload entries, leave and swap requests, and follow their proctoring duties. Instructors oversee task assignments, exam definitions, and the approval of TA activities. Authorized staff supervise user registrations and coordinate departmental proctoring operations and exam definitions. Faculty administrators are informed in case of TA shortages and can intervene to maintain adequate exam staffing across departments.

The system also provides supplementary features such as generating classroom distribution lists for exams, email verification during user registration, and email notifications that facilitate inter-user communication and information flow. Furthermore, it supports bulk data import via Excel, allowing the system to ingest TA identity data, academic backgrounds, technical competencies, and course-based workload planning. This capability enables seamless integration of assignment and reporting processes into the database.

From a technical standpoint, the system is built on a Django-based backend and a React-based frontend, using SQLite as the database management system.

1.1 State of the System

- Although the backend infrastructure and UI logic for the swap mechanism have been implemented, the feature is **not fully functional** in the current version due to unresolved integration issues. As a result, TAs are currently limited to rejecting assignments instead of initiating swap requests through the interface.

- The React-based frontend and Django backend have been successfully connected, with well-structured API communication and smooth data flow between components. User actions on the interface are effectively synchronized with backend logic and database operations.

- The system handles role-based access control and workflow separation across multiple user types; TAs, instructors, authorized staff, and faculty administrators; allowing seamless multi-role interaction throughout the platform.

- Excel-based bulk data import has been implemented to support the integration of TA information, technical backgrounds, academic history, and course-level workload requirements. While this feature works reliably, minor performance delays were observed during the upload of very large Excel files.

- User interface components including task entry, exam definition forms, and workload overviews have been tested and refined to ensure clarity, responsiveness, and user-friendliness.

- Database design with SQLite supports relational data structures for task history, exam records, availability logs, and user role mapping. The system runs in a cross-platform compatible environment and has shown stability under moderate testing conditions.

2 LESSONS LEARNED

2.1 Team Management and Collaboration

During the development of the TA Management System project, we gained valuable experience in both technical implementation and collaborative teamwork. This process not

only enhanced our software development skills but also improved our abilities in project planning and client communication.

One of the most important lessons we learned was the significance of effective team management. Through regular meetings and structured task allocation, we were able to track each member's progress and maintain transparency throughout the project. By evaluating the priority, estimated time, and efficiency of each task, we were able to divide work strategically and distribute responsibilities more effectively.

2.2 User-Centered Design and Prioritization

At the early stages of the project, we conducted one-on-one interviews with various instructors and stakeholders to understand their expectations. This helped us focus on user-centered design and implement features that aligned with real needs. We also realized that not all desired features could be included due to time constraints, which led us to prioritize the most essential functionalities for the end users.

2.3 Importance of Modeling and Design

Another key takeaway was the importance of modeling and design before diving into implementation. After preparing our initial use case diagram and receiving feedback, we clearly recognized how valuable diagrams are in conceptualizing the overall system flow and logic before writing any code. Designing early helped us better anticipate potential design issues and edge cases. Similarly, building a mockup interface early on allowed us to visualize the project layout, identify corner cases more easily, and understand how different components would interact. As a result, when we moved on to designing class, activity, and sequence diagrams, we had a much clearer understanding of system behavior. This structured approach significantly reduced complexity during coding and allowed us to build more confidently and efficiently.

2.4 Backend Architecture and Design Patterns

From a technical perspective, we found that allocating time to design the backend architecture at the beginning significantly benefited us in the long term. Carefully planning the relationships between entities and their interactions with the database made future modifications much easier and less error-prone. We also recognized the limitations of rigid, non-extensible code structures and adopted design patterns to ensure flexibility and maintainability. Specifically, we used the **Facade pattern** to simplify complex backend

interactions and encapsulate subsystems, and the **Strategy pattern** to enable interchangeable algorithms for assigning tasks and proctoring based on dynamic conditions.

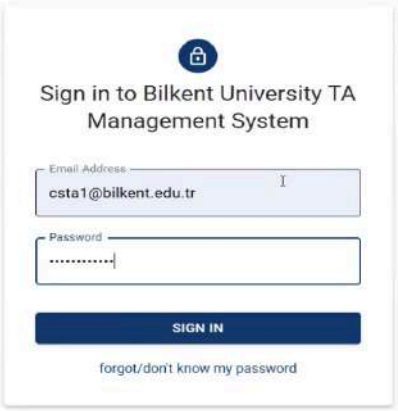
2.5 Framework Focus and Practicality

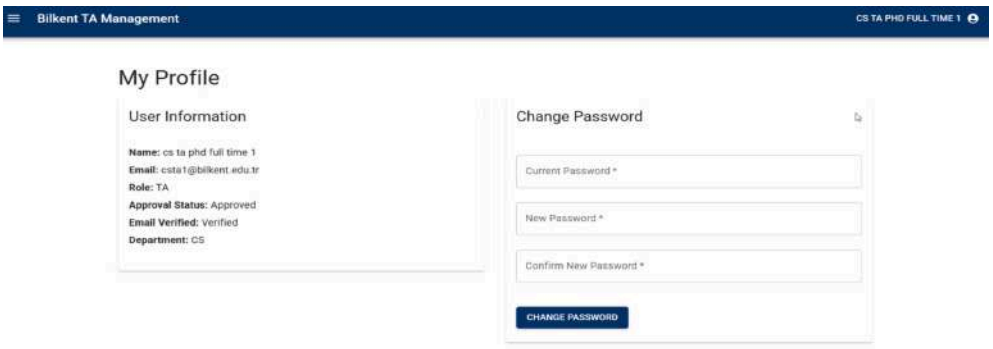
Another key insight was understanding the importance of focusing on the parts of a technology that are most relevant to the project. Rather than attempting to master every feature of a framework, we concentrated on essential components—such as routing and form handling in React, and models, views, serializers, and database logic in Django—which helped us use our time more efficiently.

3 USER'S GUIDE

3.1 TA

3.1.1 Login & Account Initialization



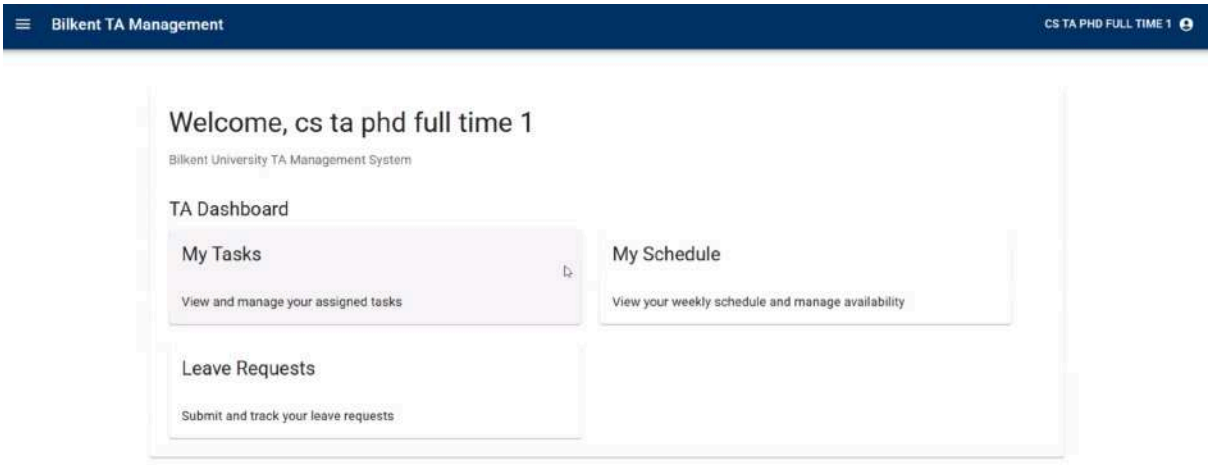


When a Teaching Assistant (TA) is first granted access to the Bilkent TA Management System, they are instructed to log in using their university-issued email address. Upon entering their email on the login screen (see Image 1), if the email already exists in the system, they can proceed to sign in with their assigned password. If they do not yet know their password, they can click the **“I Don’t Know My Password”** button. This triggers a system-generated email that includes a temporary password (Image 2). This password is valid for a limited time and allows the TA to log in for the first time securely.

Once logged in, TAs are directed to their **My Profile** page (Image 3), where they can review and verify their personal information such as name, academic level, employment type (e.g., part-time), join date, and email verification status. At this stage, they are strongly advised to update their temporary password by clicking the **“Change Password”** button provided at the bottom of the profile page. This step is mandatory to ensure secure access going forward.

Additionally, on first login, the system detects whether a TA has defined their weekly schedule. If not, a prompt appears urging them to input their availability—this is essential because no proctoring assignments can be scheduled unless the TA has clearly marked their class hours as unavailable.

3.1.2 TA Dashboard Overview



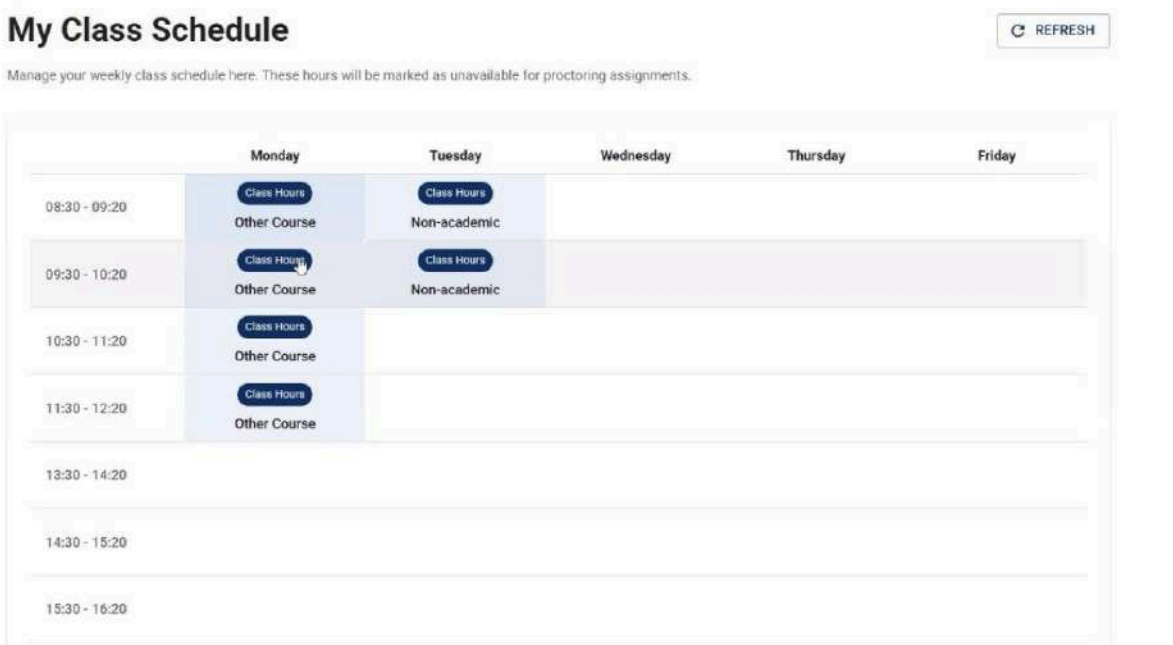
Once a Teaching Assistant successfully logs into the system using their institutional credentials or the temporary password sent to their email, they are automatically redirected to the TA Dashboard — the central landing page of the Bilkent TA Management System.

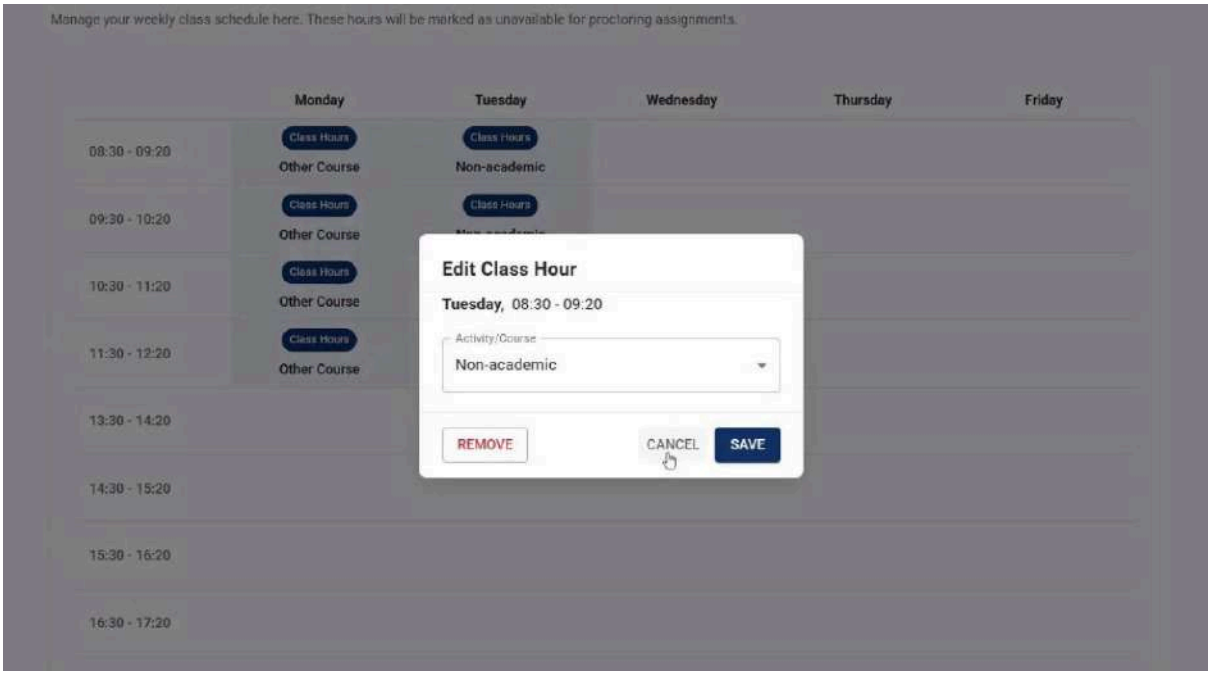
This dashboard serves as the main control panel where TAs can quickly access all major functionalities of the platform. It provides an intuitive and structured interface, allowing users to efficiently manage their academic responsibilities throughout the semester.

The TA Dashboard presents four main sections. The “My Tasks” area allows assistants to view and manage their assigned academic duties, including grading, recitations, labs, and office hours. The “My Schedule” section is where TAs enter their weekly course hours or other unavailable time slots. These declared hours prevent overlap with proctoring assignments or task scheduling.

Another key component is the “Proctoring Assignments” module, where TAs can review their upcoming or past exam invigilation duties and request swaps when necessary. Finally, the “Leave Requests” section enables TAs to submit requests for specific dates by selecting a leave type, providing a reason, and optionally uploading supporting documents. Once submitted, these requests are routed to the course instructor for approval.

3.1.3 My Schedule





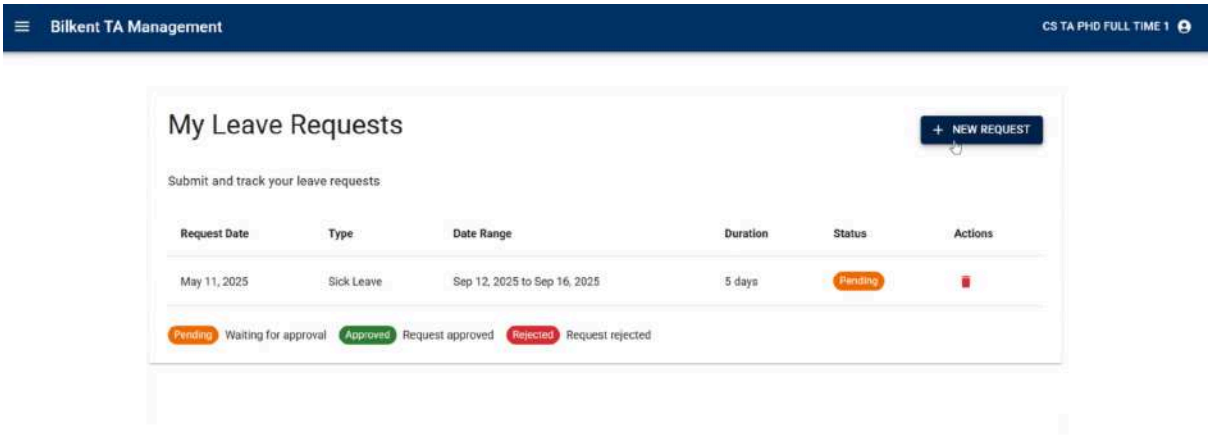
After a Teaching Assistant logs into the system for the first time, they are immediately prompted with a message reminding them to define their weekly schedule before any proctoring assignments can be created. This requirement is essential to prevent scheduling conflicts and to respect each TA's academic and personal commitments.

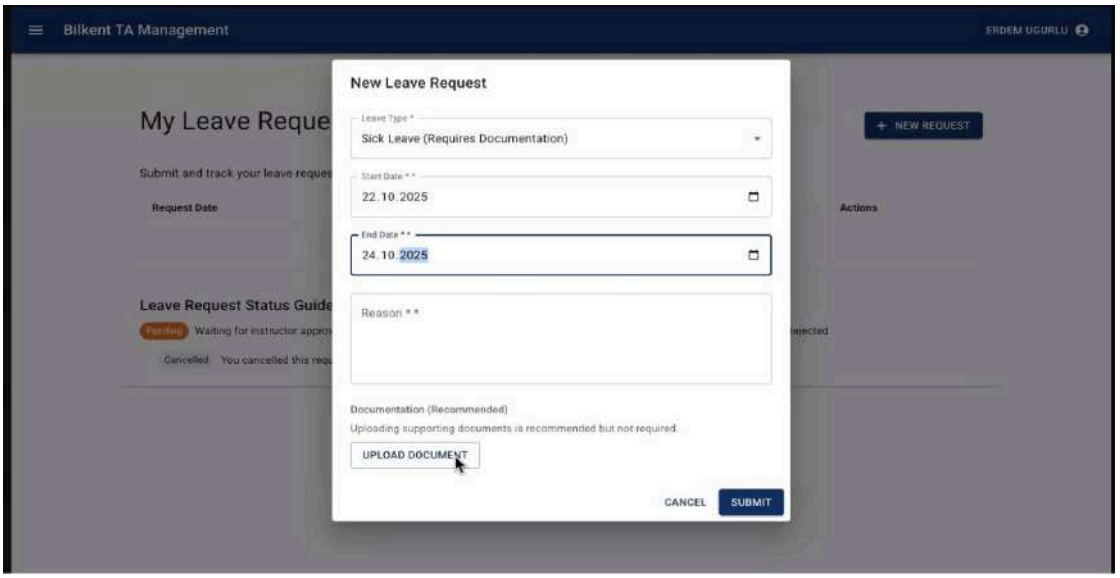
The **My Schedule** page provides a clean and interactive interface for managing weekly availability. The schedule is displayed as a grid organized by day and hour blocks. TAs can click on any blank time slot to mark it as unavailable. Upon selection, a modal window opens, allowing the TA to assign a category label to that block — such as *Class Hours*, *Non-academic*, or *Other Course*. This helps clearly indicate the reason for unavailability.

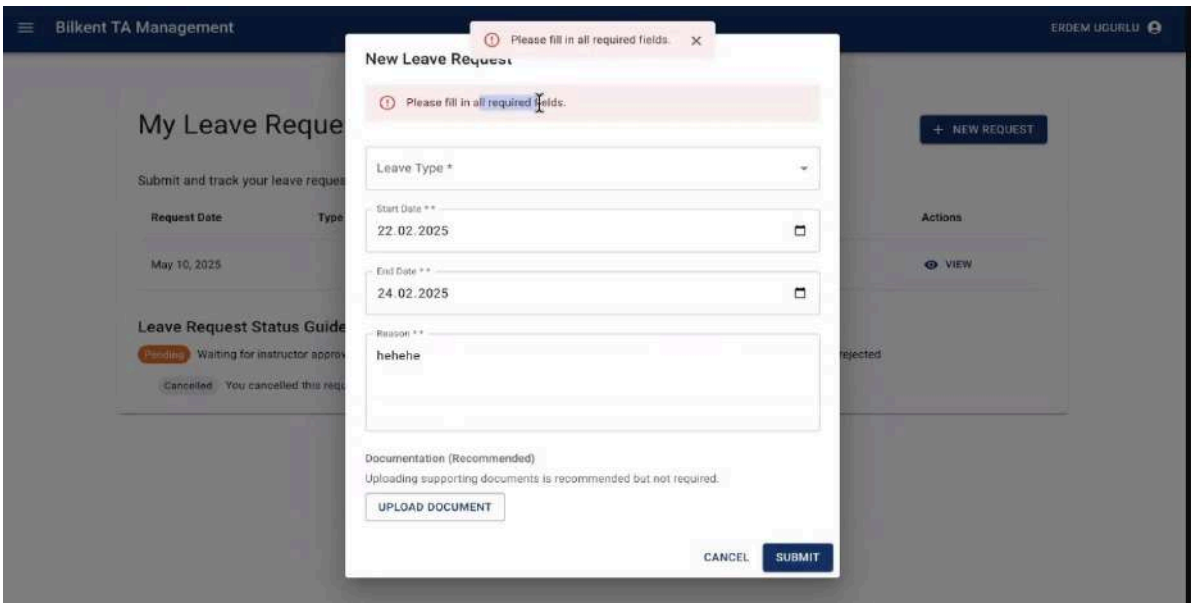
Once saved, these entries are visually reflected in the schedule with the corresponding label, and the system marks these time periods as unavailable for proctoring or other duties. If any adjustments are necessary, the TA can simply click an already added block to either **edit the label** or **remove** it entirely. These updates are processed instantly and displayed within the same interface.

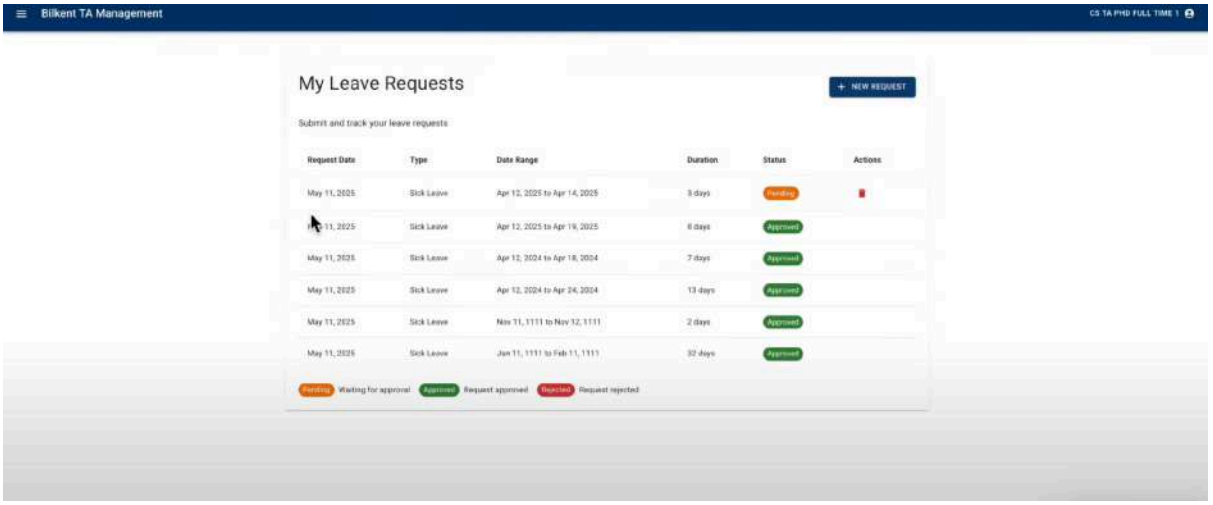
To further ensure accuracy and user confirmation, a double-check **confirmation pop-up** appears after each save operation. This prompt asks the TA to confirm their changes before they are finalized, minimizing accidental updates and enhancing reliability.

3.1.4 Leave Request









The Leave Requests page allows Teaching Assistants to formally submit absence requests and track their status throughout the approval process. This module is designed to streamline communication between TAs and course instructors while ensuring that leave periods are respected during task assignment and proctoring schedules.

To initiate a leave request, the TA clicks on the New Request button, which opens a modal window where key information must be entered. The TA is required to select a Leave Type from a predefined list — such as *Sick Leave*, *Personal Leave*, *Academic Leave*, or *Emergency Leave*. An optional Other category is also available for cases that do not fall under the standard types. Following this, the TA must enter the start and end dates of the leave period and provide a brief explanation for the request.

Additionally, the interface allows TAs to upload supporting documents, such as medical reports or conference invitations. While documentation is not mandatory, it is *highly recommended*, especially for leave types like sick leave or academic leave where justification may influence approval. Once the form is complete and the Submit button is clicked, the request is forwarded to the corresponding course instructor for review.

All fields marked with a double asterisk (**) are mandatory. If the user attempts to submit the form without filling in these required fields, the corresponding input areas will be highlighted in red, and an error message will be displayed at the top of the modal. This validation mechanism ensures completeness of the request before submission and guides the TA to correct missing information in real time.

Upon submission, the leave status is marked as Pending and the TA is informed through a confirmation message. Instructors receive the request through the system and via email notification. After reviewing the submission, they can either Approve or Reject the request. If approved, the specified time range is automatically blocked in the TA's availability, ensuring no tasks or proctoring assignments are scheduled during the leave. If rejected, the TA is notified via email along with the instructor's reasoning.

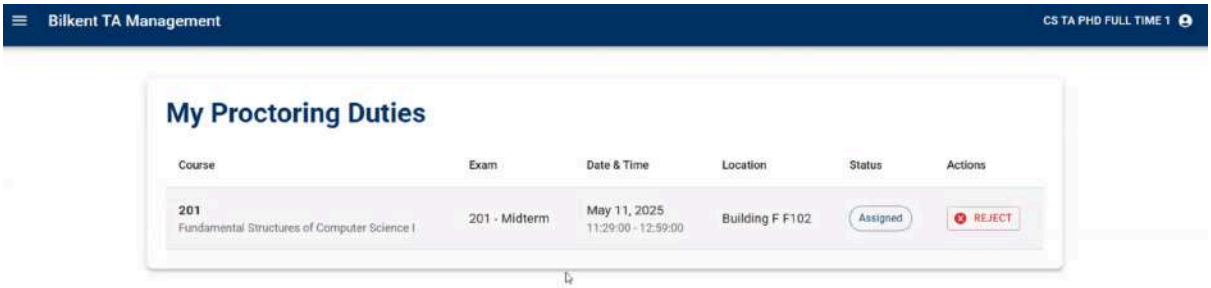
Each request's status is clearly displayed in the TA's Leave Requests table, with visual indicators for *Pending*, *Approved*, *Rejected*, or *Cancelled*. TAs also retain the option to cancel a previously submitted request, as long as it has not yet been approved. This entire flow is managed with clear status updates, institutional logging, and full transparency for both sides.

3.1.5 Task Management

The Task Management page is where Teaching Assistants can view and manage their academic responsibilities assigned by their instructors. This module is directly linked to the courses that a TA is officially assigned to, and tasks originate from the instructors of those specific courses. Each TA may be assigned to one or more courses depending on the department's distribution model.

Once a task has been created by the course instructor, it becomes visible to the TA under this page, categorized by course and task type (e.g., grading, lab session, recitation, office hours). TAs can access detailed information about each task, including the assigned date, estimated workload contribution (in hours), and any additional notes provided by the instructor. The status of each task is clearly shown as Pending, Submitted, or Approved. After completing a task, the TA is expected to update its status through the interface by marking it as complete and, if necessary, providing any comments or adjustments. Submitted tasks are then forwarded to the instructor for review and confirmation. Once confirmed, the workload is automatically added to the TA's semester tally.

3.1.6 Proctoring Assignments



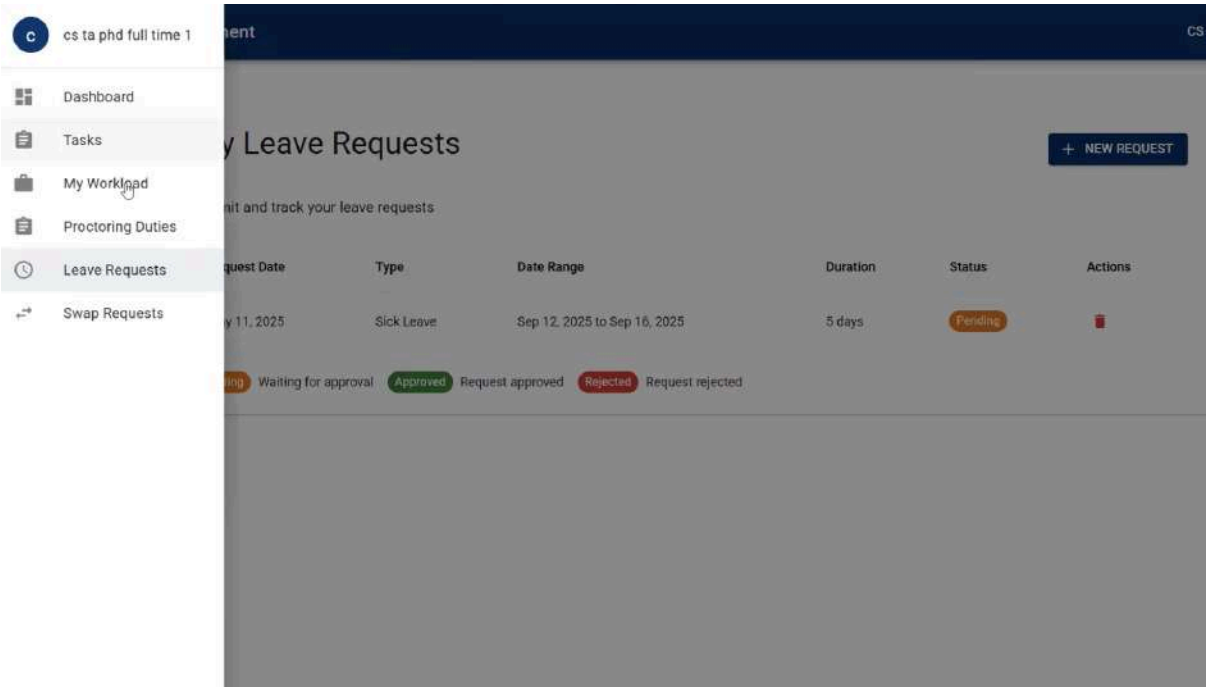
The Proctoring Assignments module enables TAs to view and manage their invigilation responsibilities for midterm, quiz, or final exams. Once the scheduling process is completed by the administrative staff or dean's office, assigned proctoring duties become visible in this module.

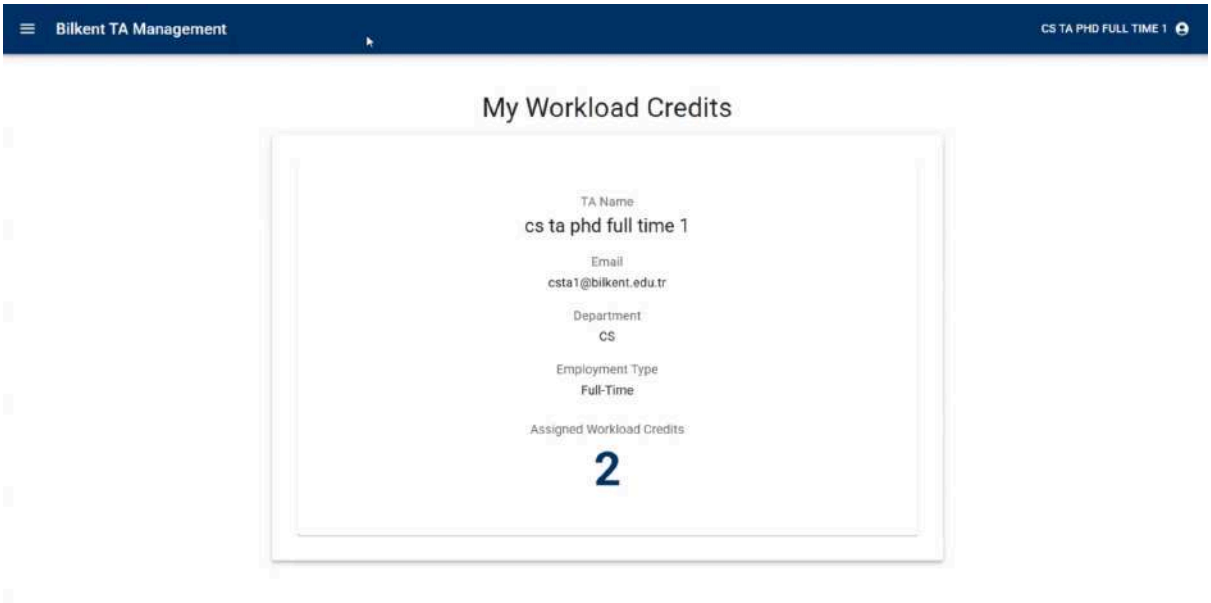
Each proctoring entry includes the course code, exam type (e.g., midterm, final), exam date and time, and the designated classroom. TAs can review whether a task is assigned manually or automatically, and can access detailed proctoring logistics, such as expected duration and location. Assignments are marked with status indicators such as Assigned, Swap Requested, or Cancelled.

If a TA has a valid scheduling conflict or is unable to attend the assigned duty, they may initiate a swap request through the system. This request is forwarded to another eligible TA, who must explicitly accept it for the swap to be finalized. Swap approvals are logged, and the system ensures that the resulting schedule remains conflict-free. Additionally, if mutual agreement cannot be reached, the Authorized Staff may perform a Force Swap Operation, overriding swap history or consent in urgent cases.

Proctoring assignments are generated based on two criteria: the TA's availability and the instructor-course linkage. The system prioritizes TAs who are officially associated with the course, and among them, selects those with the lowest current workload. If no course-specific TAs are available, broader availability and load balancing logic is used.

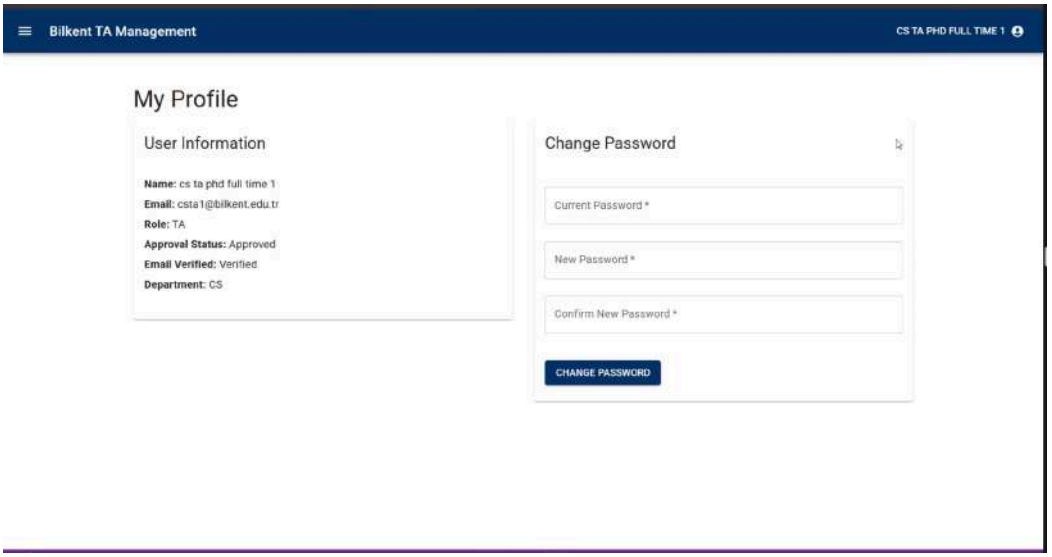
3.1.7 Workload





TA users can view their assigned workload in detail by clicking the **“My Workload”** tab from the left-hand menu. On the displayed screen, essential information such as the user’s name, email address, department, and employment type (full-time/part-time) is shown.
At the bottom of the page, the total number of workload credits assigned to the TA is prominently displayed in large font. This section allows TAs to monitor their current workload at a glance and serves as a reference when requesting additional tasks.

3.1.8 My Profile



The **My Profile** page allows users to review their account information and update their password. On the left side of the screen, users can view key details such as their full name, email address, system role (e.g., TA), approval status, email verification state, and

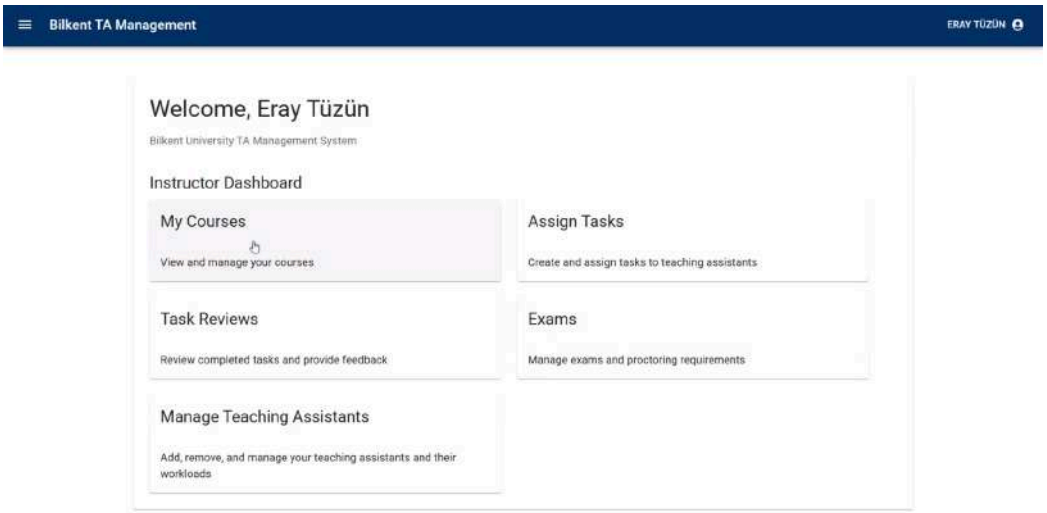
associated department. These fields are pulled directly from the backend and are typically set during the account creation and onboarding process.

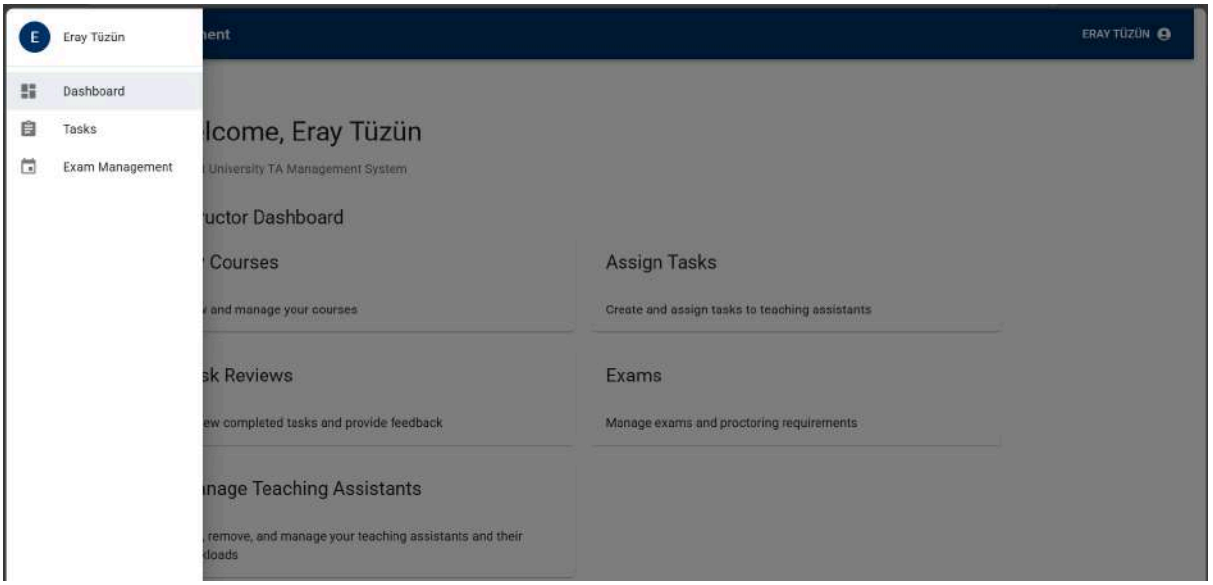
On the right side, the **Change Password** section enables users to update their credentials securely. Since all users initially receive a system-generated password via email, it is strongly recommended that they create a personalized password upon first login. This ensures greater account security and prevents potential access issues in the future.

To change the password, the user must enter their current password, choose a new password, and confirm it. The system verifies the input and applies the change only if all fields are correctly filled out and matched. This functionality promotes safe access and account ownership throughout the TA Management System.

3.2 Course Instructor

3.2.1 Instructor Dashboard Page





After logging into the system with their Instructor account, the user is directed to the main Dashboard page. This interface serves as the central hub from which instructors can access all key functionalities. While the sidebar on the left allows navigation between pages, the main area presents clickable tiles for quick access to core tasks.

The "View My Courses" tile directs instructors to the Course Management interface, where they can see all courses and sections they have been assigned. This page acts as the foundation for tasks such as defining exams or assigning duties.

The "Assign Tasks" tile allows instructors to create and assign specific tasks to their own Teaching Assistants (TAs). Each task includes fields for title, description, deadline, credit hours, and can be individually assigned to a selected TA.

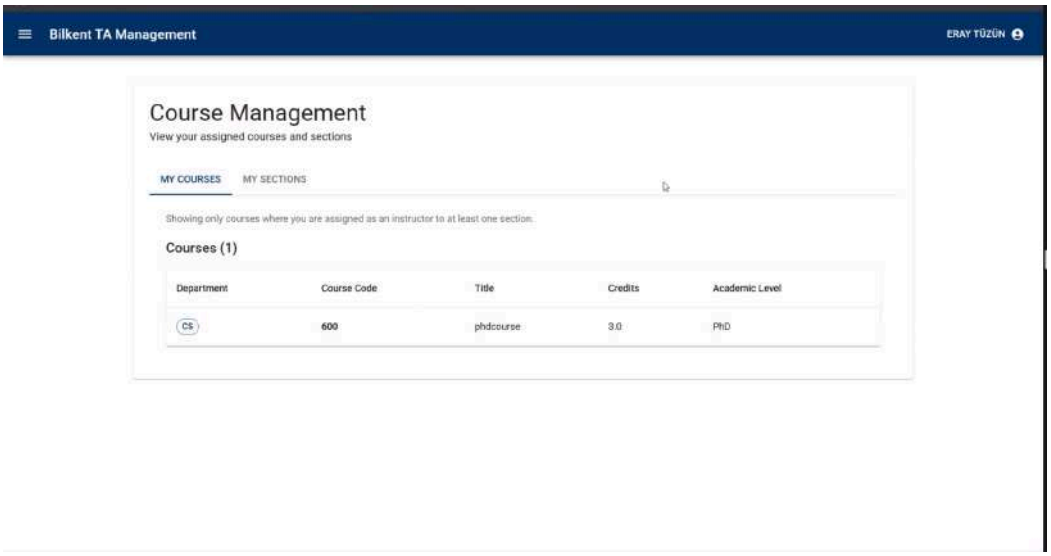
The "Task Reviews" section is used to review and approve tasks submitted by TAs. Once approved, the workload is automatically reflected in the TA's overall contribution in the system.

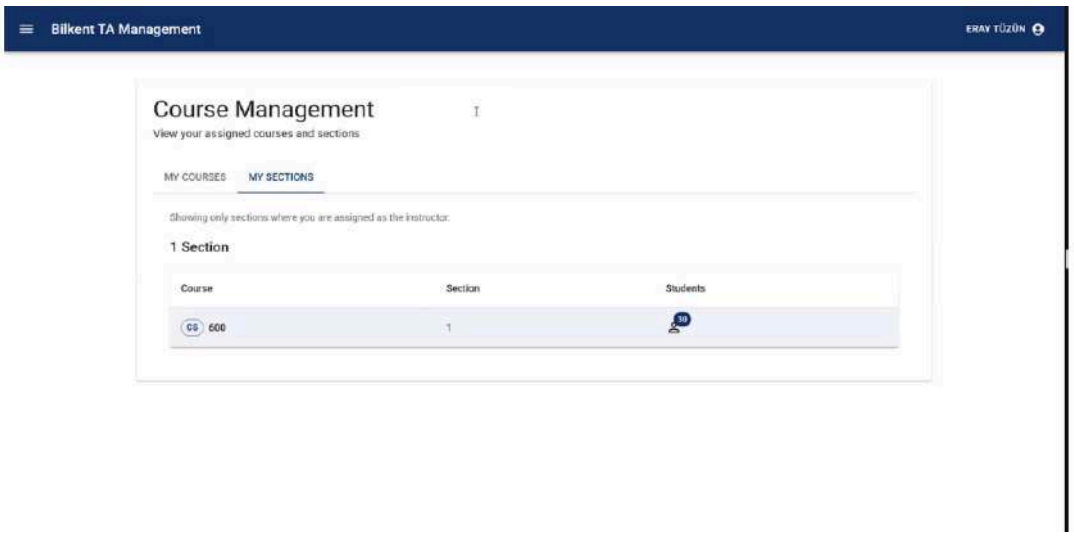
The "Exams" tile leads to the exam management module, where instructors can plan exams for their courses by entering the necessary details. However, it's important to note that proctor assignments are handled solely by the Authorised Staff.

Finally, the "Manage Teaching Assistants" section enables instructors to oversee only their own TAs. From here, they can view each TA’s assigned duties, workload, and any leave requests they have submitted.

This dashboard is designed to support instructors throughout the semester, offering a simple and intuitive interface that is easy to navigate from the very first use.

3.2.2 Course Management – My Courses & My Sections



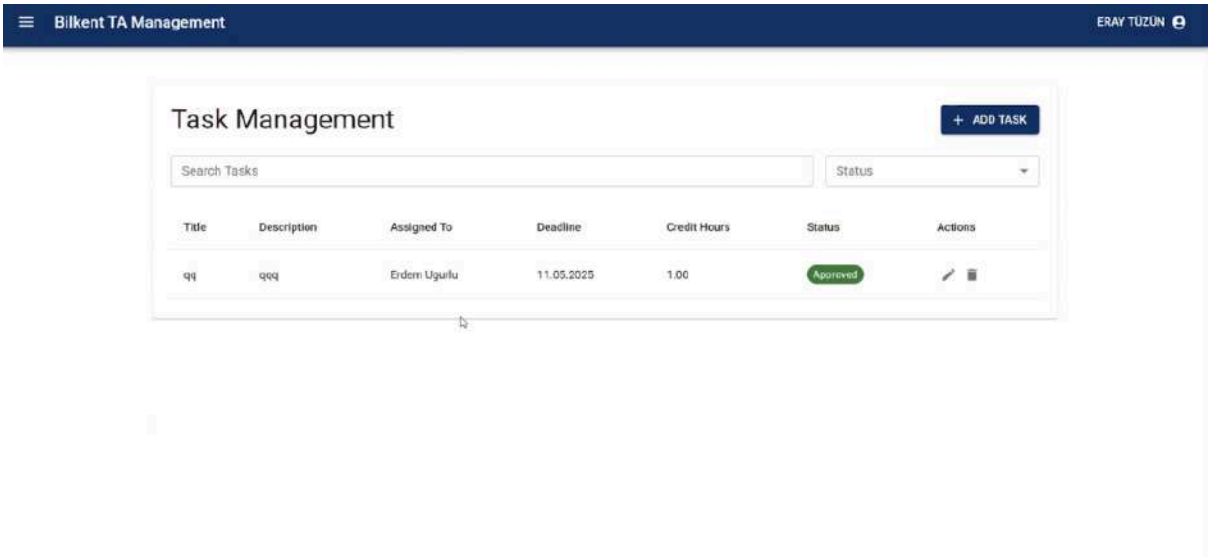


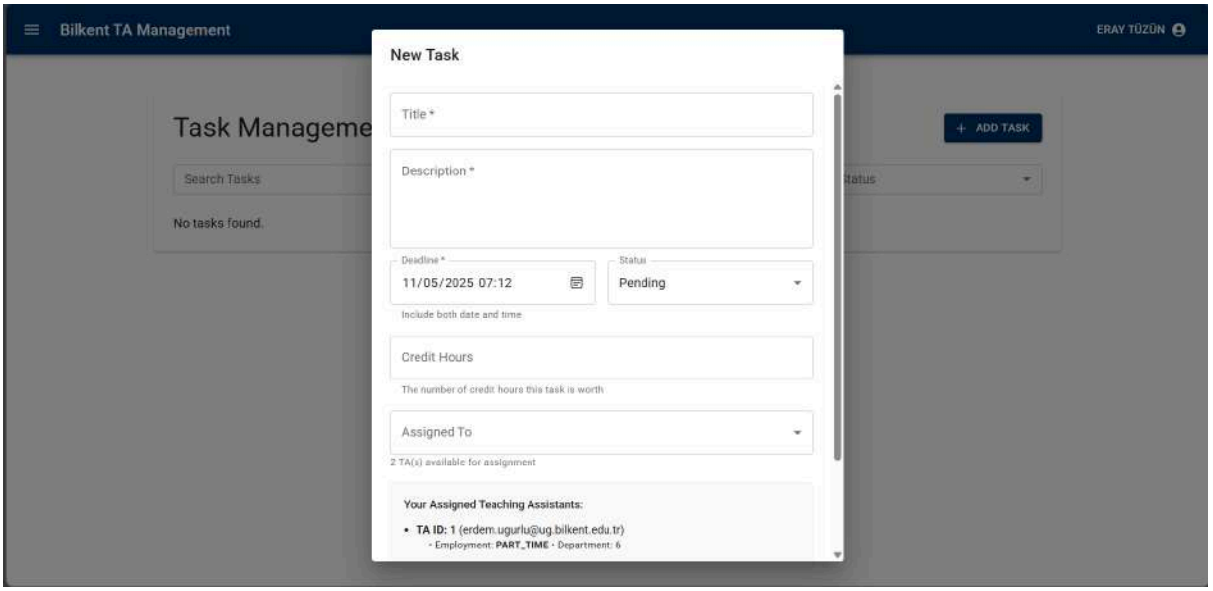
Instructors interact with the **Course Management** page to view the courses and specific sections they are officially assigned to. This interface ensures that only relevant academic data is visible, maintaining clarity and role-specific access. The interface is divided into two primary tabs: **My Courses** and **My Sections**.

The **My Courses** tab provides a summary of the instructor's officially assigned courses. For each entry, the system displays the department name, course code, course title, number of credit hours, and academic level (e.g., undergraduate, graduate). If this section is empty, it indicates that no courses have yet been assigned to the instructor, and they are not currently associated with any instructional responsibilities.

The My Sections tab drills down into the specific sections of each course the instructor is responsible for. Each section entry includes the course title, section number, and the current number of enrolled students. These section assignments play a critical role during both exam setup and task delegation processes, as only instructors of a given section are allowed to assign TAs or manage exams for that group.

3.2.3 Task Management





The **Task Management** section provides instructors with a centralized interface to create, assign, and track tasks for their assigned Teaching Assistants. When the instructor clicks the **+ Add Task** button, a modal window appears, allowing them to define the details of the task

before assigning it. The form contains several key fields, each supporting structured task definition.

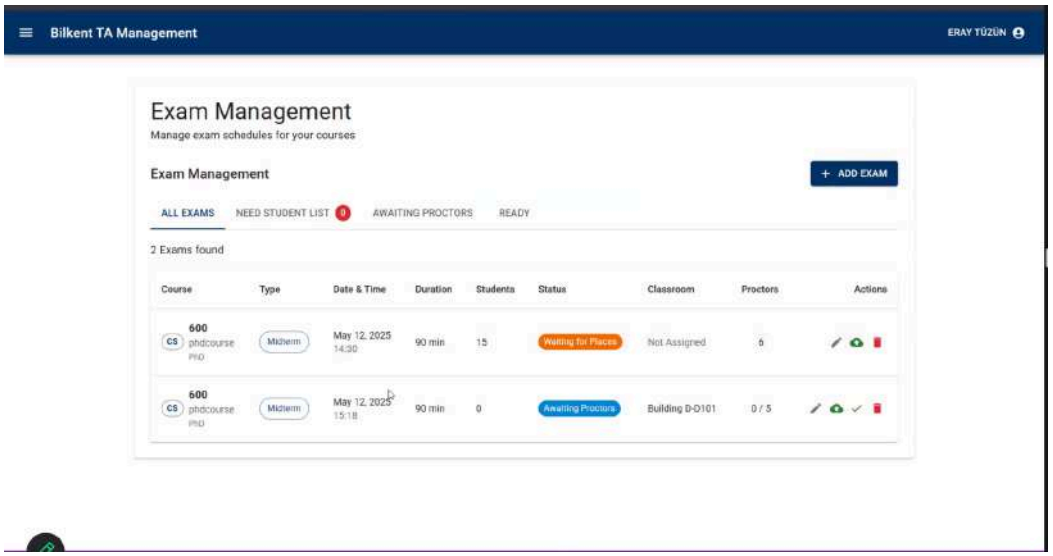
Instructors begin by entering the **Title**, which briefly names the task—for instance, “Quiz Grading” or “Assignment Review.” Next, the **Description** field allows for detailed explanations of what the task entails, including specific expectations, files to be checked, or grading schemes. The **Deadline**field captures both the date and time by which the TA should complete the task, ensuring clarity on time-sensitive duties. The **Status** is set to **Pending** by default but may later be updated depending on task progression.

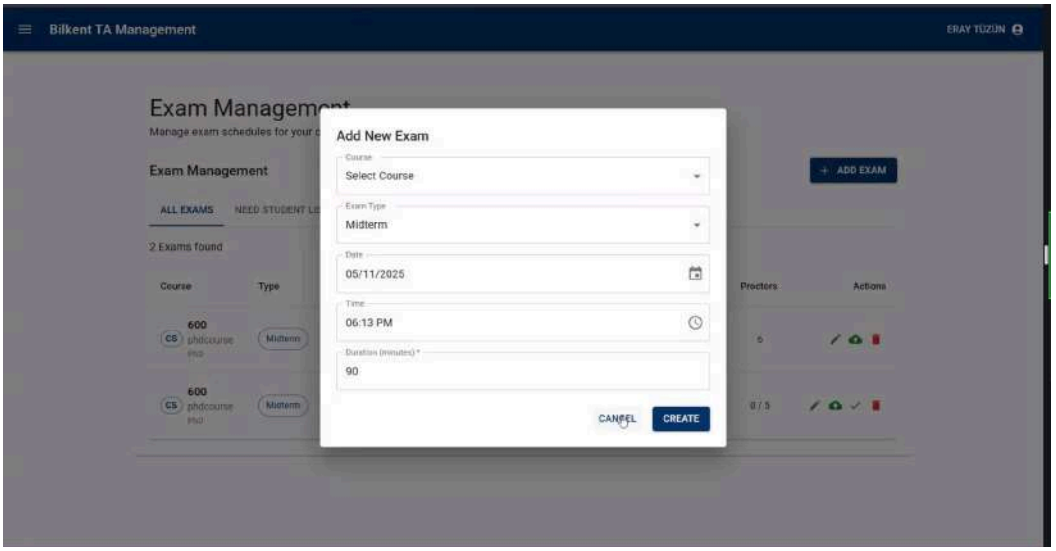
The **Credit Hours** input lets instructors define how many workload hours the task should contribute to the TA’s semester total. Beneath this, the **Assigned To** dropdown lets instructors select the relevant TA(s) from a list limited to their own assigned assistants. This restriction ensures that task delegation remains within the instructor’s authority and maintains accurate workload distribution.

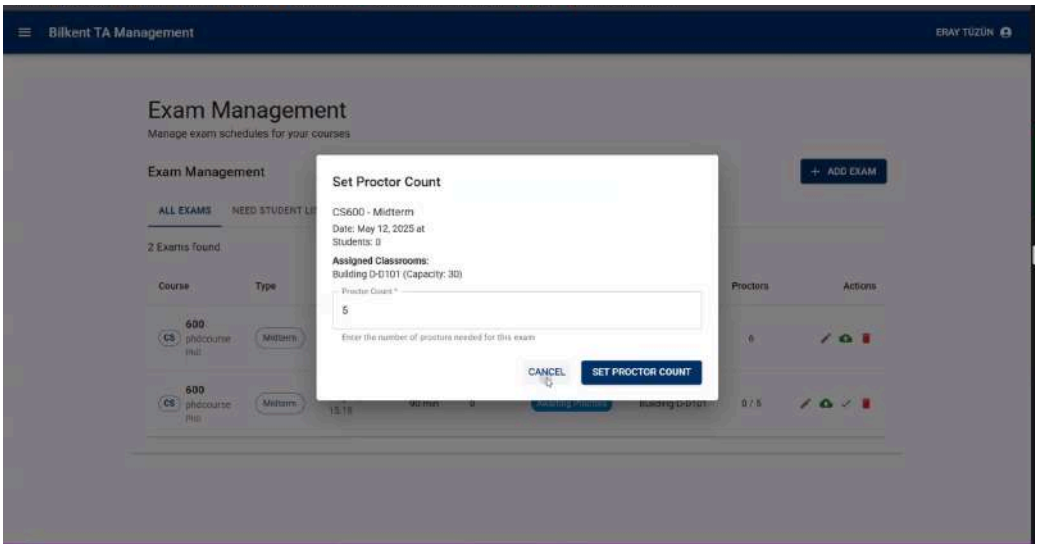
At the bottom of the form, the instructor sees a summary titled **Your Assigned Teaching Assistants**, listing their available TAs. Each entry includes the TA’s university email, employment type (e.g., Part-Time), and department or academic program. This transparency helps instructors choose the most appropriate assistant for the task.

Once a TA completes a task, they mark it as done from their own interface. The instructor is then notified and given the opportunity to review and approve the completed work. Upon approval, the system automatically updates the TA’s workload, ensuring real-time accuracy across records.

3.2.4 Exam Management







The **Exam Management** interface for Course Instructors allows them to manage and monitor the exams related to the courses they are responsible for. Instructors can create a new exam using the **+ Add Exam** button, where they select the course, exam type (Midterm, Quiz), date, time, and duration in minutes. Upon creation, the exam is added with a status of **“Need Student List.”**

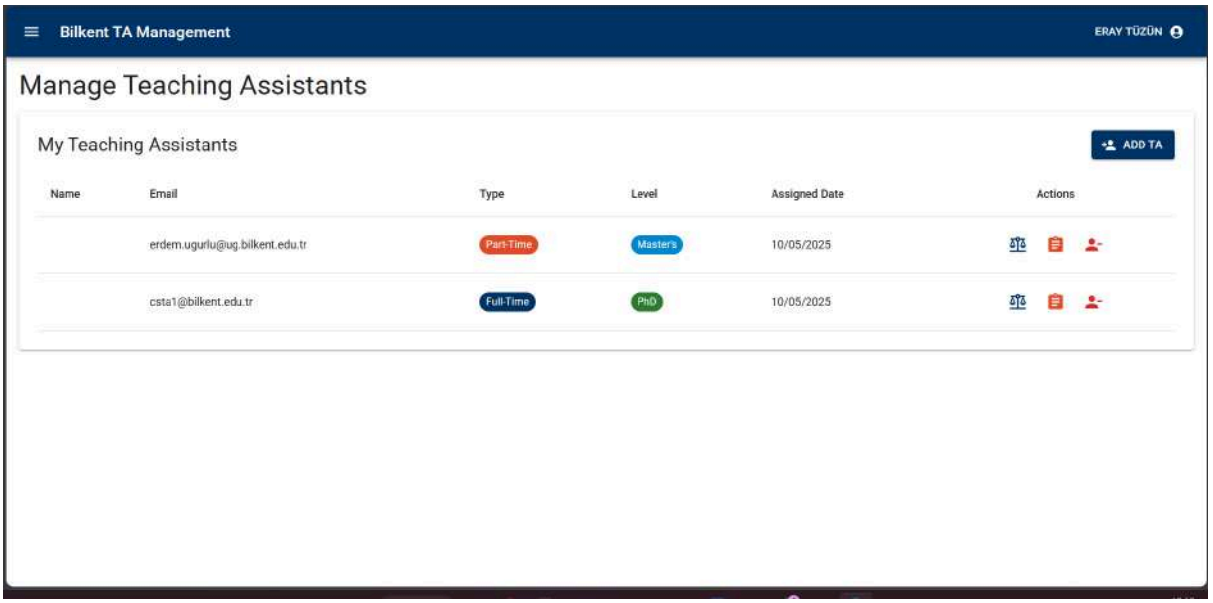
At this point, the Instructor is expected to upload a student list in Excel format, containing each student's name, surname, ID, and course enrollment information. Once uploaded successfully, the system advances the status to **“Waiting for Places,”** waiting for classroom assignment from the Dean’s Office.

After classrooms are assigned, the exam status moves to **“Awaiting Proctors.”** Here, **Instructors are allowed to specify the required number of proctors** by clicking the checkmark icon, which opens the “Set Proctor Count” modal. They can enter the number of proctors based on exam conditions and classroom capacity.

However, **Instructors do not have permission to assign Teaching Assistants as proctors.** The actual proctor assignment process—manual or automatic—is managed exclusively by the Authorized Staff. Instructors can still track the number of assigned proctors and review progress, but cannot perform the assignments themselves.

Instructors can also re-upload or update student lists after creation if needed, for example in cases of make-up exams or registration changes.

3.2.5. Manage Teaching Assistants

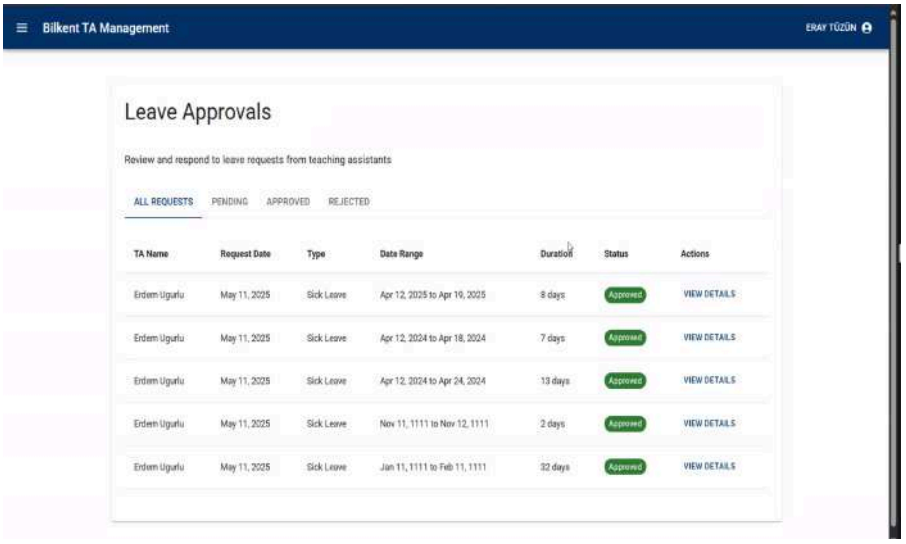


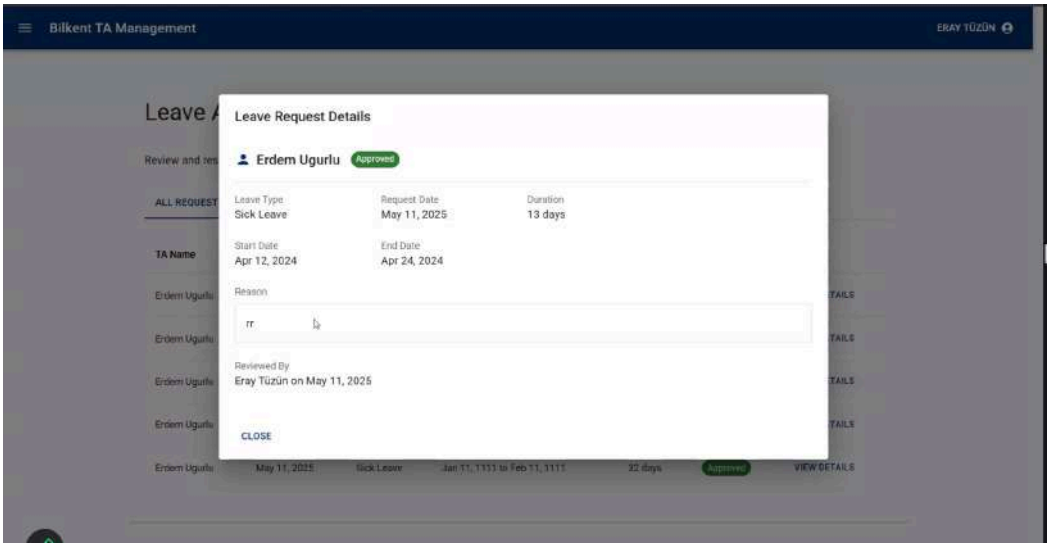
The **Manage Teaching Assistants** section enables instructors to oversee the teaching assistants (TAs) officially assigned to their courses. Only TAs directly connected to the instructor's courses appear in this list. For each TA, the system displays their **name**, **email address**, **employment type** (Full-Time or Part-Time), and **academic level** (such as Master's or PhD). Additionally, the **assigned date** indicates when the TA was officially linked to the instructor's supervision.

To facilitate management, three action icons are available beside each TA's entry. Clicking the **scales icon** opens the TA's workload summary, where instructors can track accumulated hours and duty types. The **checklist icon** provides a breakdown of all tasks assigned to that TA, including current completion status. If the instructor needs to remove a TA from their list, they can click the **red user icon**, though removal rights may be restricted and could require coordination with Authorized Staff depending on system permissions.

Beyond task assignment, instructors also play a key role in approving submissions from their TAs. Under the **Duties** workflow, TAs report the type of academic task completed, the time spent, and any relevant notes. Instructors must **approve or reject** these entries before they count toward the TA's workload record. Similarly, **Leave Requests** submitted by TAs are routed to the instructor for review. Each request includes a leave period, selected type (e.g., Sick, Academic, Personal), and an explanation. If approved, the leave dates are blocked from any further proctoring or task assignment. If rejected, the TA receives an automated notification with the stated reason. This module ensures instructors remain actively engaged in their assistants' workload and availability.

3.2.6. Leave Approvals



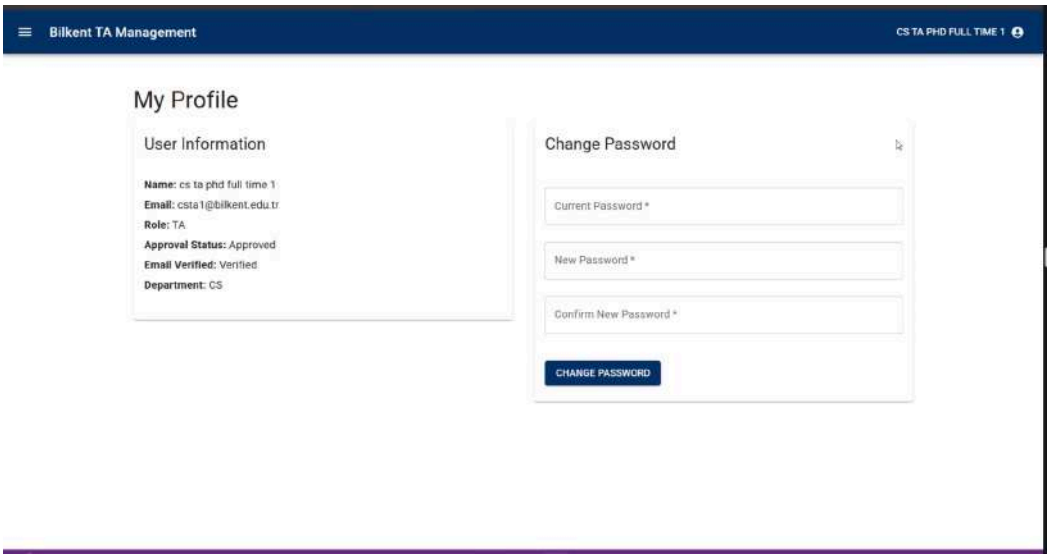


This page serves as the *Leave Approvals* panel, where instructors can review and respond to leave requests submitted by Teaching Assistants (TAs). When a TA wishes to request time off, they do so through the system by specifying the type of leave (such as *Sick Leave*), start and end dates, and the reason for the request. The reason field allows them to provide additional context—for example, “I have the flu” or “medical report attached”—to justify their absence.

All submitted requests are listed in a structured table showing the TA’s name, request date, leave type, date range, total duration, current status (Approved, Pending, or Rejected), and a link to view detailed information. By clicking the “View Details” button, instructors can see all relevant information in a pop-up window and then choose to either approve or reject the request.

An important rule enforced by the system is that **once a leave request is approved, the TA cannot be assigned to any other duty**, such as proctoring exams, during the approved leave period. This prevents scheduling conflicts and ensures that instructors respect the personal time and well-being of TAs who are officially excused. The system automatically disables any proctoring or other assignments that might overlap with the leave window.

3.2.7. My Profile



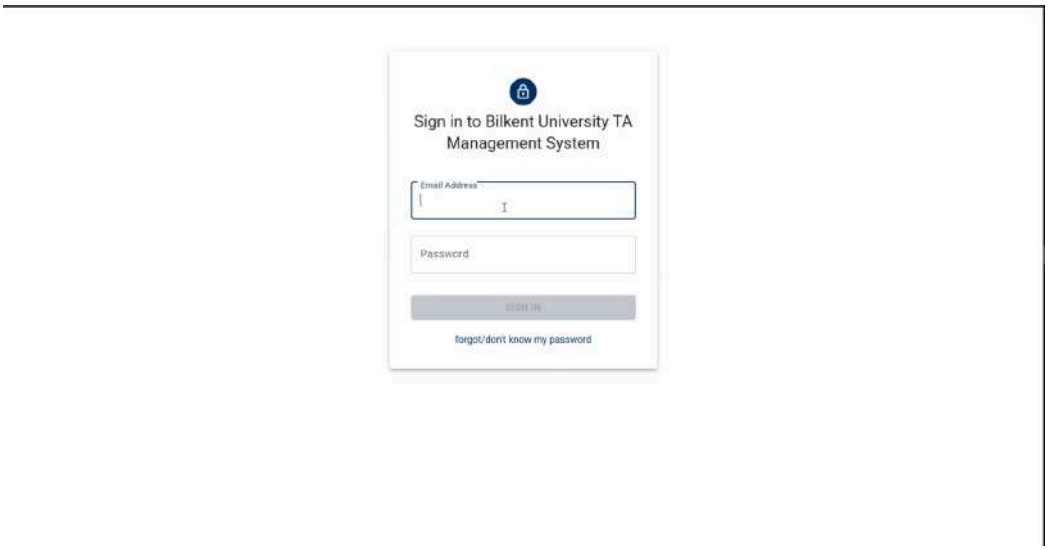
The **My Profile** page allows users to review their account information and update their password. On the left side of the screen, users can view key details such as their full name, email address, system role (e.g., TA), approval status, email verification state, and associated department. These fields are pulled directly from the backend and are typically set during the account creation and onboarding process.

On the right side, the **Change Password** section enables users to update their credentials securely. Since all users initially receive a system-generated password via email, it is strongly recommended that they create a personalized password upon first login. This ensures greater account security and prevents potential access issues in the future.

To change the password, the user must enter their current password, choose a new password, and confirm it. The system verifies the input and applies the change only if all fields are correctly filled out and matched. This functionality promotes safe access and account ownership throughout the TA Management System.

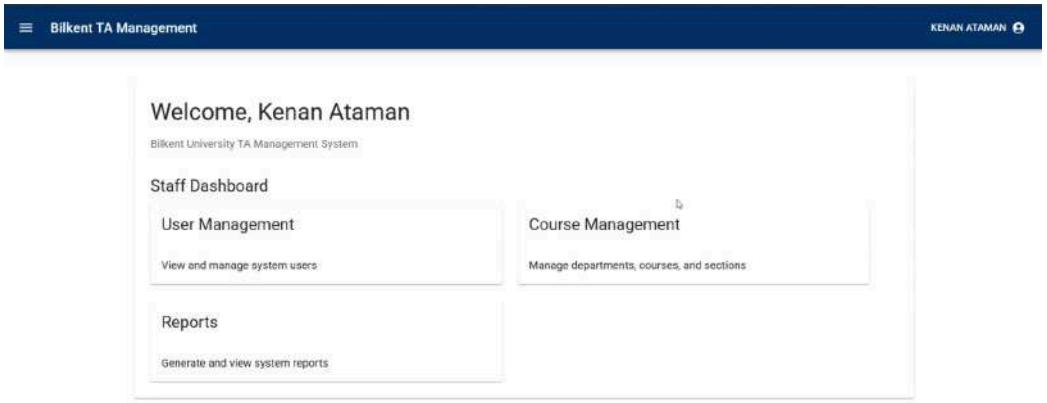
3.3 Authorized Staff

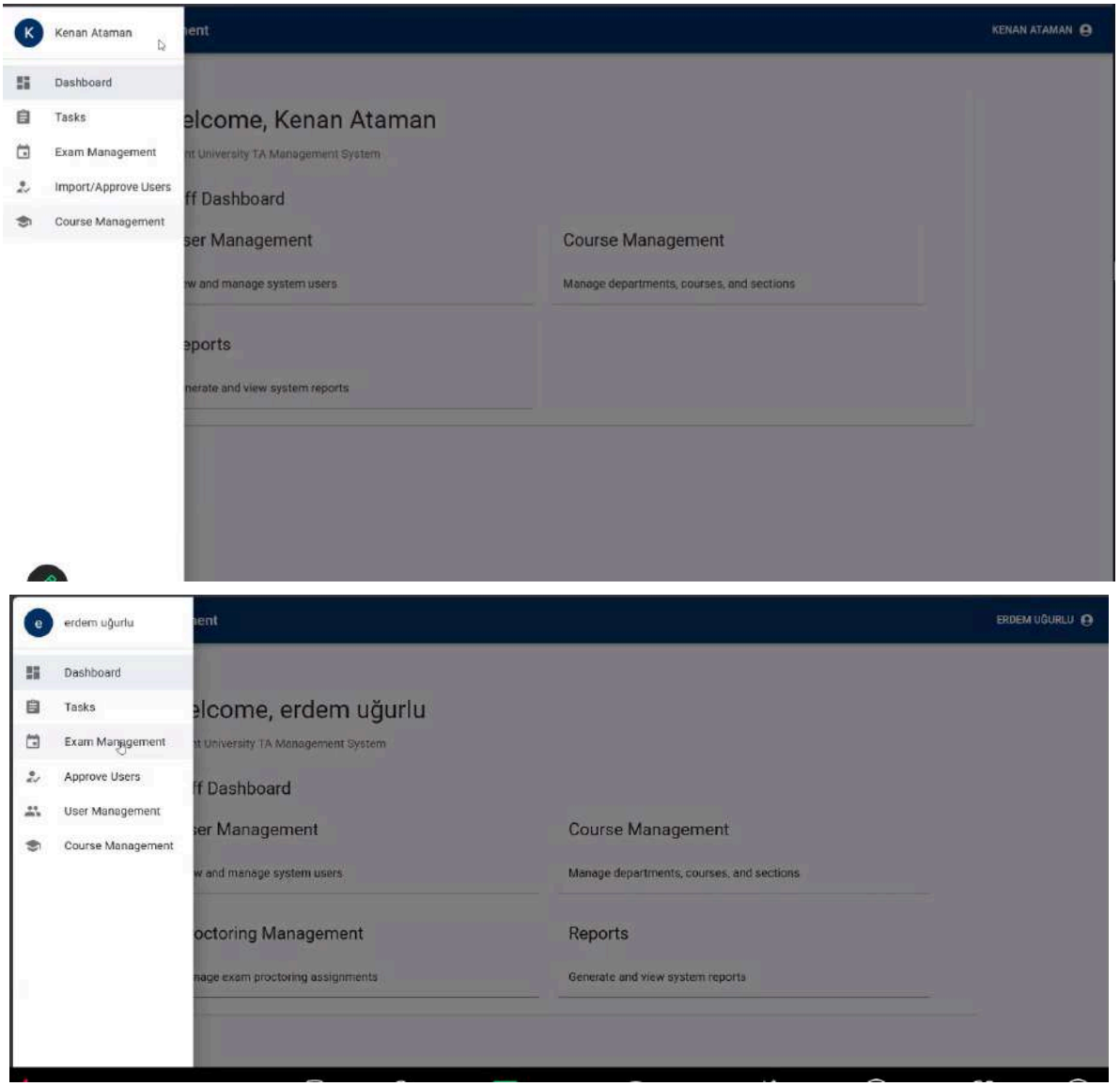
3.3.1 Sign-in



Unlike other users, Authorized Staff accounts are not self-registered instead, their credentials (including password) are directly created and provided by the Admin.

3.3.2 Dashboard





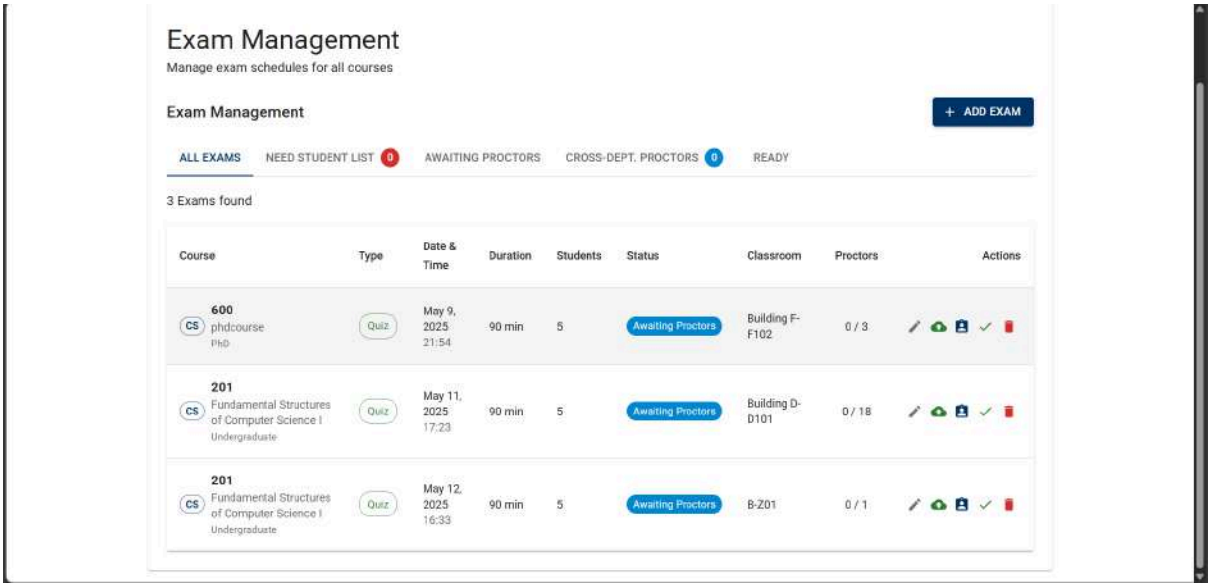
After successfully logging in, users with Authorized Staff privileges are greeted by a streamlined and focused operational dashboard designed specifically for academic coordination. The interface emphasizes clarity and utility, placing essential administrative tools within immediate reach. A collapsible sidebar on the left provides navigation to core modules, including Dashboard, Tasks, Exam Management, Import/Approve Users, and Course Management—each section representing a distinct area of responsibility for departmental staff members.

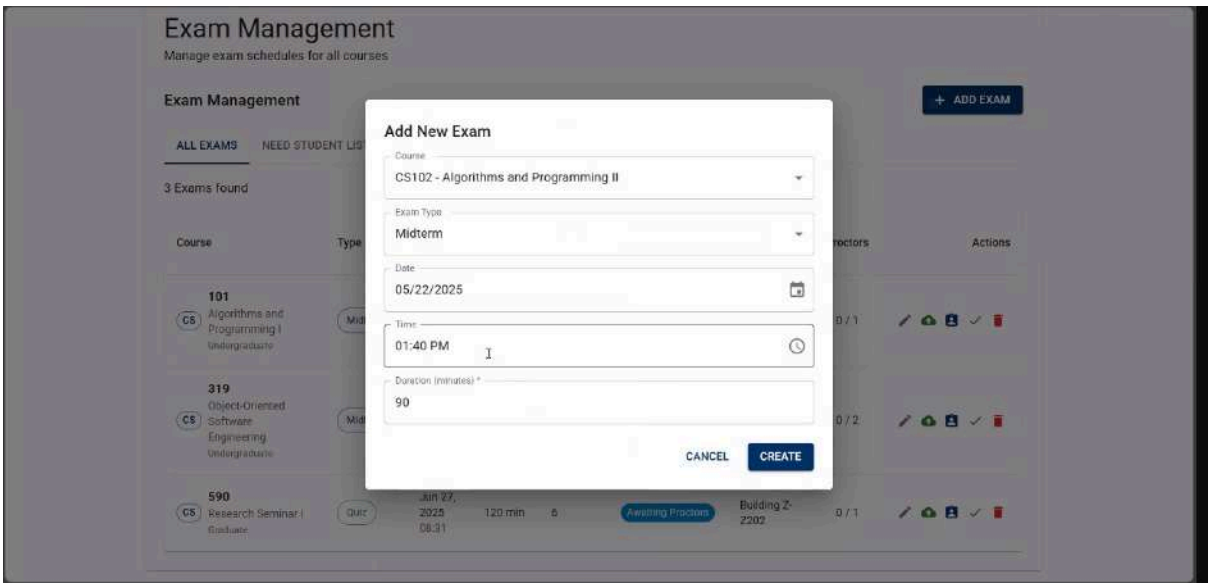
At the center of the screen lies the Staff Dashboard, visually divided into two primary focus areas. On the left, the **User Management** section enables staff to approve new sign-up requests from Teaching Assistants and Instructors, assign roles, and deactivate accounts when necessary. This process ensures that only verified users gain access to system functionalities. On the right, the **Course Management** section allows users to define and

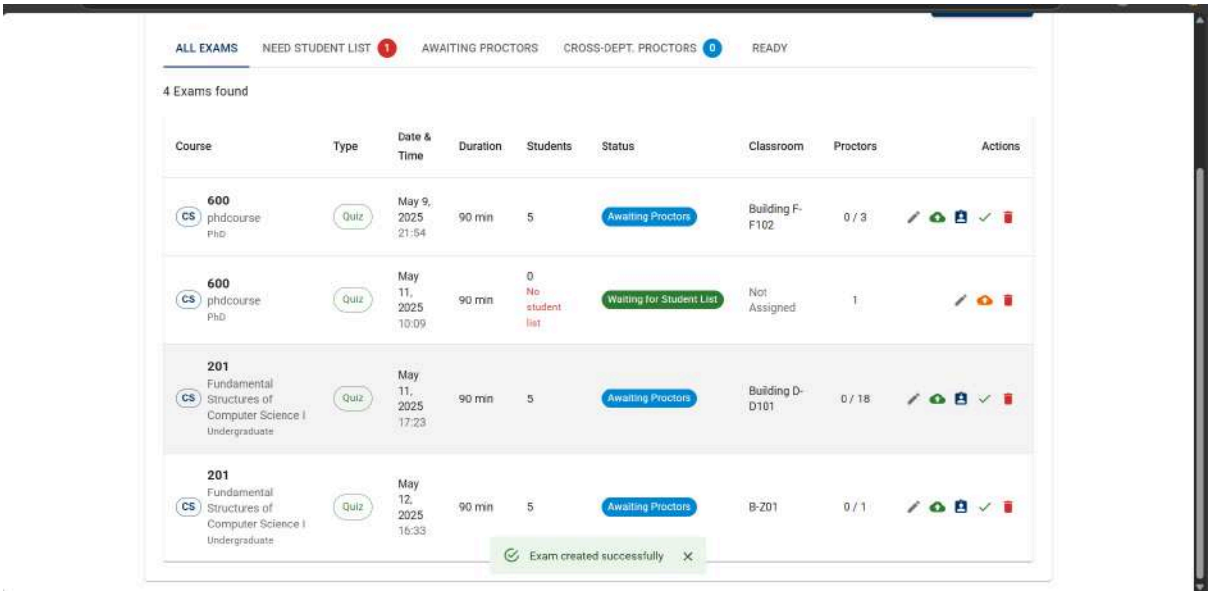
edit academic structures, including departments, individual courses, and their relevant sections, forming the foundation upon which TA assignments and exam definitions are built.

Additionally, the **Exam Management** module in the sidebar offers tools to create exams, upload student lists, and assign TA proctors either manually or automatically. The **Import/Approve Users** tab enables batch upload and verification of personnel and TA data via Excel files, streamlining bulk operations for larger departments. These features combine to offer a centralized, role-sensitive management system tailored to academic coordination needs.

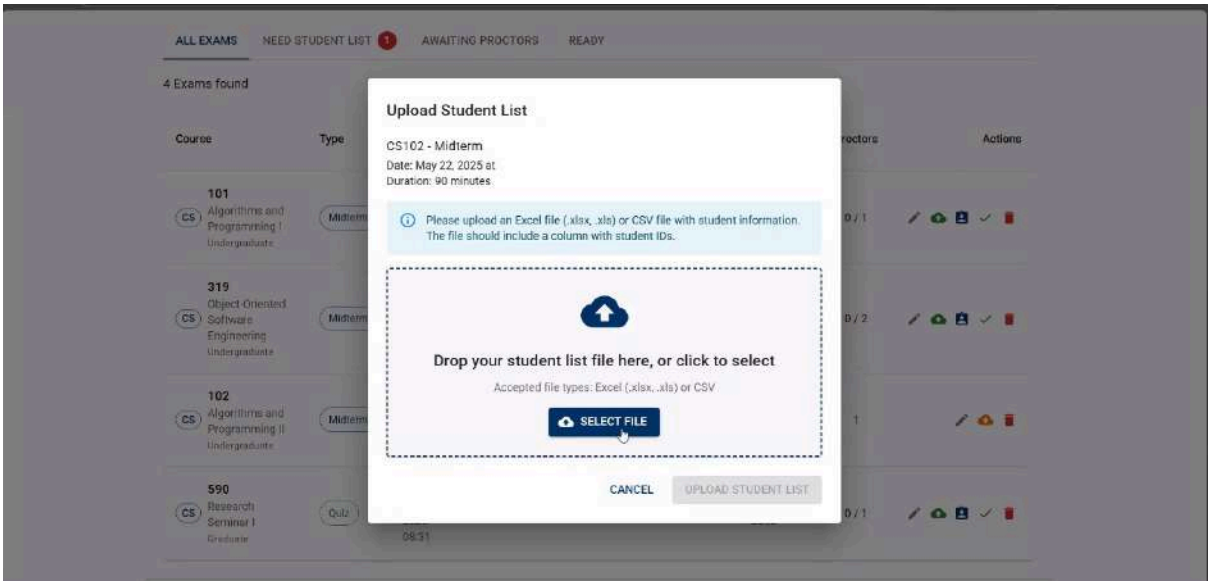
3.3.3 Exam Management

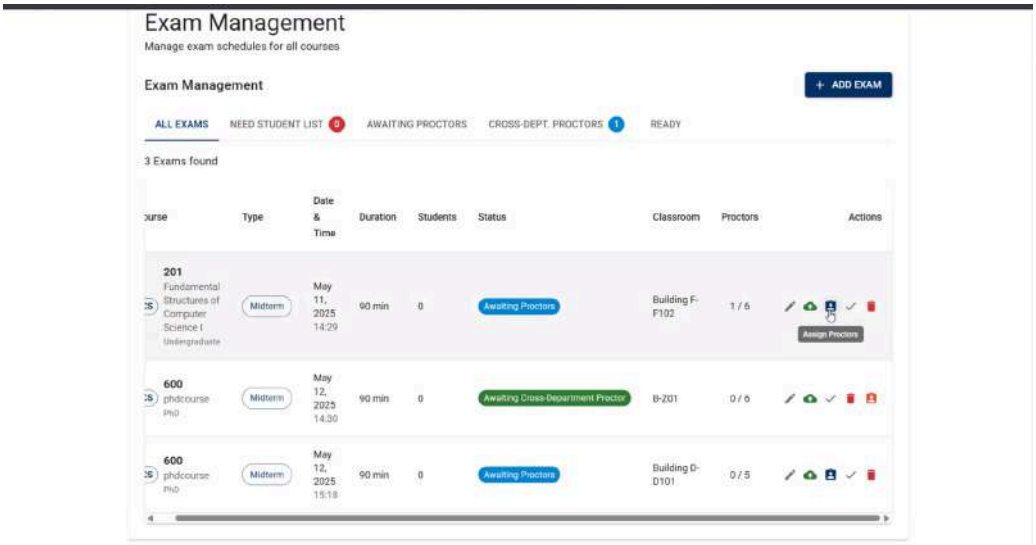




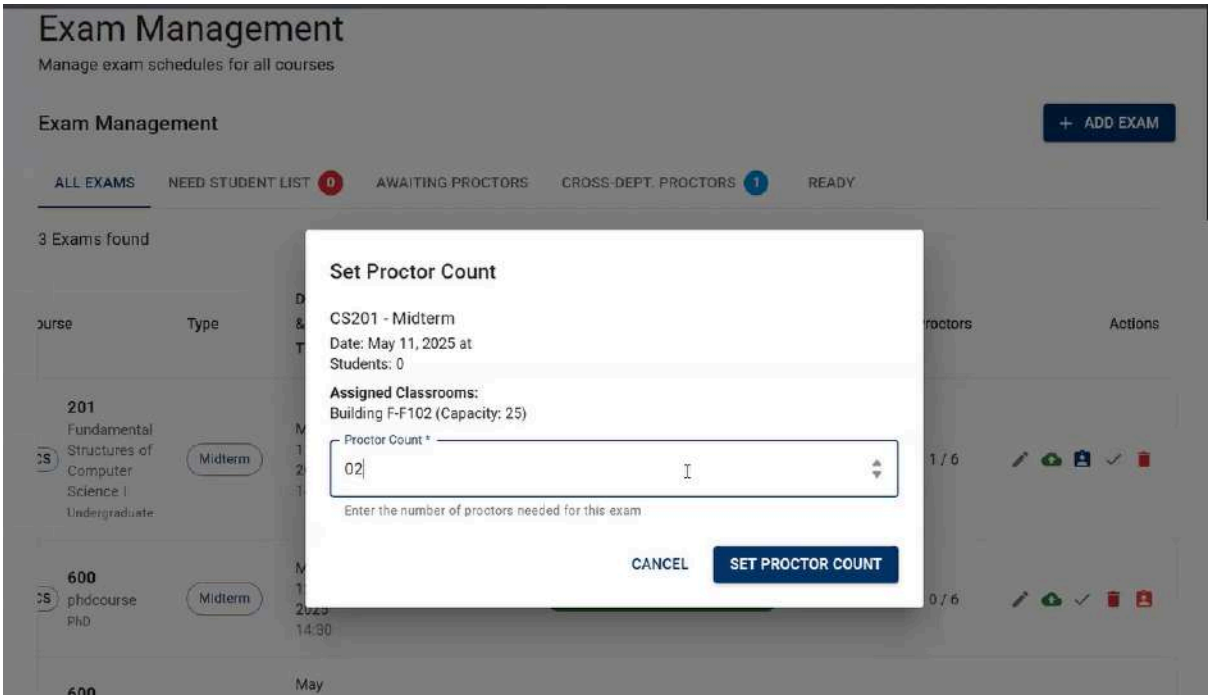


The **Exam Management** interface allows Authorized Staff to coordinate the scheduling and supervision of all departmental exams. By clicking the **+ Add Exam** button in the top right corner, users can initiate the creation of a new exam. Only courses relevant to the staff member's department (e.g., CS101, CS319) are shown in the course dropdown menu. To successfully define an exam, users must input the course, exam type (Midterm, Quiz, or Final), date and time, and duration in minutes. Once submitted, the exam is saved and automatically transitions to the **"Need Student List"** status.

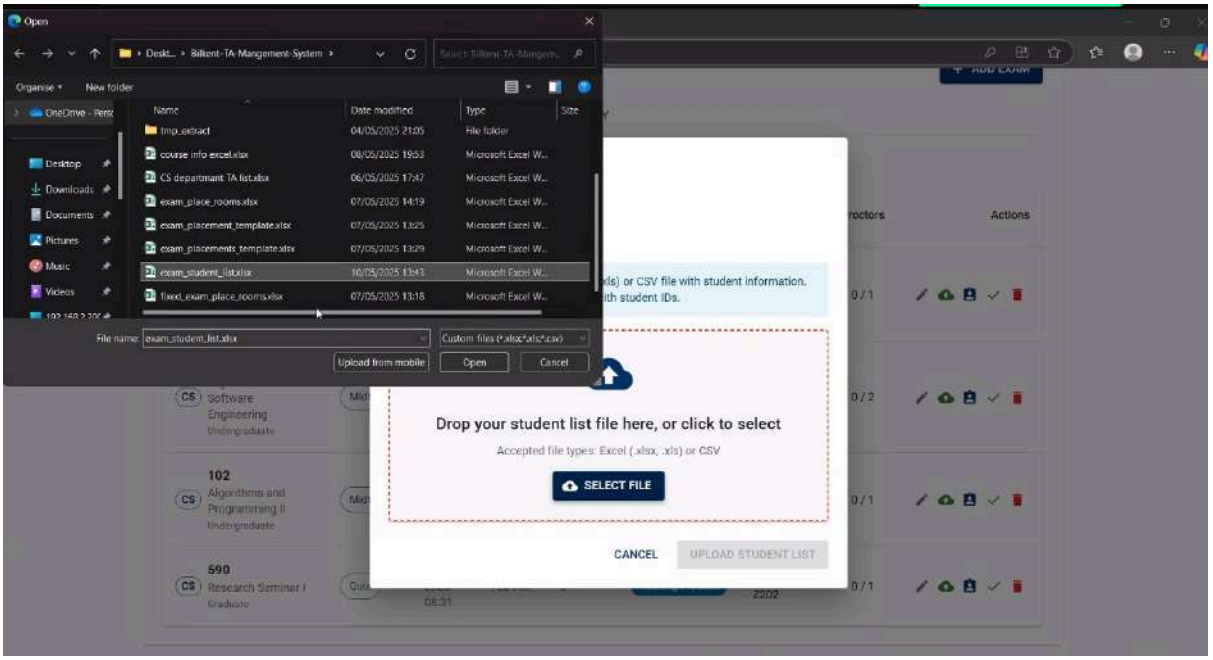




At this stage, the system prompts the user to upload a student list in Excel format. This list must include each student's name, surname, ID number, and course registration information. Once the upload is complete and verified, the system updates the exam's status to "Waiting for Places", indicating that the Dean's Office must now assign appropriate classrooms.

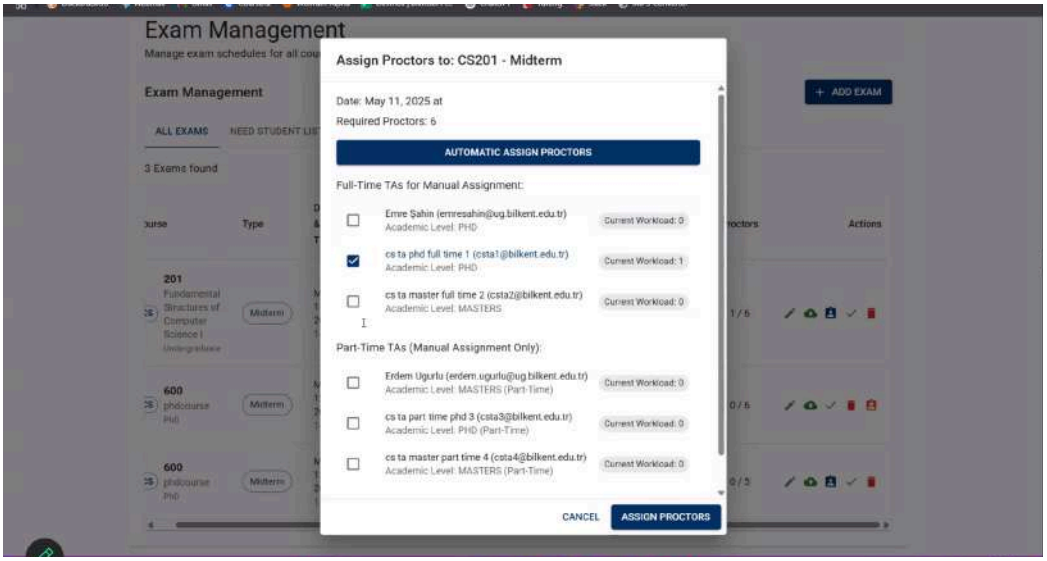


Once classrooms are assigned, the exam moves into the **“Awaiting Proctors”** state, allowing Teaching Assistant assignments to begin. However, before proctors can be assigned, the staff member must first specify the number of proctors required. This is done by clicking the **checkmark icon**, which opens the **“Set Proctor Count”** window. Here, the appropriate number of proctors based on classroom capacity is entered and saved.



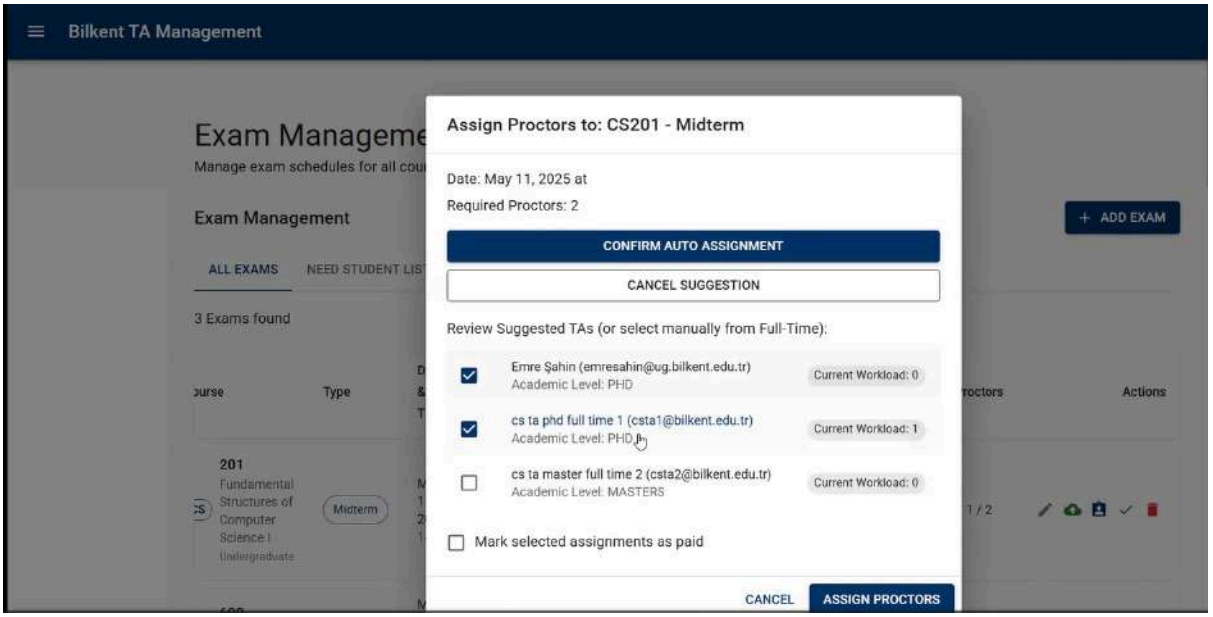
Even after an exam is created, the student list and other exam information can be re-uploaded and updated. This is especially useful in cases such as make-up exams (FZ), report-based excuses, or enrollment changes. The system allows the list to be refreshed before the exam takes place.

3.3.4 Assigning Proctoring

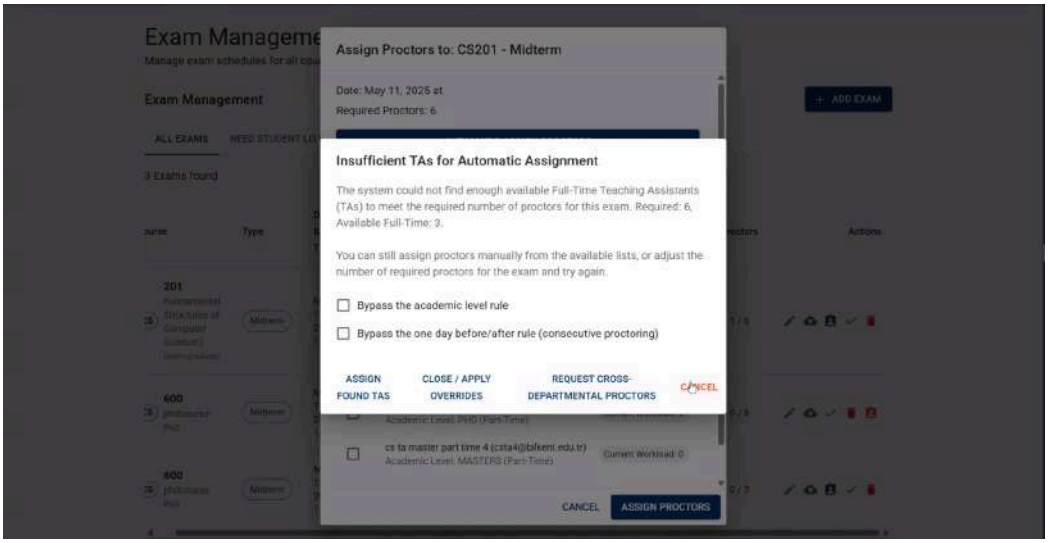


Once the count is set, the Assign Proctors action becomes available via the blue person icon. Clicking it opens a detailed list of eligible TAs, categorized by Full-Time and Part-Time, along with their current workload and academic level (Master's or PhD).

At this stage, the system allows the user to select TAs according to specific requirements. For instance, the user may choose only PhD-level TAs, or only Full-Time TAs, depending on the institutional preference for that particular exam. This enables targeted assignment based on TA profile. Users can manually assign proctors or use the Automatic Assign Proctors feature, which selects the most optimal candidates based on availability and workload.



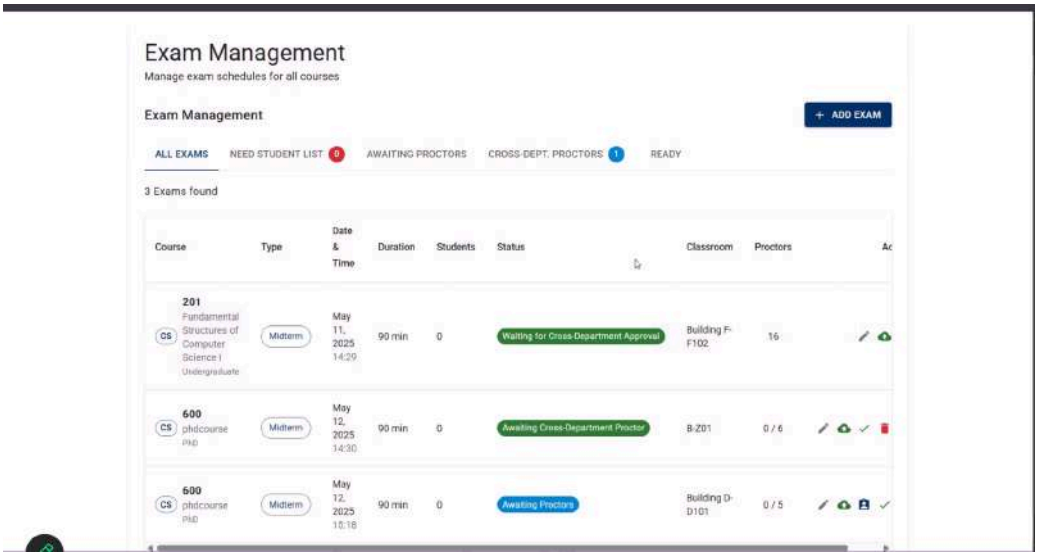
Alternatively, clicking Automatic Assign Proctors triggers a rule-based suggestion system. The algorithm first filters TAs who are officially affiliated with the course. From this filtered list, it then identifies the ones with the lowest current workload who are available during the exam time. The system also enforces important constraints: it avoids assigning TAs to exams for courses they are currently enrolled in, and tries to prevent back-to-back or same-day overloads to maintain fairness and workload balance.

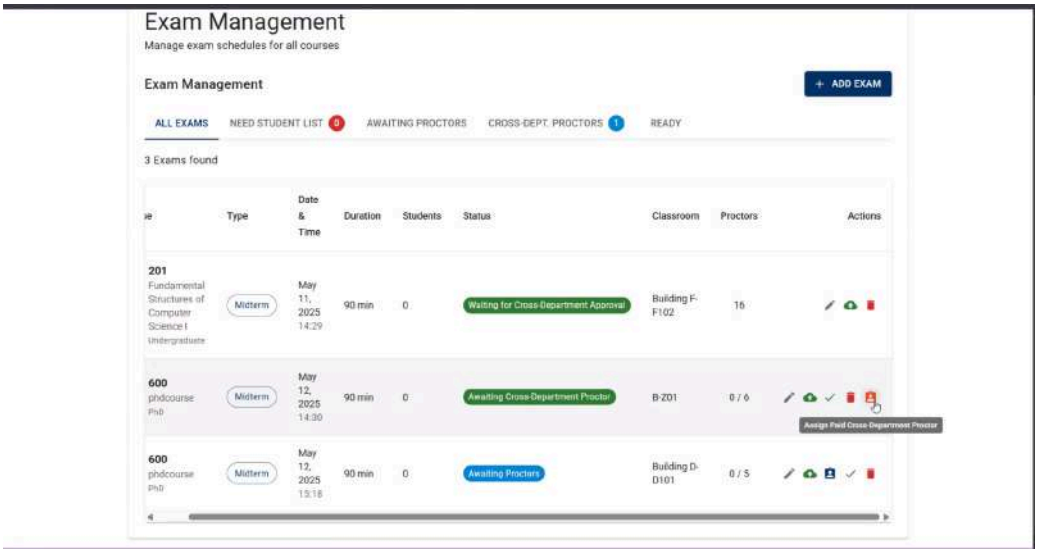


If the number of eligible TAs is insufficient, a warning modal appears. The system offers several options: override academic-level restrictions, allow assignments on consecutive days, manually select TAs, or request cross-departmental proctors if internal staff is insufficient.

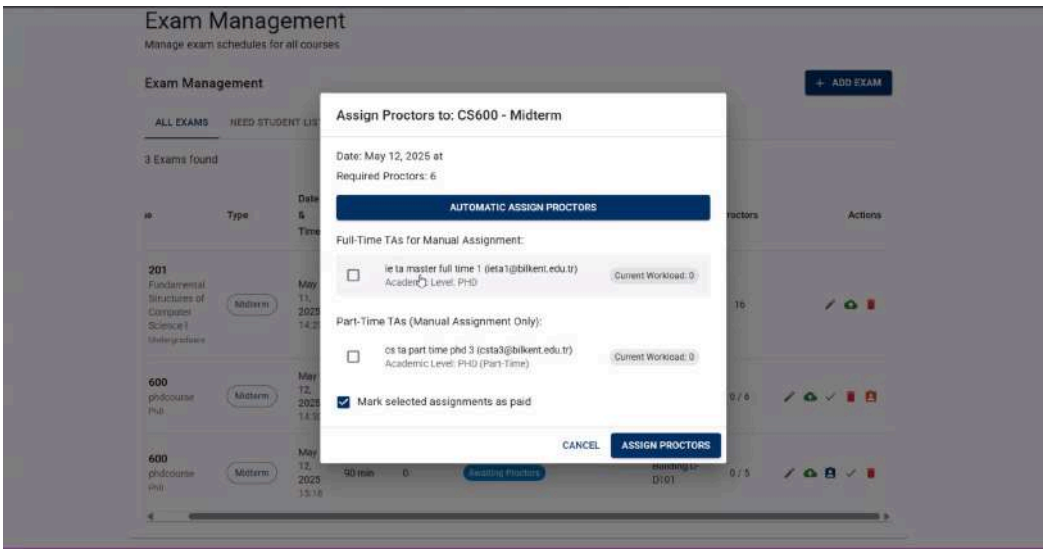
If the system fails to find enough eligible TAs, a warning modal is displayed to notify the user. Within this popup, several alternative solutions are offered. By selecting the **“Bypass the academic level rule”** checkbox, users can allow Master’s level TAs to be assigned to exams that would normally require PhD-level proctors. Similarly, enabling **“Bypass the one day before/after rule”** removes the restriction that prevents TAs from being assigned on two consecutive days. After selecting one or both options, the user must click the **“Close / Apply Overrides”** button to confirm the override rules.

Alternatively, the user may proceed with assigning only the available proctors by clicking the **“Assign Found TAs”** button. Even if the required number is not fully met, the system will assign the proctors it currently has. If there are still not enough eligible TAs within the department, the user can click **“Request Cross-Departmental Proctors”**, which triggers a request for assistance from other departments. Once this request is submitted, the exam status automatically updates to **“Waiting for Cross-Department Approval”**, indicating that the system is now waiting for external proctor assignments.





After the Dean grants approval for cross-departmental invigilation, the status of the exam is updated to **“Awaiting Cross-Department Proctor”**. At this stage, a new action icon labeled **“Assign Paid Cross-Department Proctor”** becomes visible in the Actions column.

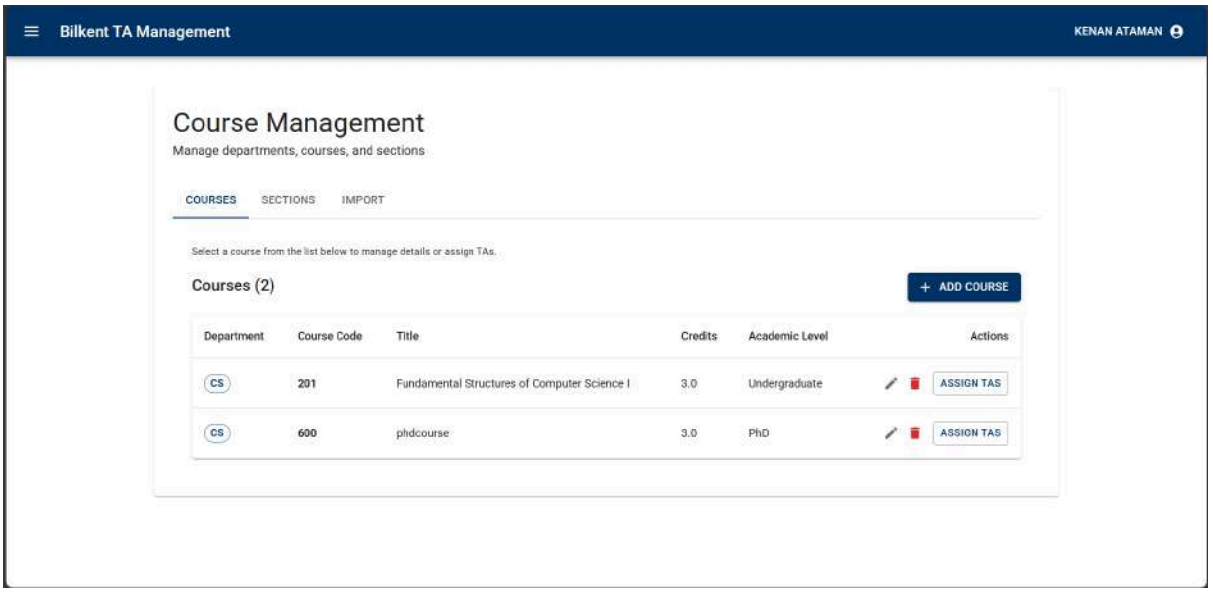


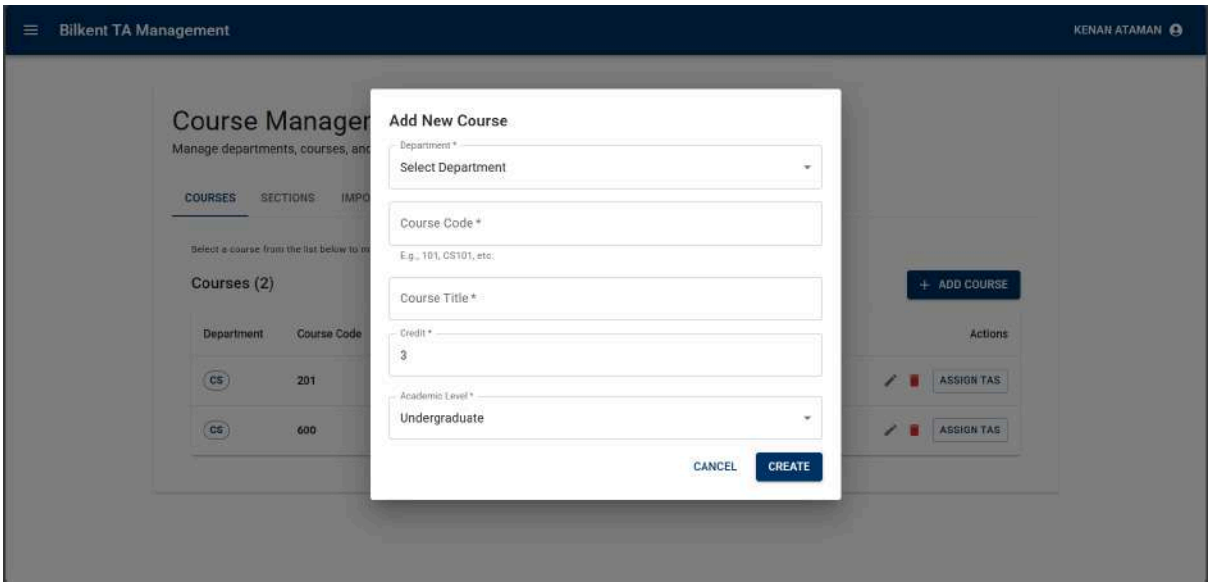
When the authorized staff clicks this icon, a proctor assignment interface opens just like the standard assignment panel, but now includes eligible TAs from other departments (such as IE, not just CS). Within this dialog, the user can manually assign any of these cross department TAs and has the option to mark the assignments as paid by checking the **“Mark selected assignments as paid”** box. This step ensures that external invigilators are properly compensated and logged accordingly. Once proctors are selected and the process is confirmed, the exam continues progressing toward the final “Ready” state.

If the assignment is finalized, the TA workload is updated accordingly, and the exam status changes to “Ready”, indicating all requirements for the exam are fulfilled. A success

message such as *“Proctors assigned successfully”* is also shown to notify the staff that the assignment has been completed.

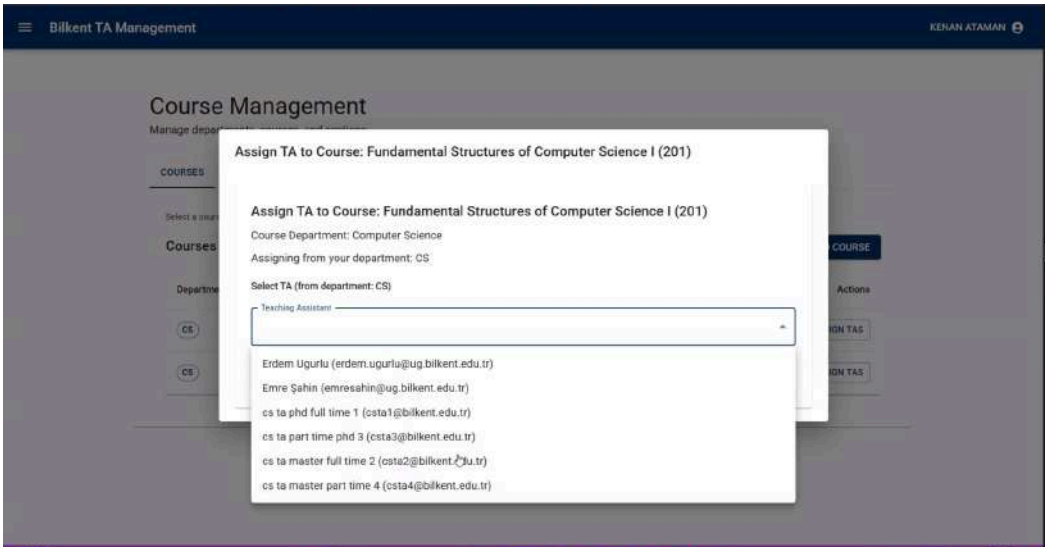
3.3.5 Course Management

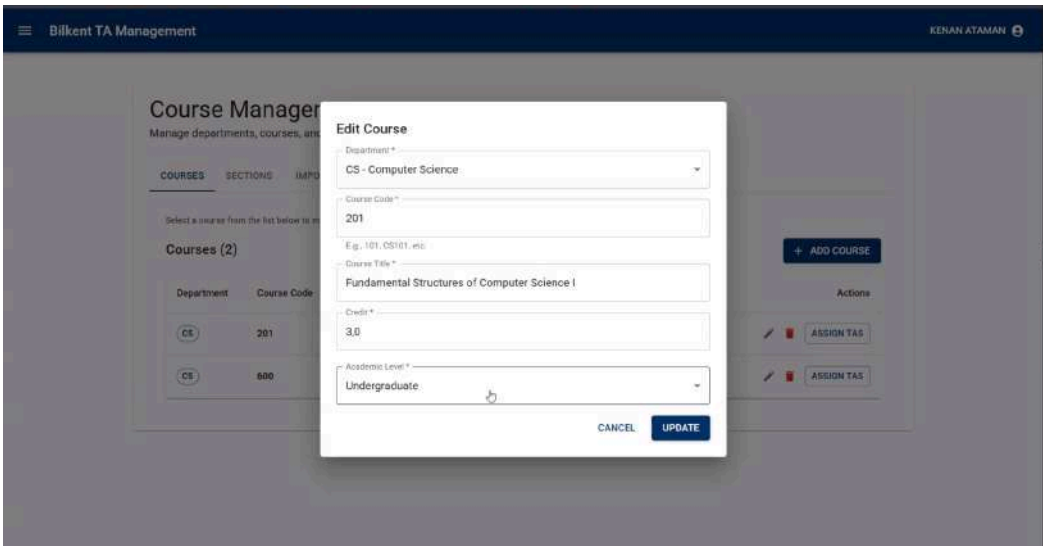




The **Course Management** interface allows Authorized Staff to oversee all courses, sections, and teaching assistant (TA) assignments within their department. Under the **“Courses”** tab, the system displays all available courses, including attributes such as department code (e.g., CS), course code (e.g., 201), course title (e.g., “Fundamental Structures of Computer Science I”), credit value, and academic level (Undergraduate, Graduate, or PhD). For each course, staff members can edit course details, delete the course entirely, or assign TAs using the **“Assign TAs”** button. This section ensures centralized control over departmental course data.

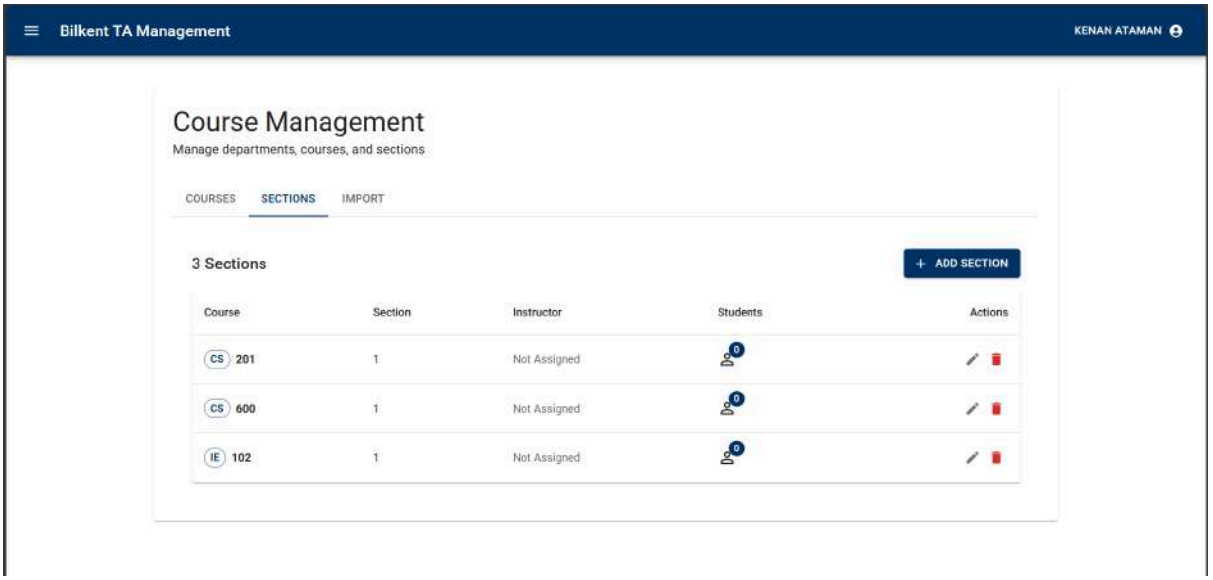
To add a new course, the user clicks the **+ Add Course** button, which opens a structured input form. This form requires essential information such as department selection, course code, title, credit value, and academic level. Once all mandatory fields are completed, clicking **Create** adds the course to the database. This ensures that every new course entry meets departmental formatting standards and is ready for section and TA association.

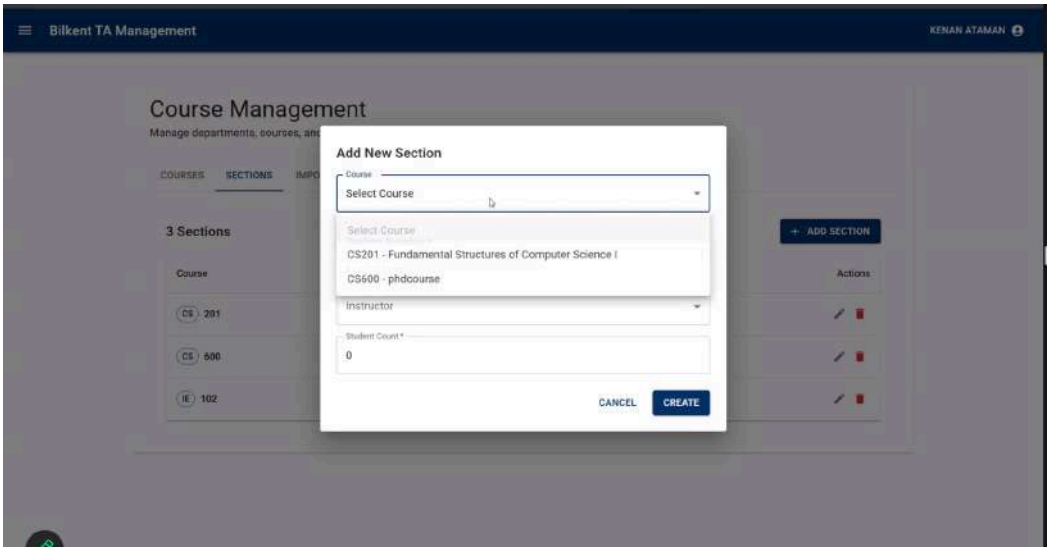




In addition to editing course data, Authorized Staff can initiate TA assignments via the integrated modal window triggered by the “Assign TAs” button. This modal automatically detects the department of the course and filters eligible TAs accordingly. The list includes detailed TA identities—name, institutional email, degree level (Master or PhD), and employment type (full-time or part-time)—to help staff make informed decisions. Once a TA is selected from the dropdown menu, the system links them to the relevant course, ensuring

that teaching responsibilities are distributed in line with departmental needs. This assignment process enforces institutional constraints and eliminates scheduling conflicts, contributing to a smooth and balanced semester workflow.



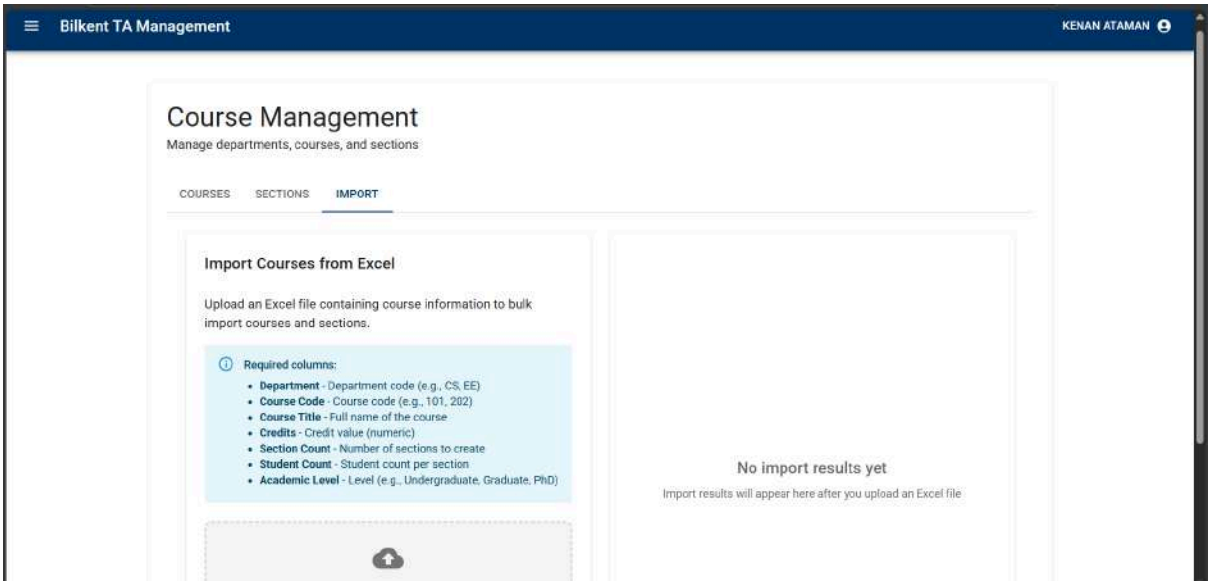


After navigating to the Course Management module, instructors or authorised staff can manage not only entire courses but also their individual sections. Clicking the **"Sections"** tab reveals a structured view of all sections currently registered in the system. Each entry is shown alongside the relevant course code, section number, instructor name, and total number of students enrolled.

This breakdown is especially important, as most operational tasks in the system—such as assigning TAs or scheduling proctoring—are conducted at the section level. For instance, even if a course like “CS 201” exists, its actual assignments and exam details will be associated with specific sections like “CS 201 - Section 1”.

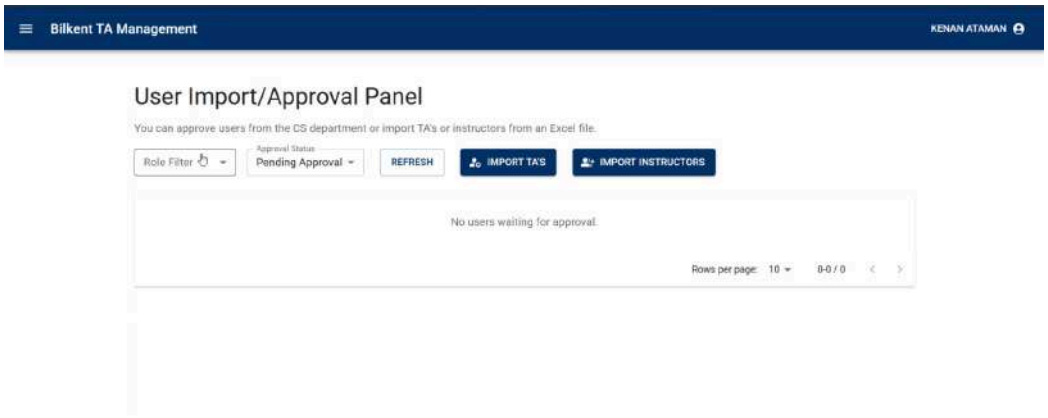
To add a new section, users click on the **“+ Add Section”** button. This opens a pop-up form where they must first select an existing course from a dropdown menu. Once a course is selected, a unique section number (such as 1, 2, or 3) must be provided. Instructors can then assign themselves or another listed instructor from a dropdown list, followed by entering the total student count in that section. Pressing “Create” finalises the operation and instantly updates the list.

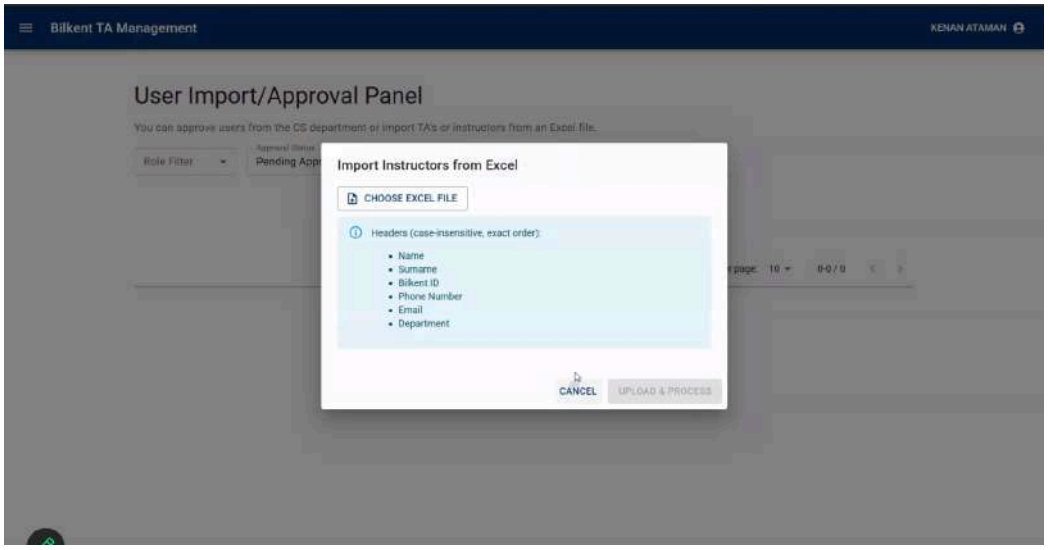
Already existing sections can also be managed through action icons beside each row. The pencil icon allows editing a section’s details, while the trash icon is used to delete a section if necessary.



The **“Import”** tab facilitates bulk creation of courses and sections through Excel or CSV file uploads. This feature supports both drag-and-drop and manual file selection. The uploaded spreadsheet must contain specific required columns: Department, Course Code, Course Title, Credits, Section Count, Student Count per section, and Academic Level. Once the file is uploaded and validated, the system auto-generates the listed courses and sections, significantly reducing manual effort during high-volume operations such as the start of a semester. This import functionality is especially valuable for large departments needing to streamline administrative setup.

3.3.5 User Import/Approval Panel





In the Bilkent TA Management System, the **User Import/Approval Panel** is designed specifically for Authorized Staff to add new Teaching Assistants (TAs) or Instructors in bulk.

This feature simplifies the process of registering new users by allowing staff members to upload an Excel file containing user information, instead of entering each record manually.

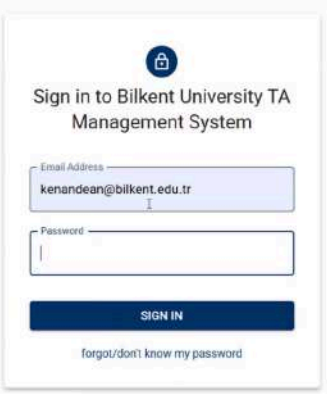
When an Authorized Staff member navigates to this panel, they are first presented with a clean interface showing pending user approvals. At the top of the page, there are two filters: the **Role Filter**, which allows narrowing the list by user type (TA or Instructor), and the **Approval Status** filter, which categorizes users based on whether they are pending, approved, or rejected. The **Refresh** button is available to reload the latest user data at any time.

To import new users into the system, the staff member may click on either the **Import TAs** or **Import Instructors** button. Selecting either opens a dialog box that clearly lists the required headers that must be included in the uploaded Excel file. For TAs, the headers include: Name, Surname, Email, Bilkent ID, IBAN, Phone Number, Employment Type (Part-Time/Full-Time), Academic Level (Master's/PhD), Undergraduate University, and Workload Number. These fields should ideally follow the specified order and are case-insensitive.

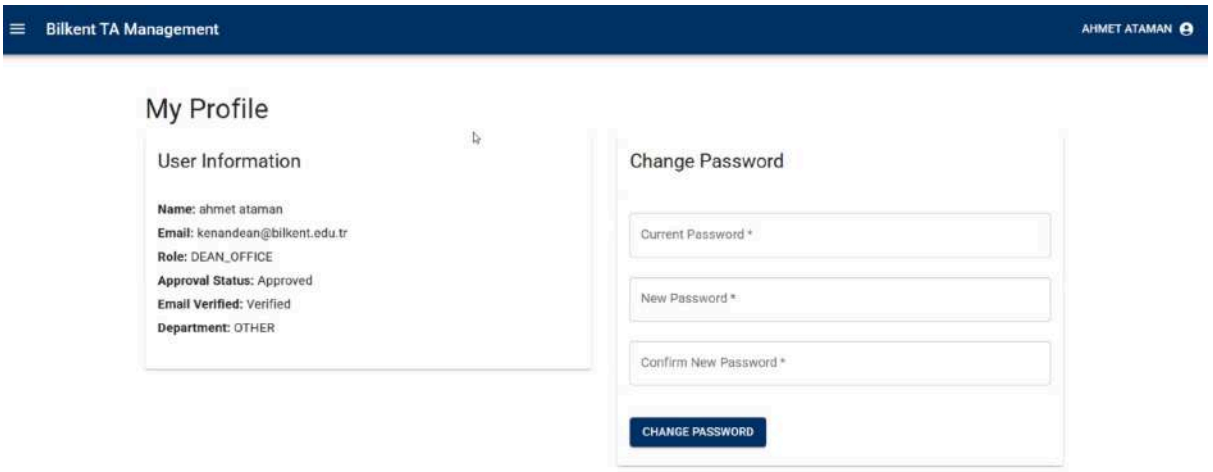
Similarly, when importing Instructors, the required Excel headers are: Name, Surname, Bilkent ID, Phone Number, Email, and Department. After selecting the Excel file, the staff member can proceed by clicking the **Upload & Process** button, which uploads the data and creates the new user records in the system.

This panel is especially useful at the beginning of a semester or during large-scale user updates. It ensures accurate, consistent data entry and significantly reduces the administrative workload. By using this panel, the Authorized Staff can maintain up-to-date user lists with minimal effort while ensuring a smooth onboarding process for TAs and Instructors.

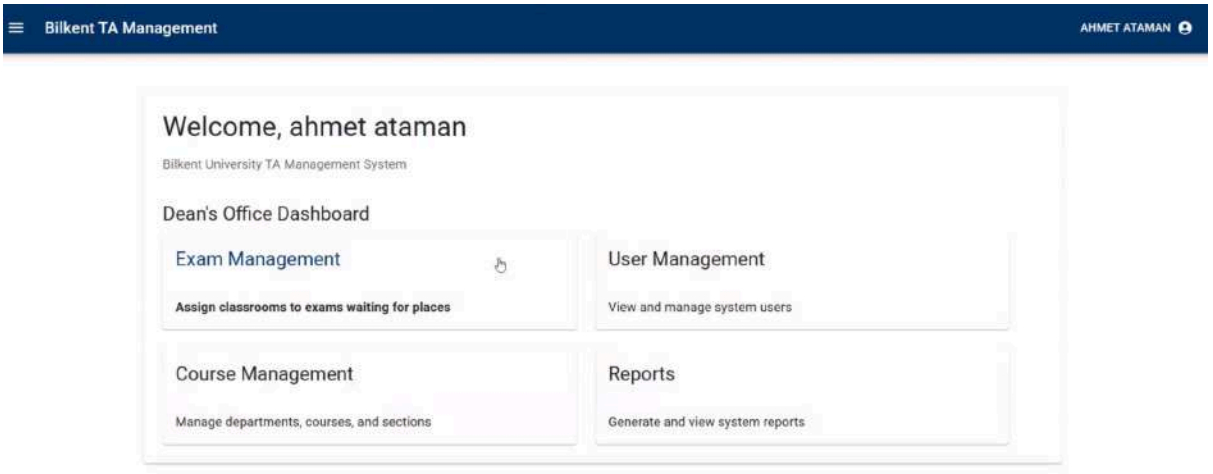
3.4 Dean Office



Unlike other users, Dean accounts are not self-registered; instead, their credentials (including password) are directly created and provided by the Admin.

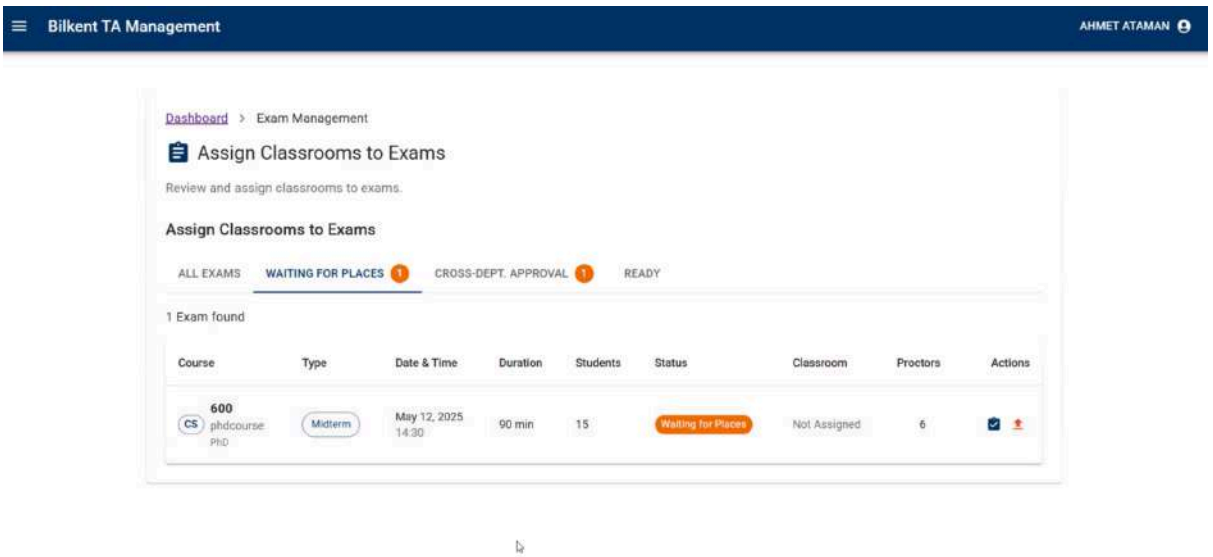


3.4.1 Assigning Classrooms



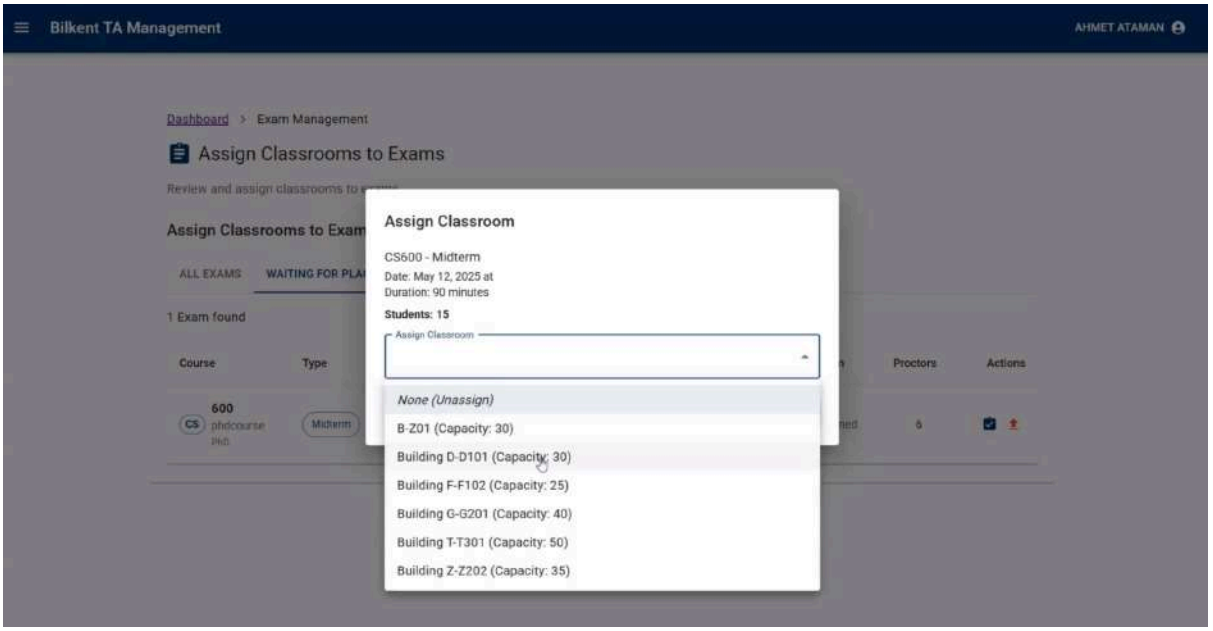
This is the initial landing page for Dean users, providing direct access to core management modules such as Exam Management, Course Management, User Management, and institutional Reports.

After an exam is created and the list of students has been uploaded by the Authorized Staff, the scheduling process transitions to the “Waiting for Places” phase. At this point, no classrooms have yet been assigned for the selected exam, and the responsibility for room allocation falls under the dean’s office.



On this screen, each exam that is in the Waiting for Places status is listed along with key metadata, including the Course Code, Exam Type (Midterm, Quiz, or Final), Date & Time, Duration, Number of Students, and Number of Required TAs. The interface highlights the

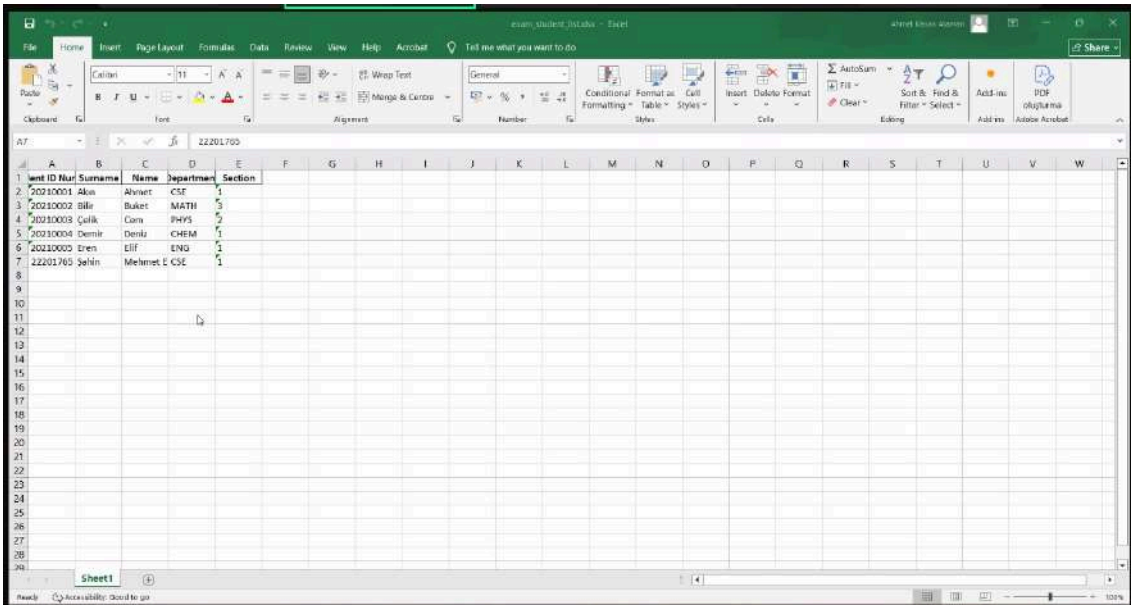
number of exams currently waiting for room assignments by displaying a count next to the “Waiting for Places” header. This real-time indicator helps administrators prioritize upcoming scheduling tasks.



Each exam entry also includes a Proctors field, which indicates how many TAs need to be assigned to that exam. Based on the number of students listed in the uploaded student (excel) file, the system automatically calculates how many rooms are necessary. This logic ensures that no manual calculation is needed by the dean’s office during this phase.

Using the Actions column, the user can assign available classrooms to each exam by selecting from a dropdown menu populated with all valid exam locations within the system.

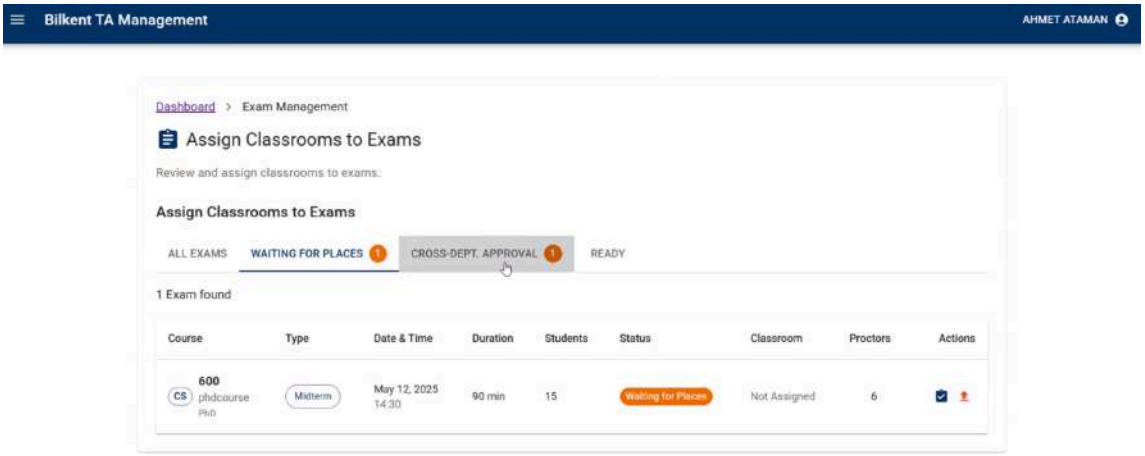
Upon successful assignment, the exam’s status is updated, and the system reflects that the room has been reserved. Once classrooms have been assigned for all upcoming exams, the interface displays a message stating “No exam found” under the Waiting for Places section, confirming that there are no remaining unassigned exams. At the same time, these exams transition to the Awaiting Proctors status under the All Exams tab, signaling that the next step—TA proctor assignment—can now begin.

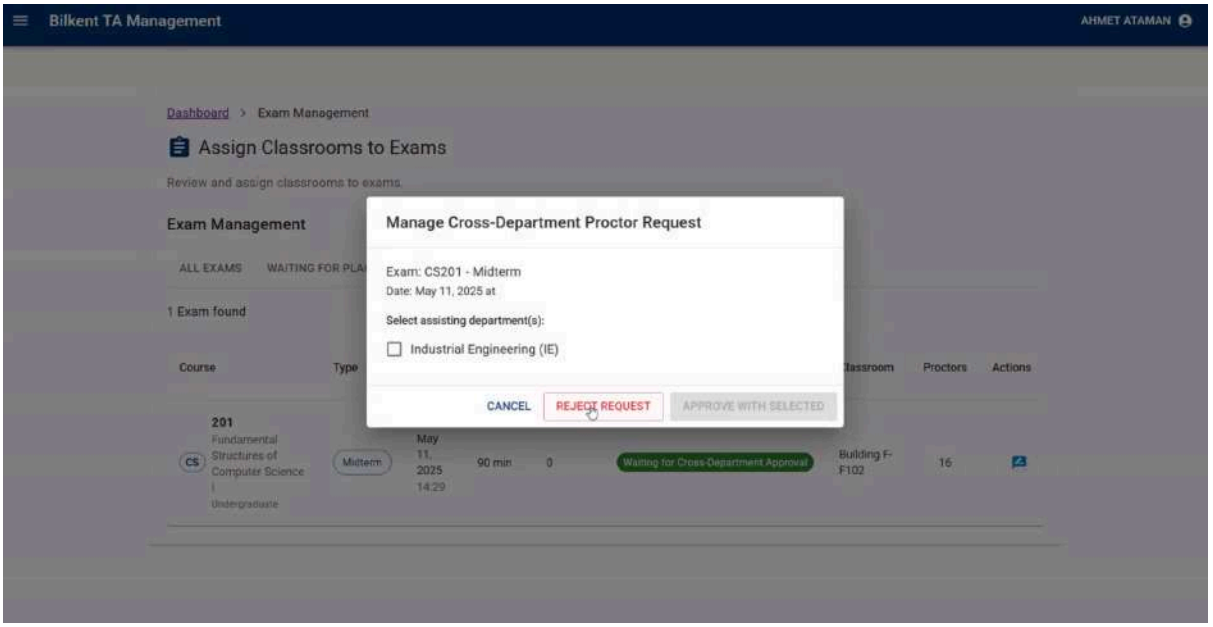


ent ID Nur	Surname	Name	Departmen	Section
20210001	Akın	Ahmet	CSE	1
20210002	Bilir	Buket	MATH	3
20210003	Çelik	Can	PHYS	2
20210004	Demir	Deniz	CHEM	1
20210005	Eren	Elif	ENG	1
22201765	Şahin	Mehmet E	CSE	1

Excel list of the students who will enter the exam

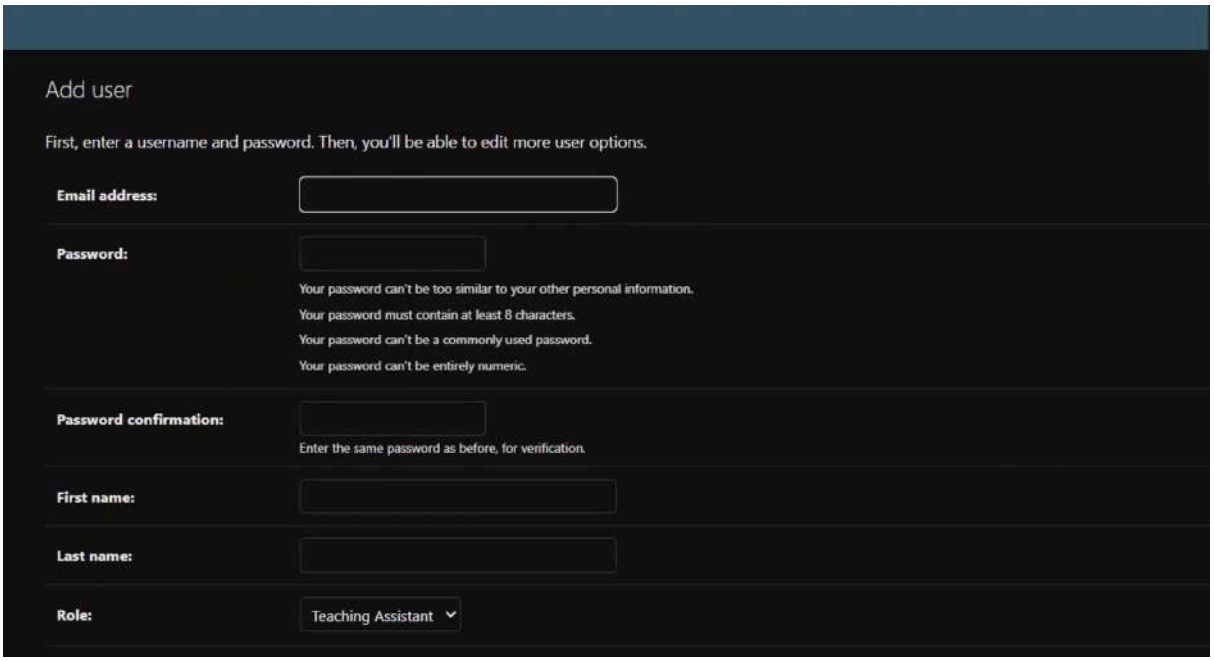
3.4.2.Cross-Department Proctor Approval





In cases where the number of required proctors exceeds the department’s capacity, a cross-department proctoring request is generated. This request is reviewed by the dean’s office. If the dean rejects the request using the red button, the status of the exam automatically returns to the requesting department’s screen and is marked as “Awaiting Proctor.” If the request is approved, the exam is forwarded to the authorized staff of the assisting department (e.g., IE), where the TA assignment process continues. This conditional workflow ensures that proctoring responsibilities are only distributed across departments with dean-level oversight and approval.

3.5 Admin



Backend demonstration of the admin's capability to create new users and assign roles

Admin users are granted the highest level of access privileges within the TA Management System. Although they do not have a dedicated page in the frontend interface, Admins can perform user management operations—such as creating and deleting accounts for Teaching Assistants, Instructors, Authorized Staff, and Dean Office members—directly through the backend (Django admin panel). They can assign roles at the time of account creation and manage login credentials.

While Admins have the technical authority to access and modify any part of the system, they do not have specific responsibilities like task approval or exam coordination. Their main role is limited to overseeing user setup and removal, ensuring that access permissions are configured correctly. Beyond this, Admins may intervene or inspect other system modules if necessary, but such actions are not part of their routine duties.

4 BUILD & EXECUTION INSTRUCTIONS

This guide walks you through setting up and running the TA Management System as a local web application, covering both backend (Django) and frontend (React) with precise commands and notes on environment differences.

4.1 Prerequisites

1. **Git**
2. **Python 3.13** (ensure `python` or `python3` maps to v3.13)
3. **Node.js & npm** (v18+ recommended)
4. **Optional but recommended:** virtualenv (for Python)

Verify on macOS/Linux:

```
git --version
python3 --version        # should report 3.13.x
node --version
npm --version
```

Verify on Windows (PowerShell):

```
git --version
python --version         # should report 3.13.x
node --version
npm --version
```

4.2 Clone the Repository

```
# From your home or dev folder:
git clone https://github.com/ErdemUgurlu/CS319-Project.git
cd CS319-Project
```

4.3 Configure Environment Variables

Both frontend and backend rely on environment variables stored in .env files. Copy and edit these templates.

1. **Backend**

   ```
   cd backend
   cp .env.example .env
   ```

   Open `backend/.env` and set:

- `SECRET_KEY`: any random secure string
- `DEBUG=True` (for development)
- `ALLOWED_HOSTS=localhost,127.0.0.1`
- Leave `DATABASE_URL` blank or pointing to SQLite

2. **Frontend**

```
cd ../frontend/ta-management-client
cp .env.example .env
```

Open `ta-management-client/.env` and set:

```
REACT_APP_API_BASE_URL=http://localhost:8000/api
```

4.4 Backend Setup (Django)

1. **Create & activate virtual environment**
   - macOS/Linux:
     - `cd ../backend`
     - `python3 -m venv venv`

     ```
     source venv/bin/activate
     ```

   - Windows (PowerShell):
     - `cd .\backend`
     - `python -m venv venv`

     ```
     .\venv\Scripts\Activate.ps1
     ```

2. **Install Python dependencies**

```
pip install --upgrade pip
pip install -r requirements.txt
```

If requirements.txt is missing:

```
pip install django djangorestframework python-dotenv
pip freeze > requirements.txt
```

3. **Apply database migrations & create superuser**

   ```
   python manage.py migrate
   python manage.py createsuperuser
   ```

   Follow prompts for email/username and password.

4. **Collect static files** *(only if deploying static assets)*

   ```
   python manage.py collectstatic --noinput
   ```

5. **Run the development server**

   ```
   python manage.py runserver 8000
   ```

   - **Backend API URL:** `http://localhost:8000/api/`
   - **Django Admin:** `http://localhost:8000/admin/`

4.5 Frontend Setup (React)

1. **Install Node dependencies**

   ```
   cd ../frontend/ta-management-client
   npm install
   ```

2. **Run development server**

   ```
   npm start
   ```

   - Opens at http://localhost:3000
   - Proxies API calls to `http://localhost:8000/api` based on `.env` setting

4.6 Unified Dev Command

If you'd like to start both servers with one command (using the provided dev script):

1. Ensure you're at the **project root** (`CS319-Project`)
2. Run:

   ```
   npm run dev
   ```

This script will:

- Launch the Django server (`backend/manage.py runserver 8000`)
- Launch the React server (`frontend/ta-management-client/npm start`)

Note: The `dev` script uses local `python` and `npm` binaries. Ensure your PATH is set up so that `python` and `npm` point to the correct executables. You can open separate terminals if preferred.

4.7 Building for Production (Frontend)

1. In `frontend/ta-management-client`:
   `npm run build`
2. A `build/` folder appears with static assets.
3. Serve these via any static server or integrate into Django's `STATICFILES_DIRS` for a unified deployment.

4.8 Common Troubleshooting

Issue	Solution
Port 3000 or 8000 in use	Change port: `npm start --port=3001` or `python manage.py runserver 8001`
Virtualenv activation fails on Windows	Run PowerShell as Administrator, or use `./venv/Scripts/activate.bat` instead of `.ps1`
Missing `.env` or env vars not loaded	Ensure `.env` is named correctly and resides in the respective `backend` or `ta-management-client` folder.

`requirements.txt` not present	Manually install dependencies and run `pip freeze > requirements.txt`
Static files 404 after deployment	Ensure `collectstatic` ran and that your web server is pointed at the correct `STATIC_ROOT` or `build` dir.

---

5 WORK ALLOCATIONS

Ahmet Kenan Ataman – 22203434

Analysis Report:
- ➢ Contributed to full Use Case Diagram design.

Design Report:
- ➢ Created Activity, Sequence, State, and Class Diagrams.
- ➢ Worked on Subsystem Decomposition and system architecture.

Implementation:
- ➢ Developed backend logic for proctoring and swap validations.
- ➢ Implemented Django APIs and email registration flow.
- ➢ Built Excel import and backend models.
- ➢ Helped connect the backend with Instructor/Admin UIs.

Final Report:
- ➢ Co-authored Instructor, Dean, and Admin User Guides.
- ➢ Documented backend build instructions.

Berfin Örtülü – 21802704

Analysis Report:
- ➢ Contributed in the Use Case Diagram and Non-Functional Requirements (reliability, performance).
- ➢ Participated in UI mock-up design.

Design Report:

- ➢ Contributed to Subsystem Diagram and Facade/Strategy pattern documentation.
- ➢ Explained UI flows in architecture.

Implementation:
- ➢ Built React pages like TA dashboard, swap, and profile.
- ➢ Created reusable components and modals.
- ➢ Worked on Course Management module and Instructor views.

Final Report:
- ➢ Wrote TA User Guide
- ➢ co-wrote the dean , lessons learned and backend build steps.

Erdem Uğurlu – 22203391

Analysis Report:
- ➢ Focused on admin and staff use cases, helped mock-up screens.

Design Report:
- ➢ Wrote Design Goals and Trade-offs.
- ➢ Revised backend-related class diagrams.

Implementation:
- ➢ Built Django-React components for dashboard, leave, swap, and assignments.
- ➢ Designed role-specific navigation and forms.
- ➢ Helped implement Excel import and email logic for login.

Final Report:
- ➢ Contributed to multiple user guides.
- ➢ Documented backend deployment steps.

Gülferiz Bayar – 21901442

Analysis Report:
- ➢ Contributed to the overall Use Case Diagram and Non-Functional Requirements (usability, performance).
- ➢ Helped design mock-ups for TA and Instructor roles.

Design Report:
- ➢ Documented the Facade and Strategy patterns.
- ➢ Contributed to Subsystem Decomposition and high-level architecture, focusing on UI-backend interaction.

Implementation:
- ➢ Built React-based interfaces for TA and Authorized Staff.
- ➢ Developed key pages like “All Exams”, role dashboards, and forms for availability, leave, and proctor swaps.
- ➢ Implemented visual feedback and structured layouts.

Final Report:
- ➢ Co-wrote the introduction and frontend build steps.
- ➢ Authored the full User Guide for the Authorized Staff role.

Mehmet Emre Şahin – 22201765

Analysis Report:
- ➢ Contributed to Use Case Diagram and flow descriptions.

Design Report:
- ➢ Created main diagrams (Activity, Sequence, State, Class).
- ➢ Documented architecture decisions and trade-offs.

Implementation:
- ➢ Implemented backend logic for assignments and logging.
- ➢ Connected services to frontend; handled Instructor/Admin flows.
- ➢ Designed SQLite models and Excel-based import features.
- ➢ Added basic email setup for password link delivery.

Final Report:
- ➢ Helped write Instructor, Dean, and Admin guides.
- ➢ Contributed to backend build and API documentation.